krishnaik06 / Advanced-House-Price-Prediction-

Code | Issues 2 | Pull requests 2 | Actions | Projects | Wiki | Security | Insights

master

Advanced-House-Price-Prediction- / Exploratory Data Analysis Part 1.ipynb

Go to file

krishnaik06 Add files via upload

Latest commit c3f4f7c on Feb 1, 2020 History

1 contributor

1.27 MB

Download

# Building Machine Learning Pipelines: Data Analysis Phase

In this and the upcoming videos we will focus on creating Machine Learning Pipelines considering all the life cycle of a Data Science Projects. This will be important for professionals who have not worked with huge dataset.

# Project Name: House Prices: Advanced Regression Techniques

The main aim of this project is to predict the house price based on various features which we will discuss as we go ahead

**Dataset to downloaded from the below link**

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)

# All the Lifecycle In A Data Science Projects

1. Data Analysis
2. Feature Engineering
3. Feature Selection
4. Model Building
5. Model Deployment

In [45]:
```
## Data Analysis Phase
## MAin aim is to understand more about the data

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
## Display all the columns of the dataframe

pd.pandas.set_option('display.max_columns',None)
```

In [46]:
```
dataset=pd.read_csv('train.csv')

## print shape of dataset with rows and columns
```

```
print(dataset.shape)
```

```
(1460, 81)
```

In [47]:
```
## print the top5 records
dataset.head()
```

Out[47]:

| | Id | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley | LotShape | LandContour | Utilities | LotConfig | LandSlope | Neighborhood | Co |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 60 | RL | 65.0 | 8450 | Pave | NaN | Reg | Lvl | AllPub | Inside | Gtl | CollgCr | No |
| **1** | 2 | 20 | RL | 80.0 | 9600 | Pave | NaN | Reg | Lvl | AllPub | FR2 | Gtl | Veenker | Fe |
| **2** | 3 | 60 | RL | 68.0 | 11250 | Pave | NaN | IR1 | Lvl | AllPub | Inside | Gtl | CollgCr | No |
| **3** | 4 | 70 | RL | 60.0 | 9550 | Pave | NaN | IR1 | Lvl | AllPub | Corner | Gtl | Crawfor | No |
| **4** | 5 | 60 | RL | 84.0 | 14260 | Pave | NaN | IR1 | Lvl | AllPub | FR2 | Gtl | NoRidge | No |

**In Data Analysis We will Analyze To Find out the below stuff**

1. Missing Values
2. All The Numerical Variables
3. Distribution of the Numerical Variables
4. Categorical Variables
5. Cardinality of Categorical Variables
6. Outliers
7. Relationship between independent and dependent feature(SalePrice)

# Missing Values

In [ ]:
```

```

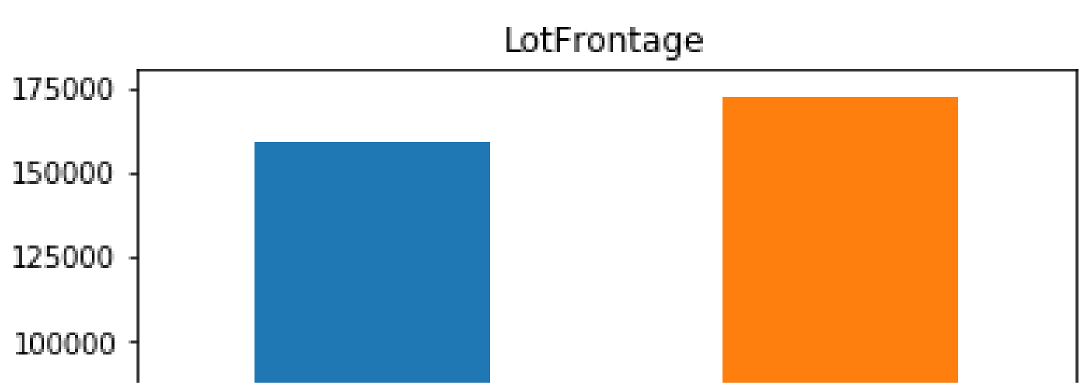
In [48]:
```
## Here we will check the percentage of nan values present in each feature
## 1 -step make the list of features which has missing values
features_with_na=[features for features in dataset.columns if dataset[features].isnull().sum()>1]
## 2- step print the feature name and the percentage of missing values

for feature in features_with_na:
    print(feature, np.round(dataset[feature].isnull().mean(), 4), ' % missing values')
```

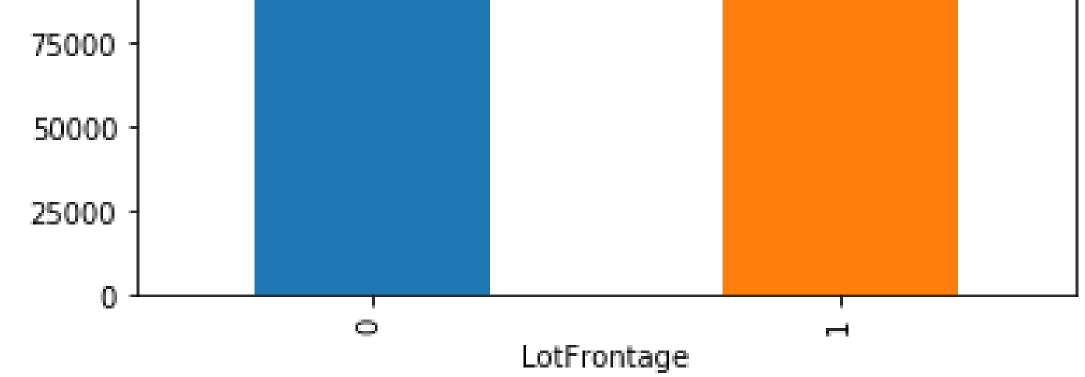
```
LotFrontage 0.1774  % missing values
Alley 0.9377  % missing values
MasVnrType 0.0055  % missing values
MasVnrArea 0.0055  % missing values
BsmtQual 0.0253  % missing values
BsmtCond 0.0253  % missing values
BsmtExposure 0.026  % missing values
BsmtFinType1 0.0253  % missing values
BsmtFinType2 0.026  % missing values
FireplaceQu 0.4726  % missing values
GarageType 0.0555  % missing values
GarageYrBlt 0.0555  % missing values
GarageFinish 0.0555  % missing values
GarageQual 0.0555  % missing values
GarageCond 0.0555  % missing values
PoolQC 0.9952  % missing values
Fence 0.8075  % missing values
MiscFeature 0.963  % missing values
```

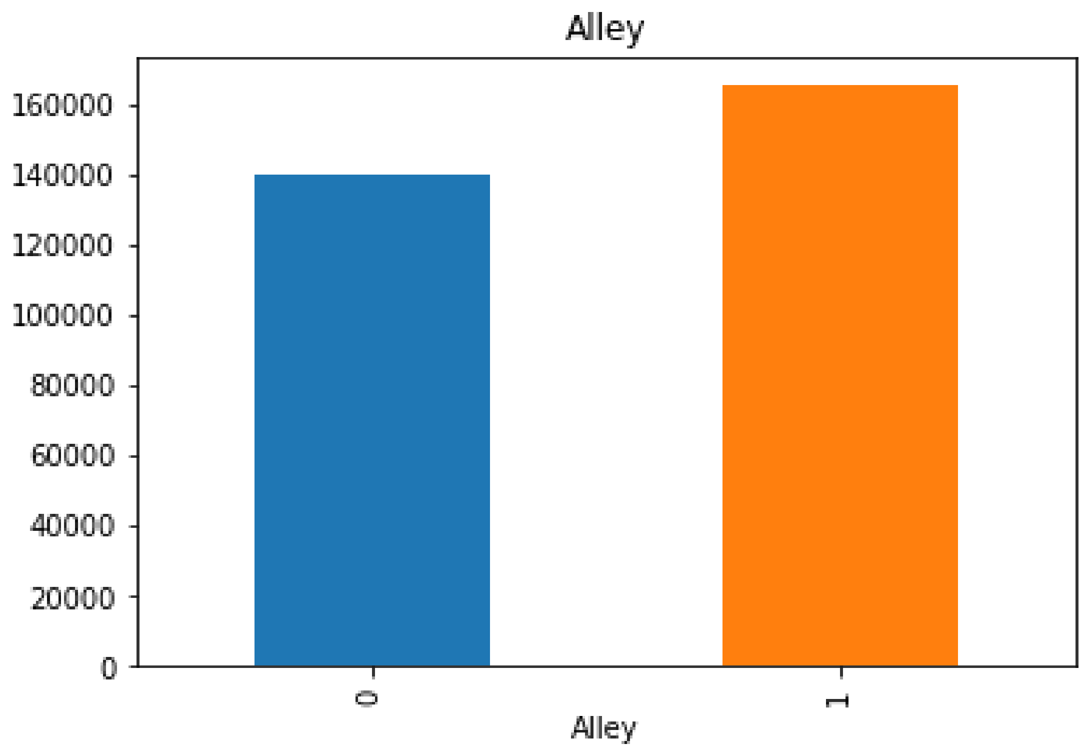
### Since they are many missing values, we need to find the relationship between missing values and Sales Price

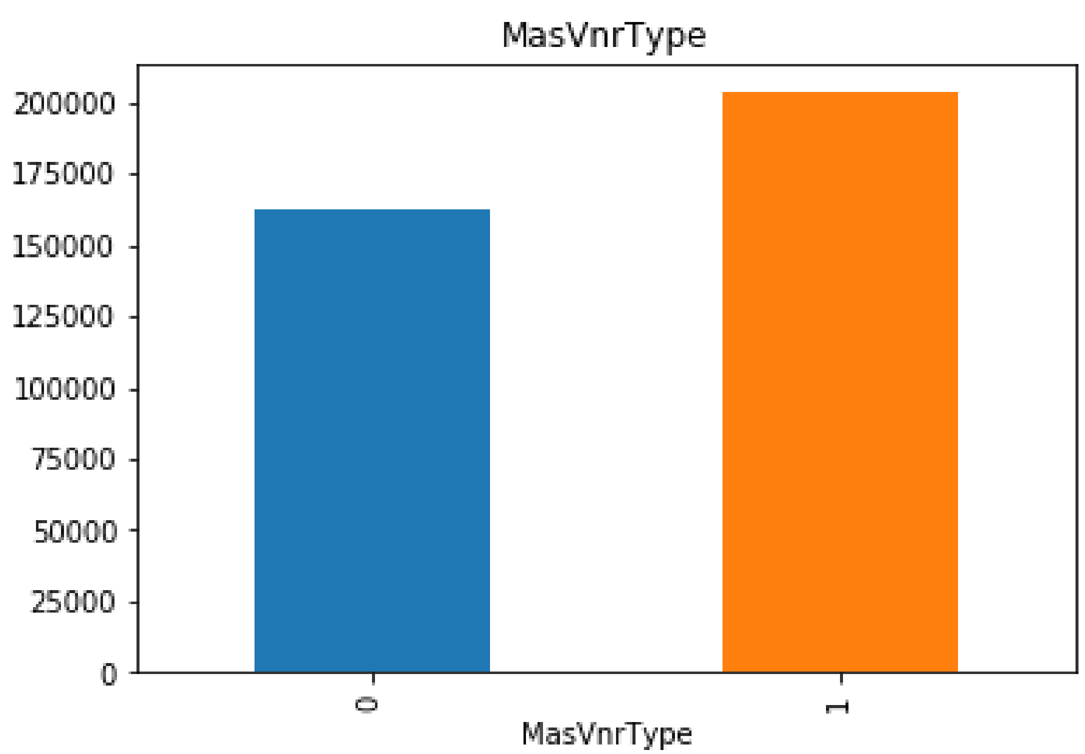
Let's plot some diagram for this relationship

In [49]:
```
for feature in features_with_na:
    data = dataset.copy()

    # let's make a variable that indicates 1 if the observation was missing or zero otherwise
    data[feature] = np.where(data[feature].isnull(), 1, 0)

    # let's calculate the mean SalePrice where the information is missing or present
    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.title(feature)
    plt.show()
```

