Housing Price Prediction

Submitted by:
ABDULMATEEN DHALAYAT

ACKNOWLEDGMENT

“House Price Prediction Using Machine Learning” by G.NagaSatish,Ch.V. Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu in the Year july2019

This paper tells that Machine learning plays a major role from past years inimagedetection,spamreorganization,normalspeechcommand,productrecommendation and medical diagnosis. Present machine learning algorithmhelpsusinenhancingsecurityalerts,ensuringpublicsafetyandimprovemedical enhancements. Machine learning system also provides better customerservice and safer automobile systems. In this paper They discuss about thepredictionoffuturehousingpricesthatisgeneratedbymachinelearningalgorithm. For the selection of prediction methods we compare and explorevarious prediction methods. We utilize lasso regression as our model because ofits adaptable and probabilistic methodology on model selection. Their resultexhibit that their approach of the issue need to be successful, and has the abilitytoprocesspredictionsthatwouldbecomparativewithotherhousecostpredictionmodels.Moreoveronotherhandhousingvalueindices,theadvancement of a housing cost prediction that tend to the advancement of realestate policies schemes. This study utilizes machine learning algorithms as aresearch method that develops housing price prediction models. We create ahousingcostpredictionmodelInviewofmachinelearningalgorithmmodelsfor example, XGBoost, lasso regression and neural system on look at their orderprecision execution. We in that point recommend a housing cost predictionmodel to support a house vender or a real estate agent for better informationbasedonthevaluationofhouse.Thoseexaminationsexhibitthatlassoregression algorithm, in view of accuracy, reliably outperforms alternate modelsinthe execution ofhousingcostprediction.

“Predicting House Price With a Memristor-Based Artificial Neural Network” by J.J Wangi, S.G.Hu, X.TZhani, Q.LUO1, Q.YU1, Zhen Liu, T.PChen, Y.Yin, SumioHosaka, Y.Liu in the year 2018

This paper tells about that Synaptic memristor has attracted much attention forits potential applications in artificial neural networks (ANNs). However usefulapplications in real life with such memristor-based networks have seldom beenreported. In this paper, an ANN based on memristors is designed to learn amulti-variable regression model with a back-propagation algorithm. A weight unit circuit based omemristor, which can be programmed as an excitatory synapse orinhibitory synapse, isintroduced. The weight of the electronic synapse is determined by the conductance of the memristor, and the current ofthe synapse follows the charge-dependent relationship. The ANN has the abilityto learn from labelled samples and make predictions after online training. As anexample, the ANN was used to learn a regression model of the house prices ofseveral Boston towns in the USA and the predicted results are found to be closetothe targetdata

“House Planning and Price Prediction System using Machine Learning” by Mr.Rushikesh Naikare, Mr.Girish Gahandule, Mr.AkashDumbre, Mr. KaushalAgrawal, Prof. Chaitanya Mankarin the year December 2019

The housing sector has hike as it is the one of the basic need. Housing the maindomain of real estate. In the major metropolitan cities and the cities with manyprestigious Educational institutions and IT Parks have reasonable price increasein housing. Home buying plans can derails the family’s financial planning andother goals. Now a day’s house price changing rapidly according to variousparameters. The buyer gets confused in choosing his dream home as differencein price making it challenging. Both the buyer and seller should satisfy so they do not over estimate or underestimate price. So to build the platform where buyer can find home according to its needs and friendly to its financial condition. House price prediction on different parameters is our goal. Doing that we are going to use regression algorithms using machine learning on datasets it can extract features from dataset. Result of this approach provide maximumefficiency and minimum errors. We also propose to determine the plane forhousebuilding.

“House Price Prediction Using Machine Learning and RPA” by Prof.Pradnya Patil Assistant Professor, Computer Engineering Department Technology, Darshil Shah, Harshad Rajput, Jay Chheda in the year March 2020.

In today’s world, everyone wishes for a house that suits their lifestyle and provide serenities according to the needs. House prices keep on changing very frequently which proves that house prices are often exaggerated. There aremany factors that have to be taken into consideration for predicting house pricessuch as location, number of rooms, carpet area, how old the property is? andother basic local amenities. We will be using Cat Boost algorithm along with Robotic Process Automation for real-time data extraction. Robotic Process Automation involves the use of software robots to automate the tasks of dataextraction while machine learning algorithm is used to predict house prices with respect to the dataset.

INTRODUCTION

Problem statement:-

Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modeling, Market mix modeling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

A US-based housing company named **Surprise Housing** has decided to enterthe Australian market. The company uses data analytics to purchase houses at aprice below their actual values and flip them at a higher price. For the same purpose, the company has collected a dataset from the sale of houses in Australia. The company is looking at prospective properties to buy houses to enter themarket. You are required to build a model using Machine Learning in order topredict the actual value of the prospective properties and decide whether toinvestinthem or not

ExploratoryDataAnalysis

1. Checking the Missing Values

2. All the numerical Variables

3.Distribution of the Numerical Variables

4. categorical variables

5. cardinality of the categorical variables

6.Outliers

7.Relationship between dependent and independent feature (SalePrice)

1.Checking the missing values

Missing values in the dataset can be checked by below python code:-

```
missing_values=[x for x in df.columns if
df[x].isnull().sum()>1]print('Number of missing variable
columns:', len(missing_values))print("Missingvaluesin
thedataset:\n ",missing_values)
print("-
"*125)df[missing_values]
.head()
```