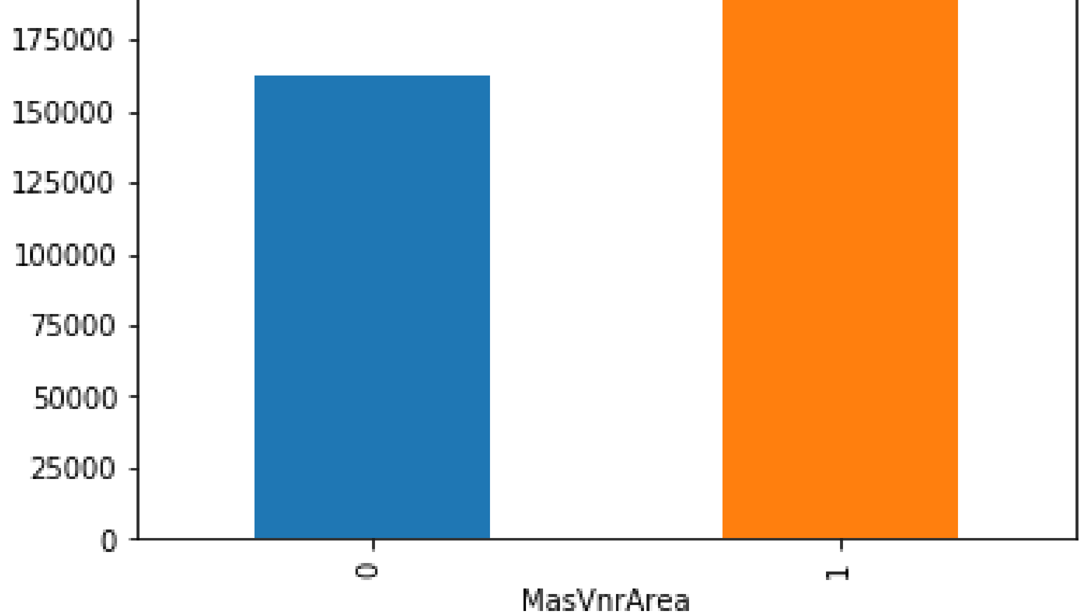


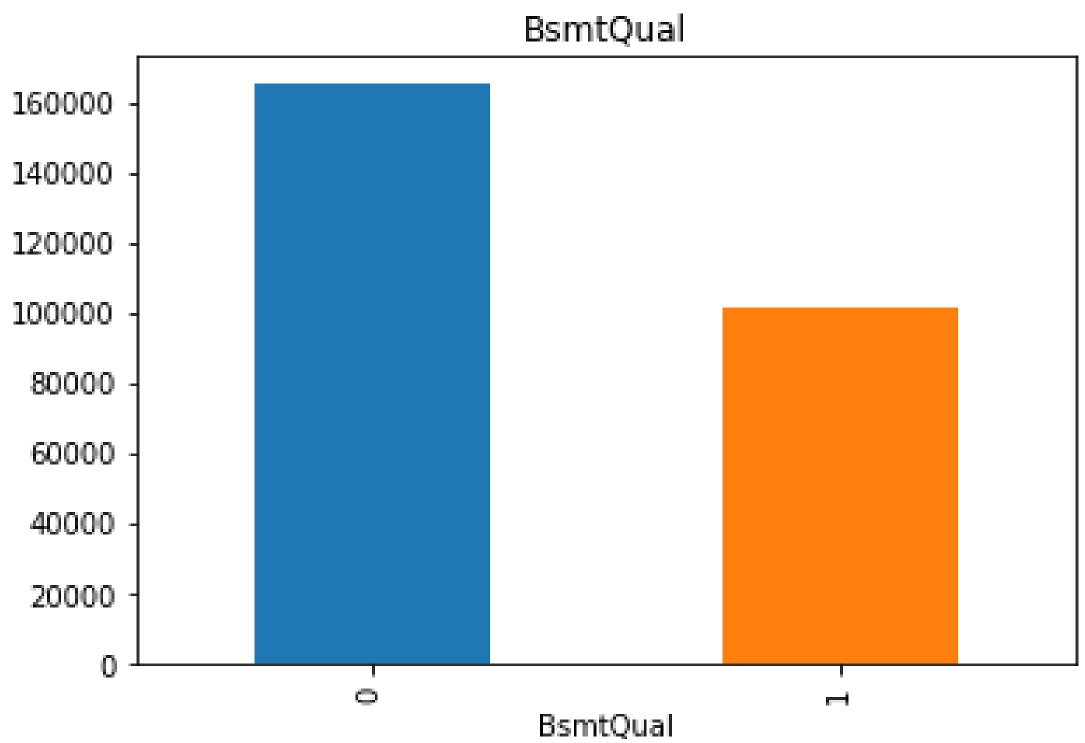

75000
50000
25000
0
0
1
LotFrontage

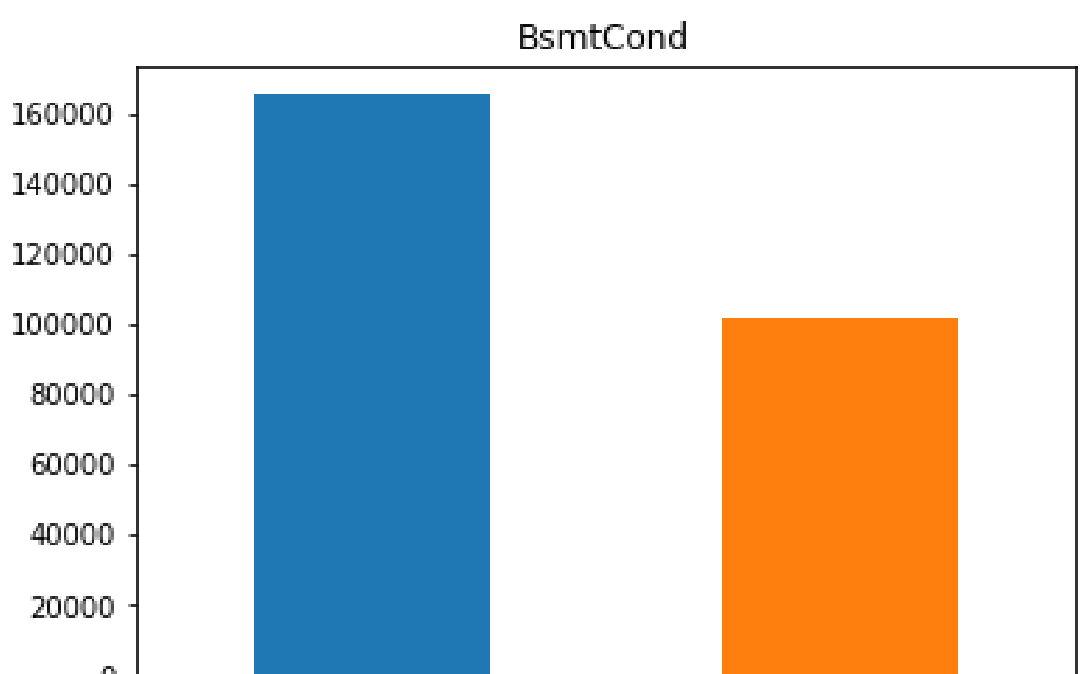

Alley
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
Alley


MasVnrType
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1
MasVnrType

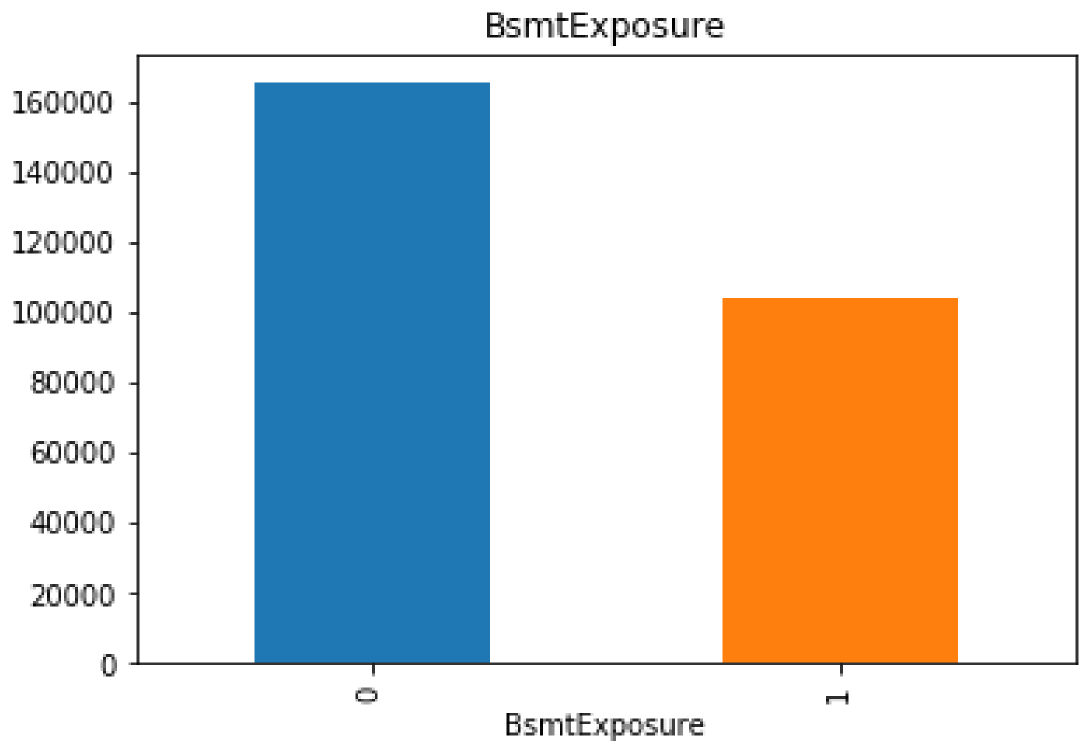

MasVnrArea
200000

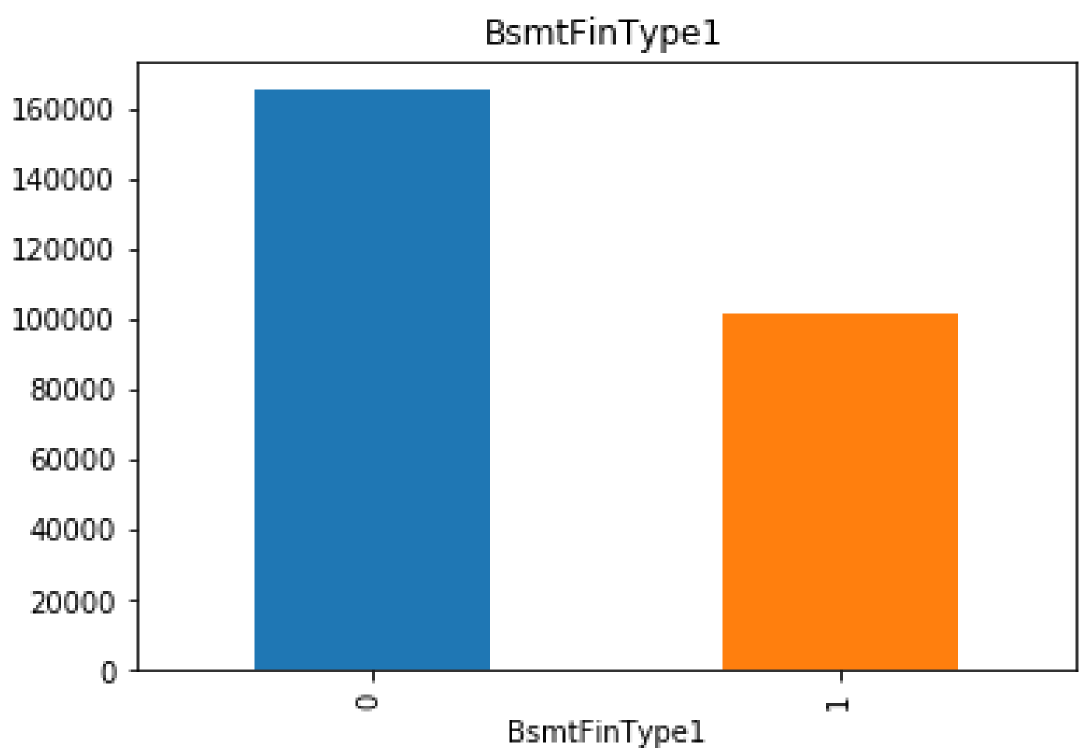

175000
150000
125000
100000
75000
50000
25000
0
0
1
MasVnrArea

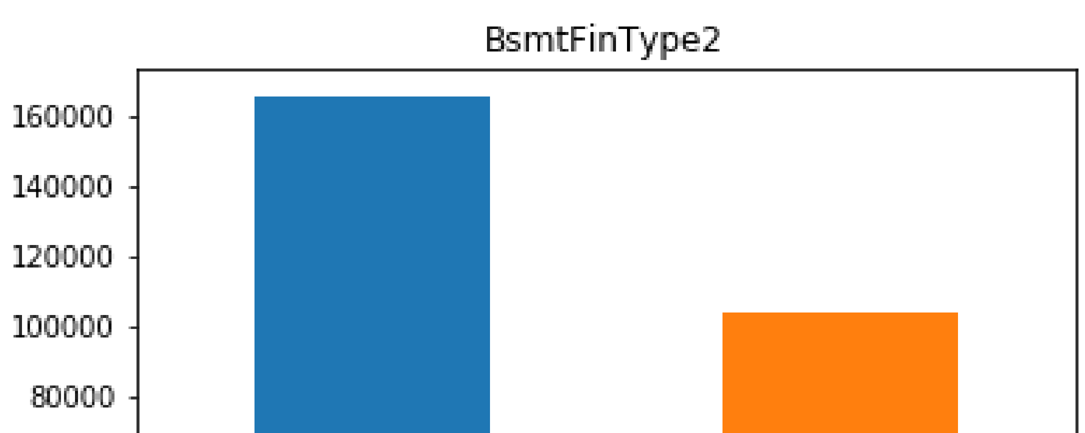

BsmtQual
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
BsmtQual

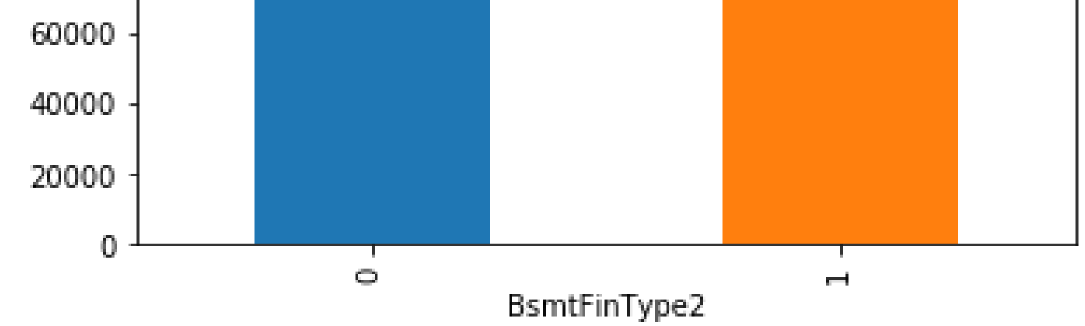

BsmtCond
160000
140000
120000
100000
80000
60000
40000
20000

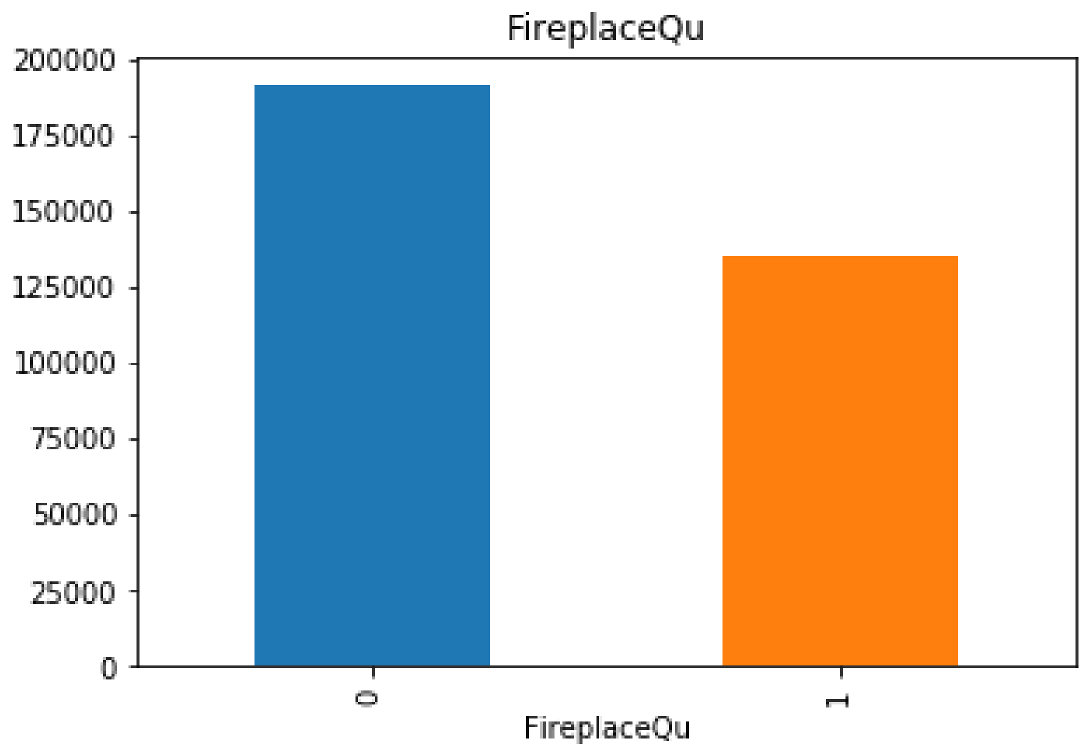

0
0
1
BsmtCond

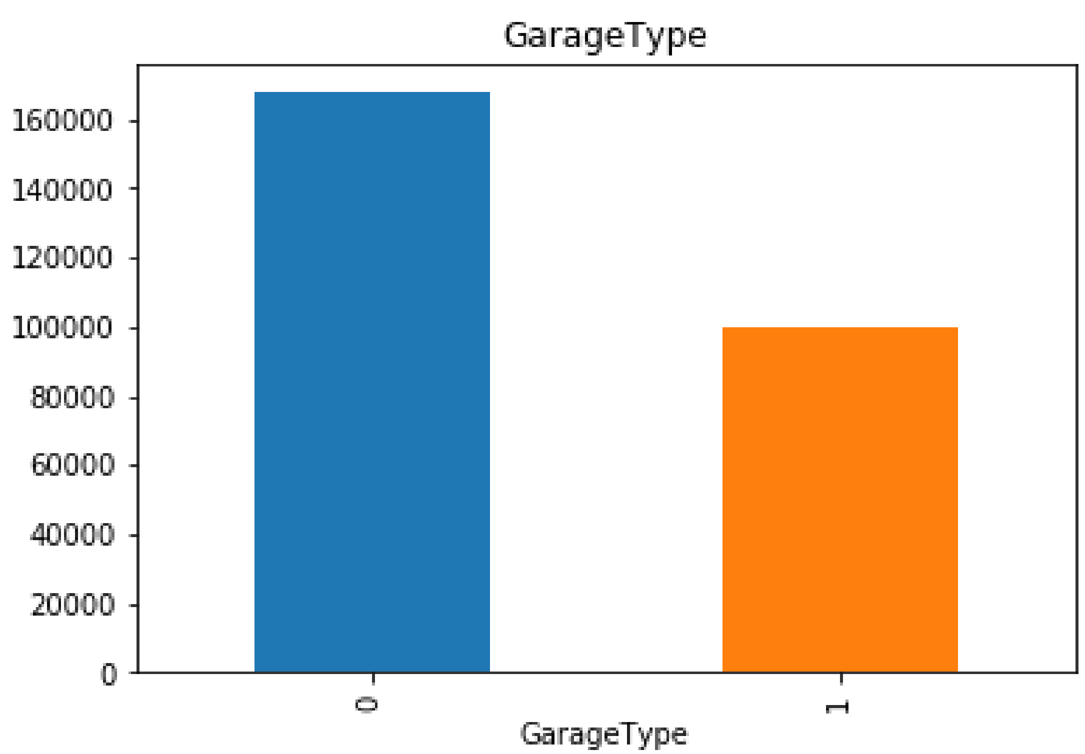

BsmtExposure
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
BsmtExposure

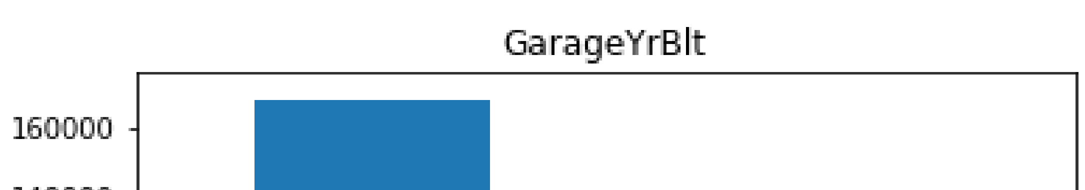

BsmtFinType1
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
BsmtFinType1

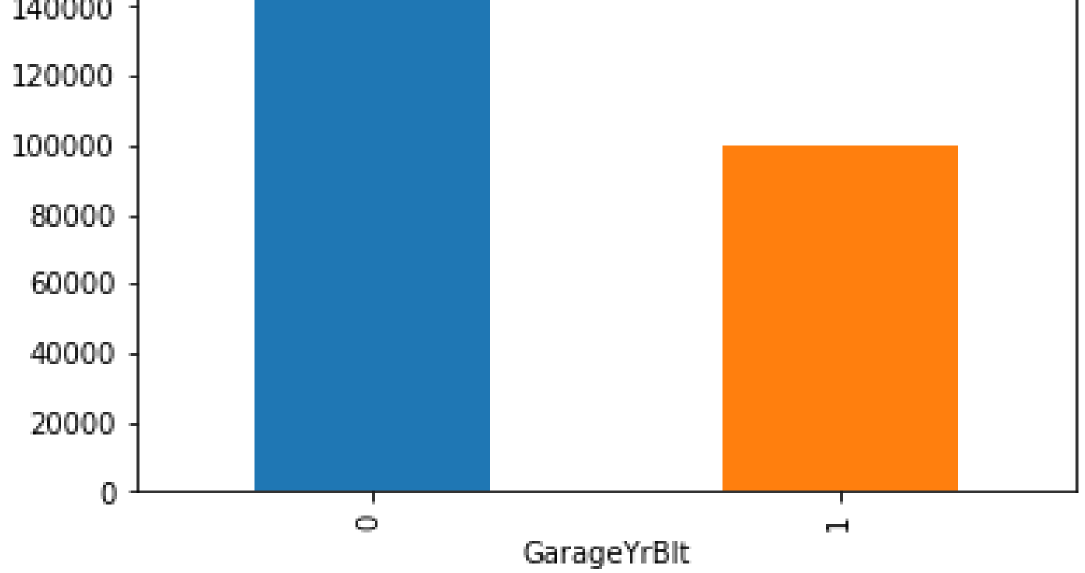

BsmtFinType2
160000
140000
120000
100000
80000

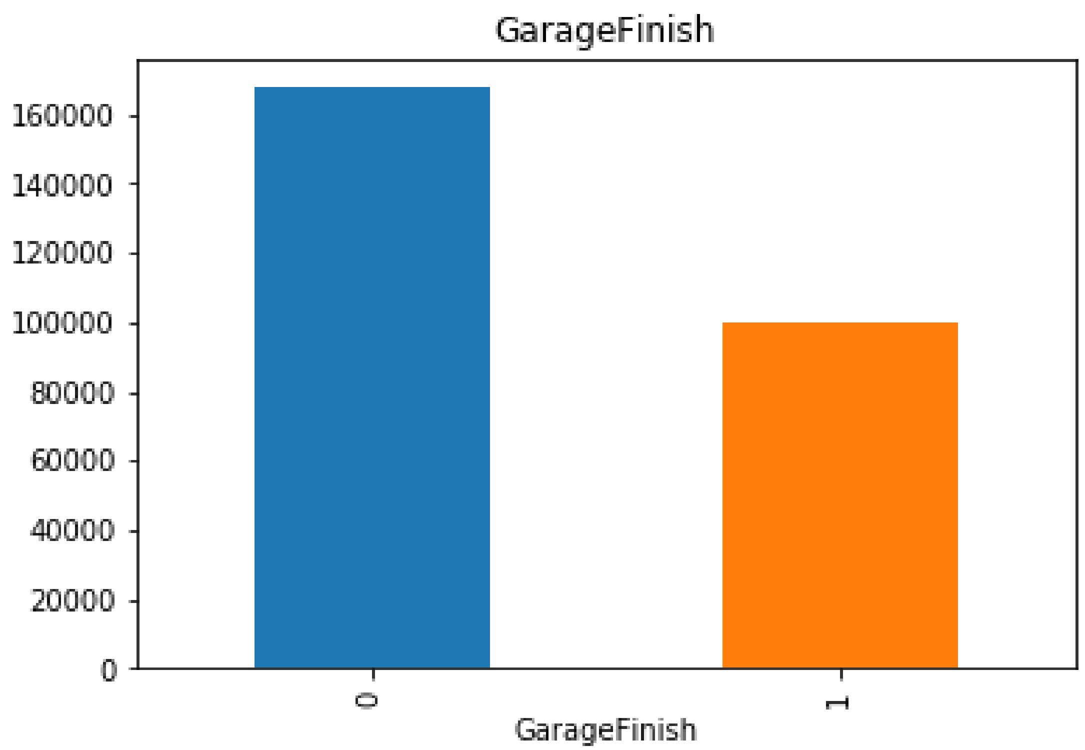

60000
40000
20000
0
0
1
BsmtFinType2

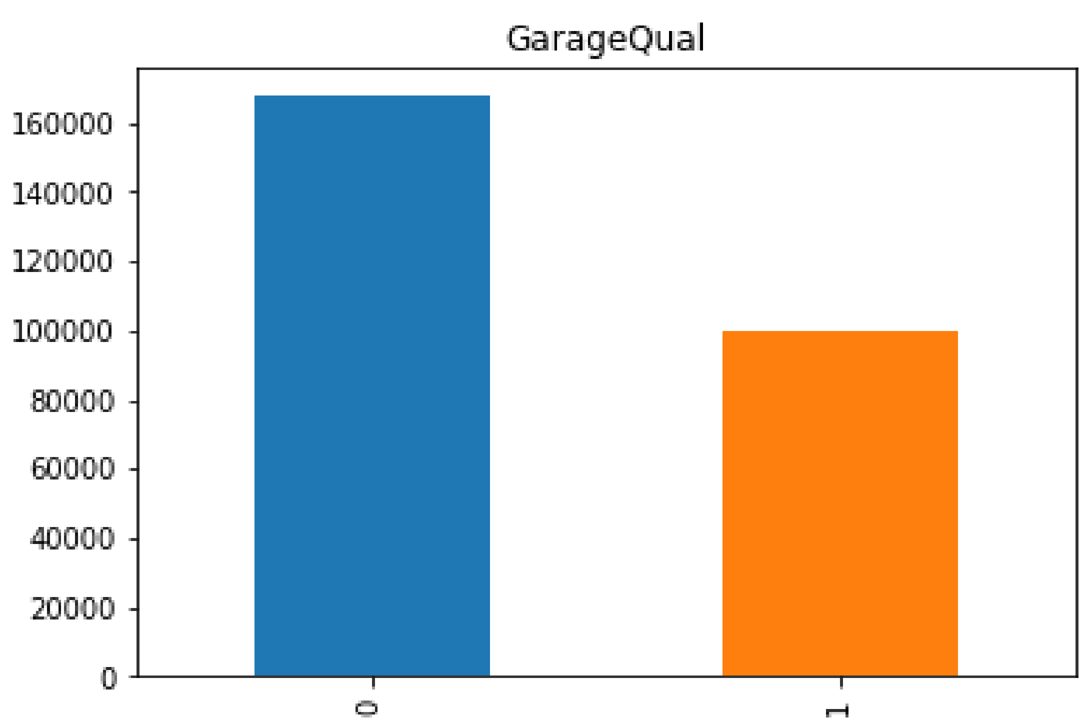

FireplaceQu
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1
FireplaceQu


GarageType
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
GarageType

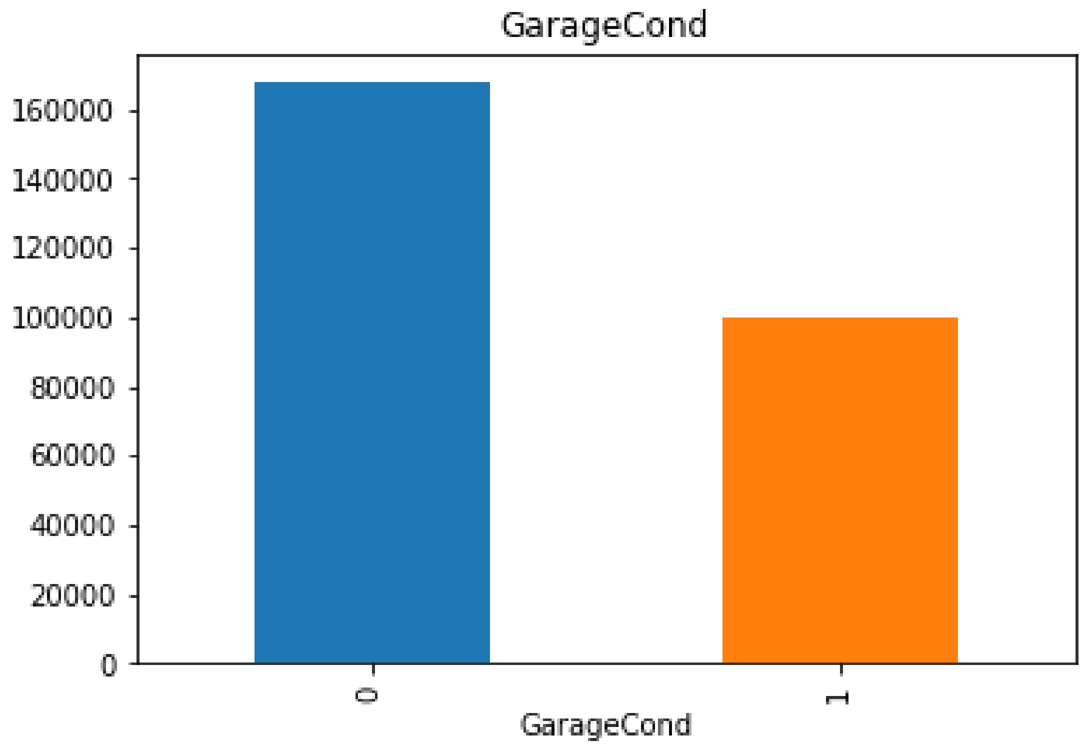

GarageYrBlt
160000

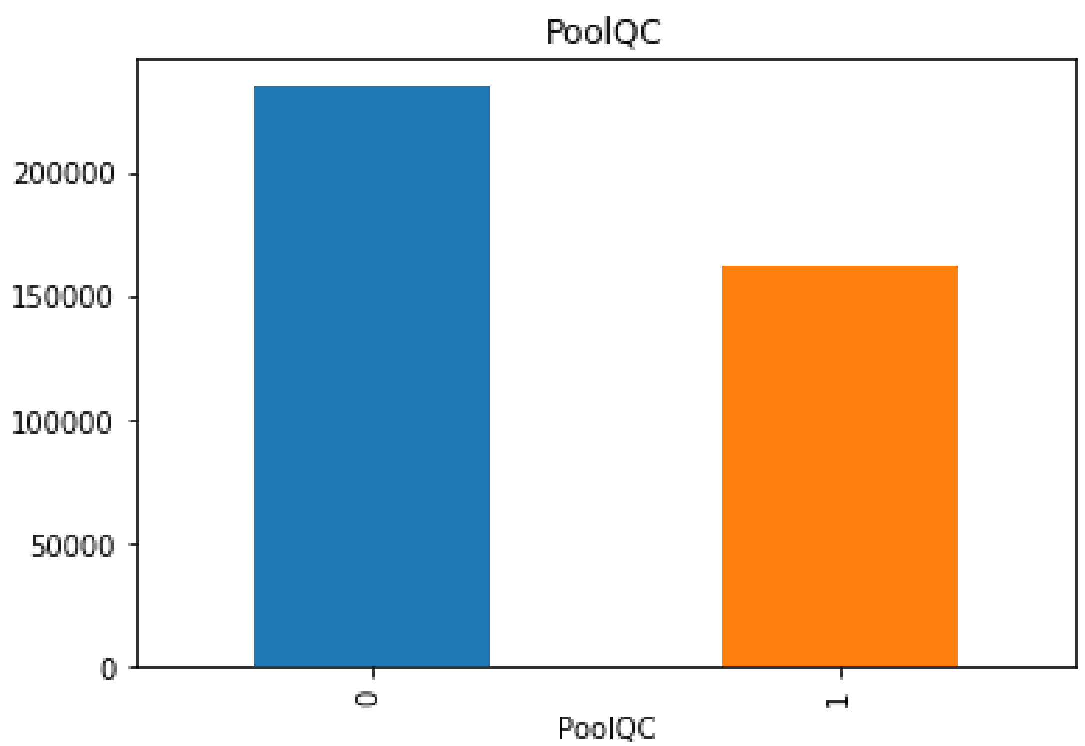

140000
120000
100000
80000
60000
40000
20000
0
0
1
GarageYrBlt

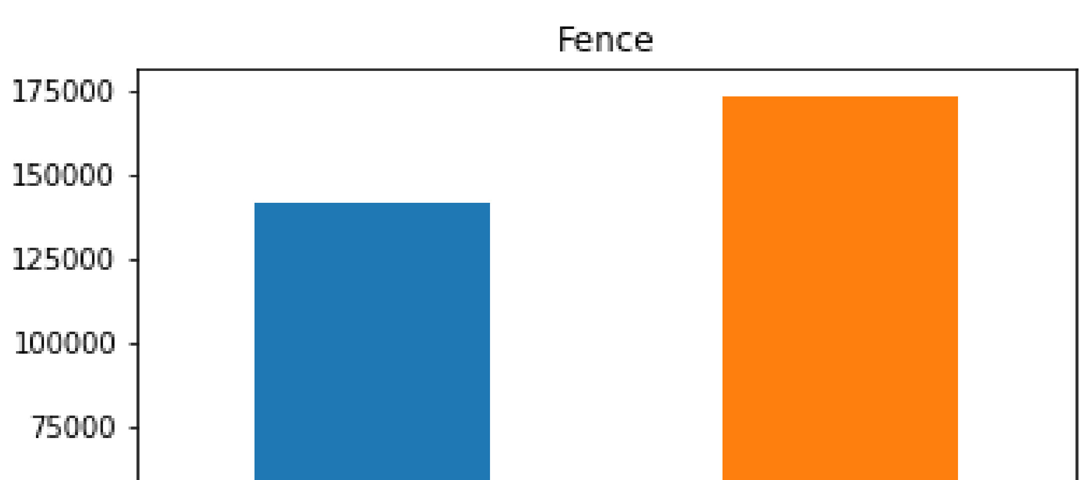

GarageFinish
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
GarageFinish

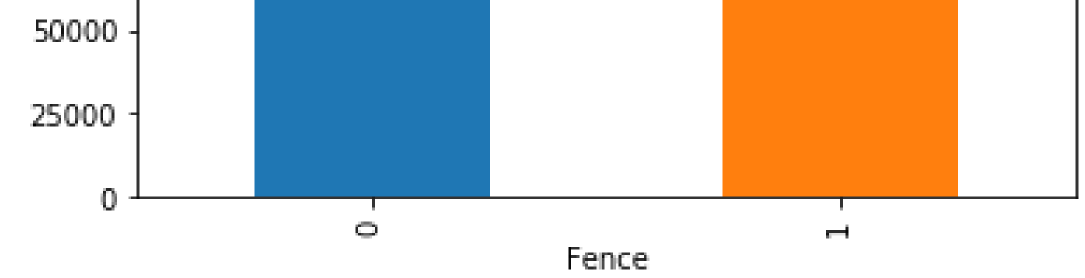

GarageQual
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1

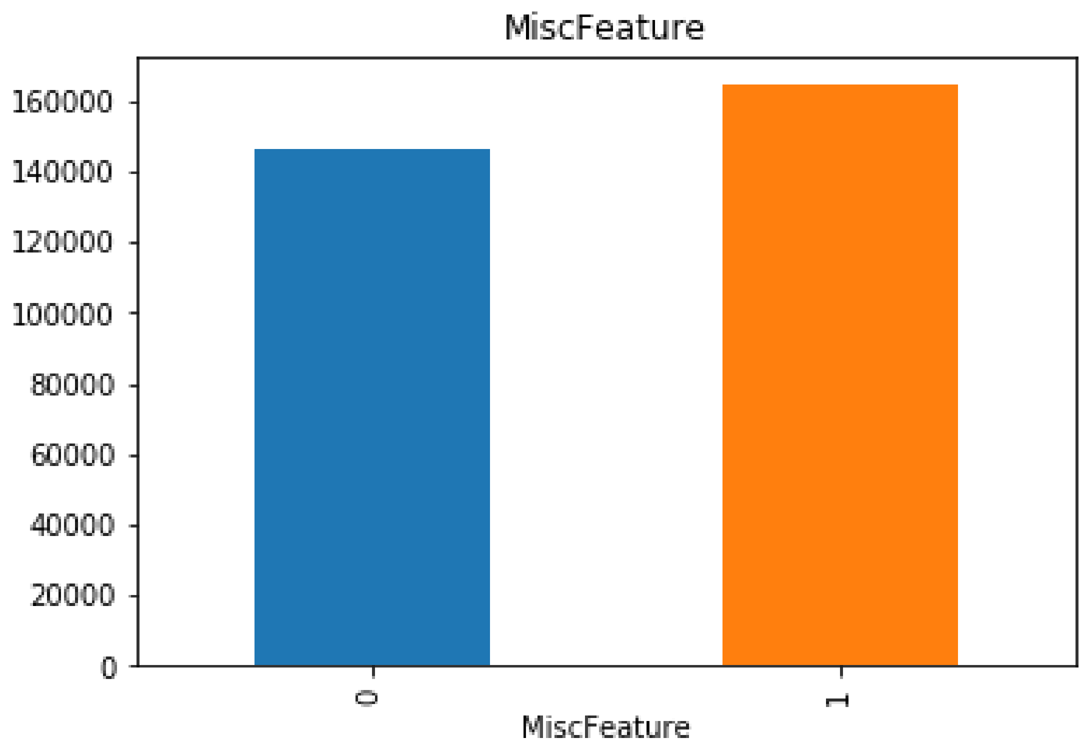
GarageQual


GarageCond
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
GarageCond


PoolQC
200000
150000
100000
50000
0
0
1
PoolQC


Fence
175000
150000
125000
100000
75000




Here With the relation between the missing values and the dependent variable is clearly visible.So We need to replace these nan values with something meaningful which we will do in the Feature Engineering section

From the above dataset some of the features like Id is not required

In [50]:
```
print("Id of Houses {}".format(len(dataset.Id)))
```

```
Id of Houses 1460
```

## Numerical Variables

In [51]:
```
# list of numerical variables
numerical_features = [feature for feature in dataset.columns if dataset[feature].dtypes != 'O']

print('Number of numerical variables: ', len(numerical_features))

# visualise the numerical variables
dataset[numerical_features].head()
```

```
dataset[numerical_features].head()
```

```
Number of numerical variables:  38
```

Out[51]:

| | Id | MSSubClass | LotFrontage | LotArea | OverallQual | OverallCond | YearBuilt | YearRemodAdd | MasVnrArea | BsmtFinSF1 | BsmtFinSF2 | Bsmt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 60 | 65.0 | 8450 | 7 | 5 | 2003 | 2003 | 196.0 | 706 | 0 | 150 |
| **1** | 2 | 20 | 80.0 | 9600 | 6 | 8 | 1976 | 1976 | 0.0 | 978 | 0 | 284 |
| **2** | 3 | 60 | 68.0 | 11250 | 7 | 5 | 2001 | 2002 | 162.0 | 486 | 0 | 434 |
| **3** | 4 | 70 | 60.0 | 9550 | 7 | 5 | 1915 | 1970 | 0.0 | 216 | 0 | 540 |
| **4** | 5 | 60 | 84.0 | 14260 | 8 | 5 | 2000 | 2000 | 350.0 | 655 | 0 | 490 |

**Temporal Variables(Eg: Datetime Variables)**

From the Dataset we have 4 year variables. We have extract information from the datetime variables like no of years or no of days. One example in this specific scenario can be difference in years between the year the house was built and the year the house was sold. We will be performing this analysis in the Feature Engineering which is the next video.

In [52]:

```
# list of variables that contain year information
year_feature = [feature for feature in numerical_features if 'Yr' in feature or 'Year' in feature]

year_feature
```

Out[52]:

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

In [53]:

```
# let's explore the content of these year variables
for feature in year_feature:
    print(feature, dataset[feature].unique())
```

```
YearBuilt [2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 1965 2005 1962 2006
 1960 1929 1970 1967 1958 1930 2002 1968 2007 1951 1957 1927 1920 1966
 1959 1994 1954 1953 1955 1983 1975 1997 1934 1963 1981 1964 1999 1972
 1921 1945 1982 1998 1956 1948 1910 1995 1991 2009 1950 1961 1977 1985
 1979 1885 1919 1990 1969 1935 1988 1971 1952 1936 1923 1924 1984 1926
 1940 1941 1987 1986 2008 1908 1892 1916 1932 1918 1912 1947 1925 1900
 1980 1989 1992 1949 1880 1928 1978 1922 1996 2010 1946 1913 1937 1942
 1938 1974 1893 1914 1906 1890 1898 1904 1882 1875 1911 1917 1872 1905]
YearRemodAdd [2003 1976 2002 1970 2000 1995 2005 1973 1950 1965 2006 1962 2007 1960
 2001 1967 2004 2008 1997 1959 1990 1955 1983 1980 1966 1963 1987 1964
 1972 1996 1998 1989 1953 1956 1968 1981 1992 2009 1982 1961 1993 1999
```

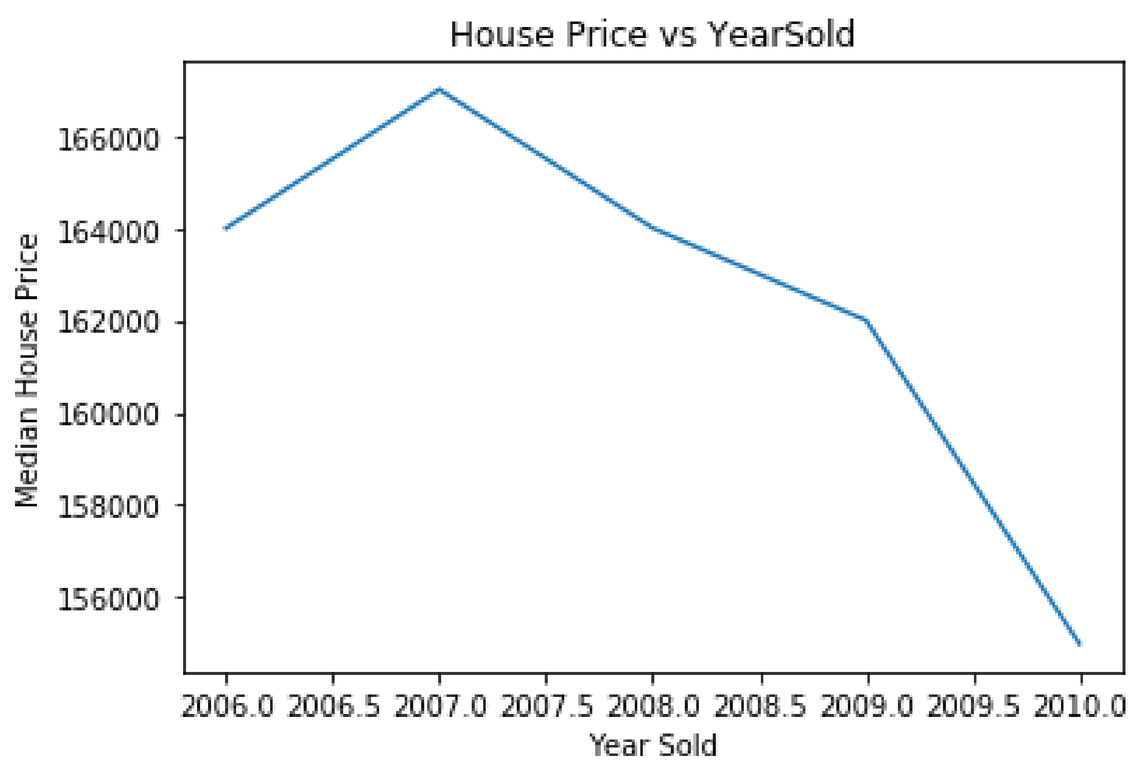
```
 1985 1979 1977 1969 1958 1991 1971 1952 1975 2010 1984 1986 1994 1988
 1954 1957 1951 1978 1974]
GarageYrBlt [2003. 1976. 2001. 1998. 2000. 1993. 2004. 1973. 1931. 1939. 1965. 2005.
 1962. 2006. 1960. 1991. 1970. 1967. 1958. 1930. 2002. 1968. 2007. 2008.
 1957. 1920. 1966. 1959. 1995. 1954. 1953.   nan 1983. 1977. 1997. 1985.
 1963. 1981. 1964. 1999. 1935. 1990. 1945. 1987. 1989. 1915. 1956. 1948.
 1974. 2009. 1950. 1961. 1921. 1900. 1979. 1951. 1969. 1936. 1975. 1971.
 1923. 1984. 1926. 1955. 1986. 1988. 1916. 1932. 1972. 1918. 1980. 1924.
 1996. 1940. 1949. 1994. 1910. 1978. 1982. 1992. 1925. 1941. 2010. 1927.
 1947. 1937. 1942. 1938. 1952. 1928. 1922. 1934. 1906. 1914. 1946. 1908.
 1929. 1933.]
YrSold [2008 2007 2006 2009 2010]
```

In [54]:

```
## Lets analyze the Temporal Datetime Variables
## We will check whether there is a relation between year the house is sold and the sales price

dataset.groupby('YrSold')['SalePrice'].median().plot()
plt.xlabel('Year Sold')
plt.ylabel('Median House Price')
plt.title("House Price vs YearSold")
```

Out[54]:

```
Text(0.5, 1.0, 'House Price vs YearSold')
```



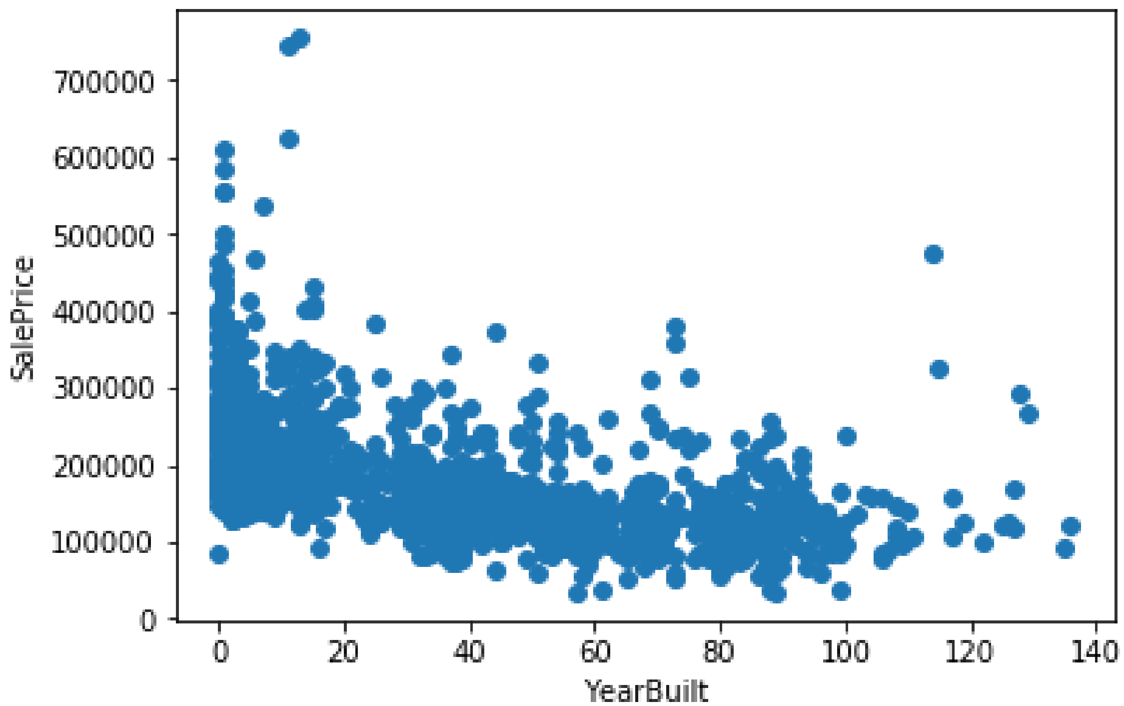

In [55]:

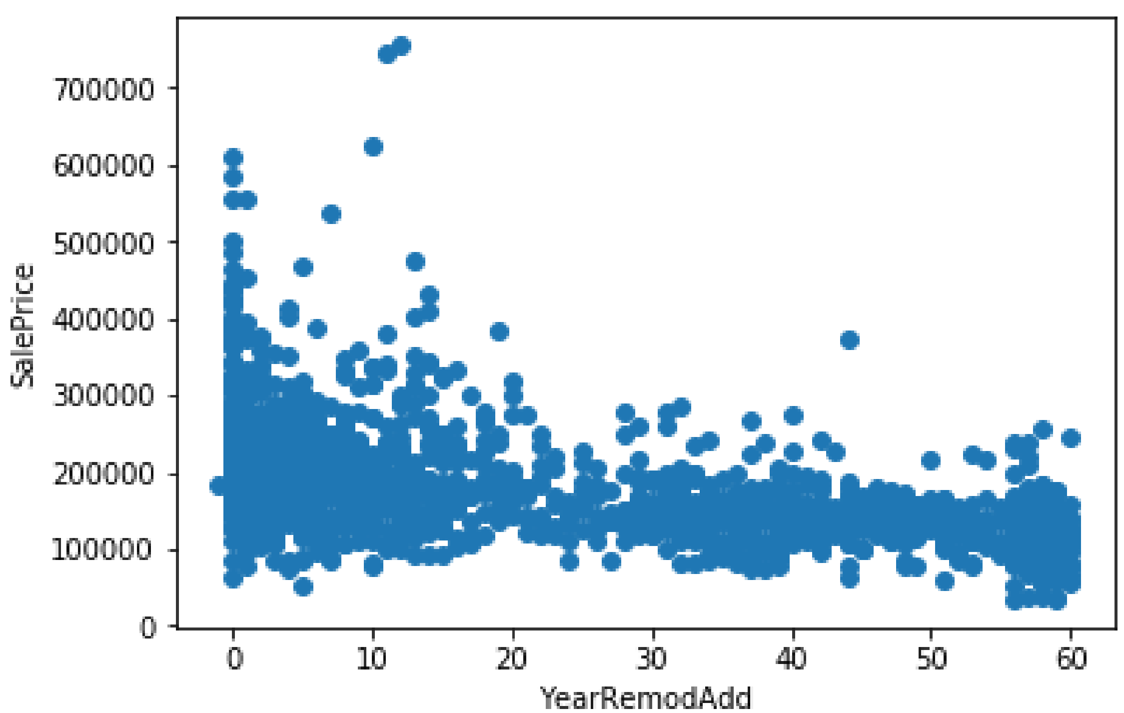
```
year_feature
```

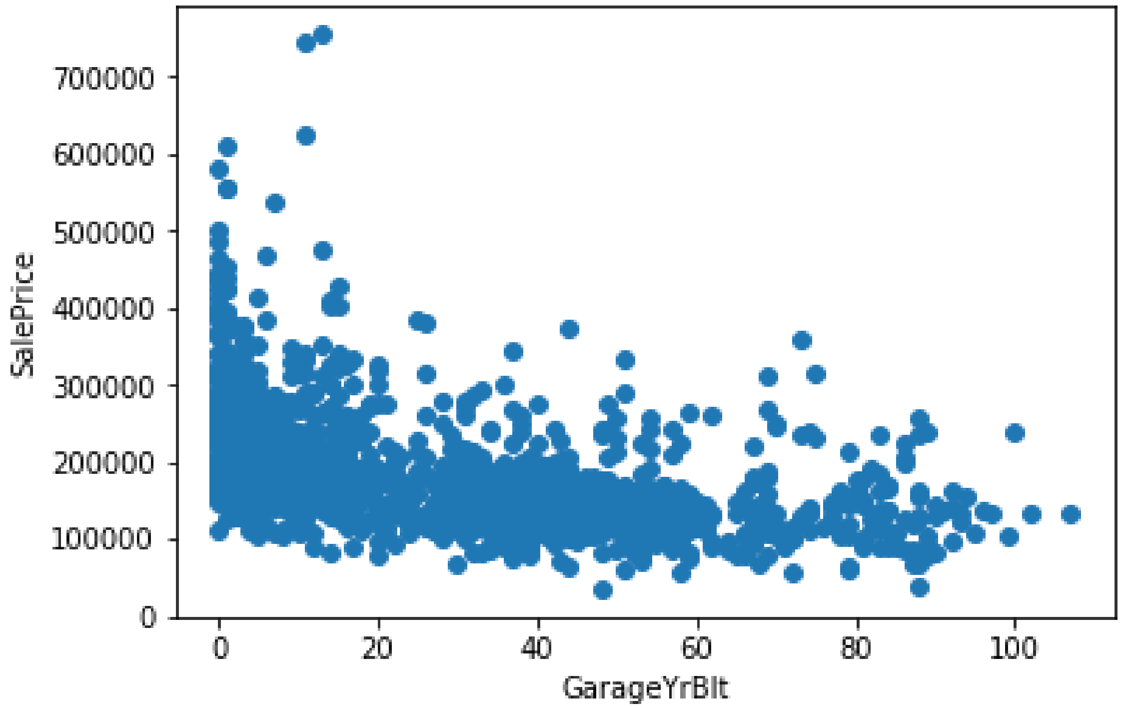
Out[55]:

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

In [56]:
```
## Here we will compare the difference between All years feature with SalePrice

for feature in year_feature:
    if feature!='YrSold':
        data=dataset.copy()
        ## We will capture the difference between year variable and year the house was sold for
        data[feature]=data['YrSold']-data[feature]

        plt.scatter(data[feature],data['SalePrice'])
        plt.xlabel(feature)
        plt.ylabel('SalePrice')
        plt.show()
```






In [57]:
```
## Numerical variables are usually of 2 type
## 1. Continous variable and Discrete Variables

discrete_feature=[feature for feature in numerical_features if len(dataset[feature].unique())<25 and feature not in year_fea
ture+['Id']]
print("Discrete Variables Count: {}".format(len(discrete_feature)))
```

```
Discrete Variables Count: 17
```

In [58]:
```
discrete_feature
```

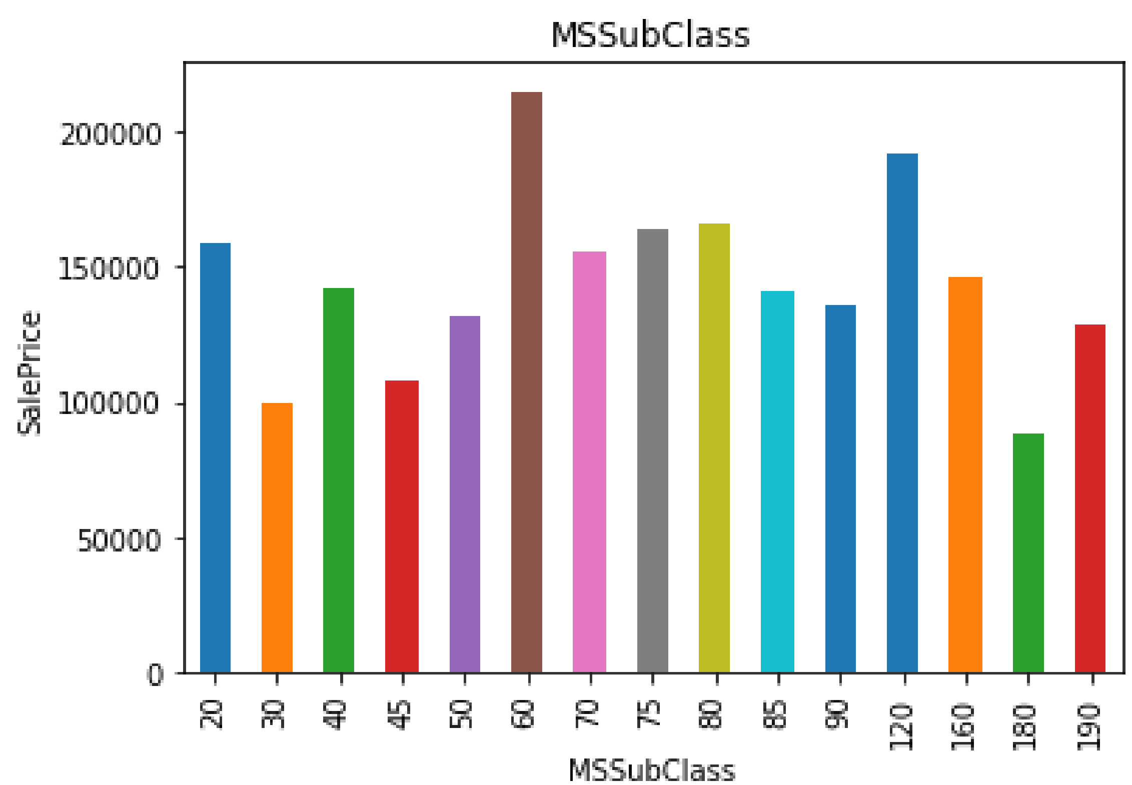
Out[58]:
```
['MSSubClass',
 'OverallQual',
 'OverallCond',
 'LowQualFinSF',
 'BsmtFullBath',
 'BsmtHalfBath',
 'FullBath',
 'HalfBath',
 'BedroomAbvGr',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageCars',
 '3SsnPorch',
 'PoolArea',
 'MiscVal',
 'MoSold']
```

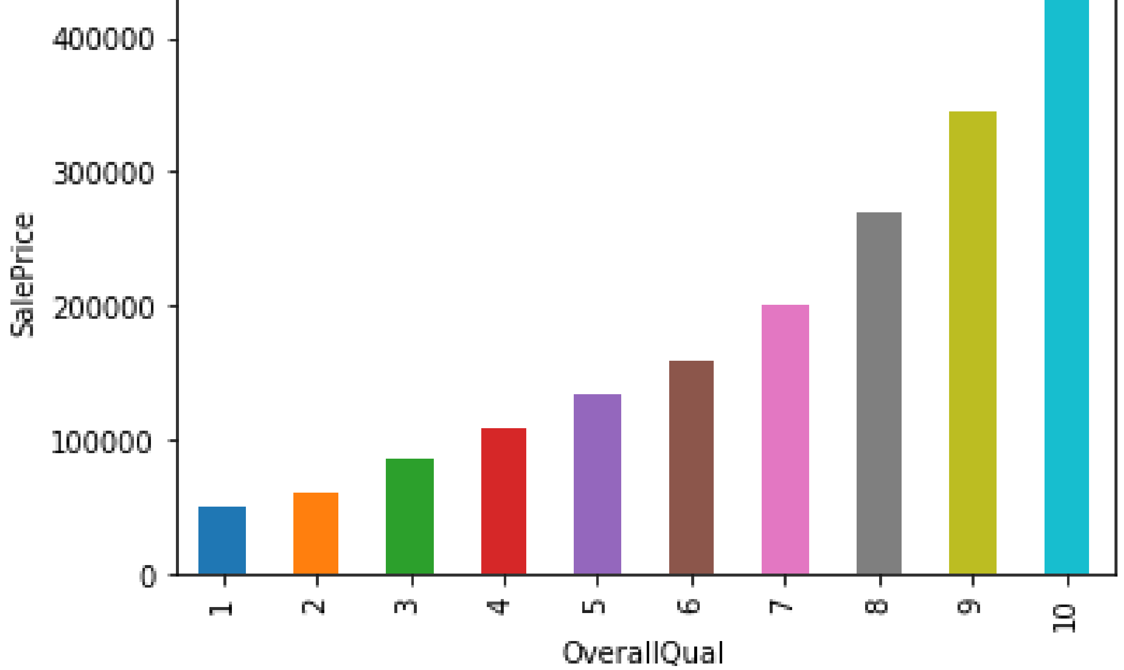
In [59]:
```
dataset[discrete_feature].head()
```

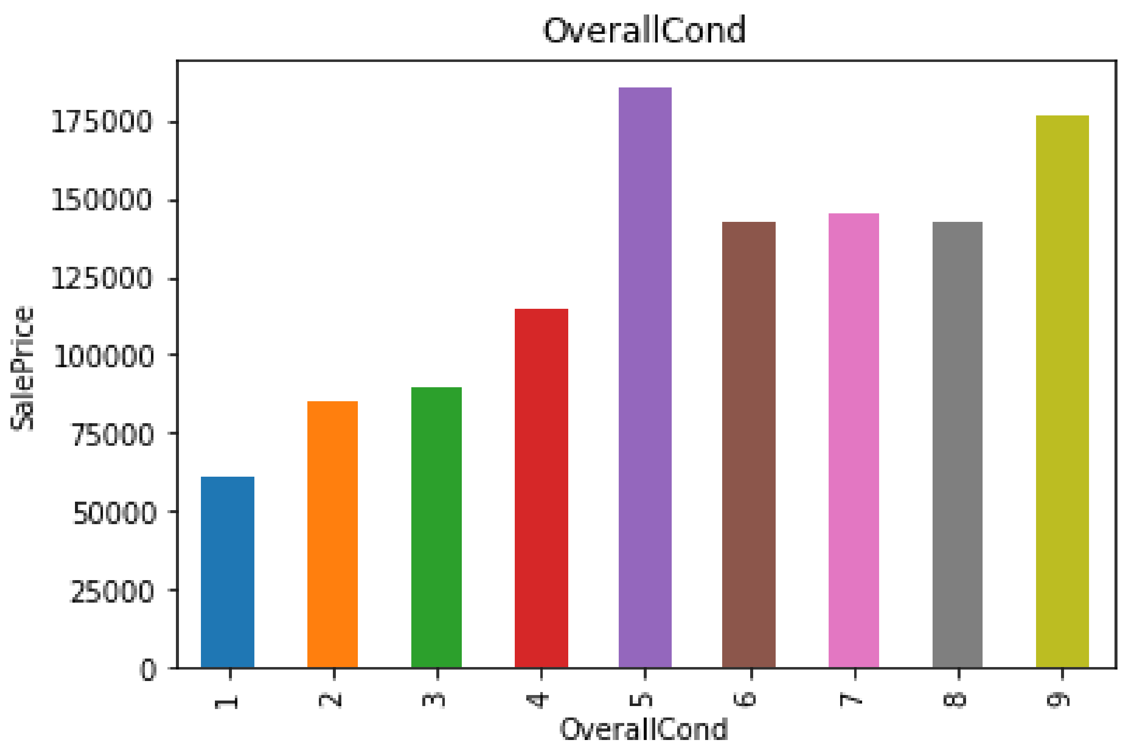
Out[59]:

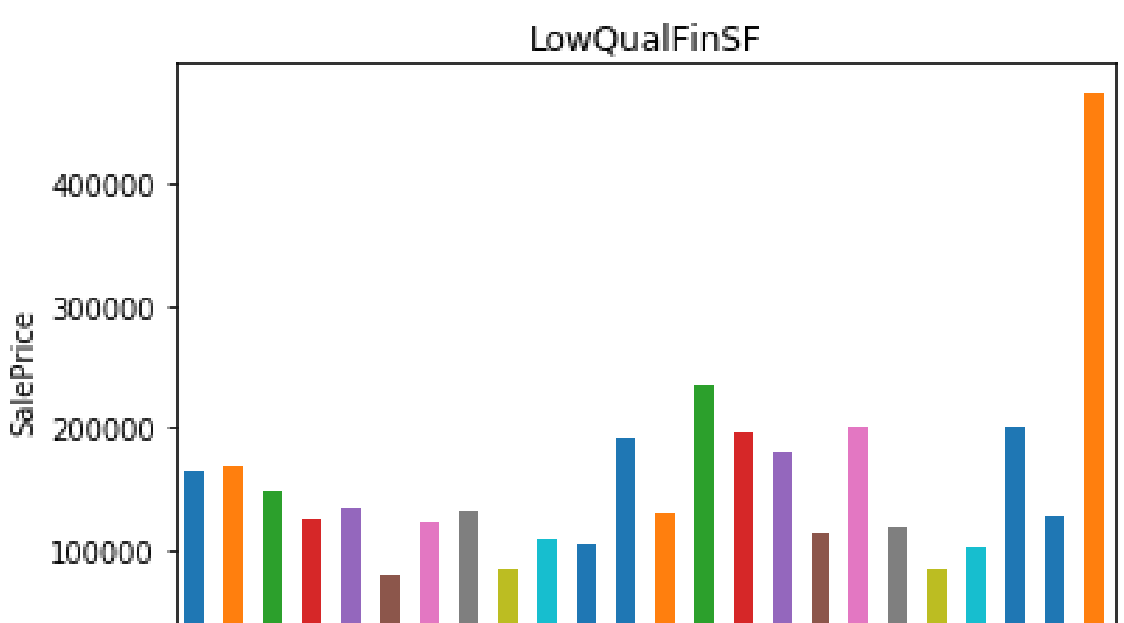
| | MSSubClass | OverallQual | OverallCond | LowQualFinSF | BsmtFullBath | BsmtHalfBath | FullBath | HalfBath | BedroomAbvGr | KitchenAbvGr | T |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 60 | 7 | 5 | 0 | 1 | 0 | 2 | 1 | 3 | 1 | 8 |
| **1** | 20 | 6 | 8 | 0 | 0 | 1 | 2 | 0 | 3 | 1 | 6 |
| **2** | 60 | 7 | 5 | 0 | 1 | 0 | 2 | 1 | 3 | 1 | 6 |
| **3** | 70 | 7 | 5 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | 7 |
| **4** | 60 | 8 | 5 | 0 | 1 | 0 | 2 | 1 | 4 | 1 | 9 |

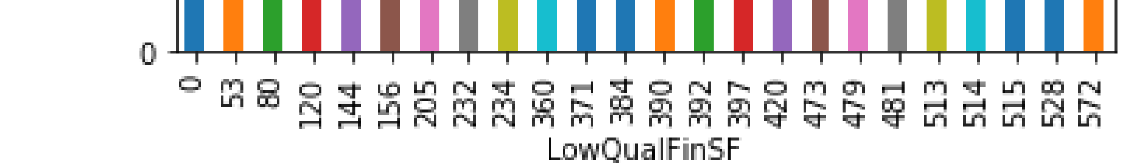
In [60]:
```
## Lets Find the realtionship between them and Sale PRice

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

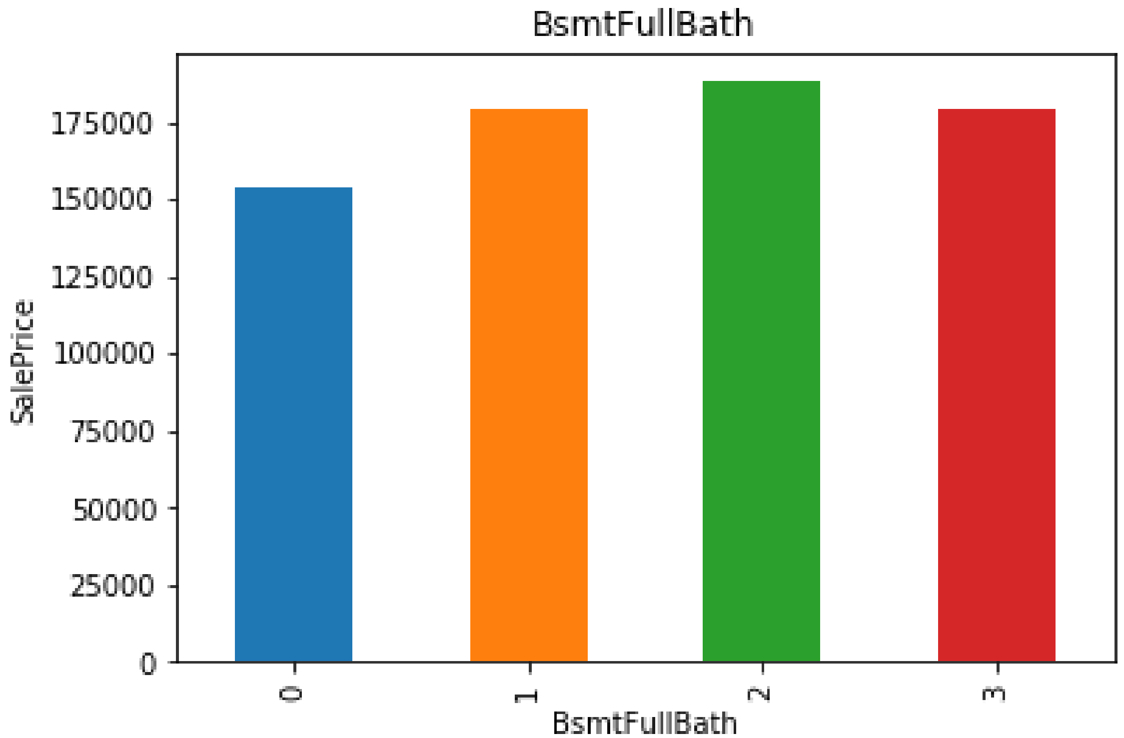


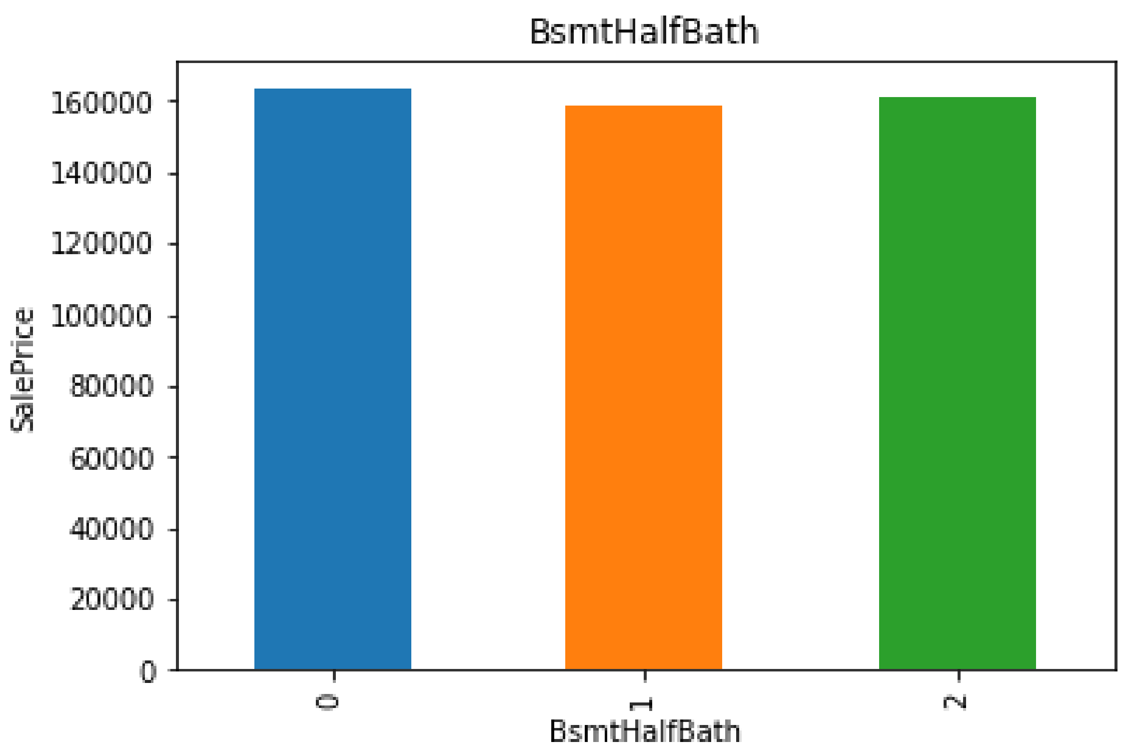


400000
300000
200000
100000
0
SalePrice
1
2
3
4
5
6
7
8
9
10
OverallQual

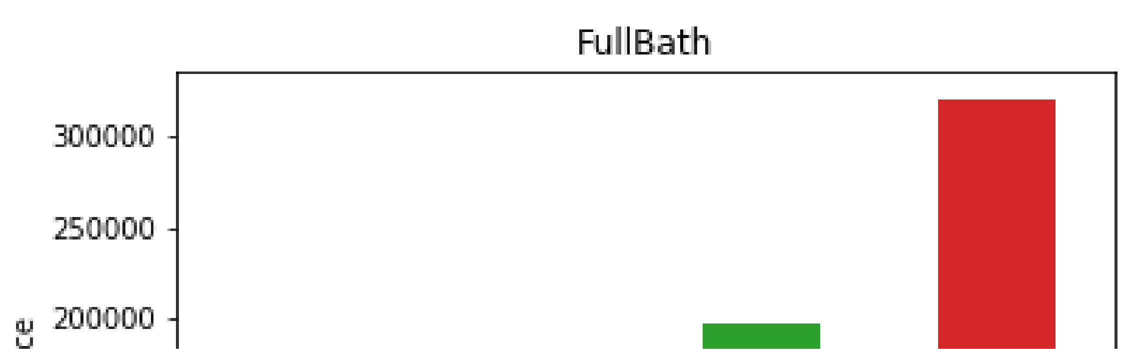

OverallCond
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
1
2
3
4
5
6
7
8
9
OverallCond

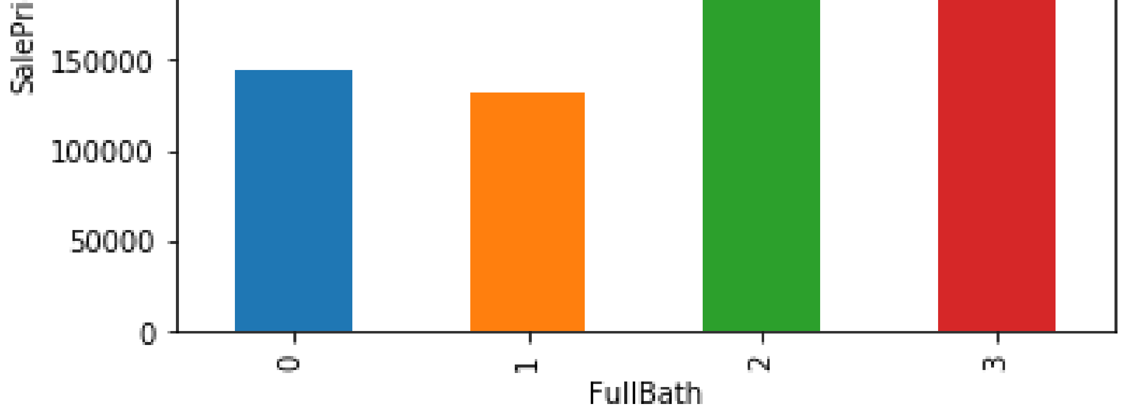

LowQualFinSF
400000
300000
200000
100000
SalePrice

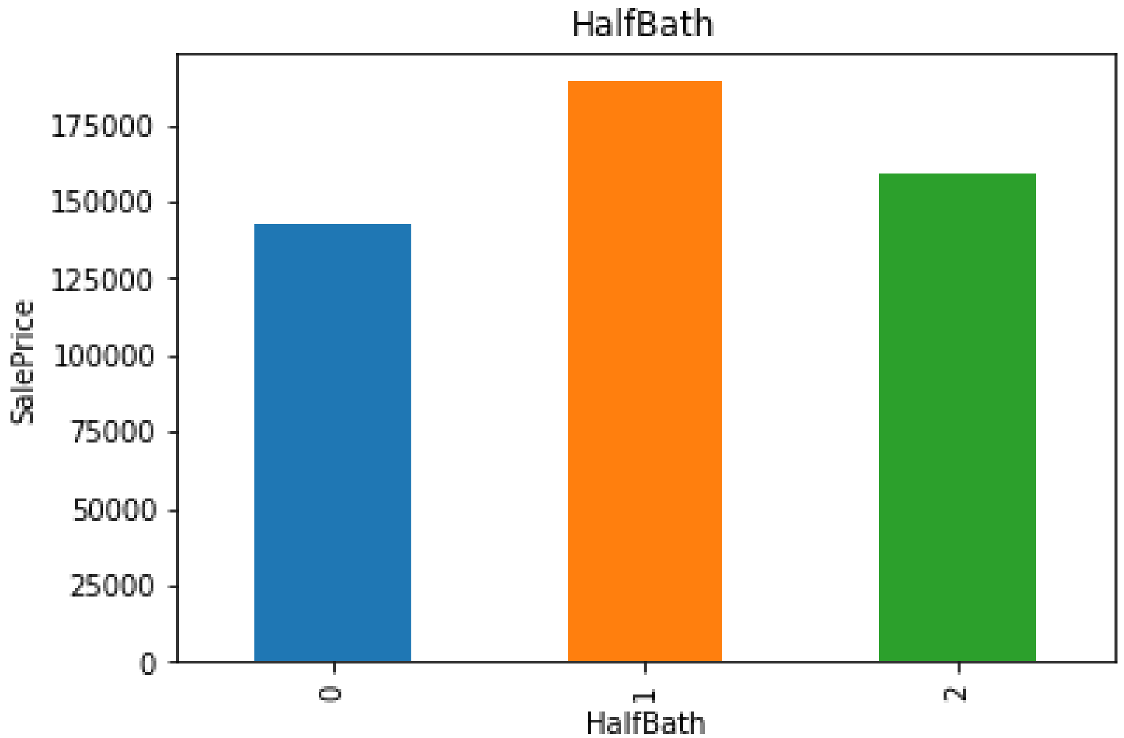

0
0
53
80
120
144
156
205
232
234
360
371
384
390
392
397
420
473
479
481
513
514
515
528
572
LowQualFinSF

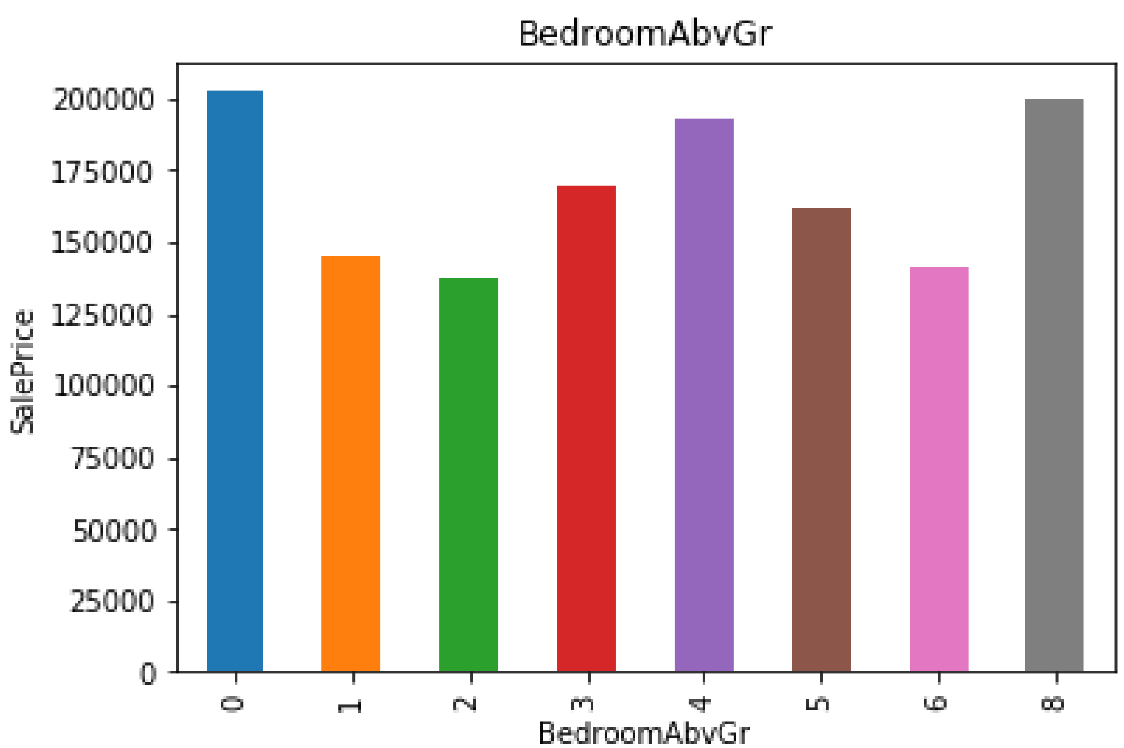

BsmtFullBath
SalePrice
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
3
BsmtFullBath


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

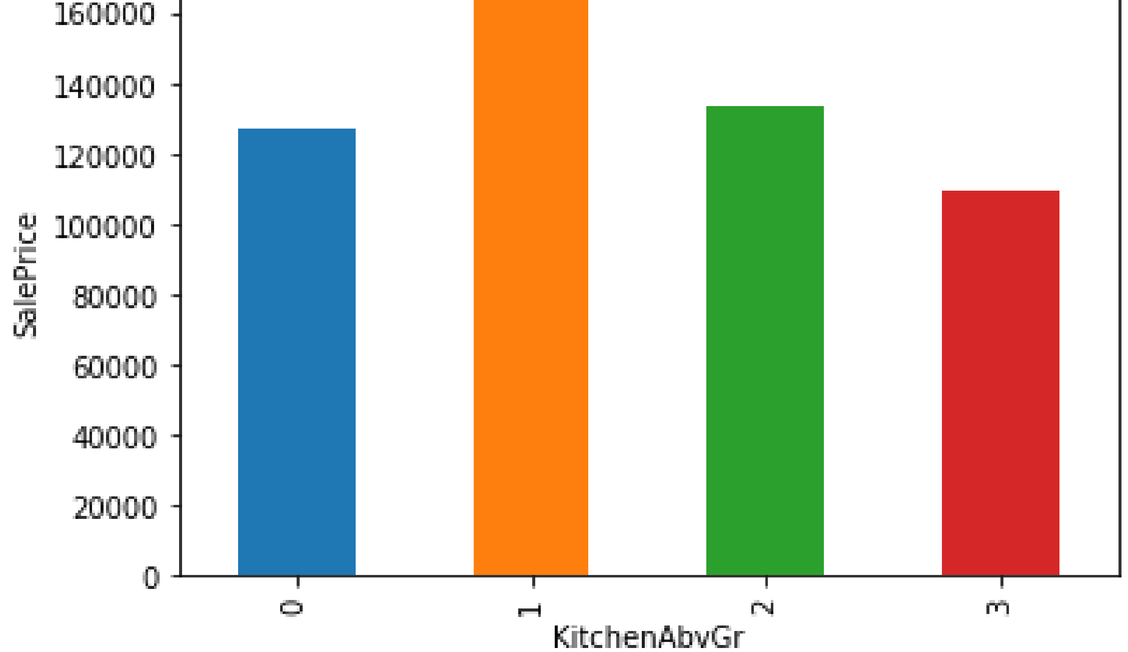

FullBath
300000
250000
200000


SalePri
150000
100000
50000
0
0
1
2
3
FullBath

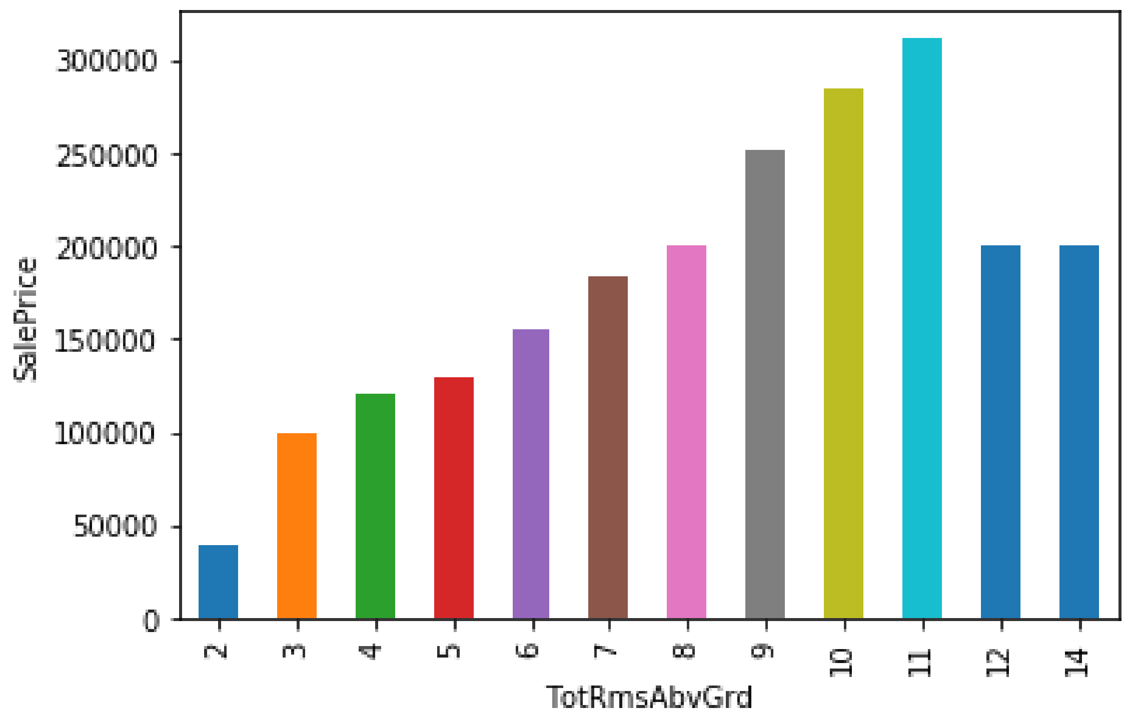

HalfBath
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
0
1
2
HalfBath


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

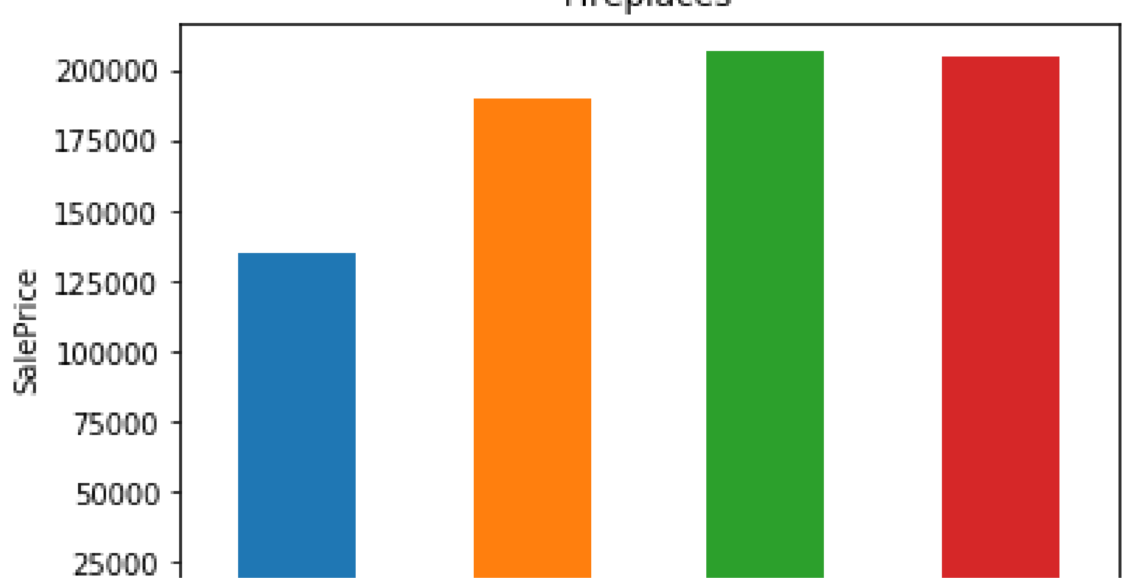

KitchenAbvGr



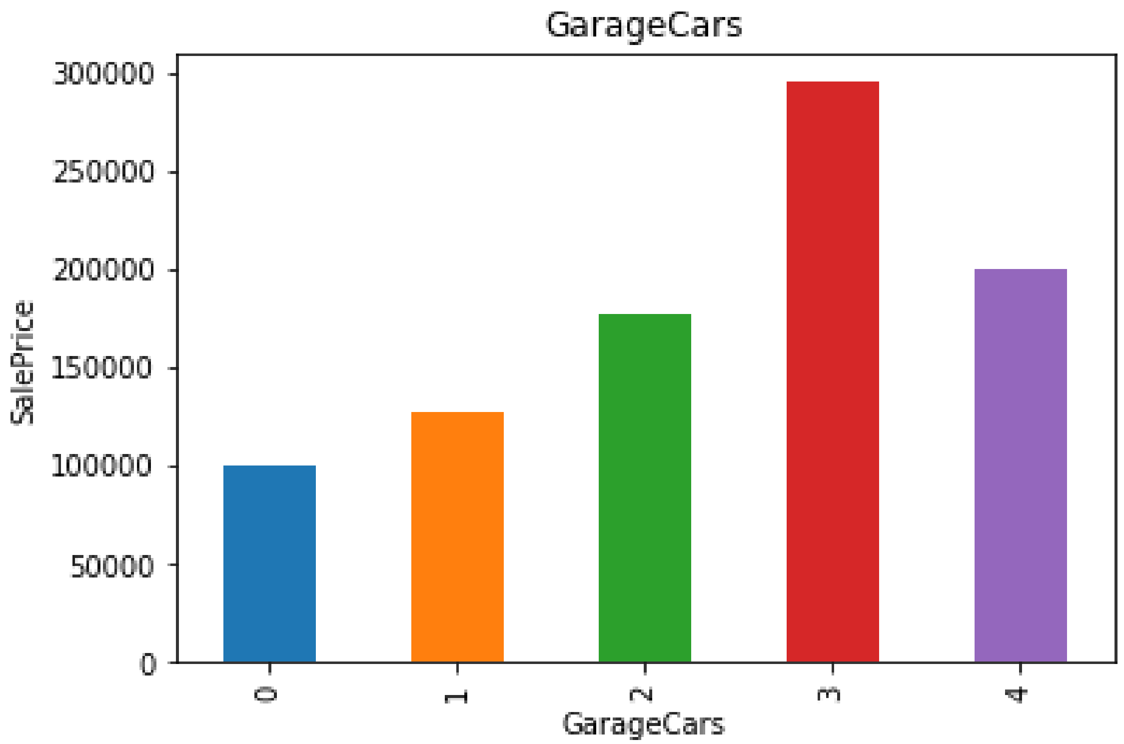

TotRmsAbvGrd

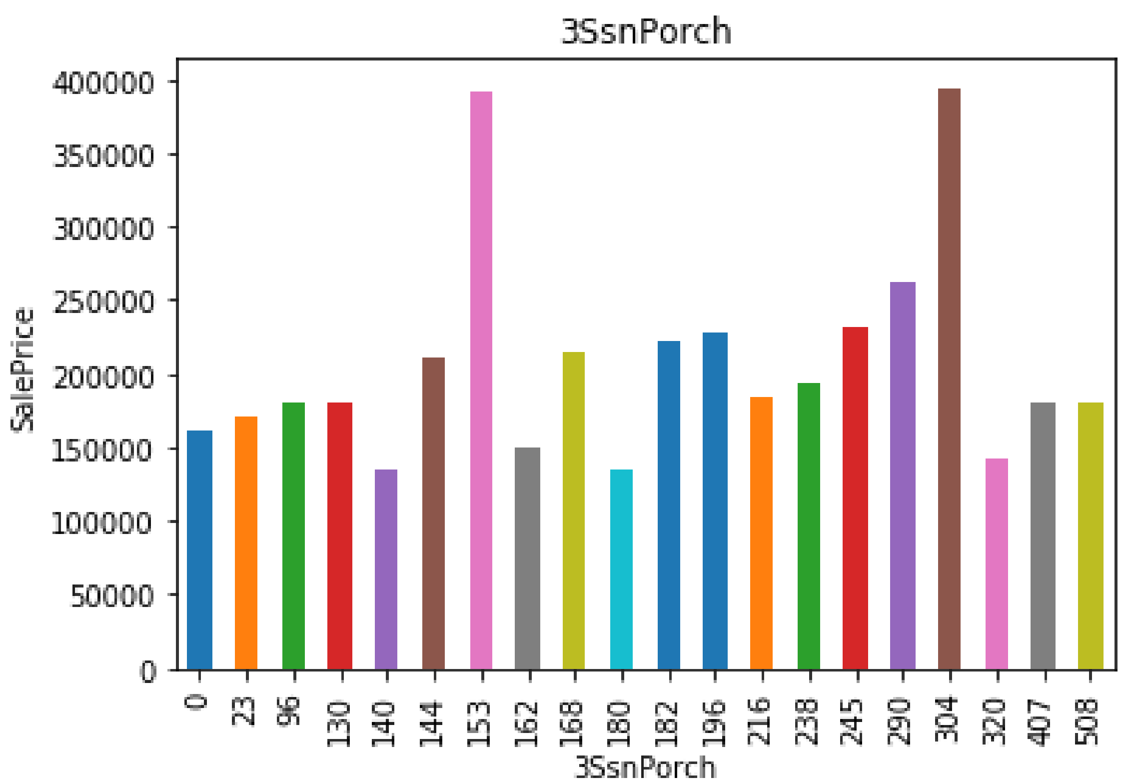


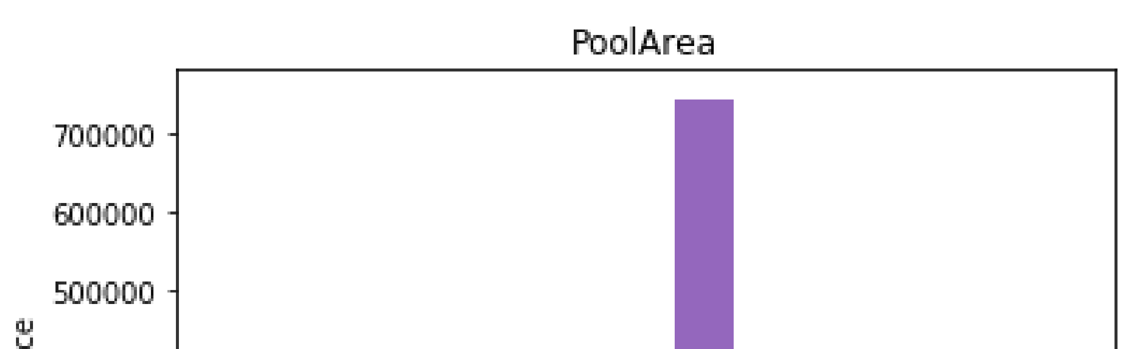

Fireplaces

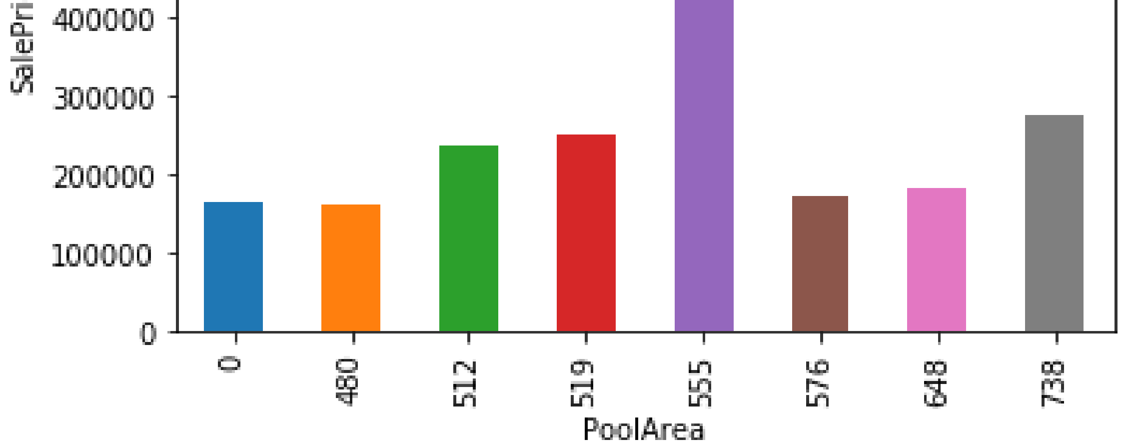



0
0
1
2
3
Fireplaces

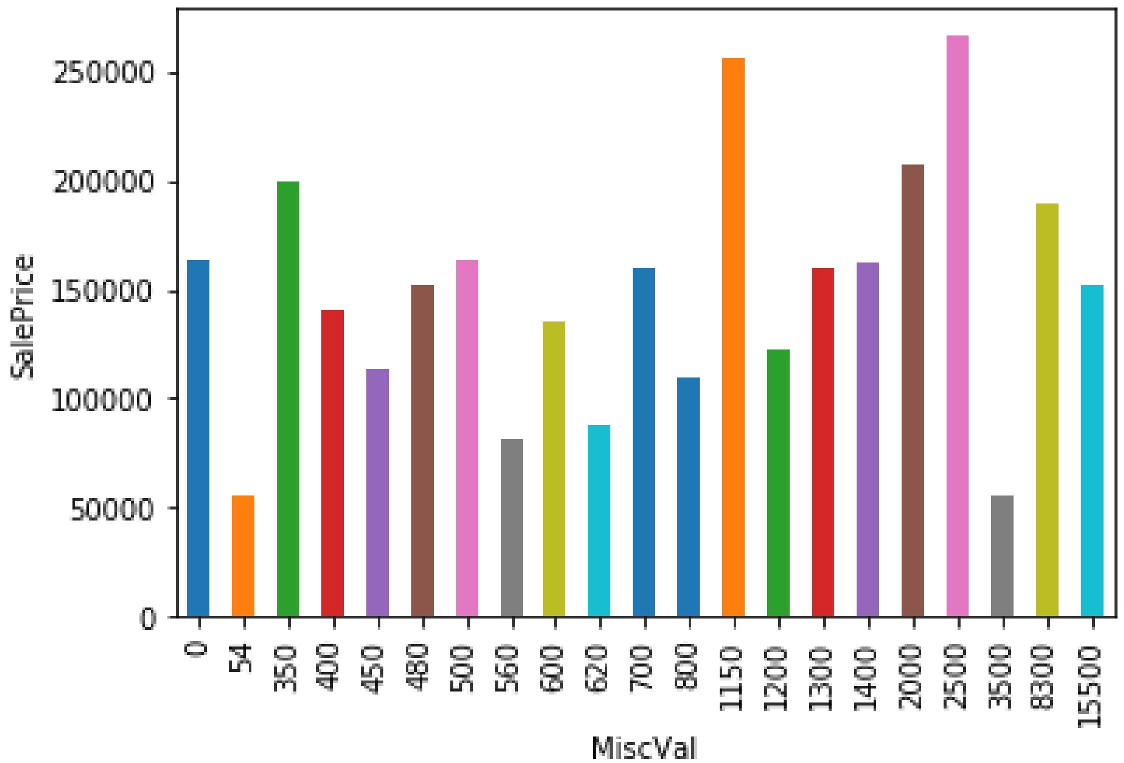

GarageCars
300000
250000
200000
150000
100000
50000
0
SalePrice
0
1
2
3
4
GarageCars


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

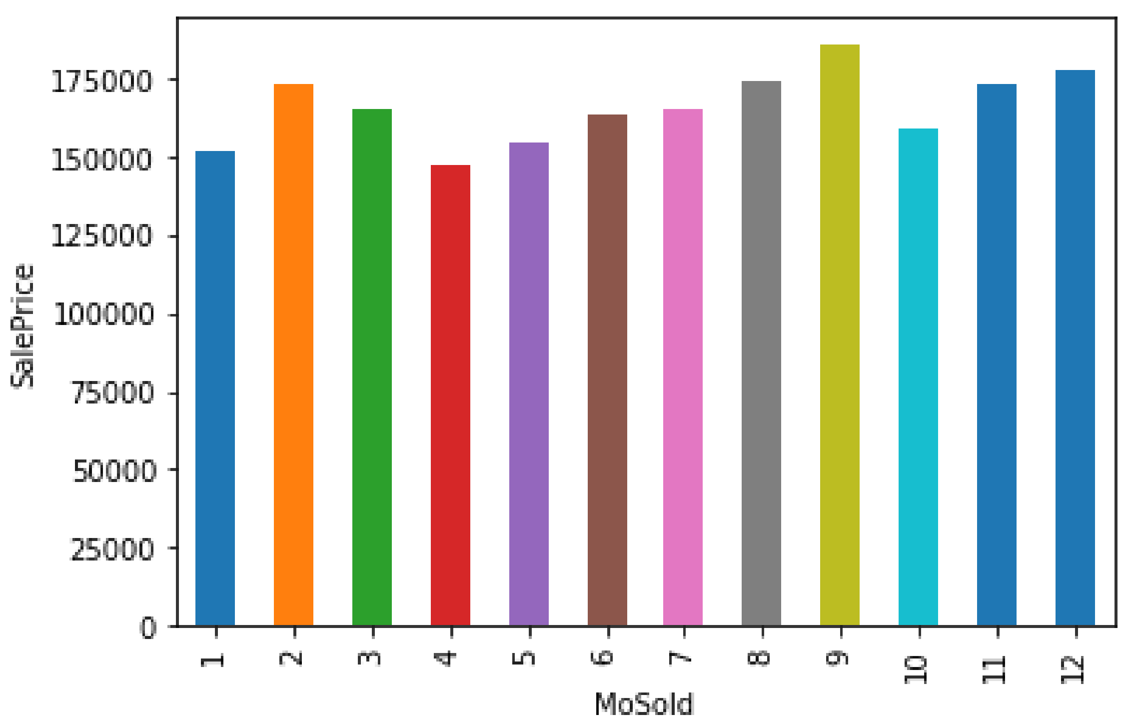

PoolArea
700000
600000
500000




MiscVal




MoSold



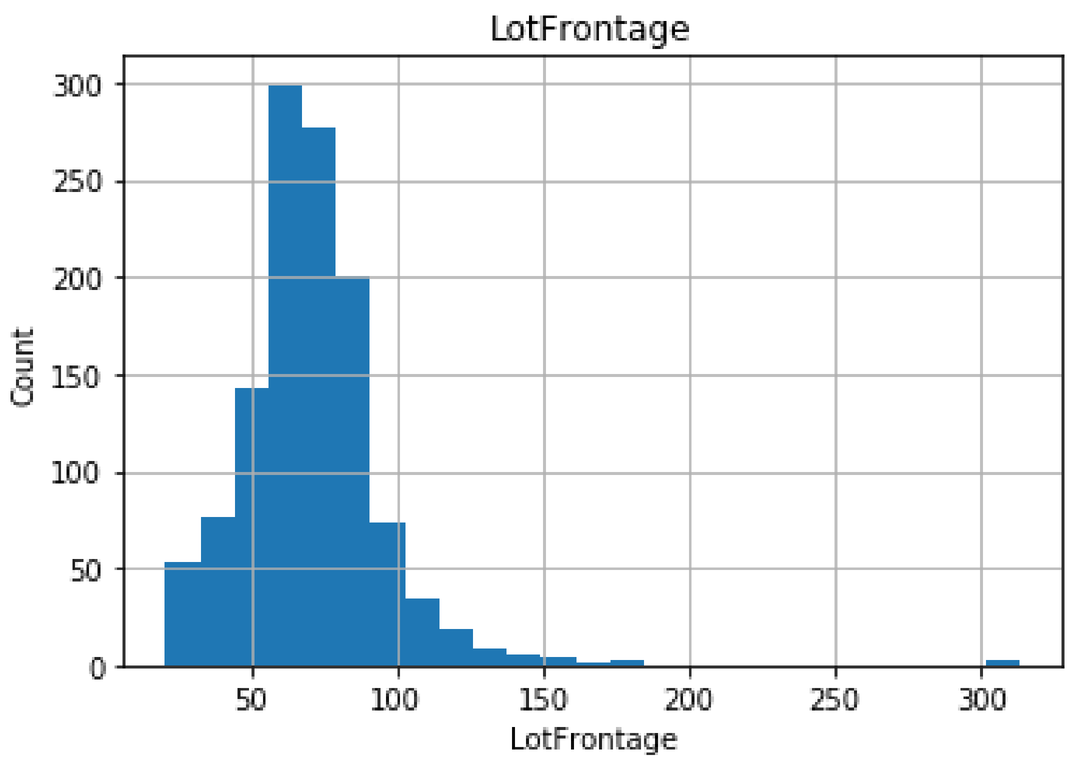
In [42]:
```
## There is a relationship between variable number and SalePrice
```

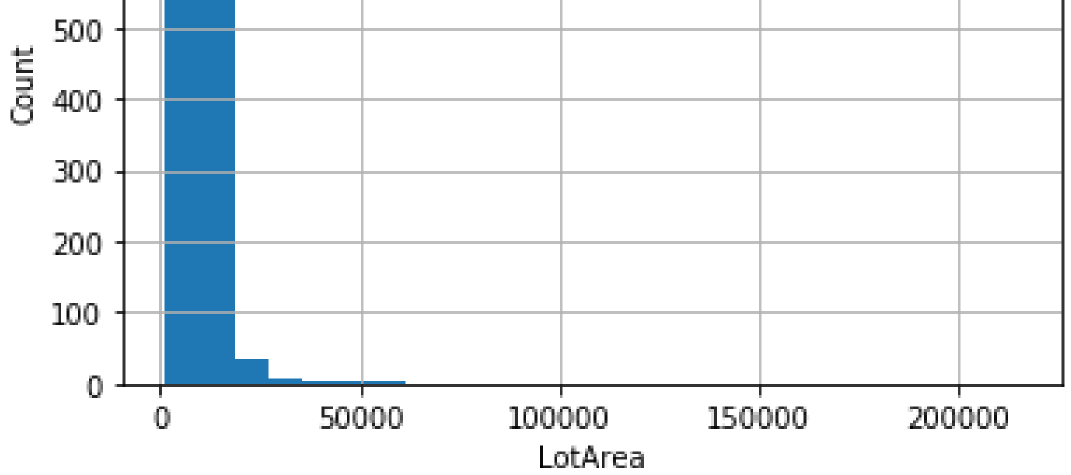
#### Continuous Variable

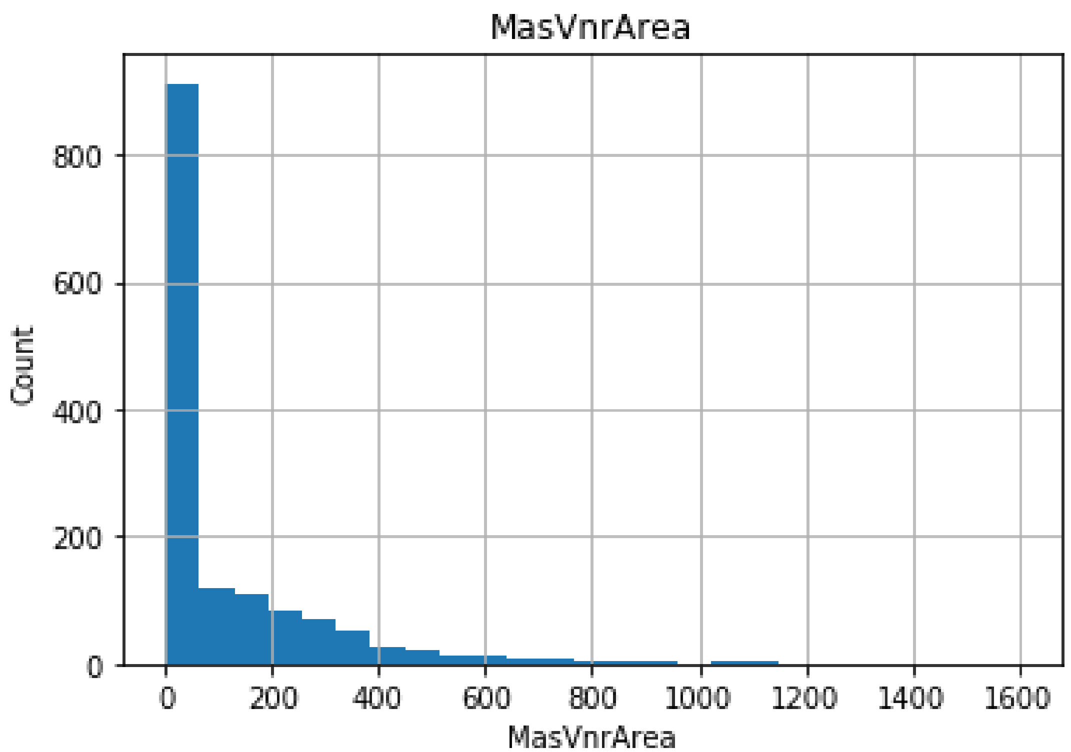
In [61]:
```
continuous_feature=[feature for feature in numerical_features if feature not in discrete_feature+year_feature+['Id']]
print("Continuous feature Count {}".format(len(continuous_feature)))
```

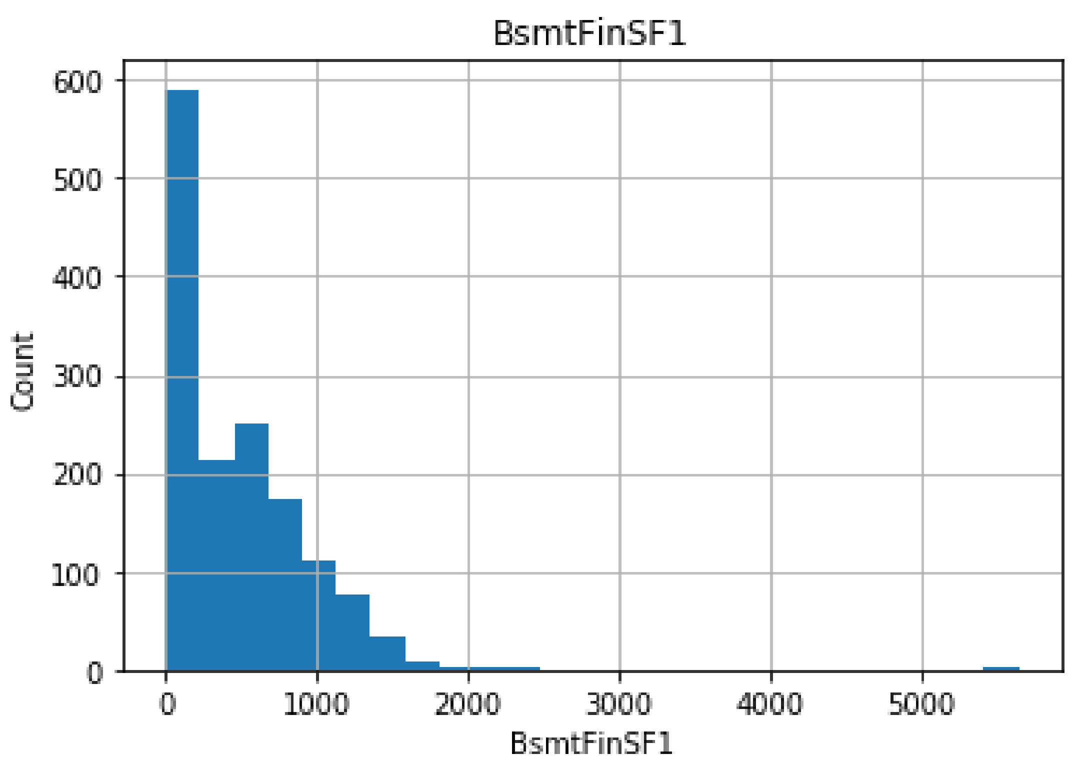
```
Continuous feature Count 16
```

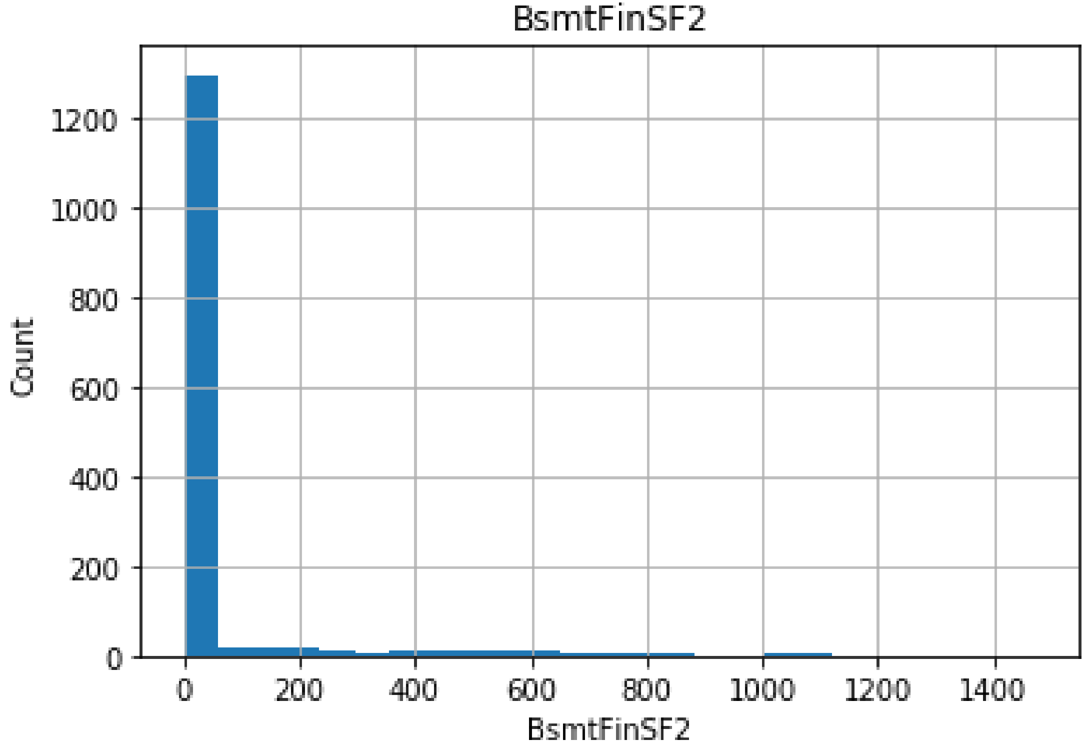
In [62]:
```
## Lets analyse the continuous values by creating histograms to understand the distribution

for feature in continuous_feature:
    data=dataset.copy()
    data[feature].hist(bins=25)
    plt.xlabel(feature)
    plt.ylabel("Count")
    plt.title(feature)
    plt.show()
```

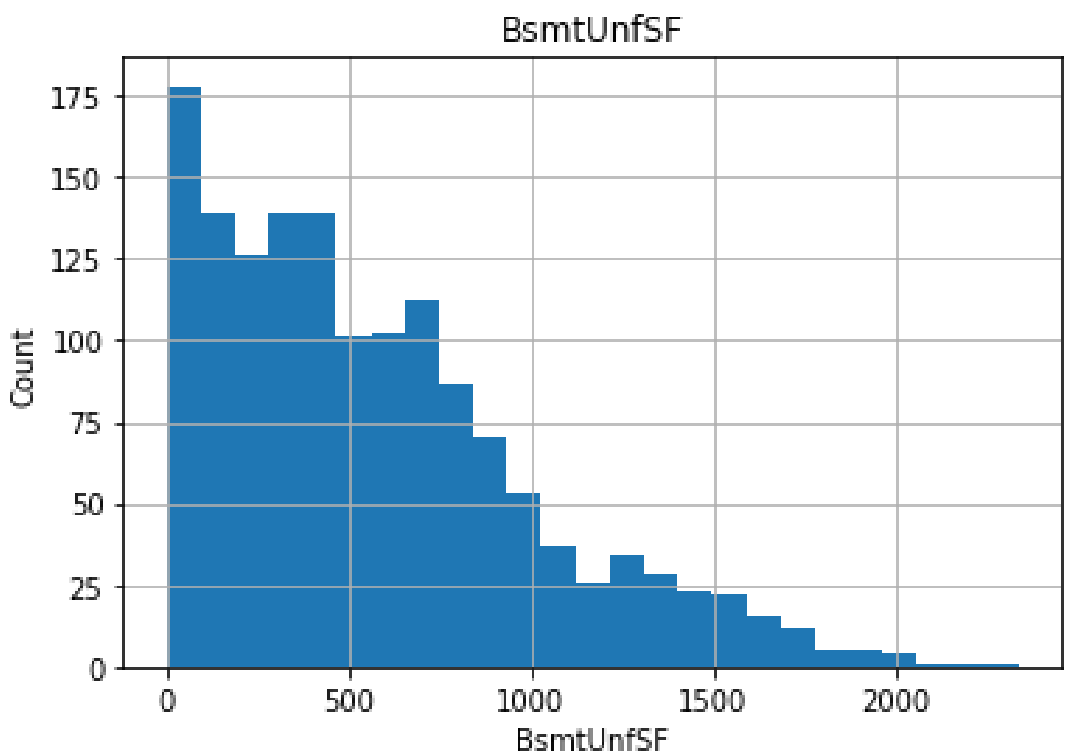


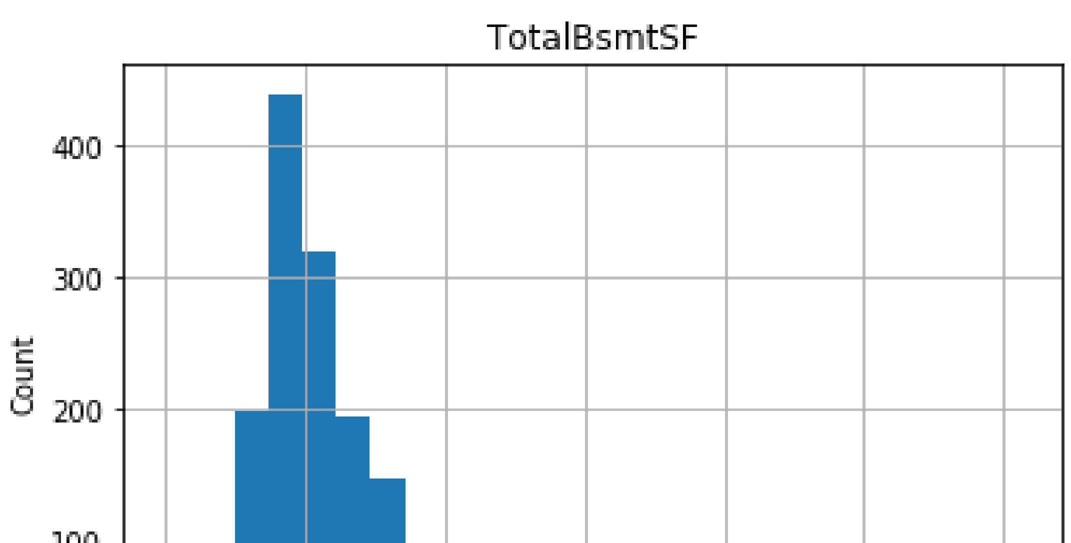


Count
LotArea

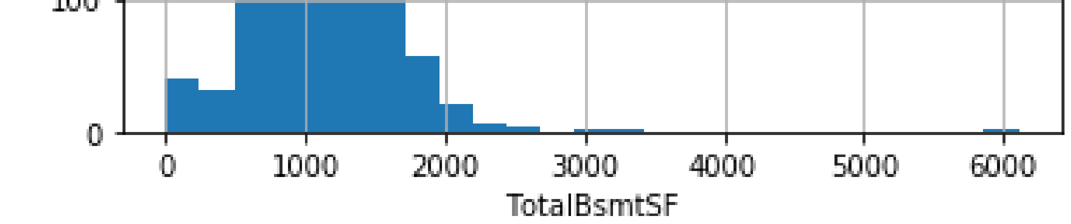

MasVnrArea
Count
MasVnrArea

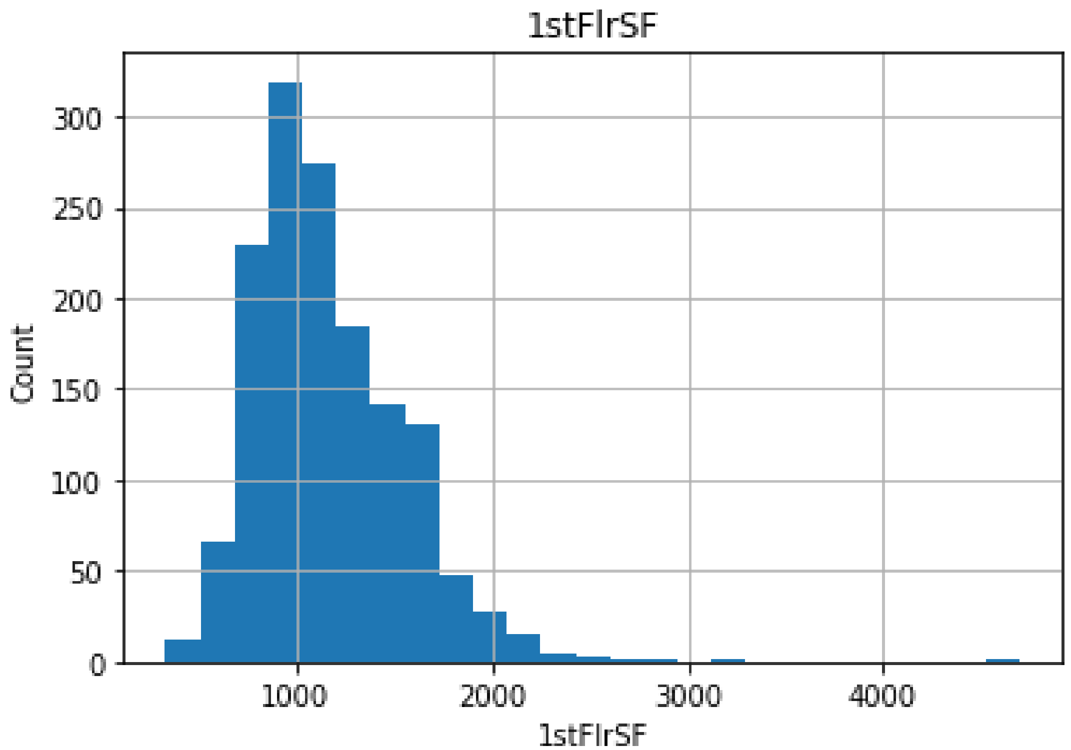

BsmtFinSF1
Count
BsmtFinSF1

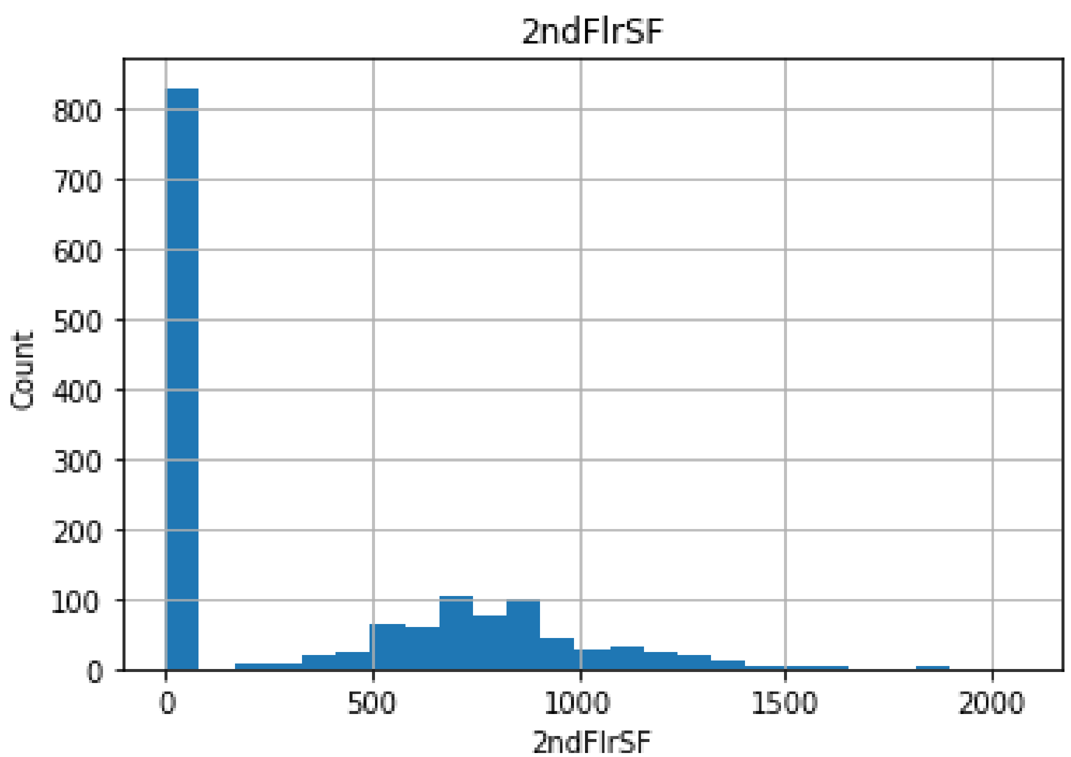

BsmtFinSF2
Count
BsmtFinSF2

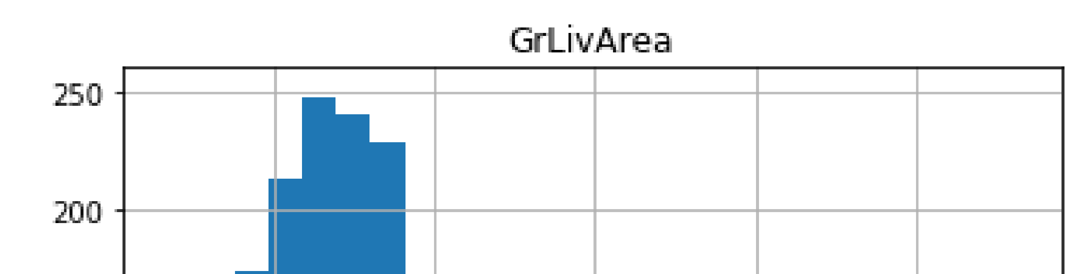

BsmtUnfSF
Count
BsmtUnfSF

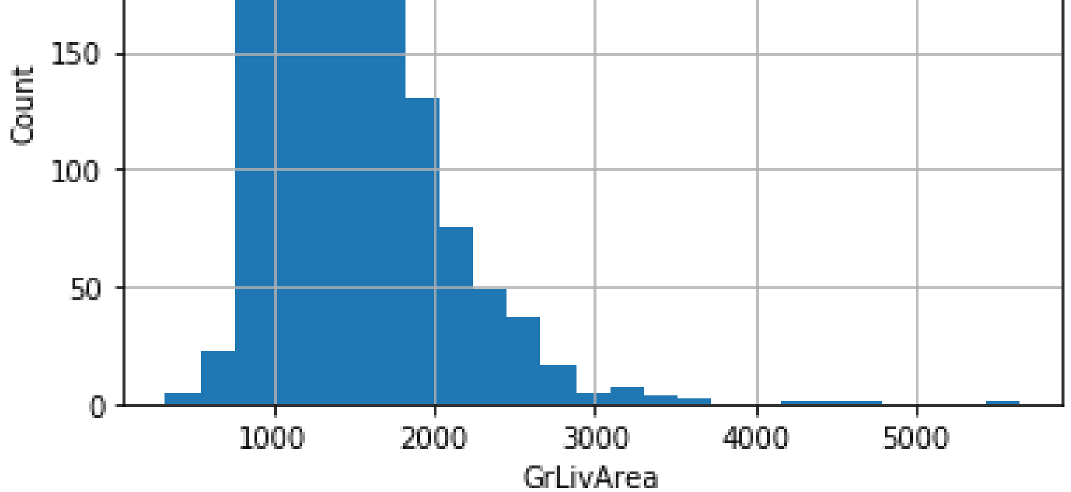

TotalBsmtSF
Count

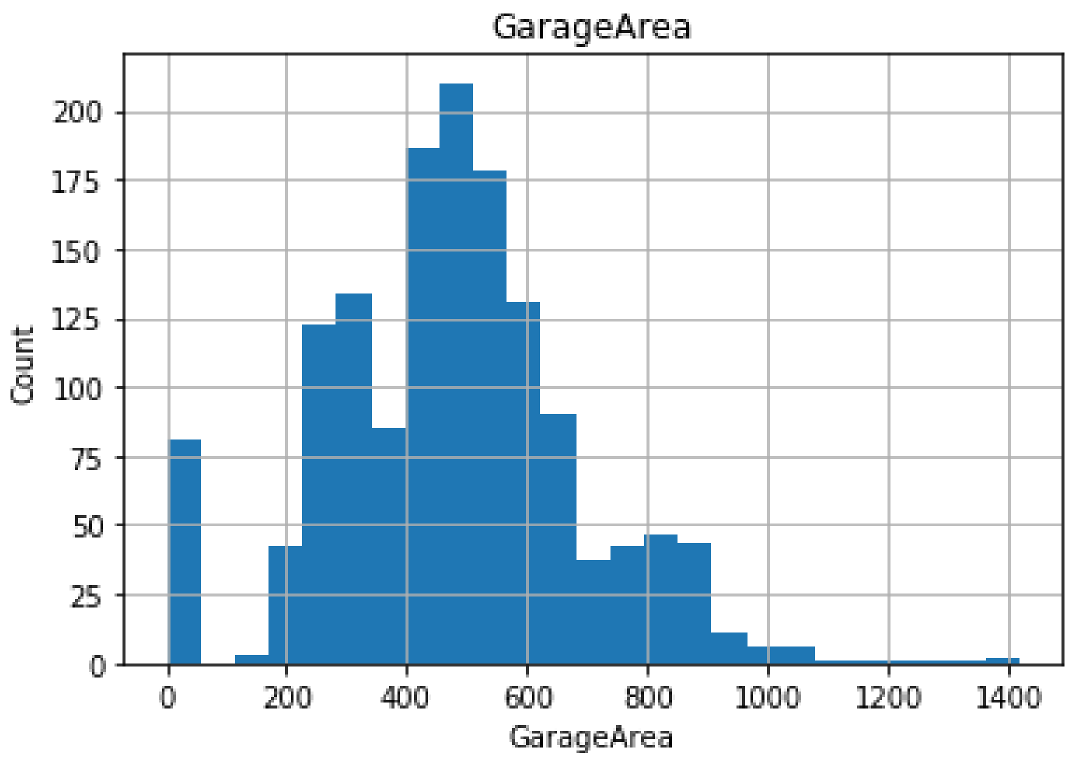

0
0
1000
2000
3000
4000
5000
6000
TotalBsmtSF

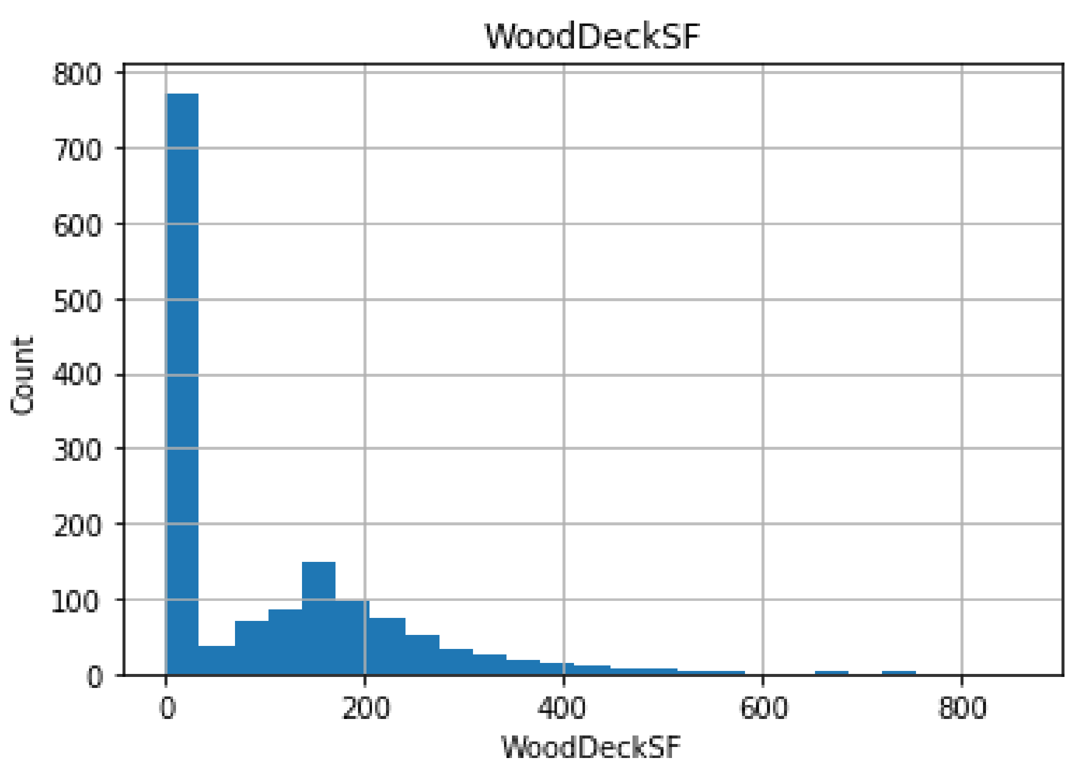

1stFlrSF
Count
0
50
100
150
200
250
300
1000
2000
3000
4000
1stFlrSF

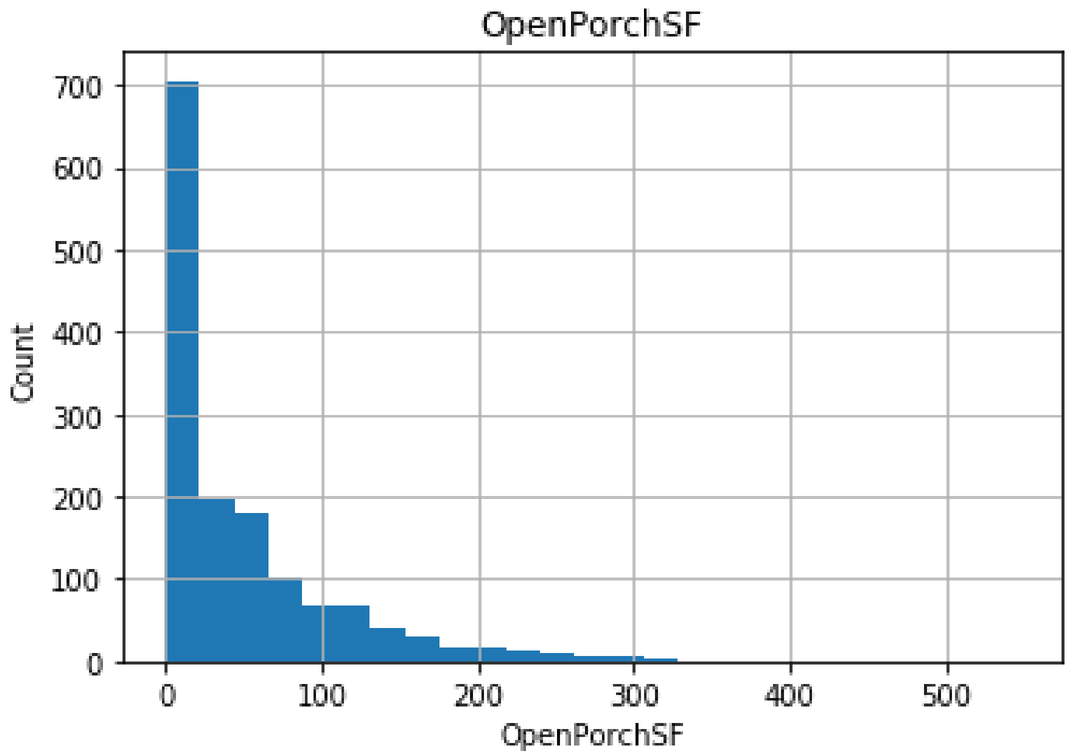

2ndFlrSF
Count
0
100
200
300
400
500
600
700
800
0
500
1000
1500
2000
2ndFlrSF

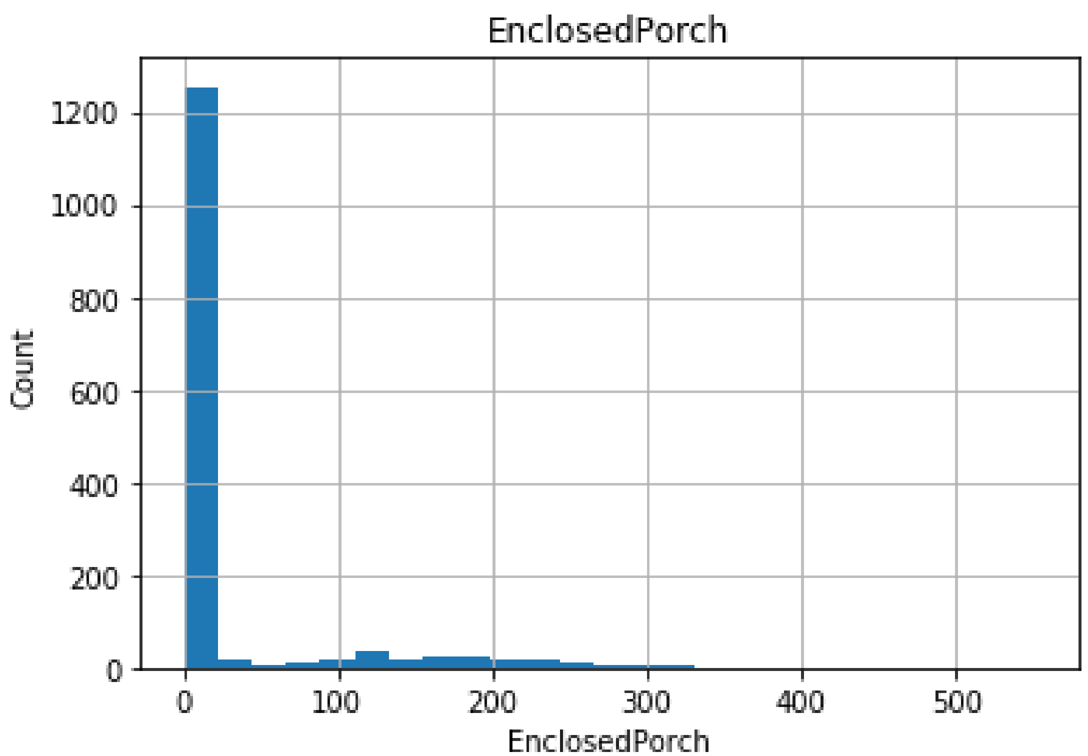

GrLivArea
200
250

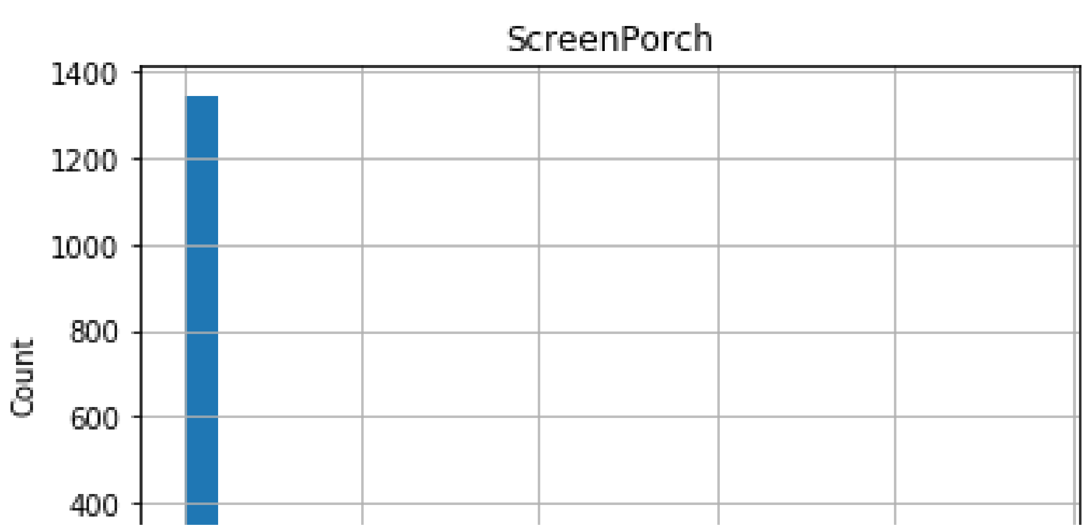

Count
GrLivArea

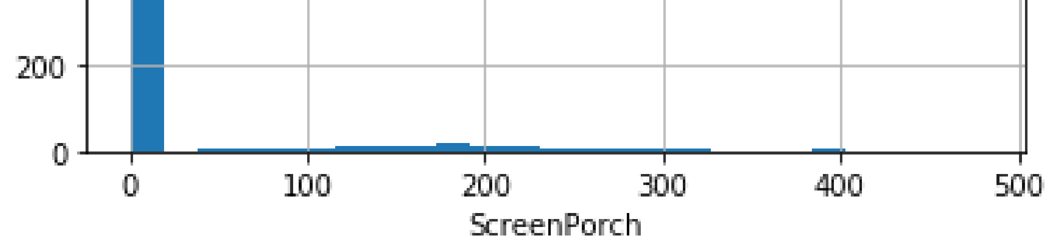

GarageArea
Count
GarageArea

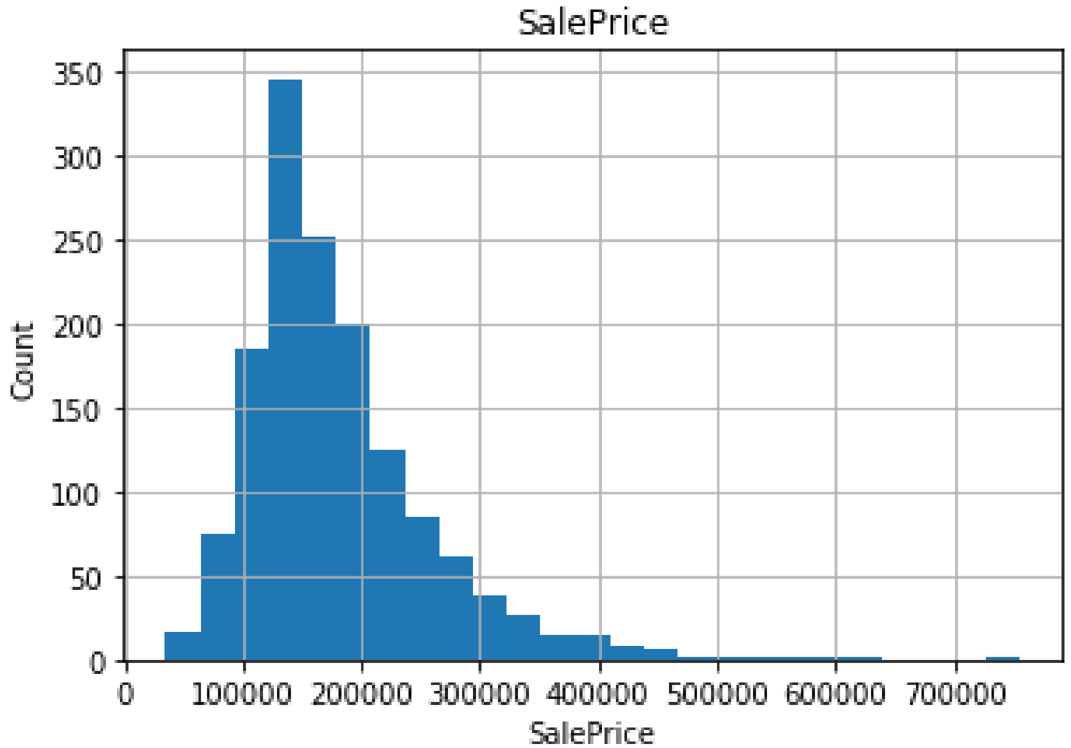

WoodDeckSF
Count
WoodDeckSF


OpenPorchSF
Count
0
100
200
300
400
500
600
700
0
100
200
300
400
500
OpenPorchSF


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

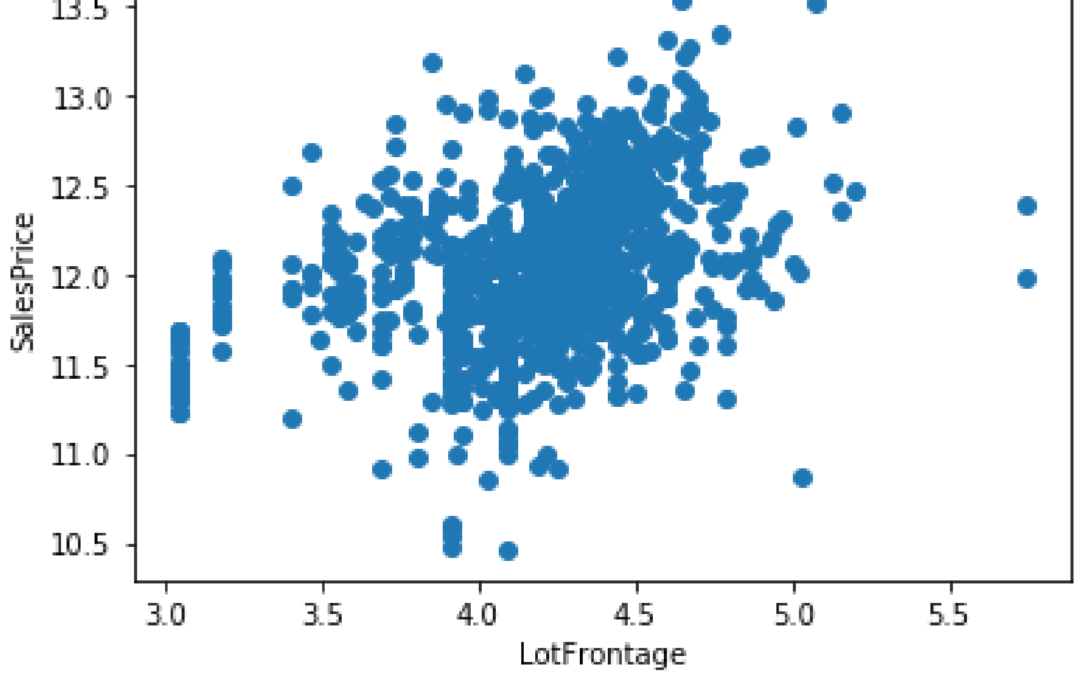

ScreenPorch
Count
400
600
800
1000
1200
1400



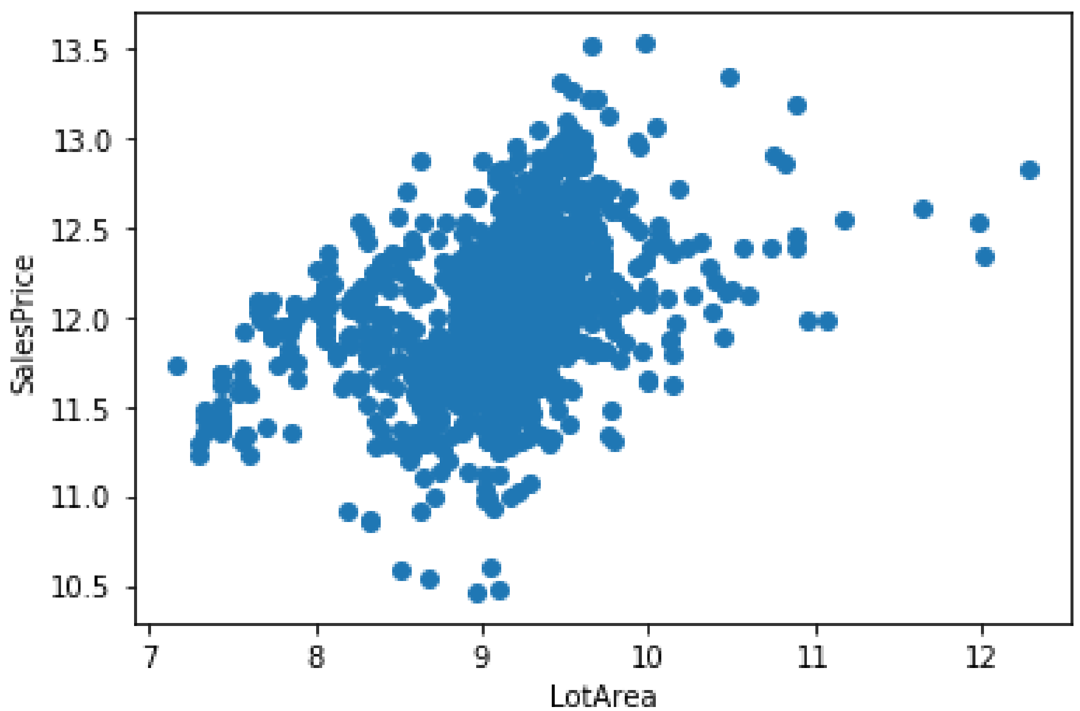




## Exploratory Data Analysis Part 2

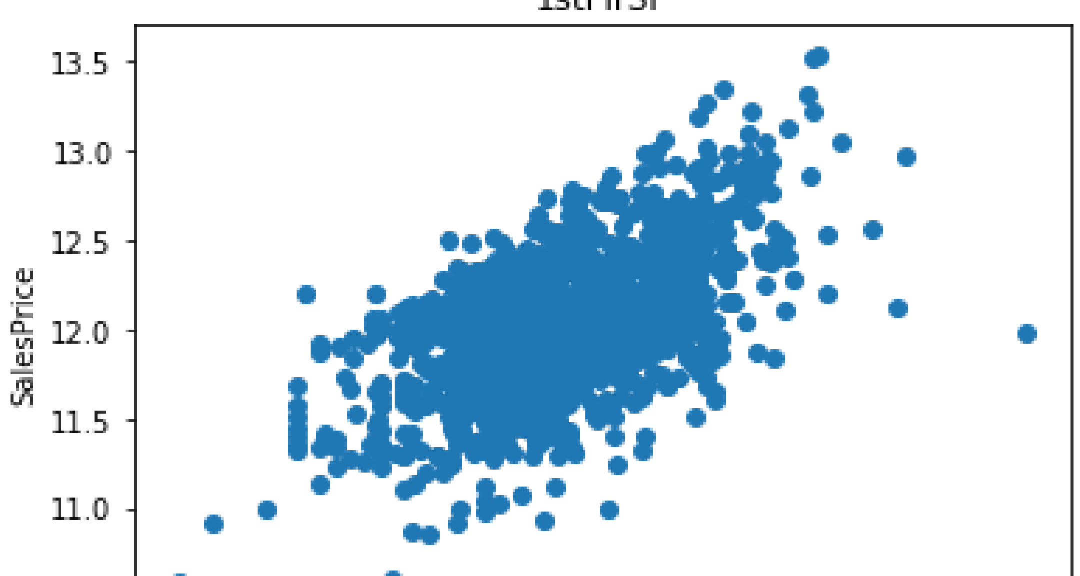
In [63]:
```
## We will be using logarithmic transformation


for feature in continuous_feature:
    data=dataset.copy()
    if 0 in data[feature].unique():
        pass
    else:
        data[feature]=np.log(data[feature])
        data['SalePrice']=np.log(data['SalePrice'])
        plt.scatter(data[feature],data['SalePrice'])
        plt.xlabel(feature)
        plt.ylabel('SalesPrice')
        plt.title(feature)
        plt.show()
```

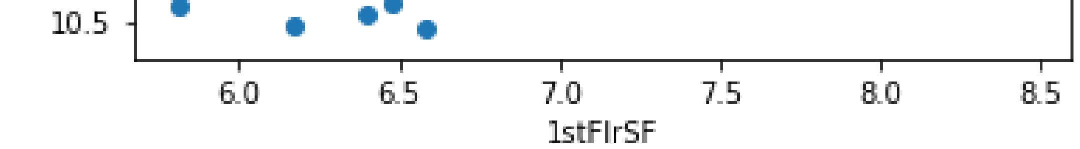


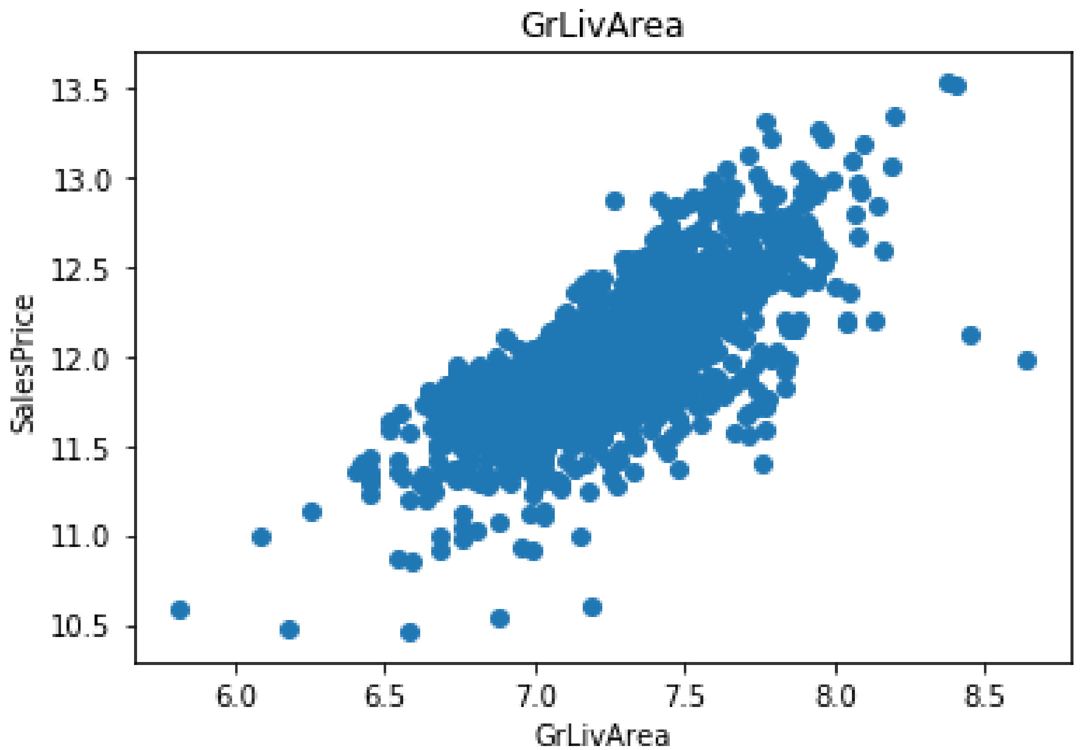

LotArea

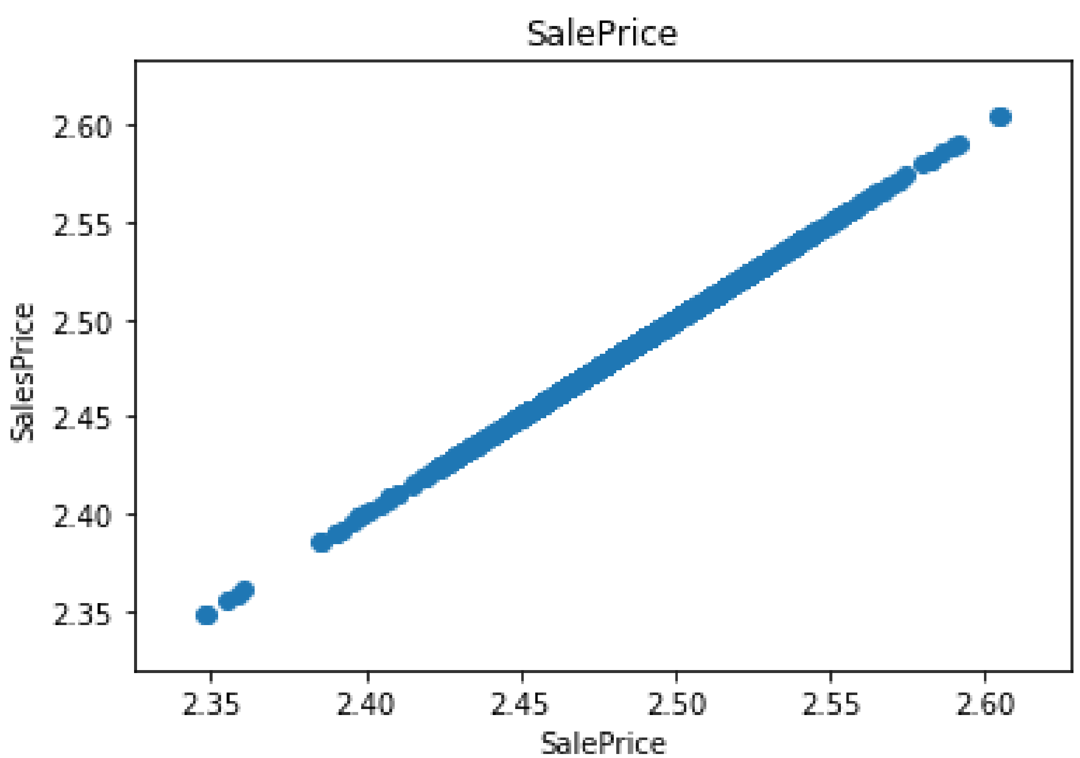



1stFlrSF





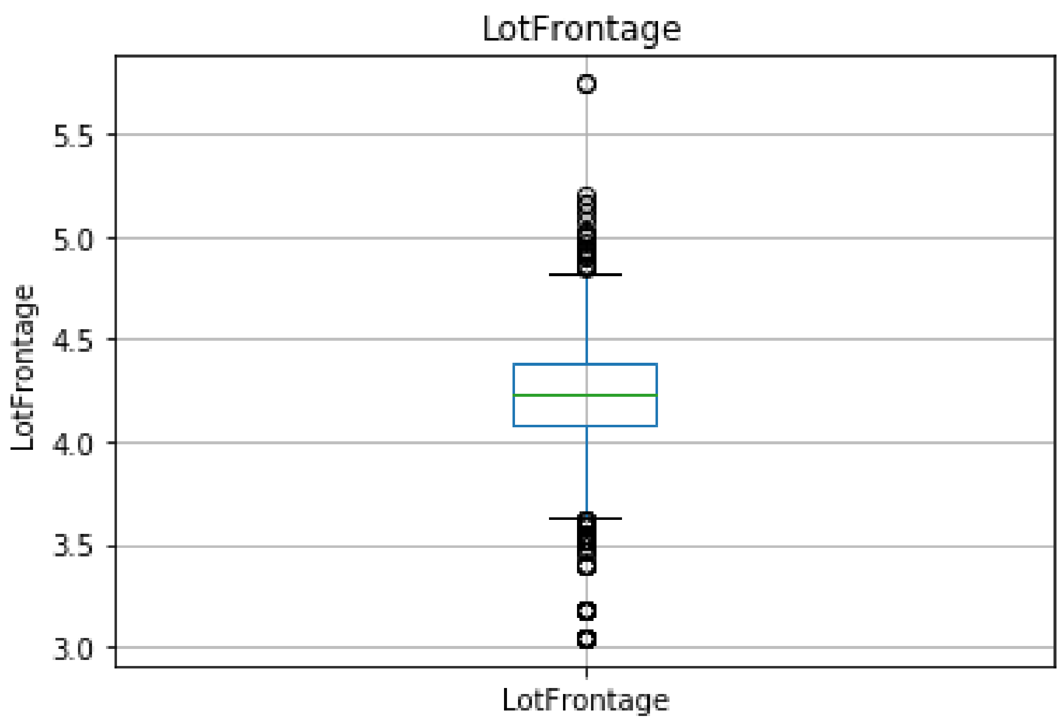



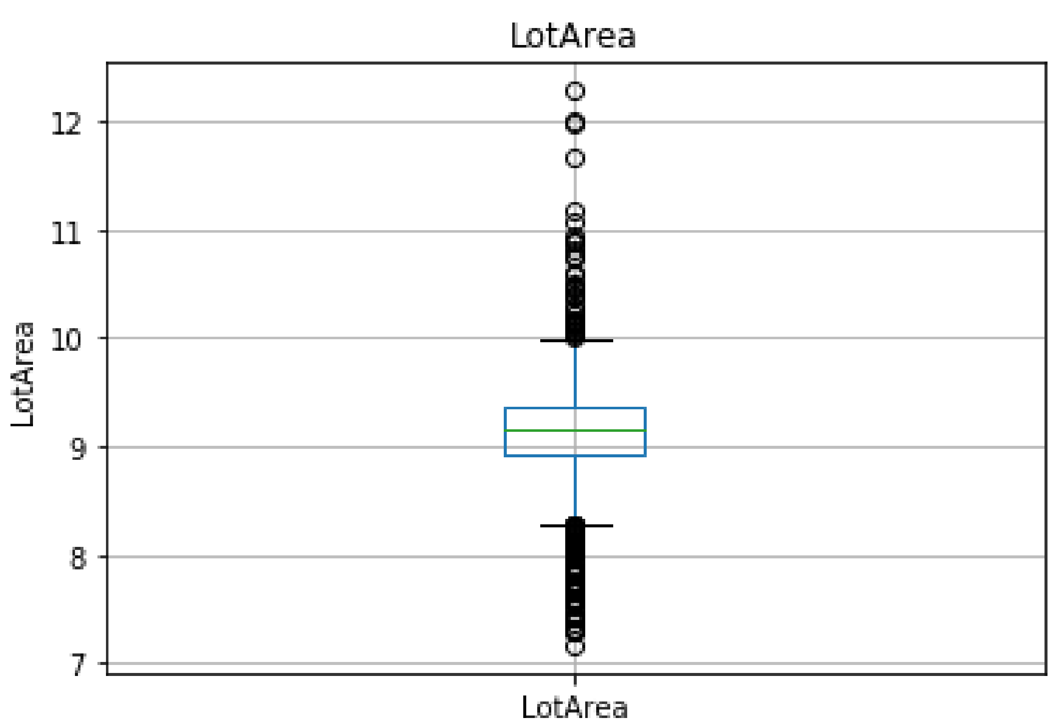



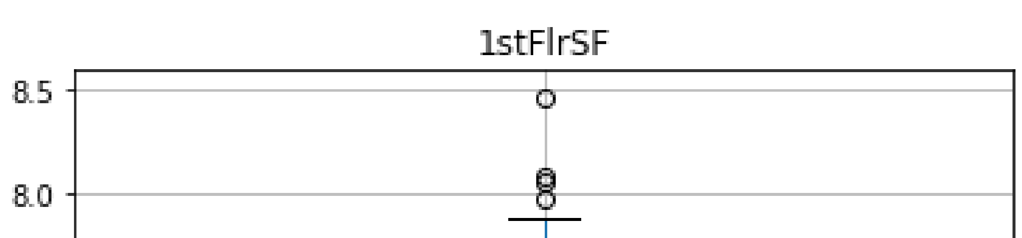

## Outliers

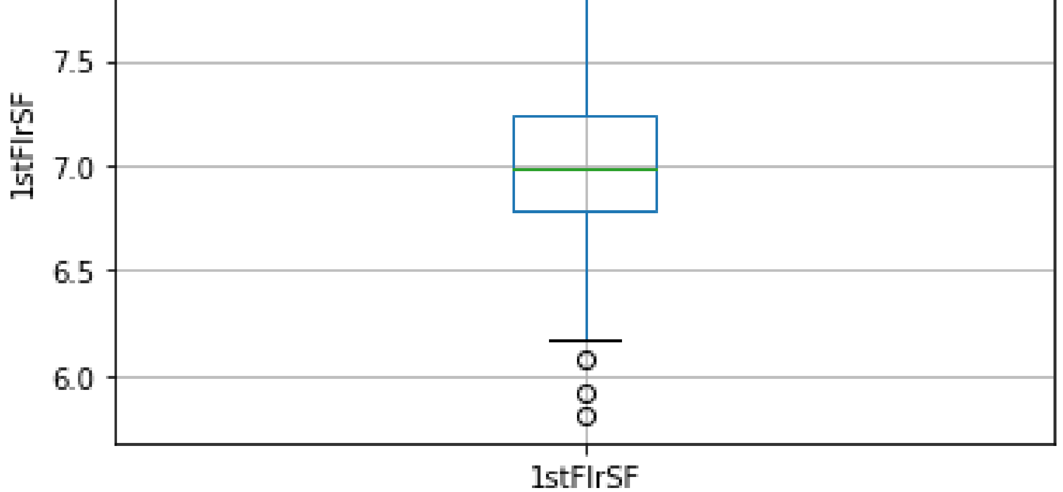
In [64]:
```
for feature in continuous_feature:
    data=dataset.copy()
    if 0 in data[feature].unique():
```

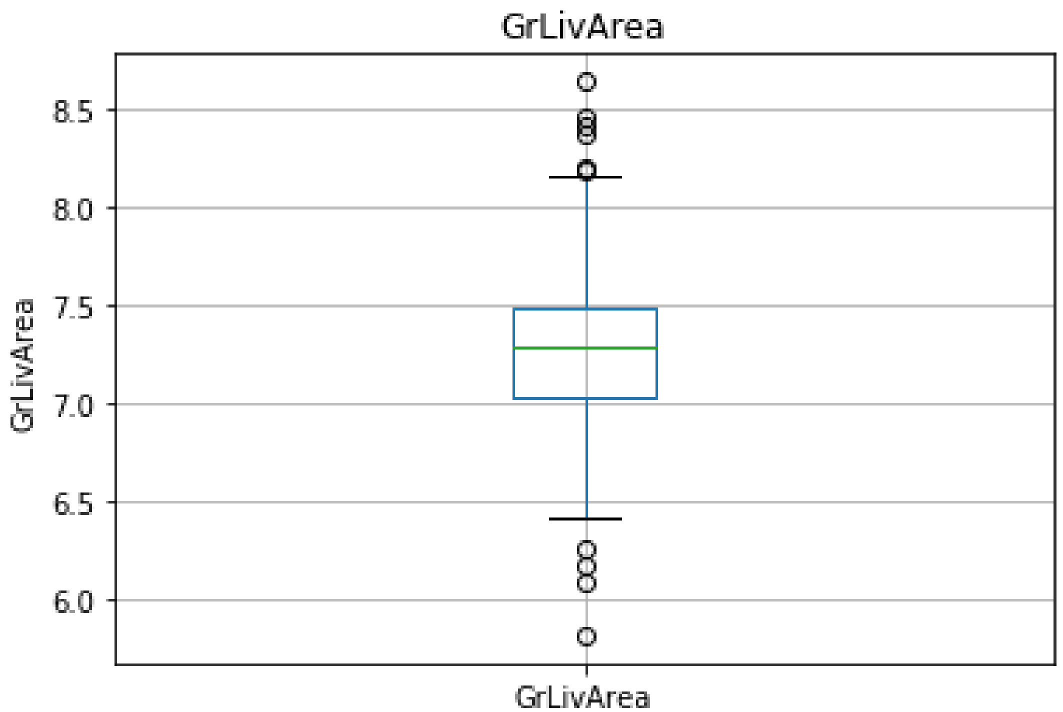
```
        pass
    else:
        data[feature]=np.log(data[feature])
        data.boxplot(column=feature)
        plt.ylabel(feature)
        plt.title(feature)
        plt.show()
```

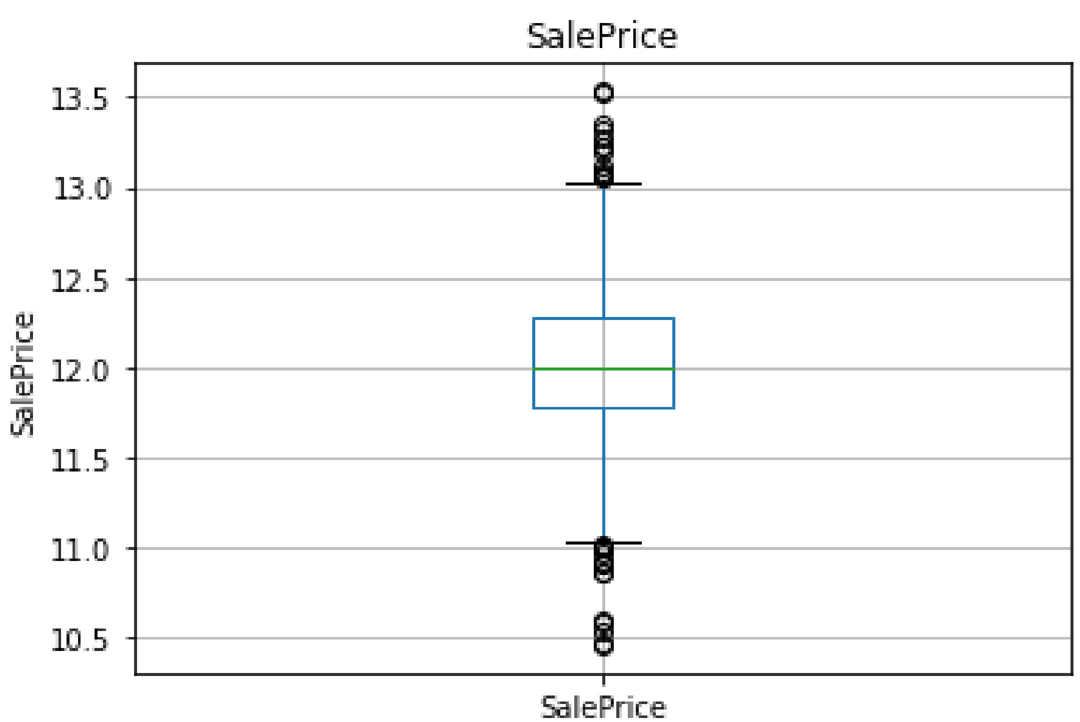



1stFlrSF
7.5
7.0
6.5
6.0
1stFlrSF


GrLivArea
8.5
8.0
7.5
7.0
6.5
6.0
GrLivArea
GrLivArea


SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
SalePrice
SalePrice

## Categorical Variables

In [67]:
```
categorical_features=[feature for feature in dataset.columns if data[feature].dtypes=='O']
categorical_features
```

Out[67]:
```
['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']
```

In [69]:
```
dataset[categorical_features].head()
```

Out[69]:

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

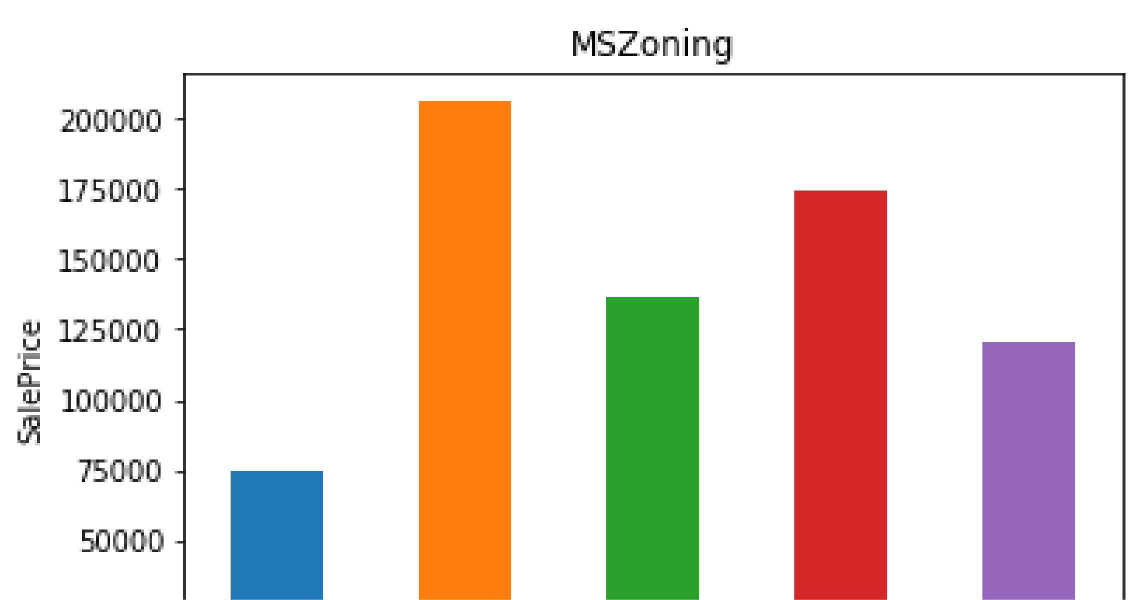
In [73]:
```
for feature in categorical_features:
    print('The feature is {} and number of categories are {}'.format(feature,len(dataset[feature].unique())))
```

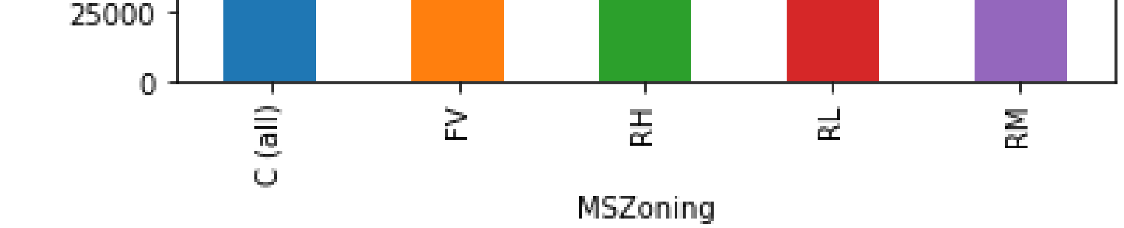
```
The feature is MSZoning and number of categories are 5
The feature is Street and number of categories are 2
The feature is Alley and number of categories are 3
The feature is LotShape and number of categories are 4
The feature is LandContour and number of categories are 4
The feature is Utilities and number of categories are 2
The feature is LotConfig and number of categories are 5
The feature is LandSlope and number of categories are 3
The feature is Neighborhood and number of categories are 25
The feature is Condition1 and number of categories are 9
The feature is Condition2 and number of categories are 8
The feature is BldgType and number of categories are 5
The feature is HouseStyle and number of categories are 8
The feature is RoofStyle and number of categories are 6
The feature is RoofMatl and number of categories are 8
The feature is Exterior1st and number of categories are 15
The feature is Exterior2nd and number of categories are 16
The feature is MasVnrType and number of categories are 5
The feature is ExterQual and number of categories are 4
The feature is ExterCond and number of categories are 5
The feature is Foundation and number of categories are 6
The feature is BsmtQual and number of categories are 5
The feature is BsmtCond and number of categories are 5
```

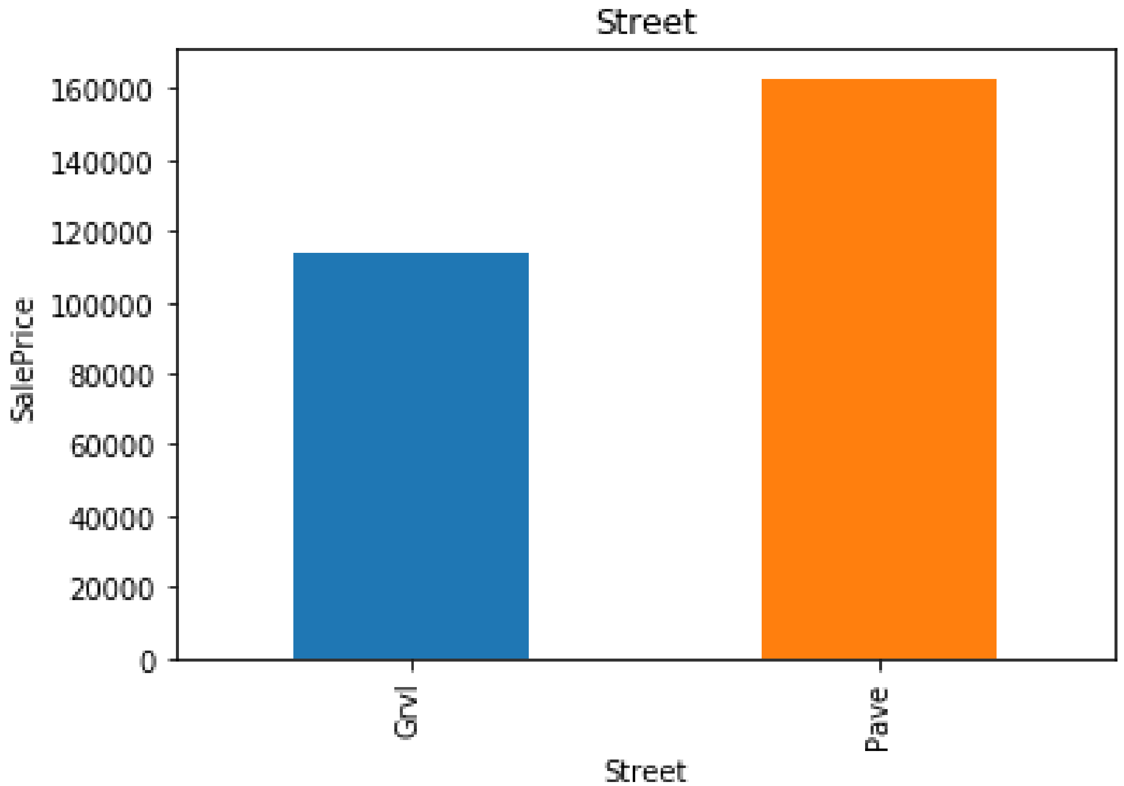
```
The feature is BsmtCond and number of categories are 5
The feature is BsmtExposure and number of categories are 5
The feature is BsmtFinType1 and number of categories are 7
The feature is BsmtFinType2 and number of categories are 7
The feature is Heating and number of categories are 6
The feature is HeatingQC and number of categories are 5
The feature is CentralAir and number of categories are 2
The feature is Electrical and number of categories are 6
The feature is KitchenQual and number of categories are 4
The feature is Functional and number of categories are 7
The feature is FireplaceQu and number of categories are 6
The feature is GarageType and number of categories are 7
The feature is GarageFinish and number of categories are 4
The feature is GarageQual and number of categories are 6
The feature is GarageCond and number of categories are 6
The feature is PavedDrive and number of categories are 3
The feature is PoolQC and number of categories are 4
The feature is Fence and number of categories are 5
The feature is MiscFeature and number of categories are 5
The feature is SaleType and number of categories are 9
The feature is SaleCondition and number of categories are 6
```

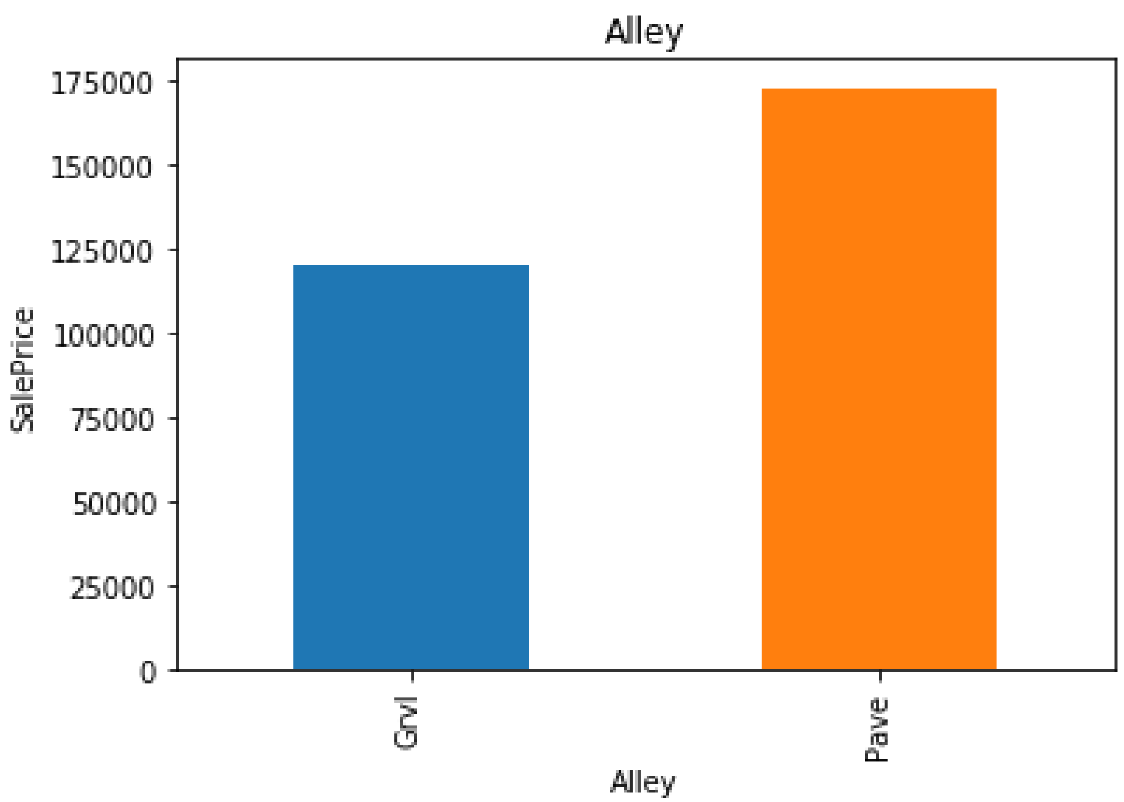
In [74]:
```
## Find out the relationship between categorical variable and dependent feature SalesPrice
```

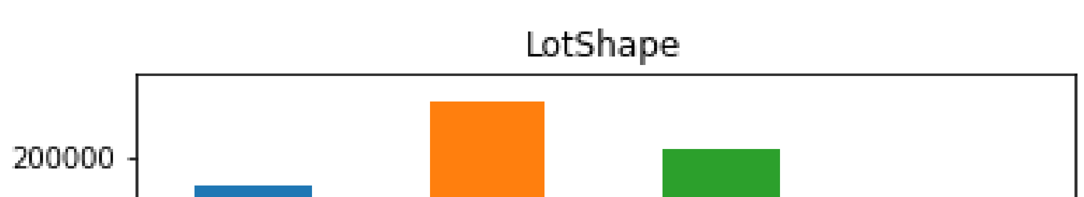
In [75]:
```
for feature in categorical_features:
    data=dataset.copy()
    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.title(feature)
    plt.show()
```

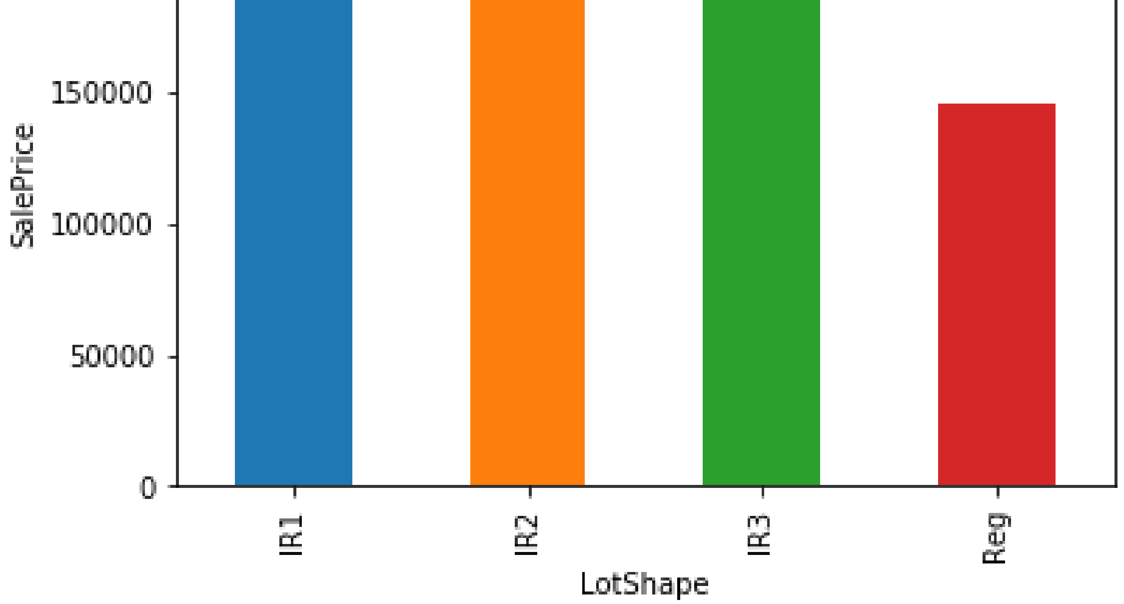


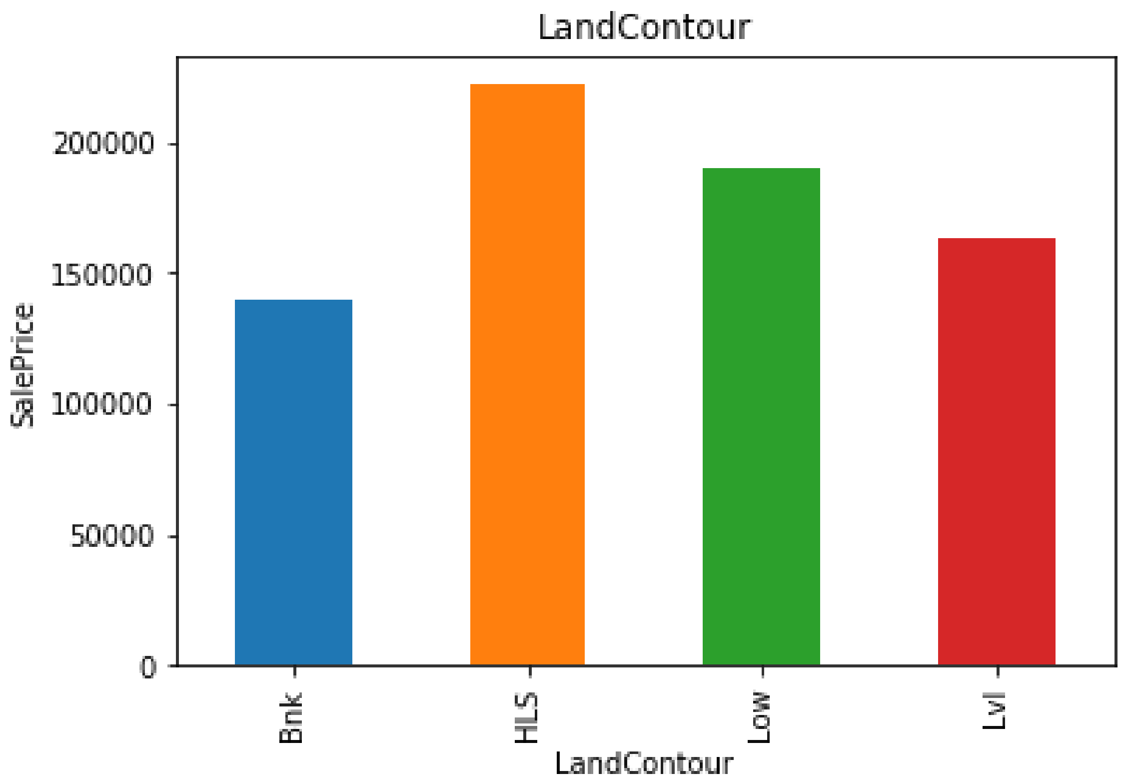

25000
0
C (all)
FV
RH
RL
RM
MSZoning

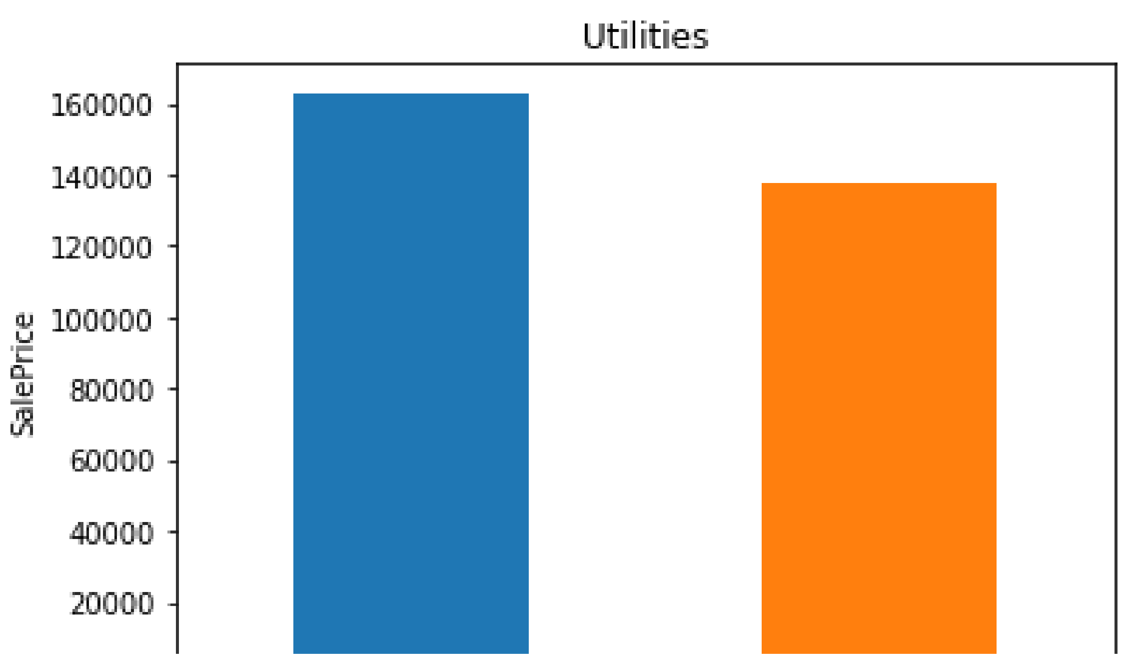

Street
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Grvl
Pave
Street


Alley
SalePrice
175000
150000
125000
100000
75000
50000
25000
0
Grvl
Pave
Alley

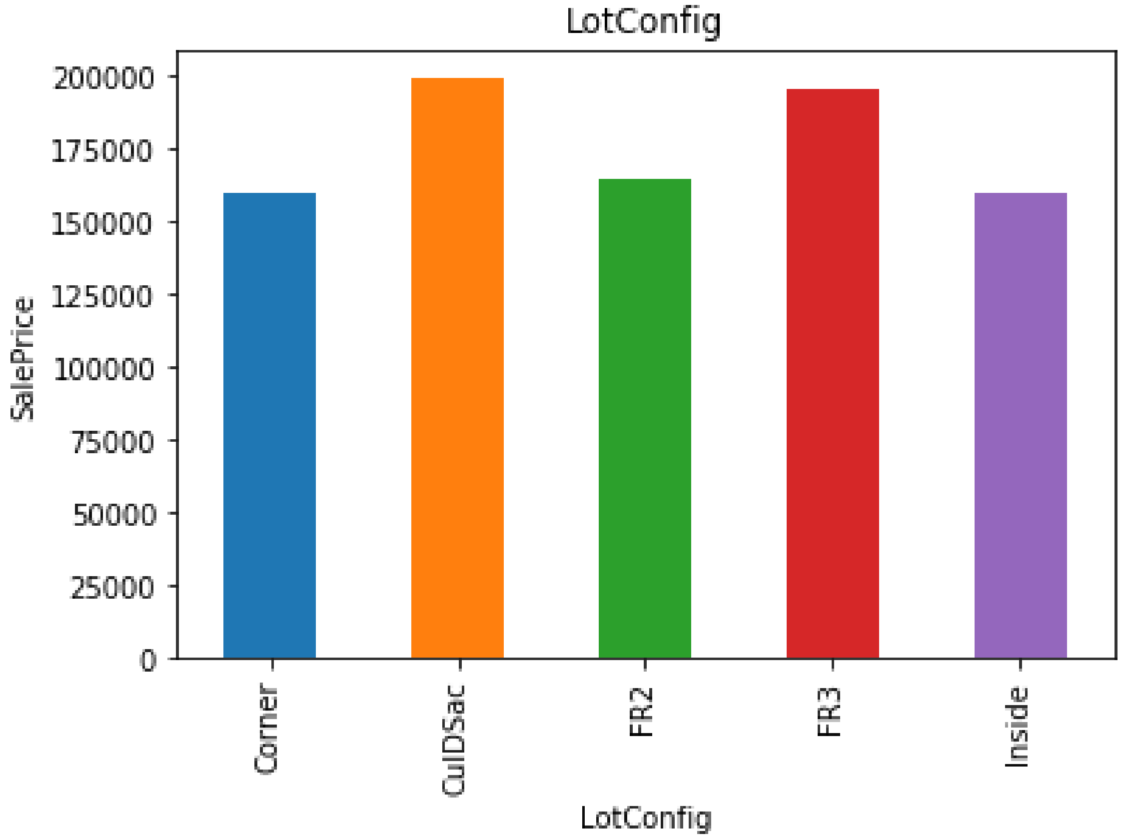

LotShape
200000

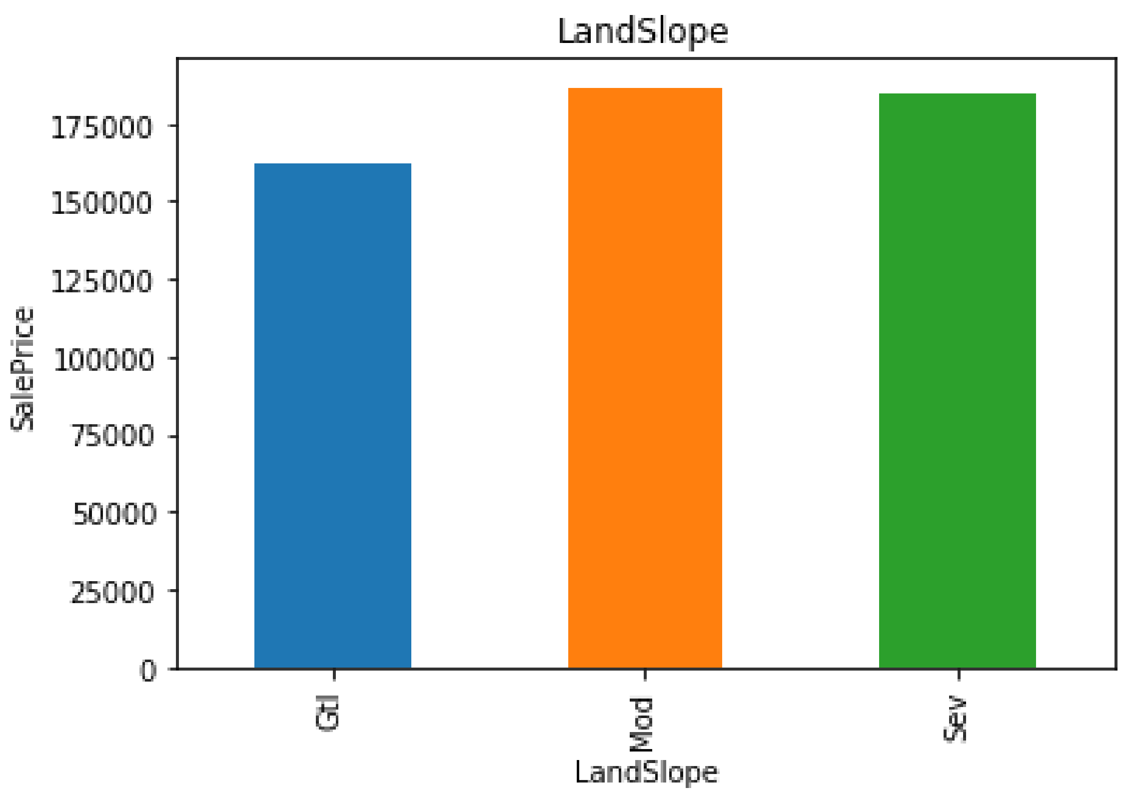

150000
100000
50000
0
SalePrice
IR1
IR2
IR3
Reg
LotShape

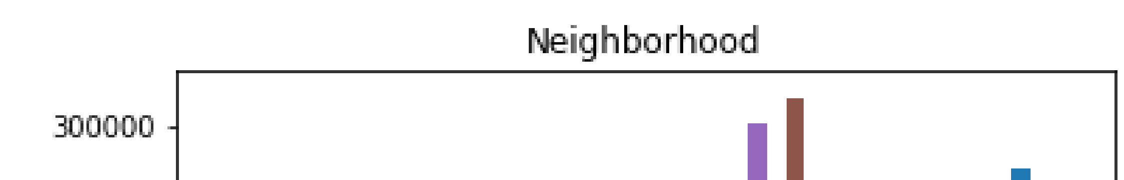

LandContour
200000
150000
100000
50000
0
SalePrice
Bnk
HLS
Low
Lvl
LandContour

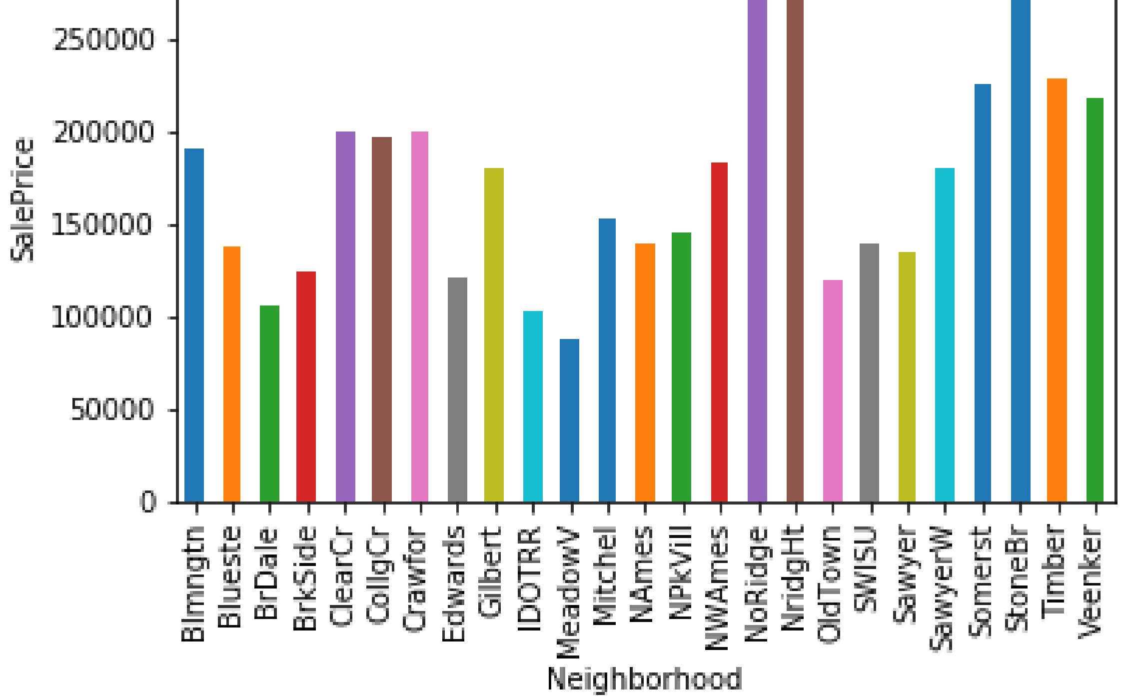

Utilities
160000
140000
120000
100000
80000
60000
40000
20000
SalePrice

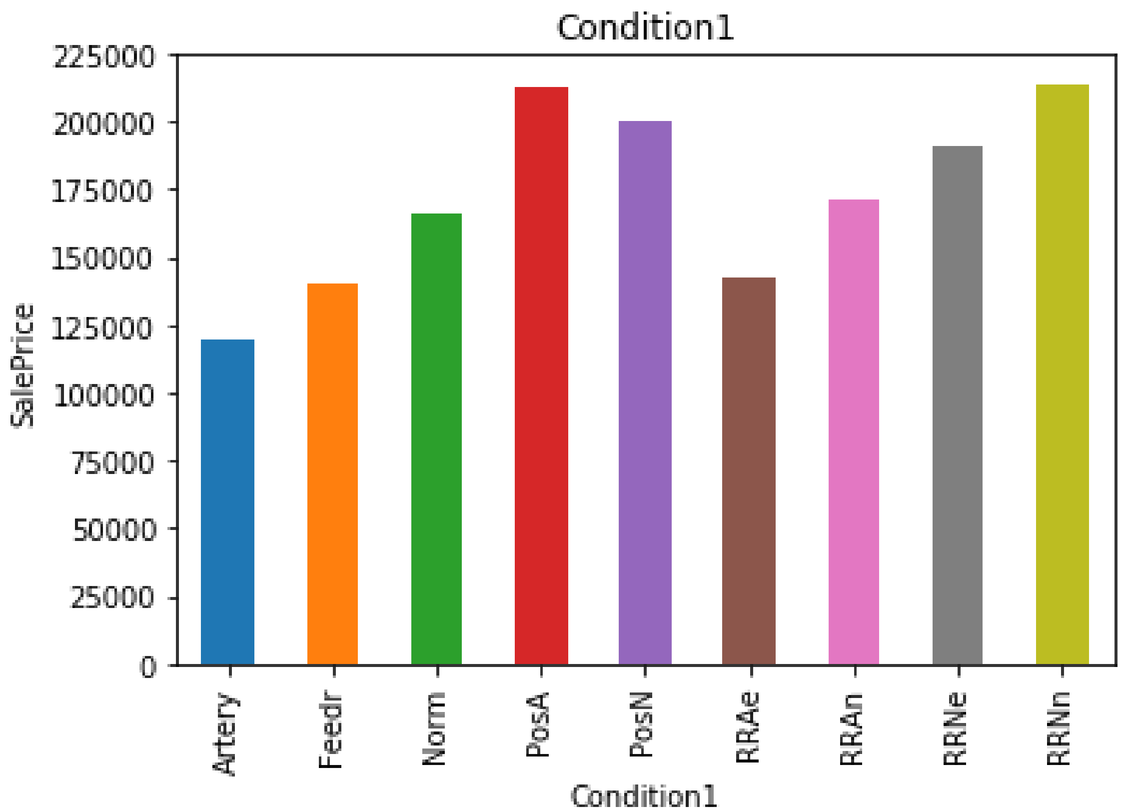

0
AllPub
NoSeWa
Utilities

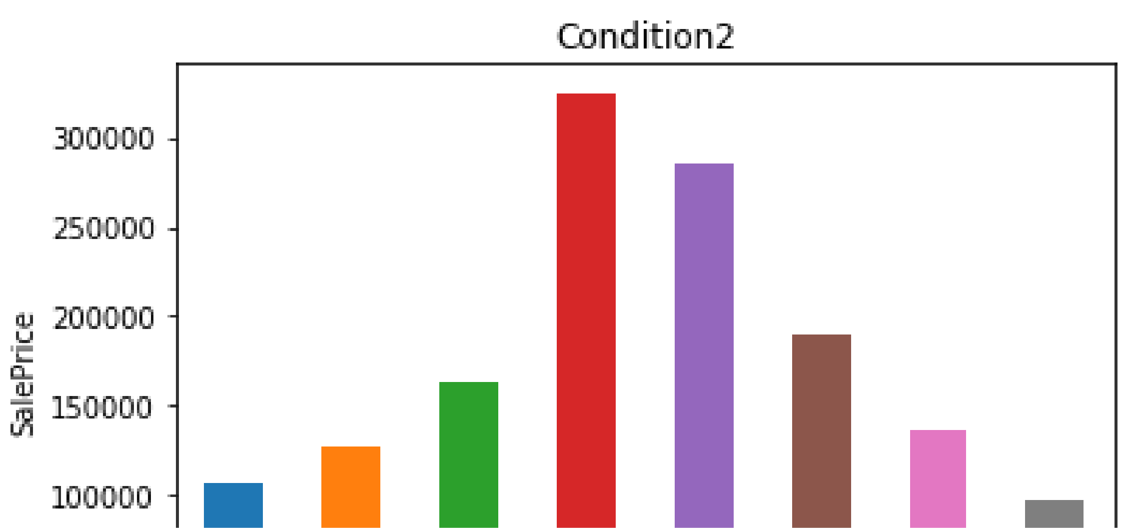

LotConfig
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
Corner
CulDSac
FR2
FR3
Inside
LotConfig

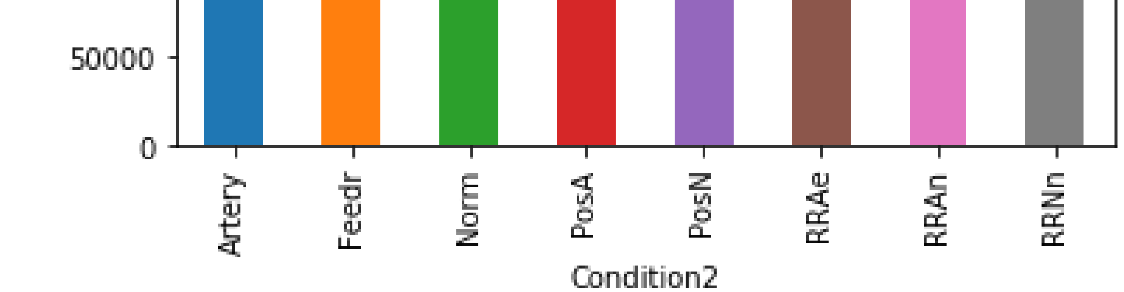

LandSlope
SalePrice
175000
150000
125000
100000
75000
50000
25000
0
Gtl
Mod
Sev
LandSlope

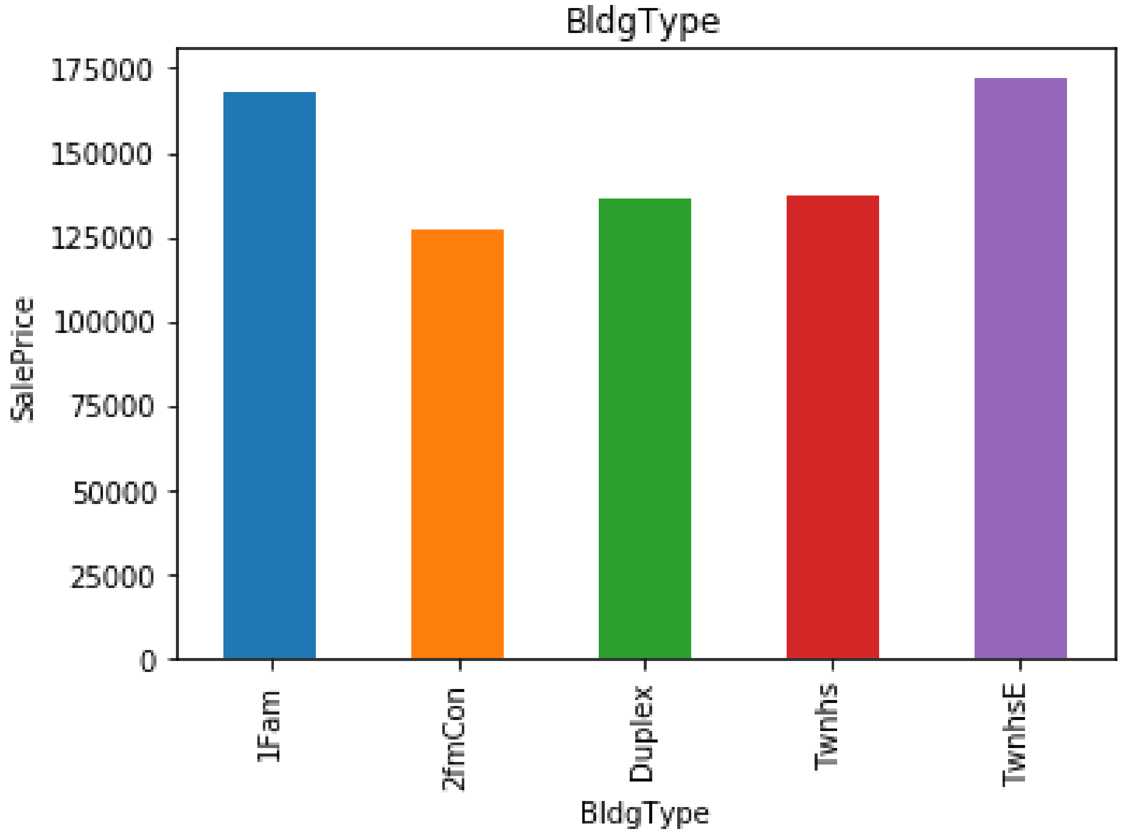

Neighborhood
300000

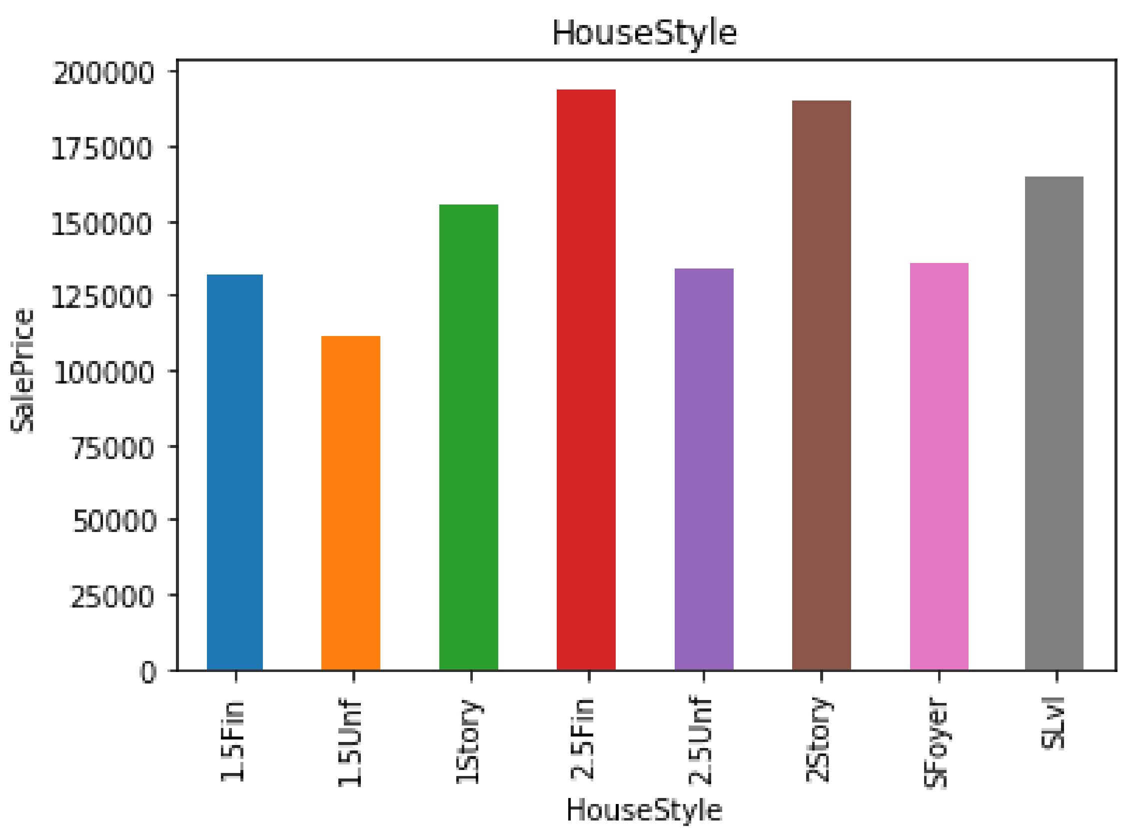

250000
200000
150000
100000
50000
0
SalePrice
Blmngtn
Blueste
BrDale
BrkSide
ClearCr
CollgCr
Crawfor
Edwards
Gilbert
IDOTRR
MeadowV
Mitchel
NAmes
NPkVill
NWAmes
NoRidge
NridgHt
OldTown
SWISU
Sawyer
SawyerW
Somerst
StoneBr
Timber
Veenker
Neighborhood


Condition1
225000
200000
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNe
RRNn
Condition1

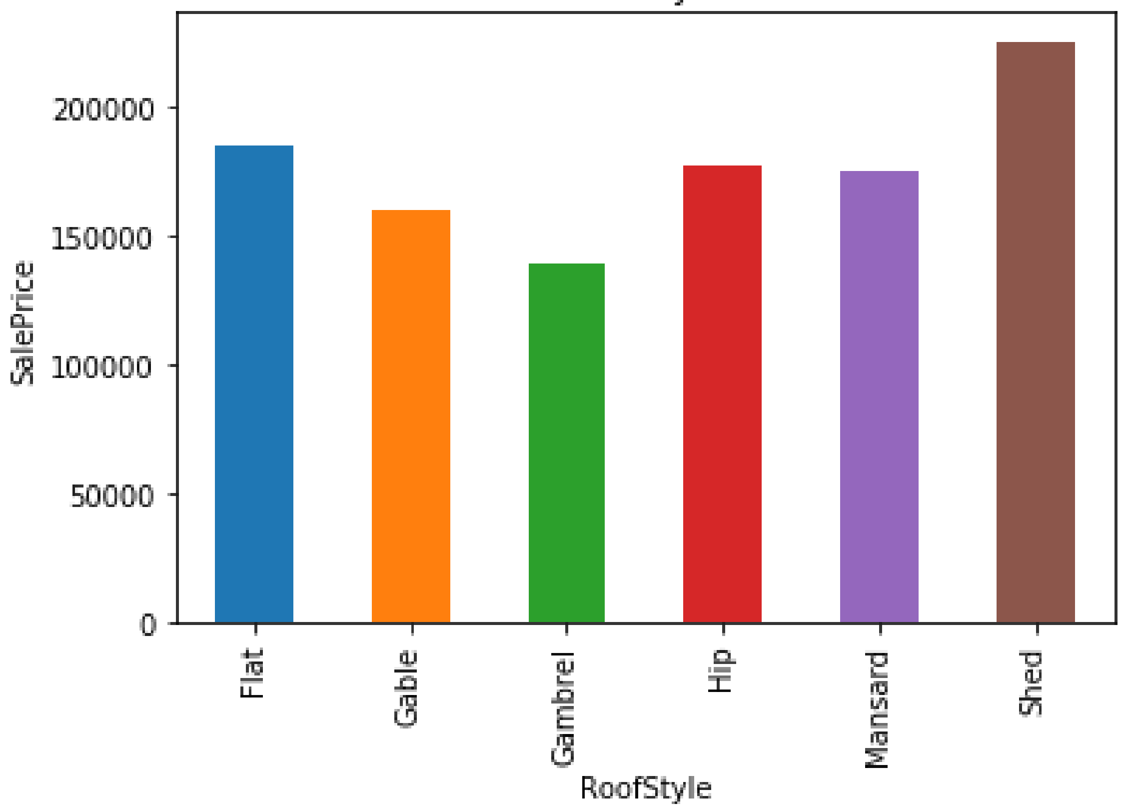

Condition2
300000
250000
200000
150000
100000
SalePrice


50000
0
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNn
Condition2

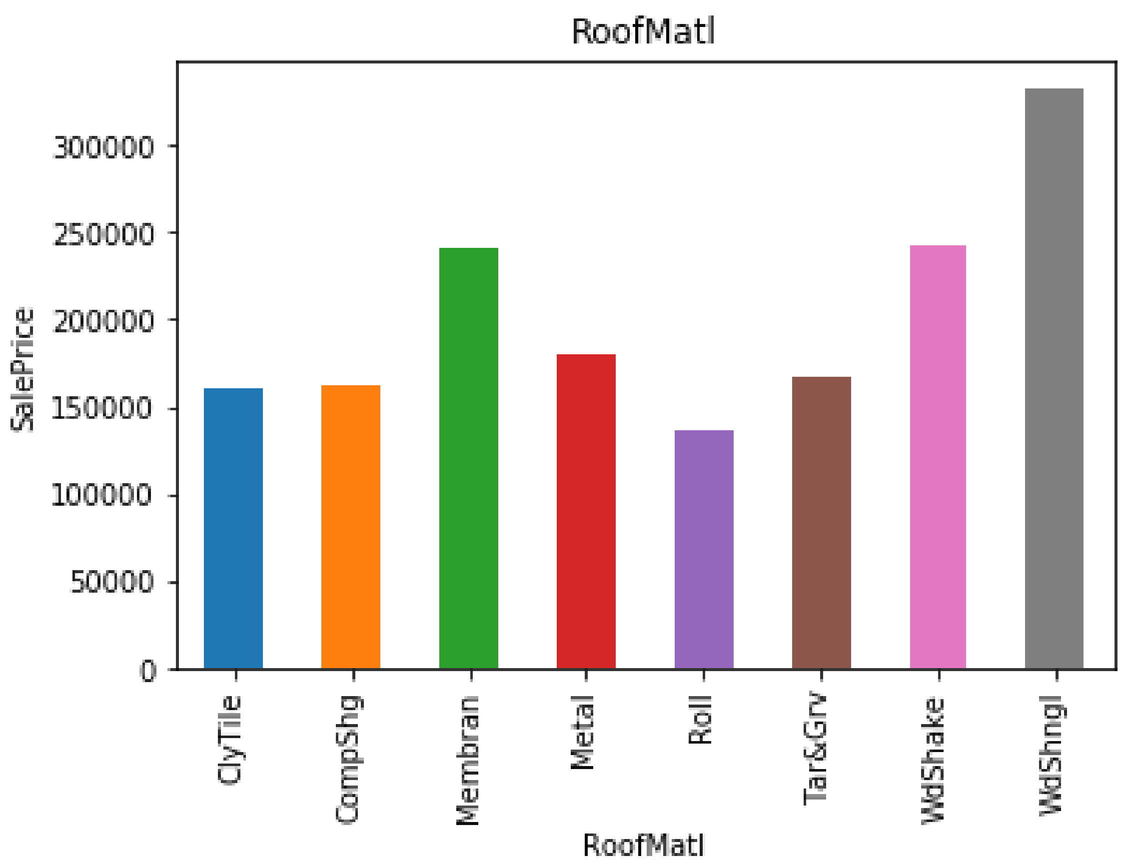

BldgType
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
1Fam
2fmCon
Duplex
Twnhs
TwnhsE
BldgType


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

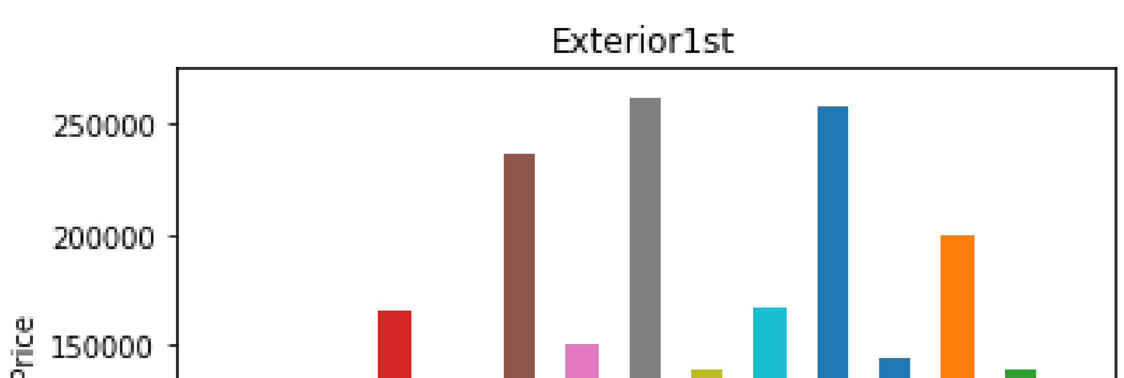
RoofStyle



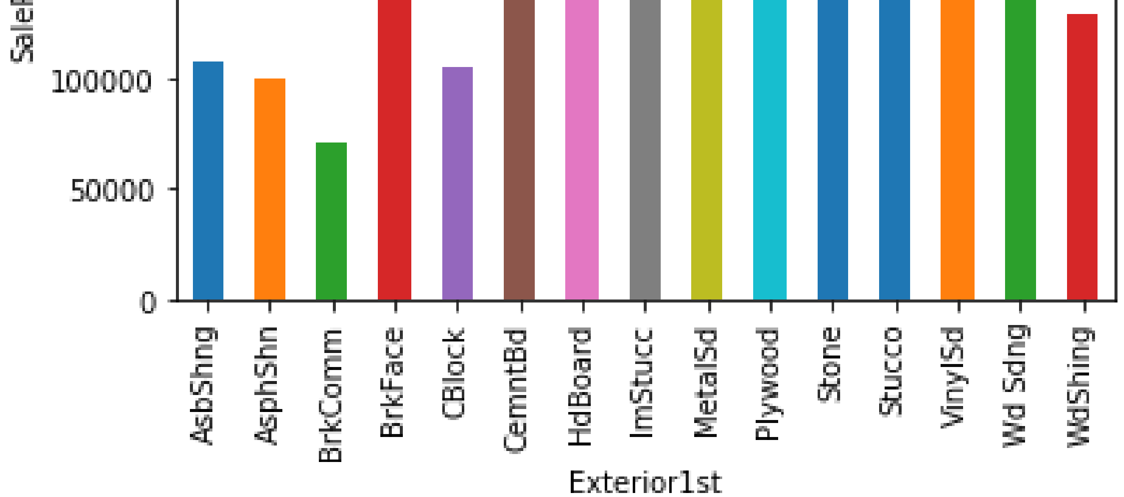
200000
150000
100000
50000
0
SalePrice
Flat
Gable
Gambrel
Hip
Mansard
Shed
RoofStyle

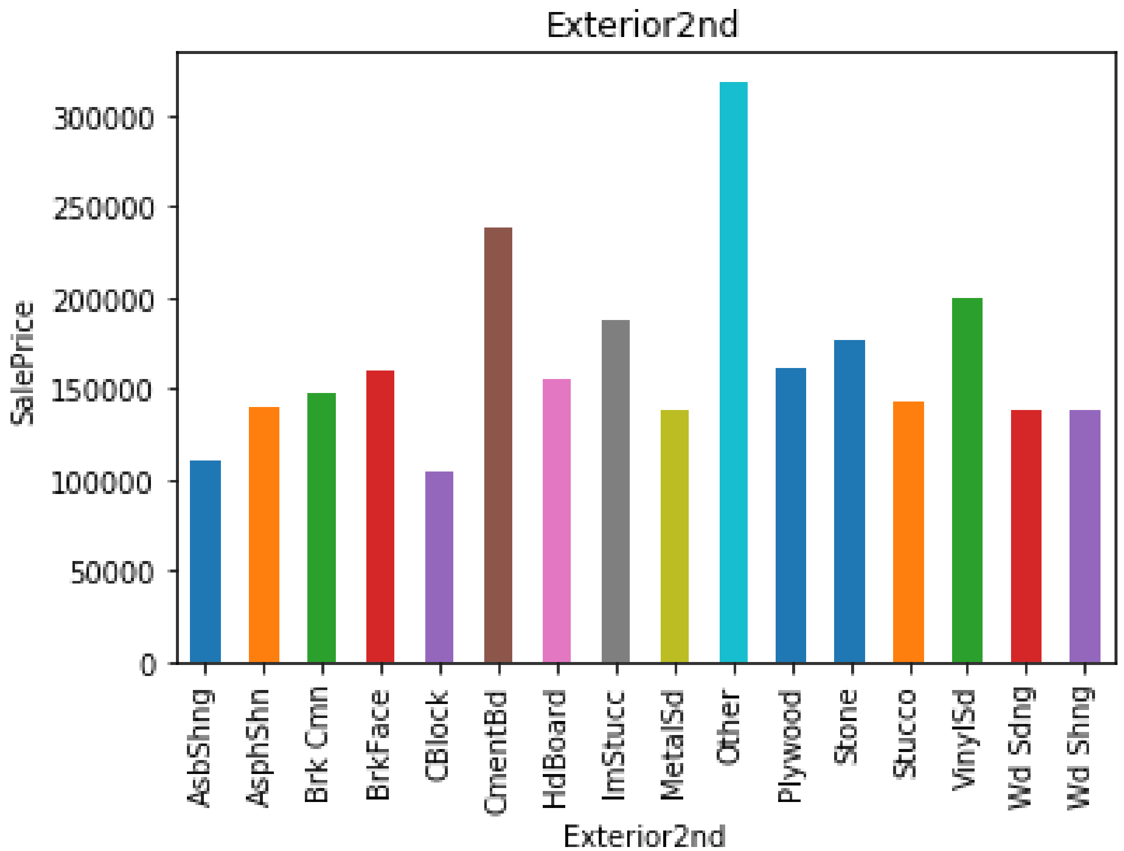



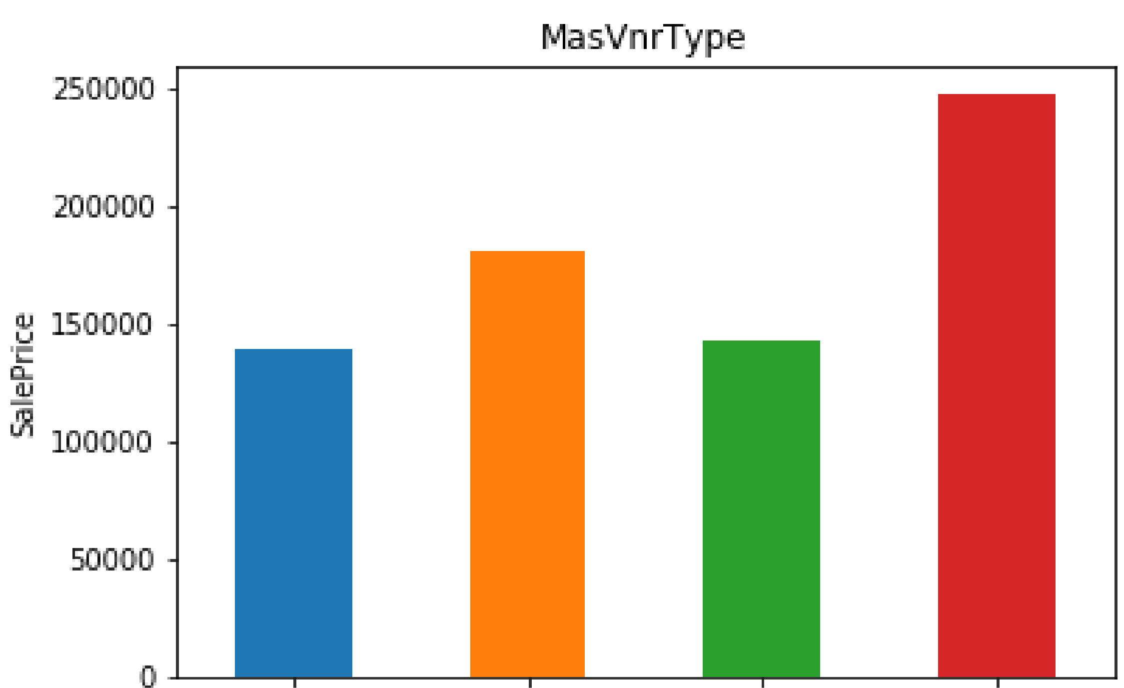
RoofMatl
300000
250000
200000
150000
100000
50000
0
SalePrice
ClyTile
CompShg
Membran
Metal
Roll
Tar&Grv
WdShake
WdShngl
RoofMatl

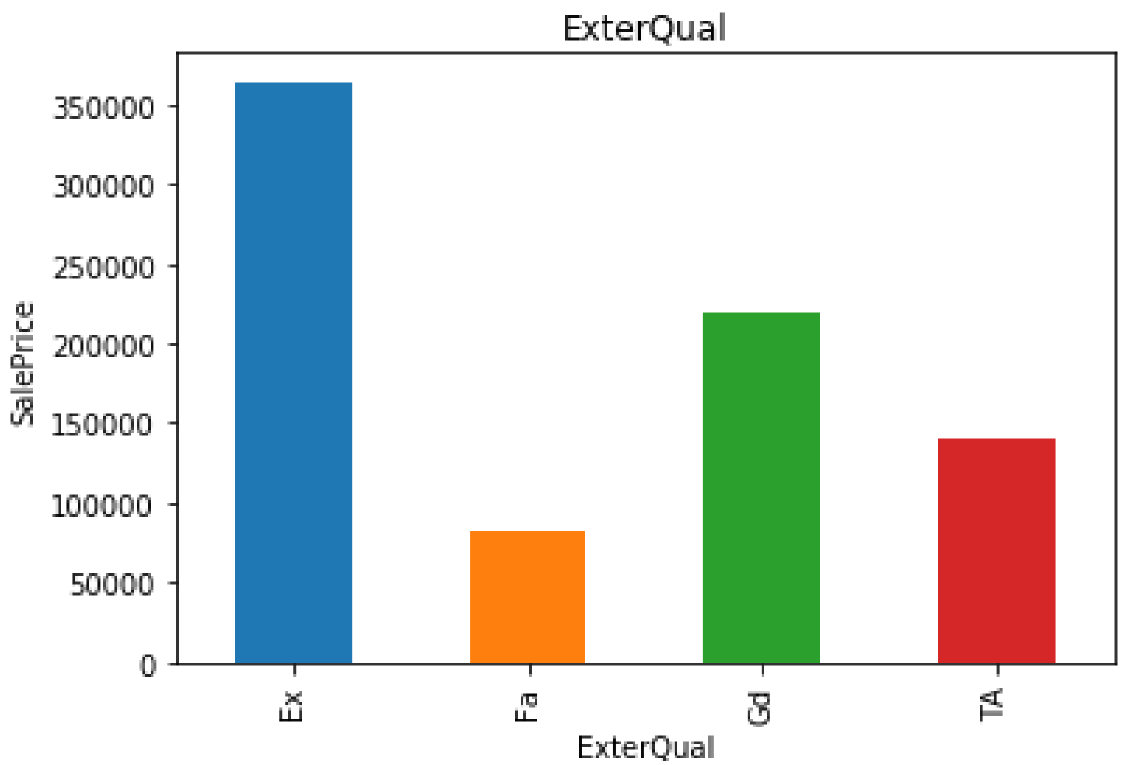
Exterior1st
250000
200000
150000

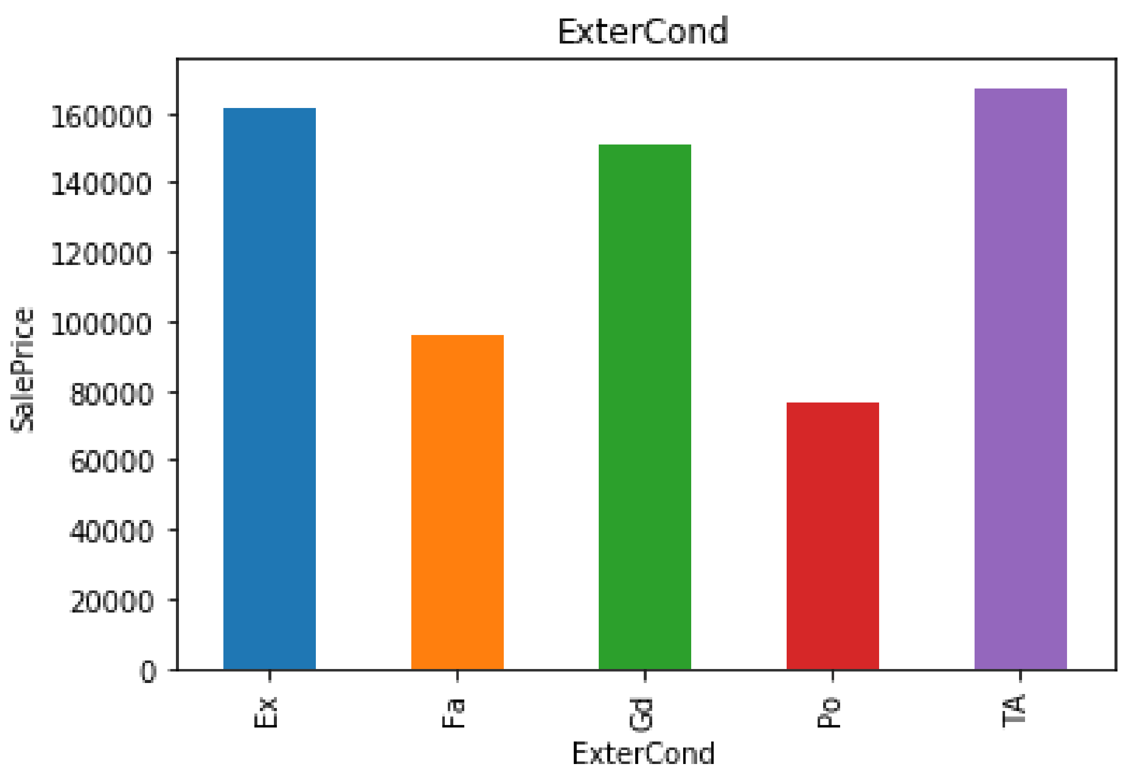

100000
50000
0
AsbShng
AsphShn
BrkComm
BrkFace
CBlock
CemntBd
HdBoard
ImStucc
MetalSd
Plywood
Stone
Stucco
VinylSd
Wd Sdng
WdShing
Exterior1st

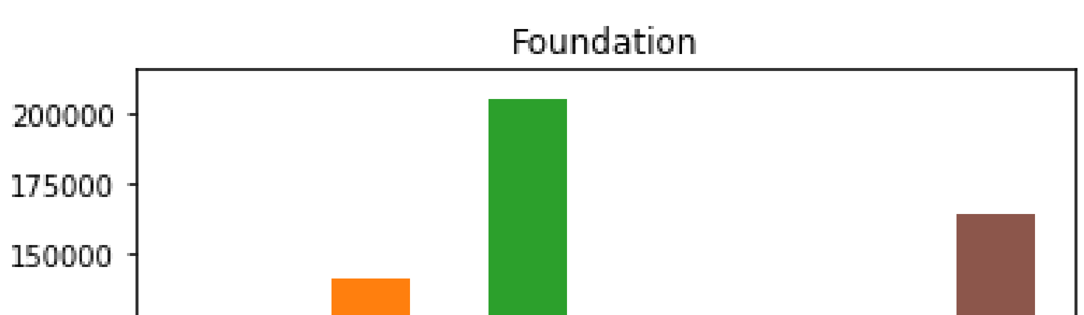

Exterior2nd
300000
250000
200000
150000
100000
50000
0
SalePrice
AsbShng
AsphShn
Brk Cmn
BrkFace
CBlock
CmentBd
HdBoard
ImStucc
MetalSd
Other
Plywood
Stone
Stucco
VinylSd
Wd Sdng
Wd Shng
Exterior2nd

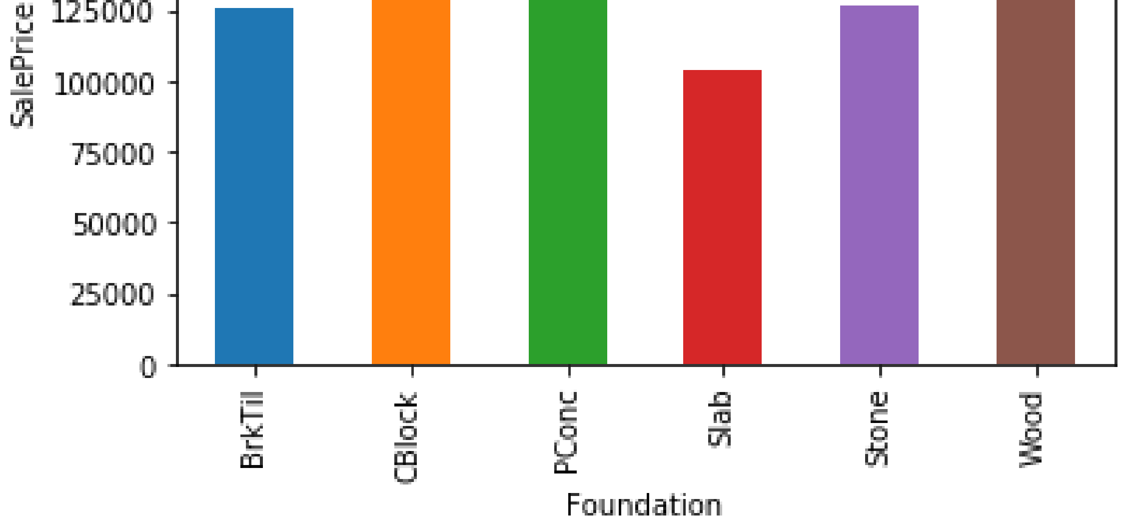

MasVnrType
250000
200000
150000
100000
50000
0
SalePrice

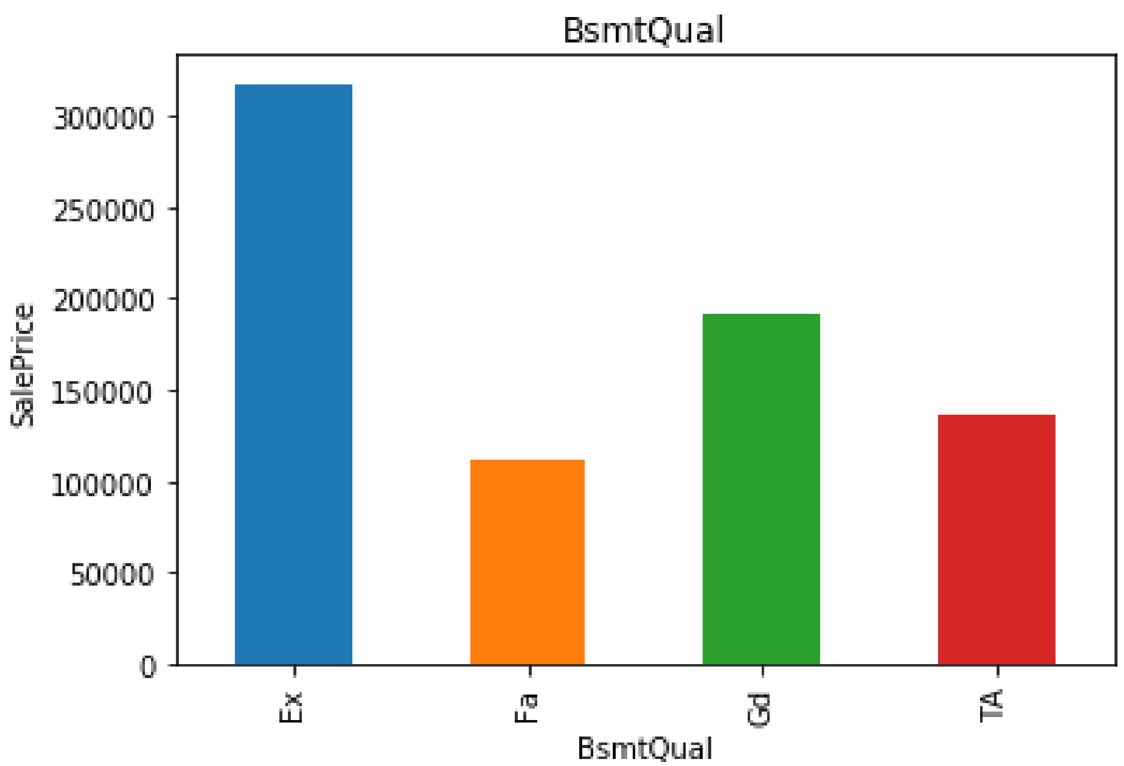

BrkCmn
BrkFace
None
Stone
MasVnrType

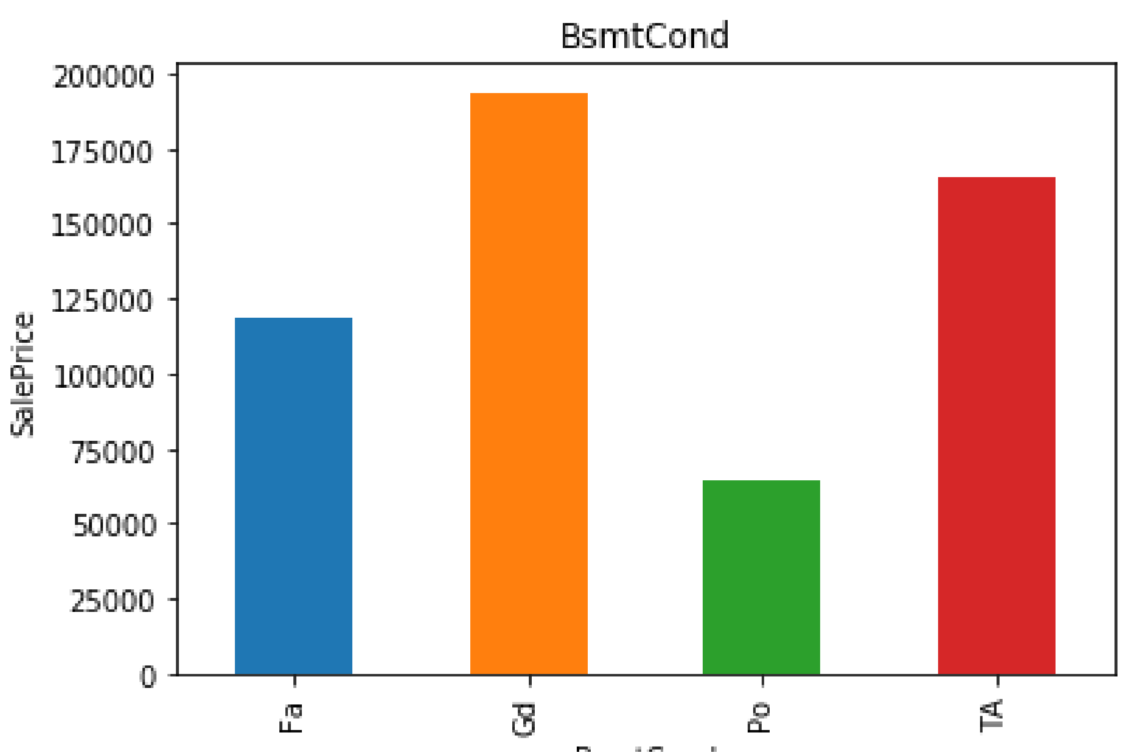

ExterQual
SalePrice
350000
300000
250000
200000
150000
100000
50000
0
Ex
Fa
Gd
TA
ExterQual


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

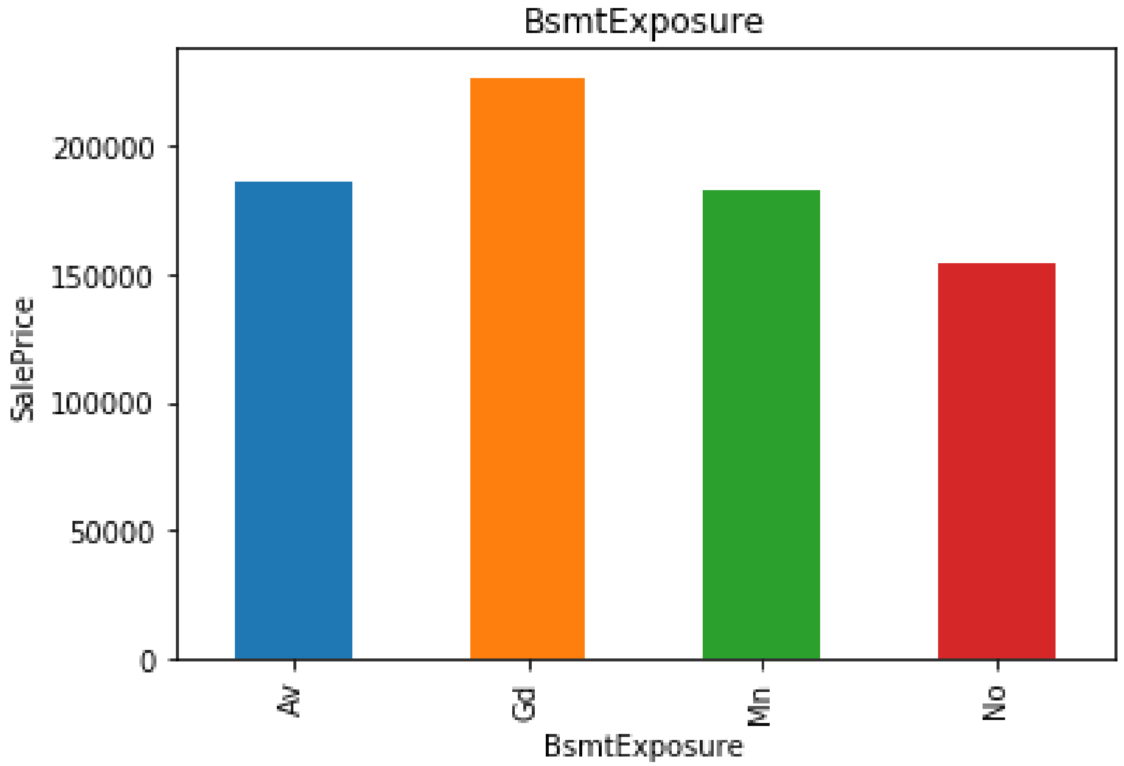

Foundation
200000
175000
150000

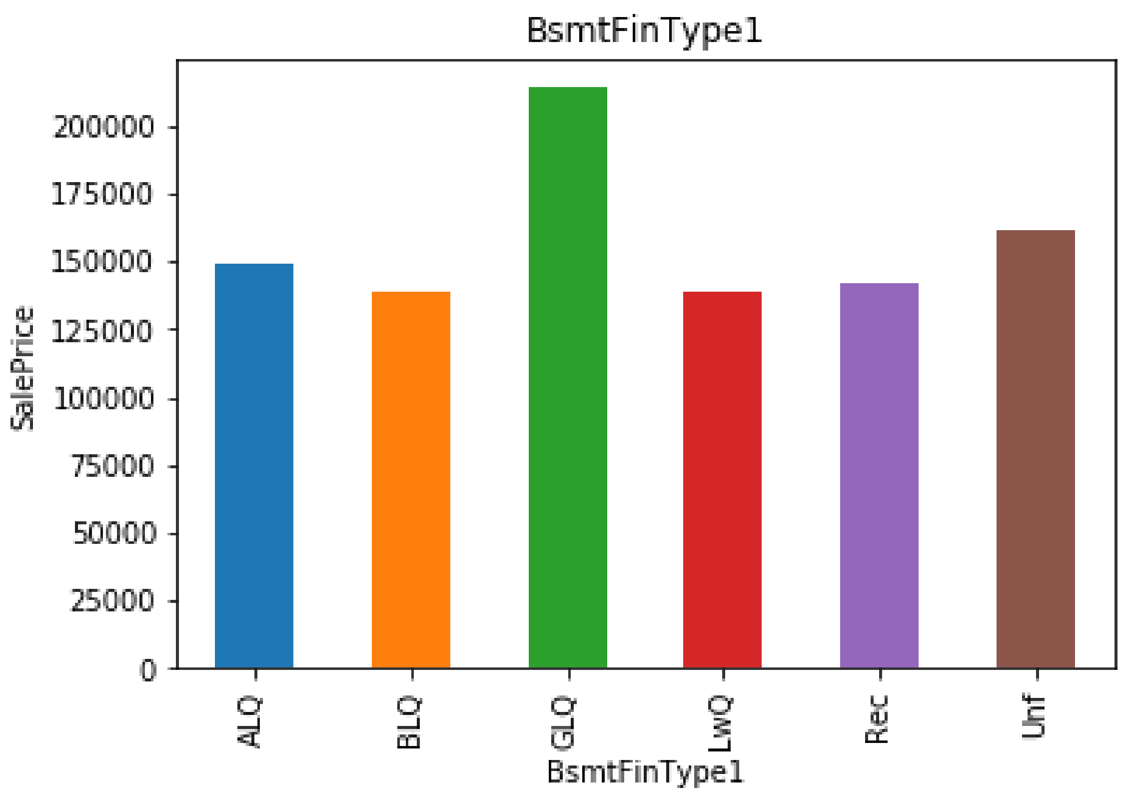

SalePrice
125000
100000
75000
50000
25000
0
BrkTil
CBlock
PConc
Slab
Stone
Wood
Foundation

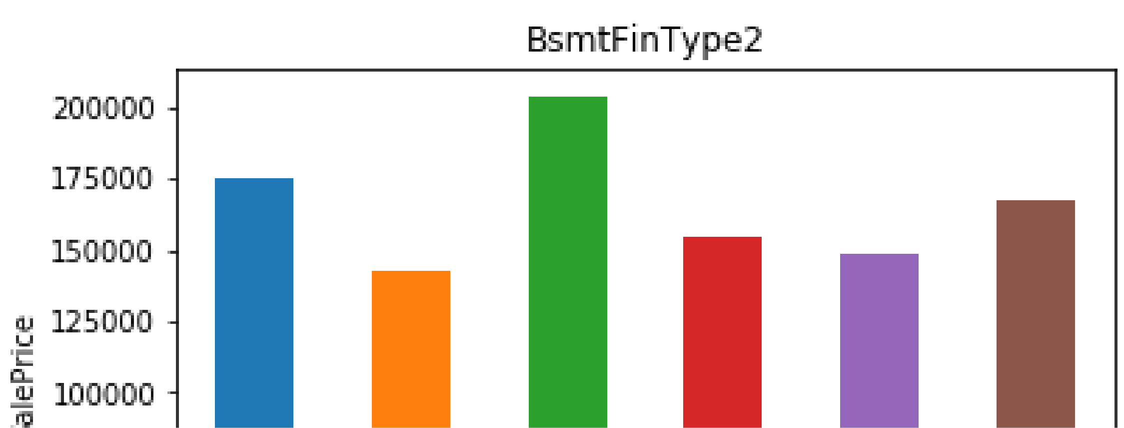

BsmtQual
SalePrice
300000
250000
200000
150000
100000
50000
0
Ex
Fa
Gd
TA
BsmtQual

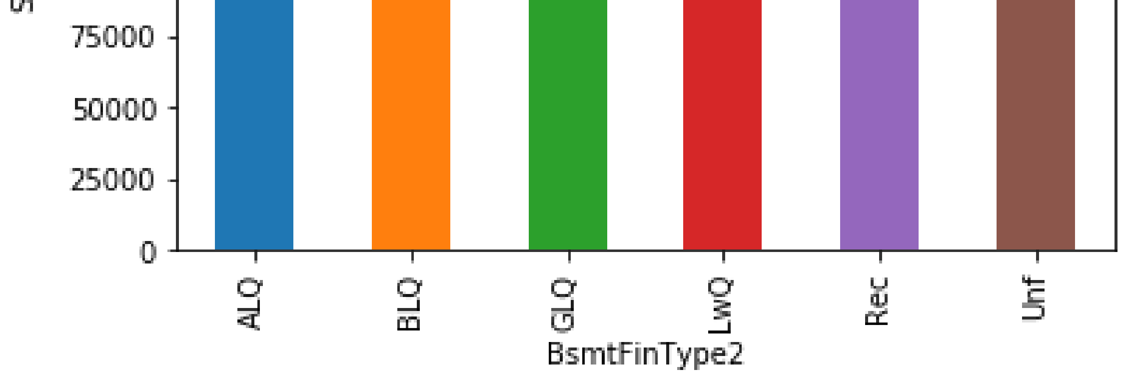

BsmtCond
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
Fa
Gd
Po
TA

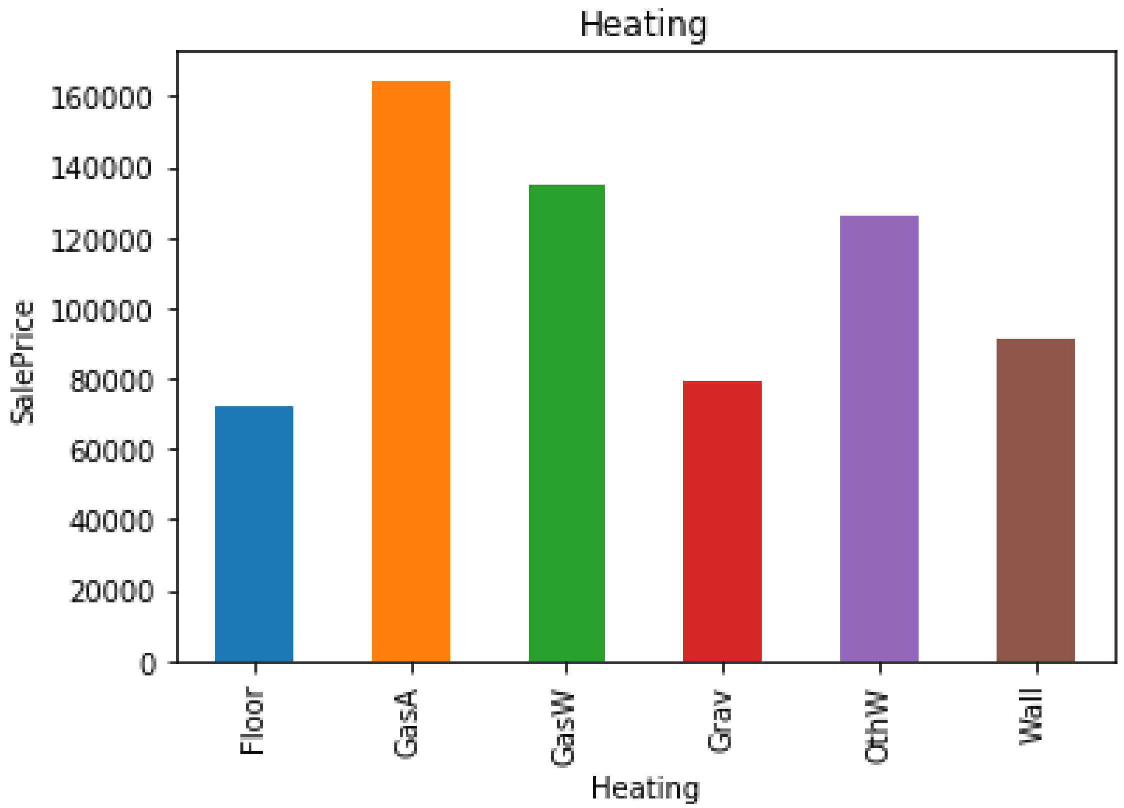
BsmtCond

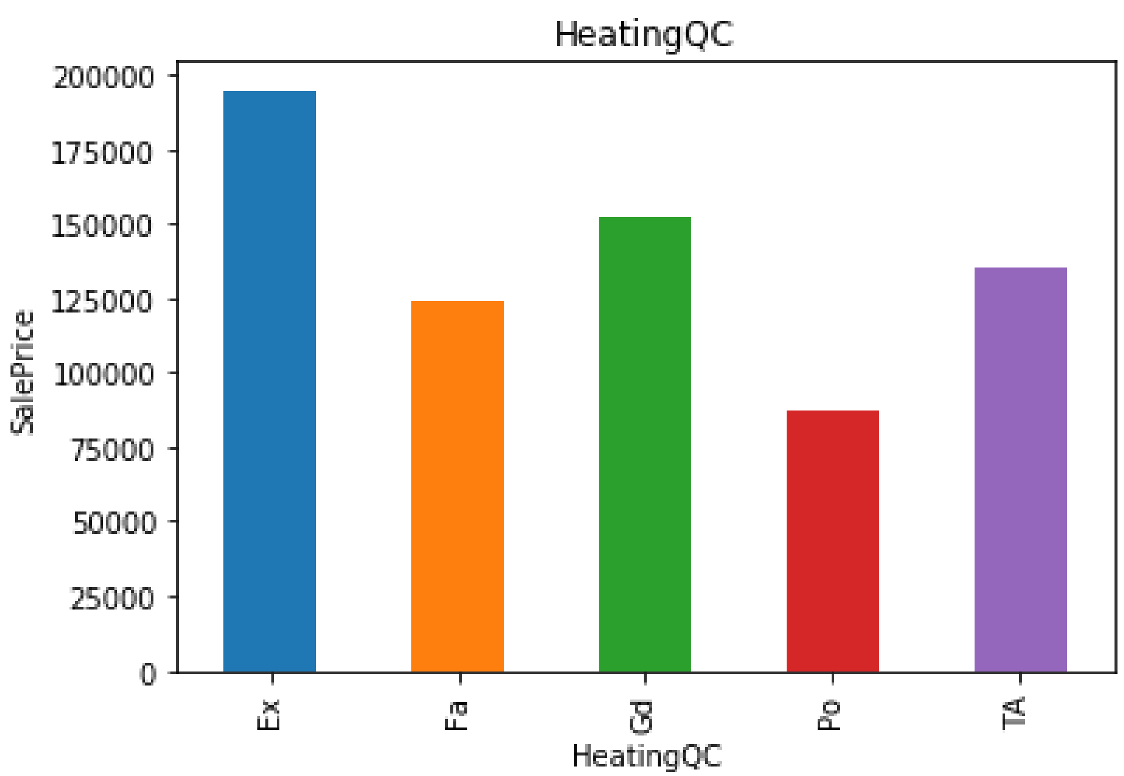

BsmtExposure
SalePrice
200000
150000
100000
50000
0
Av
Gd
Mn
No
BsmtExposure


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

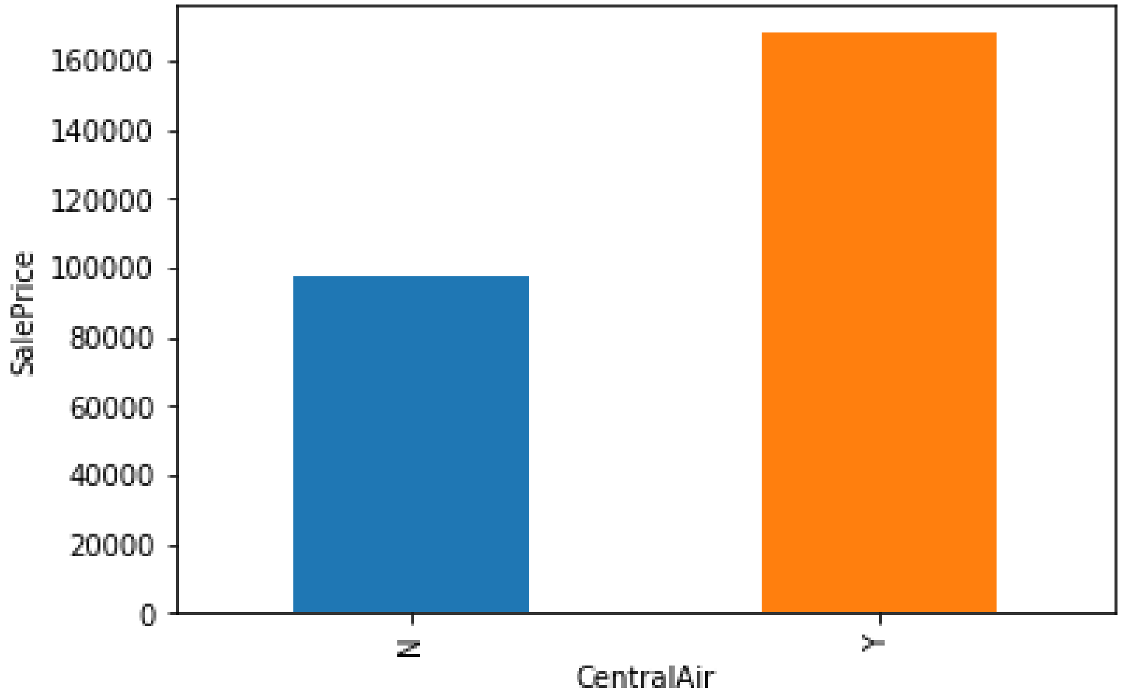

BsmtFinType2
200000
175000
150000
125000
100000


75000
50000
25000
0
ALQ
BLQ
GLQ
LwQ
Rec
Unf
BsmtFinType2

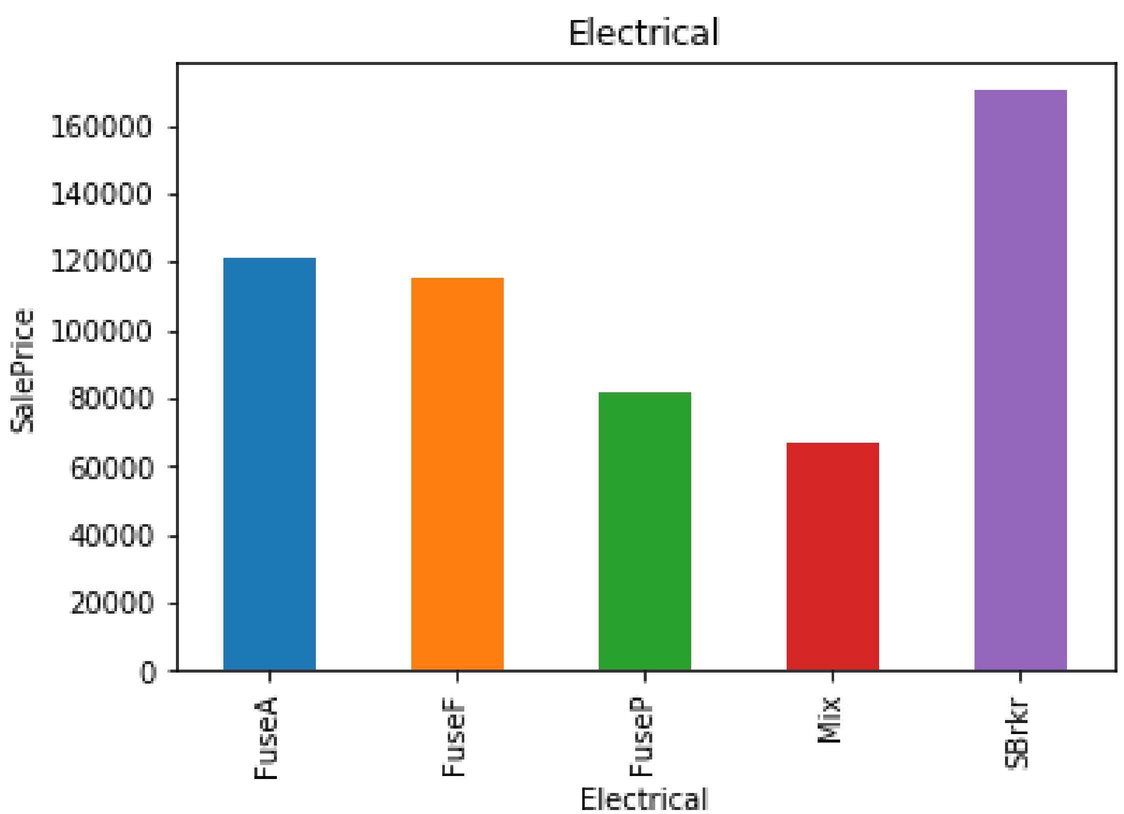

Heating
160000
140000
120000
100000
80000
60000
40000
20000
0
SalePrice
Floor
GasA
GasW
Grav
OthW
Wall
Heating


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

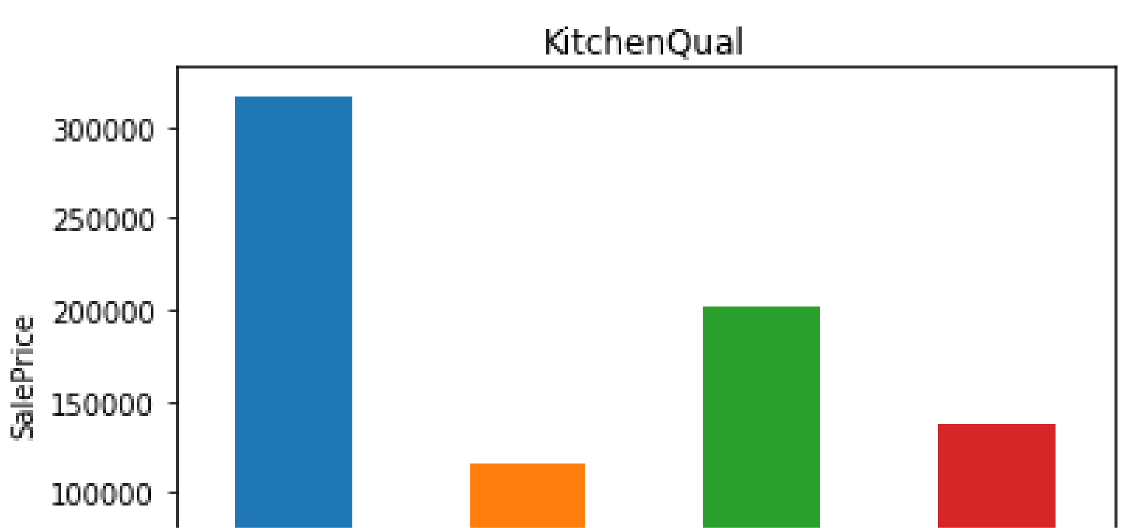
CentralAir



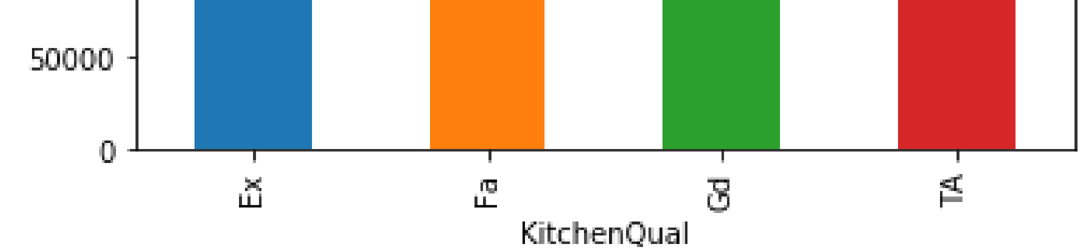
SalePrice
N
Y
CentralAir

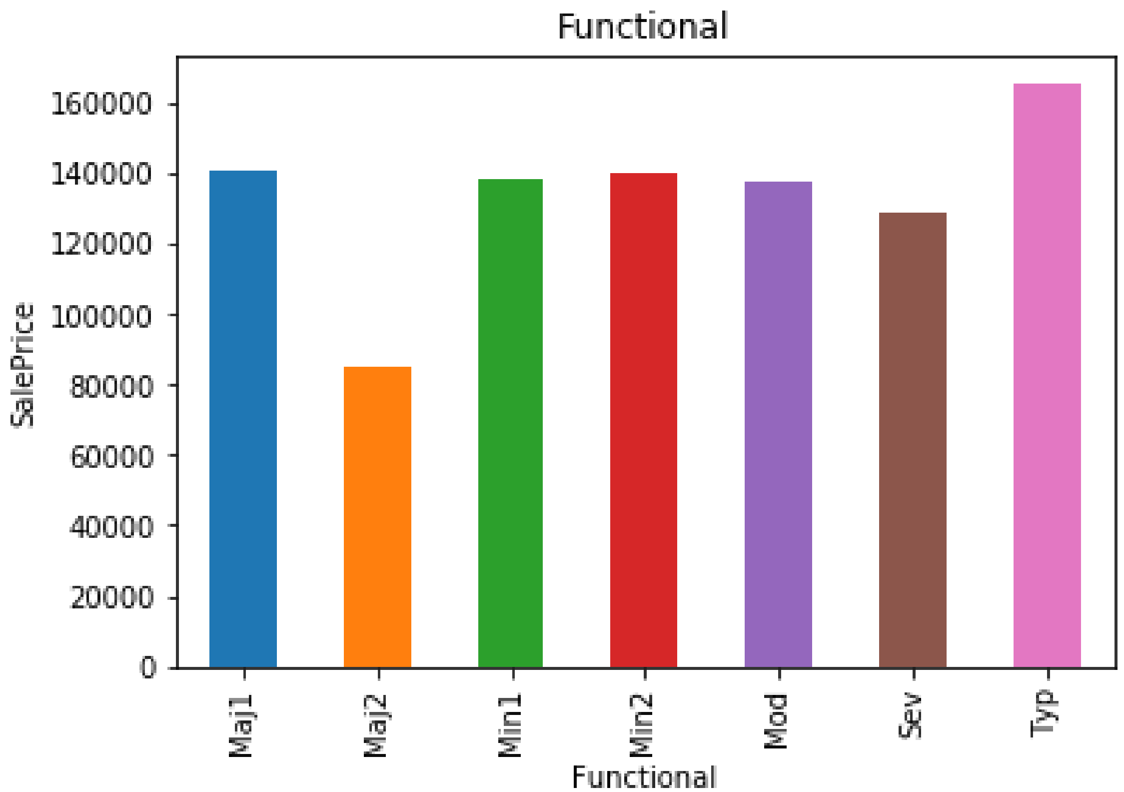



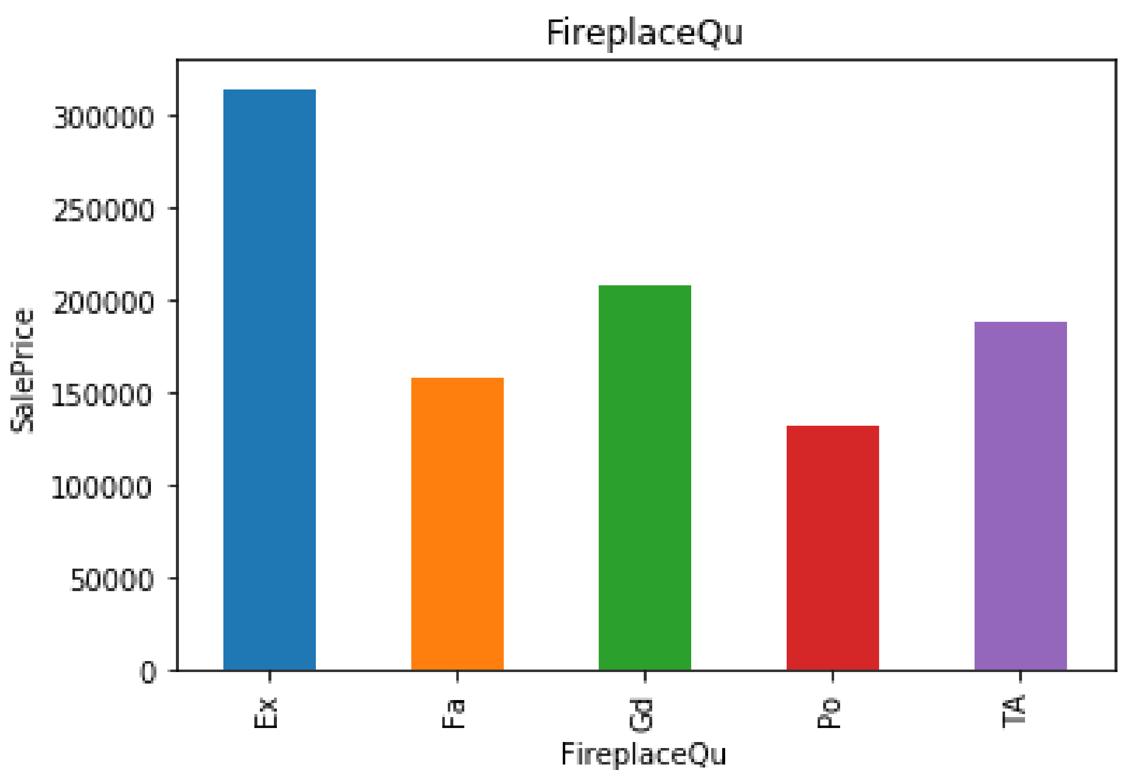
Electrical
SalePrice
FuseA
FuseF
FuseP
Mix
SBrkr
Electrical

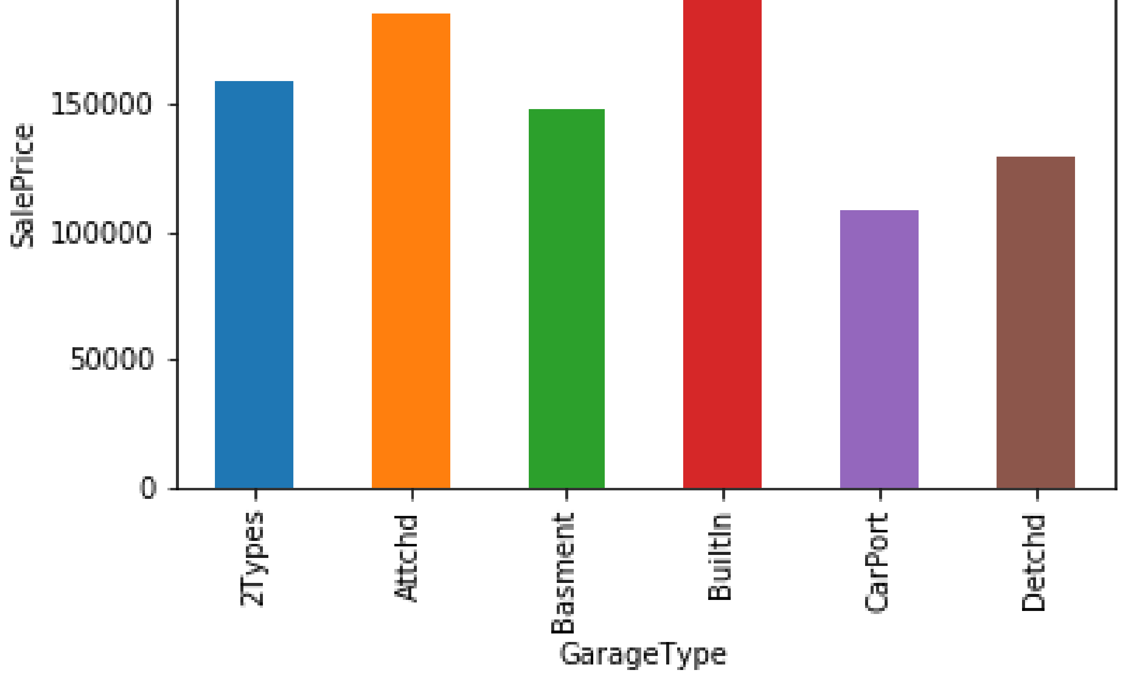
KitchenQual
SalePrice

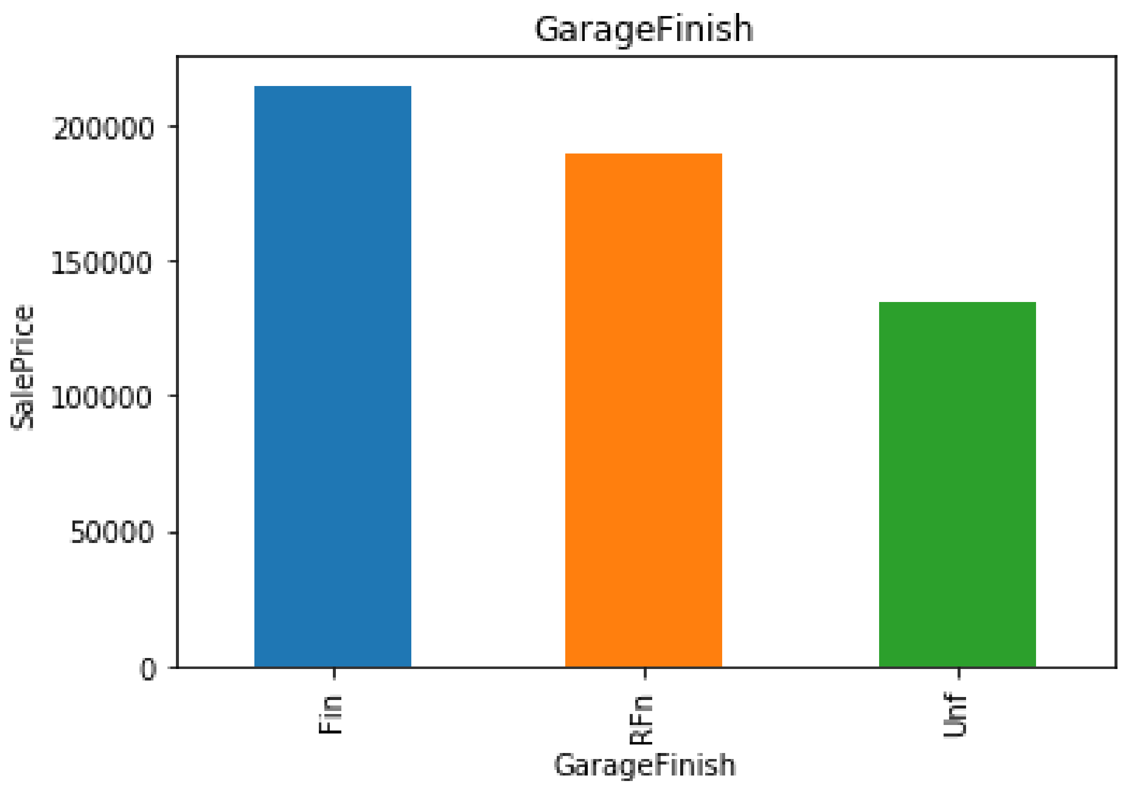

50000
0
Ex
Fa
Gd
TA
KitchenQual

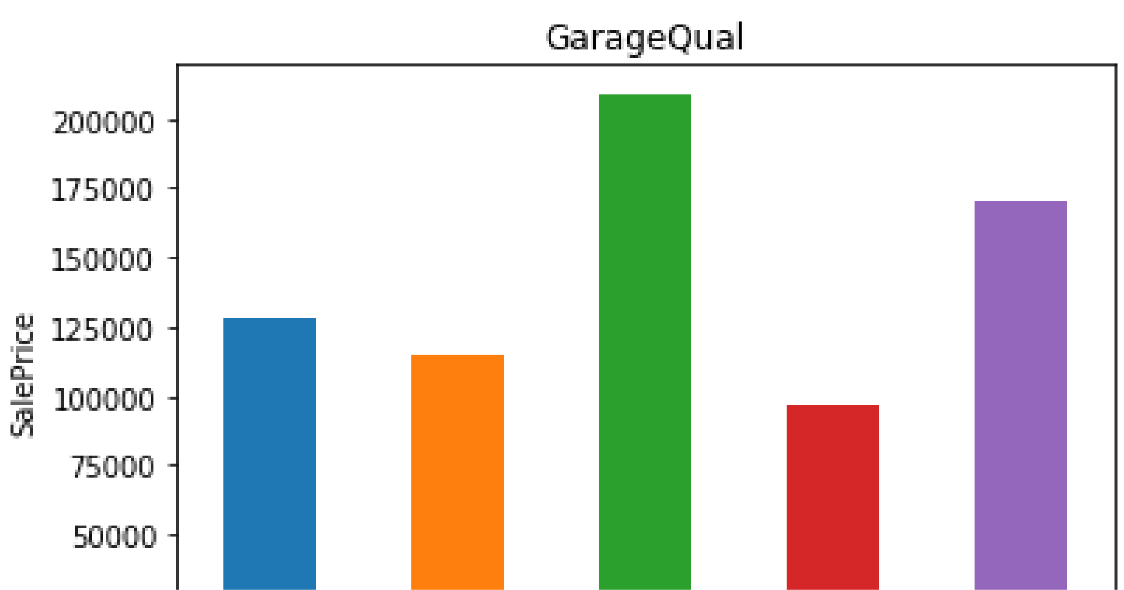

Functional
160000
140000
120000
100000
80000
60000
40000
20000
0
SalePrice
Maj1
Maj2
Min1
Min2
Mod
Sev
Typ
Functional

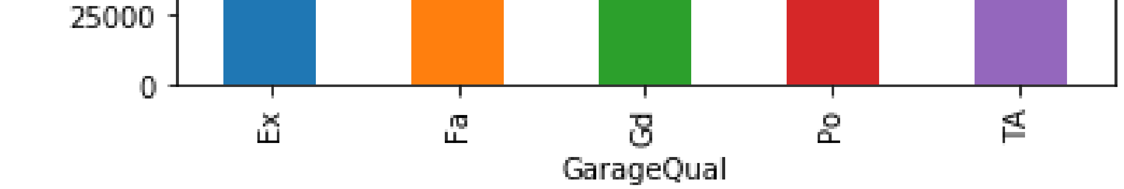

FireplaceQu
300000
250000
200000
150000
100000
50000
0
SalePrice
Ex
Fa
Gd
Po
TA
FireplaceQu

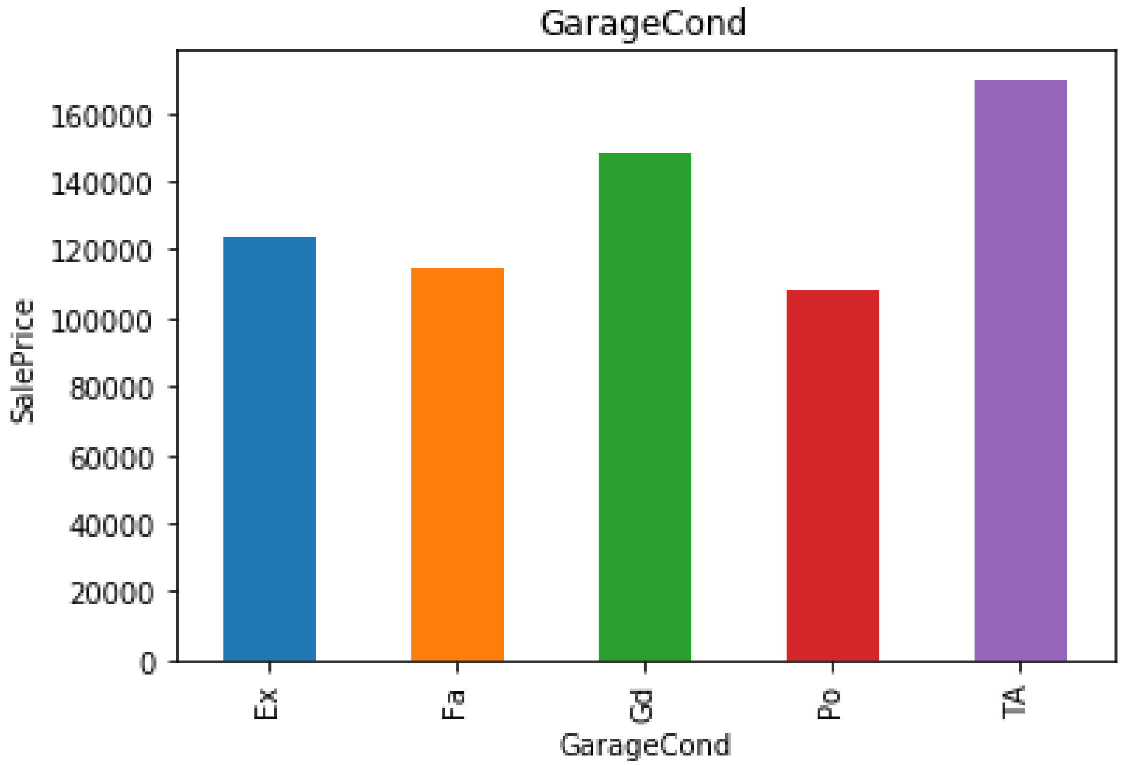

GarageType
200000

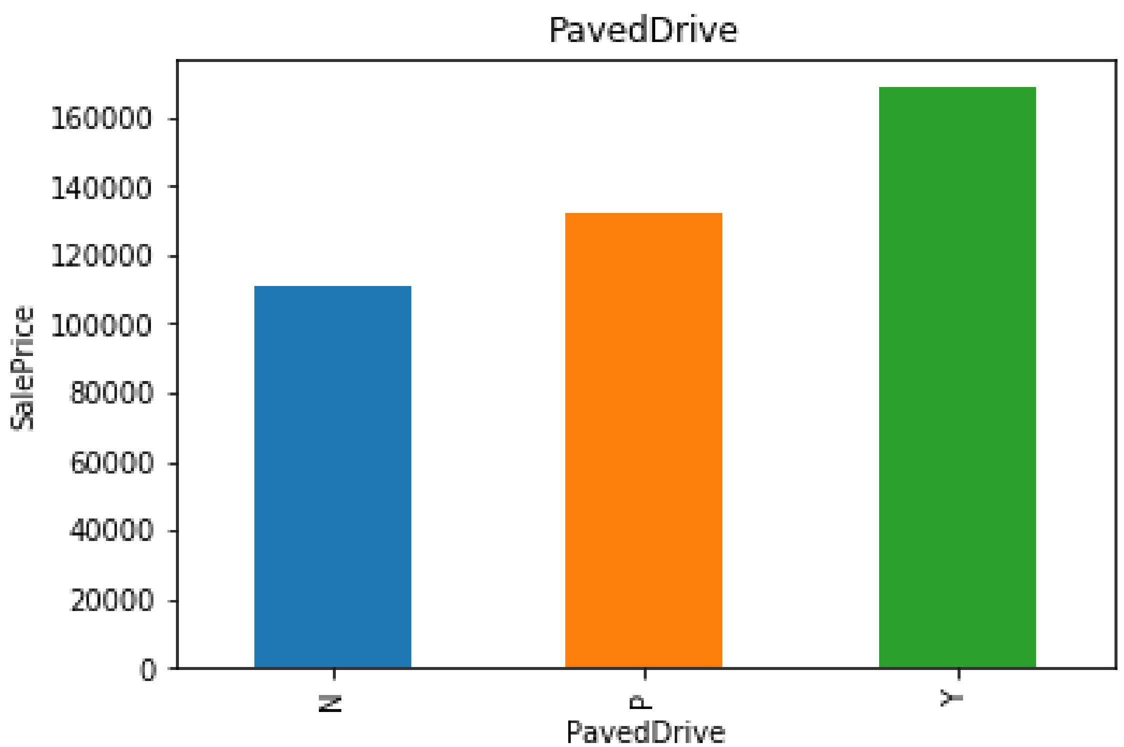

SalePrice
150000
100000
50000
0
2Types
Attchd
Basment
BuiltIn
CarPort
Detchd
GarageType

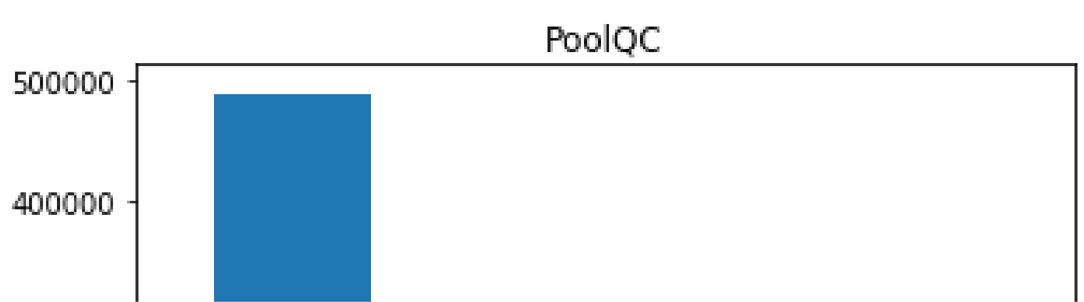

GarageFinish
SalePrice
200000
150000
100000
50000
0
Fin
RFn
Unf
GarageFinish

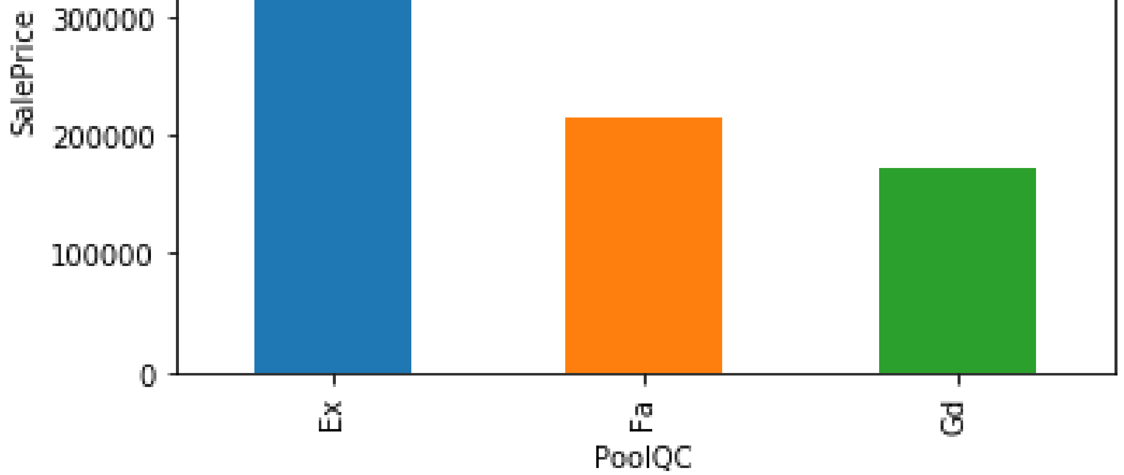

GarageQual
SalePrice
200000
175000
150000
125000
100000
75000
50000

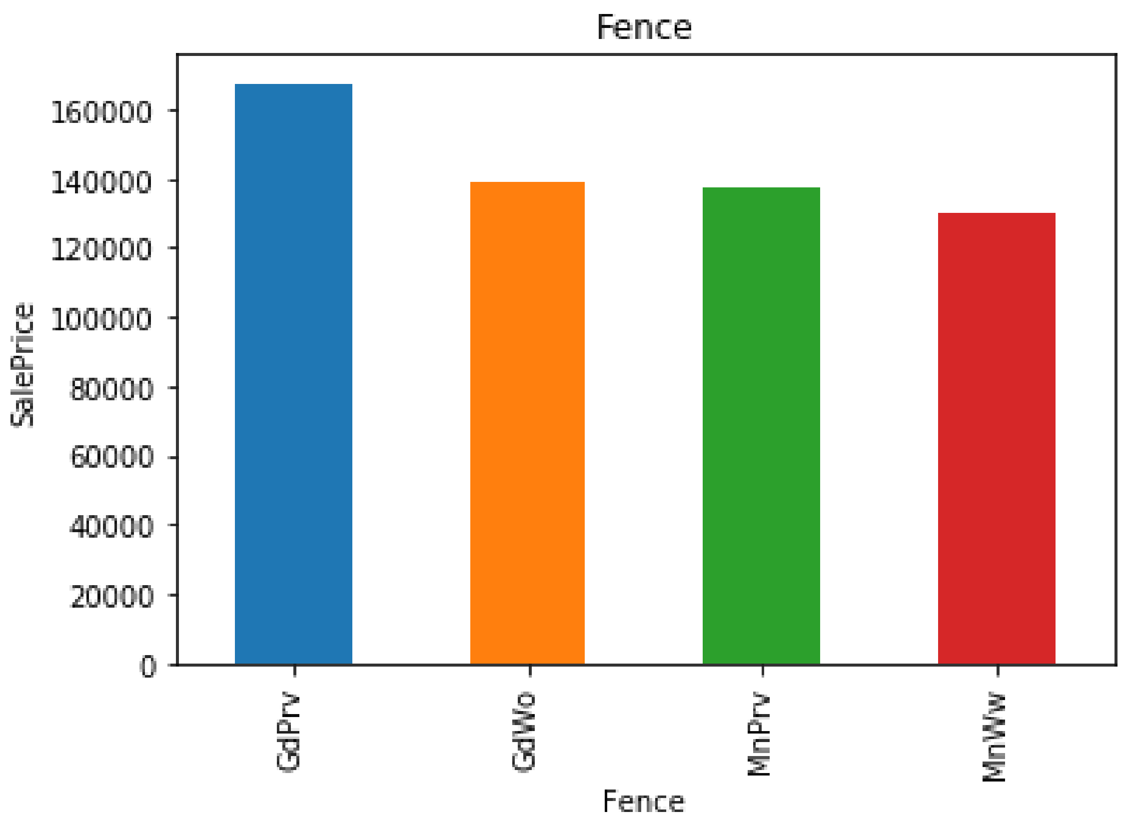

25000
0
Ex
Fa
Gd
Po
TA
GarageQual

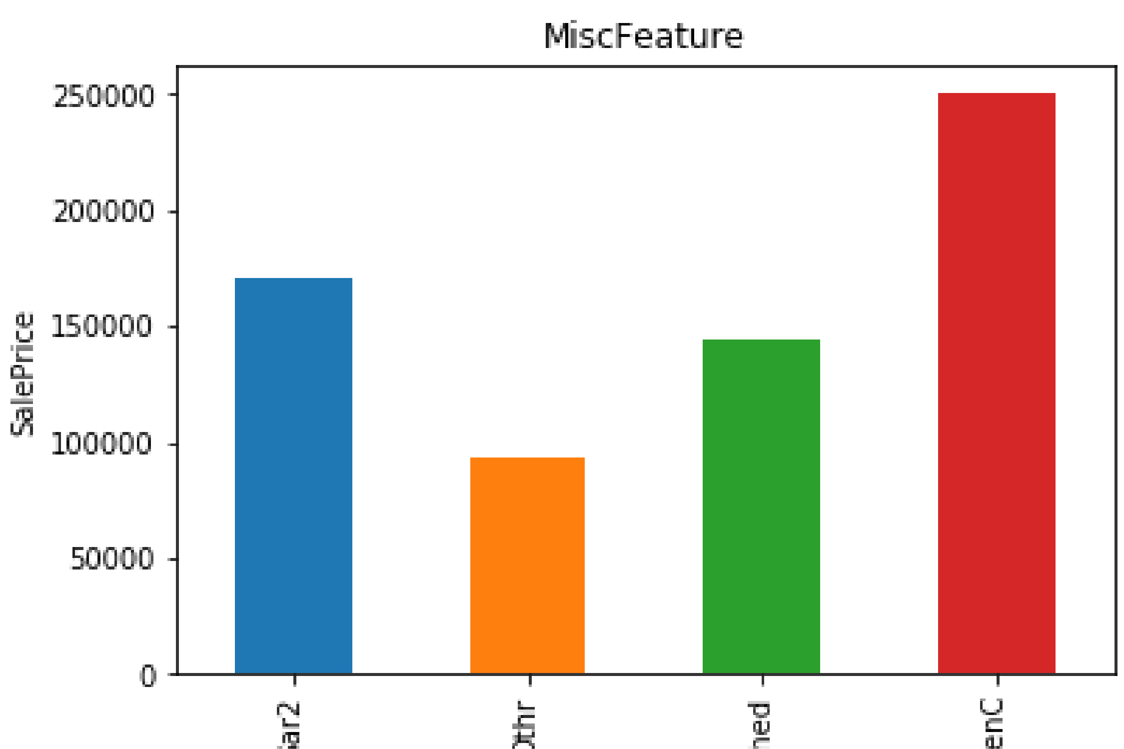

GarageCond
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Ex
Fa
Gd
Po
TA
GarageCond


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

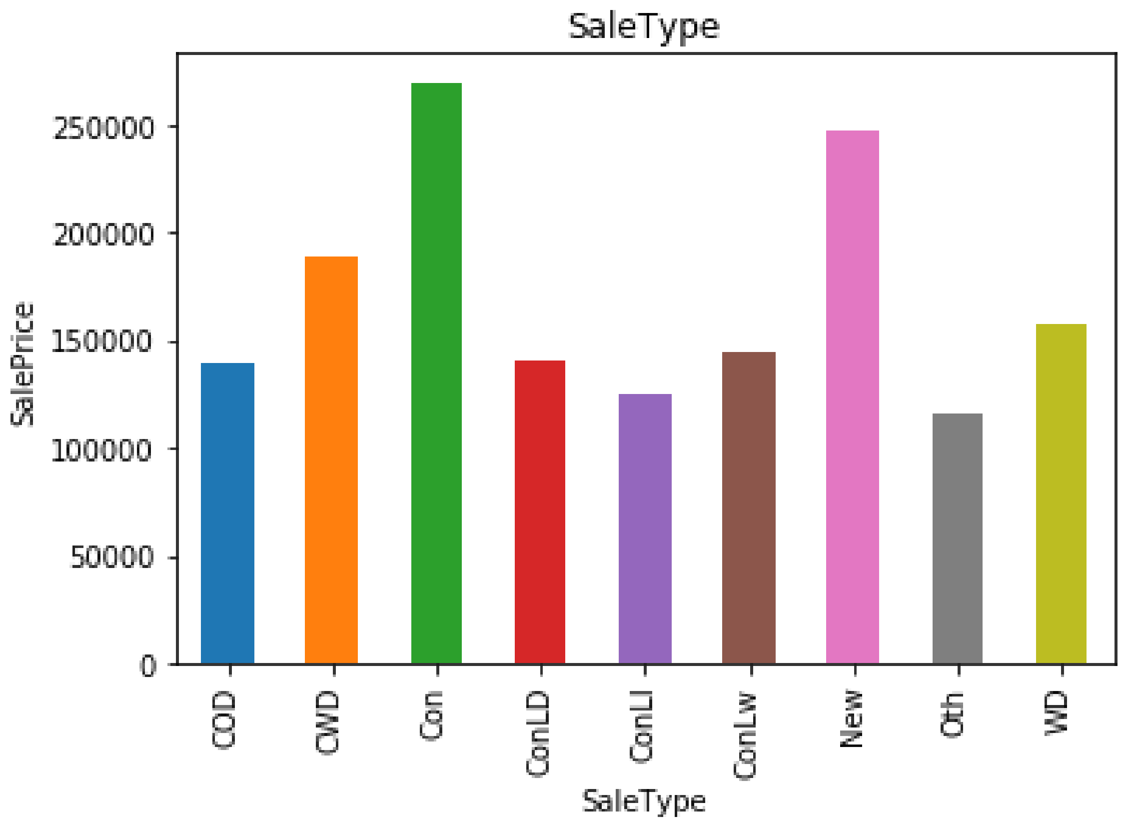

PoolQC
500000
400000

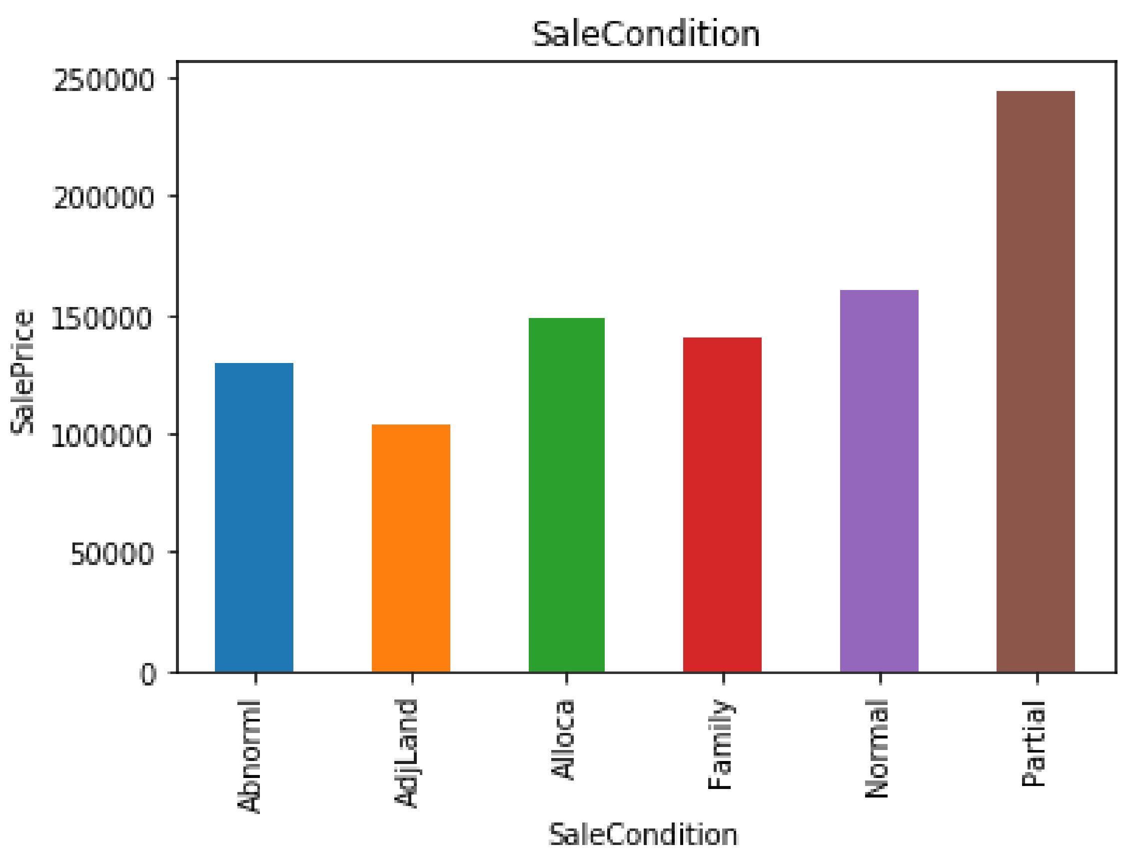

SalePrice
300000
200000
100000
0
Ex
Fa
Gd
PoolQC


Fence
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
GdPrv
GdWo
MnPrv
MnWw
Fence


MiscFeature
SalePrice
250000
200000
150000
100000
50000
0


MiscFeature


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


SaleCondition
250000
200000
150000
100000
50000
0
SalePrice
Abnorml
AdjLand
Alloca
Family
Normal
Partial
SaleCondition

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```

```

In [ ]:

```

```