Observation:-

Number of missing variable columns:
18 Missing values in thedataset :
['LotFrontage', 'Alley', 'MasVnrType', 'MasVnrArea', 'BsmtQual', 'BsmtCond','BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'FireplaceQu', 'GarageType','GarageYrBlt', 'GarageFinish', 'GarageQual', 'GarageCond', 'PoolQC' , 'Fence', MiscFeature']

2. Checking the percentage of the missing values

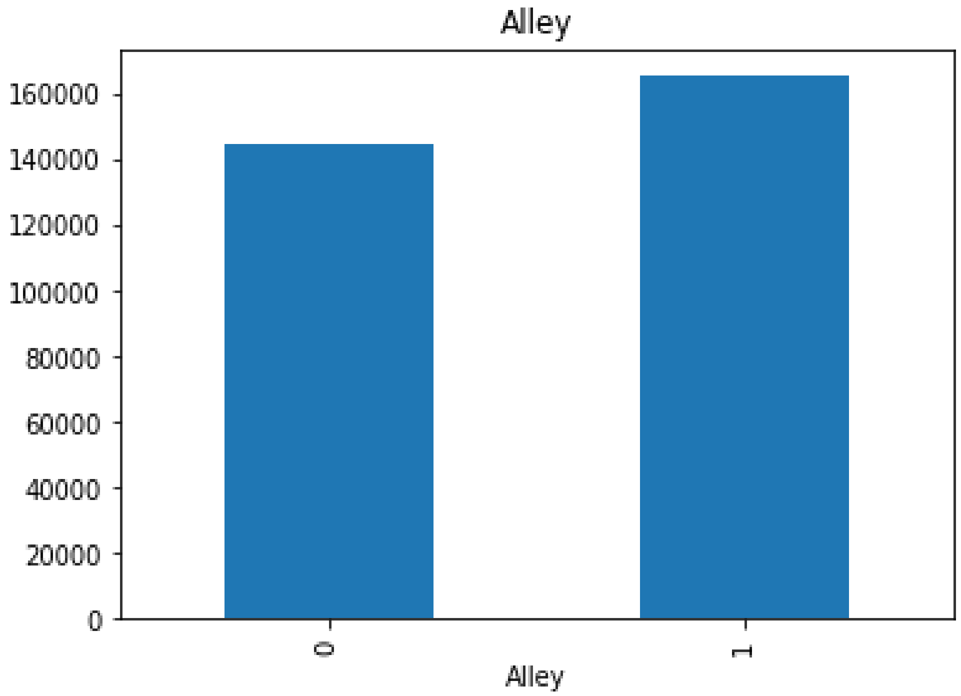
Missing values percentage can be checked by the below python code:-

```
For feature in missing_values:
    print(feature, np.round(df[feature].isnull().mean()*100,4), "%
MissingValues")
```

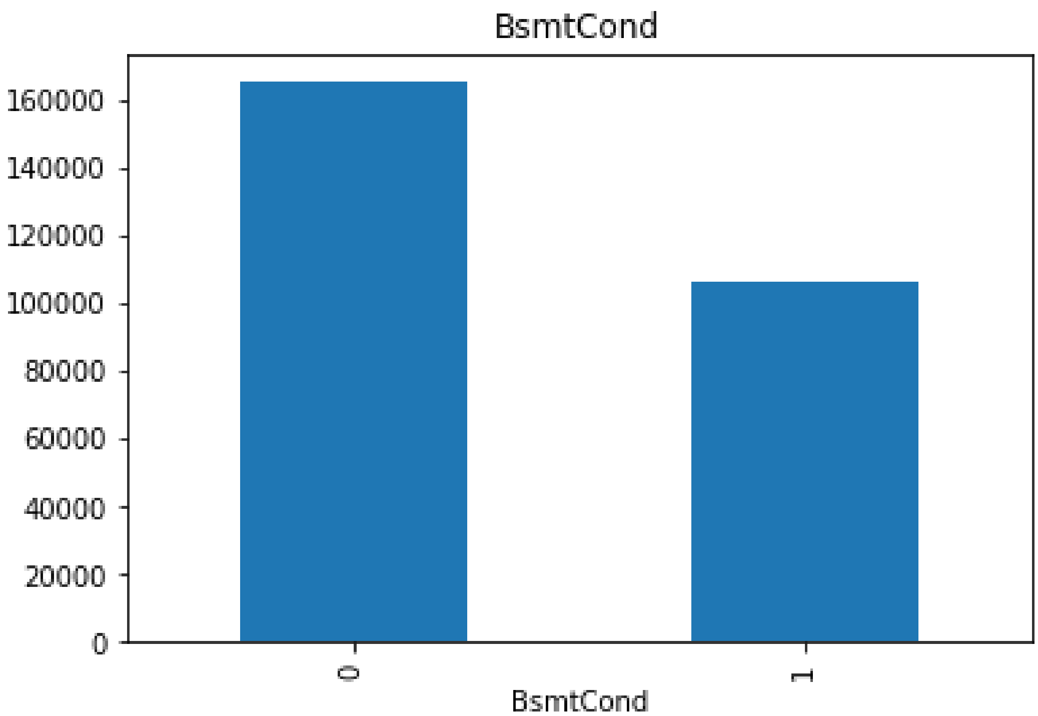
Observation:-

1. There are many missing values in the columns of the dataset
2. Hence need to check the relationship with sales price

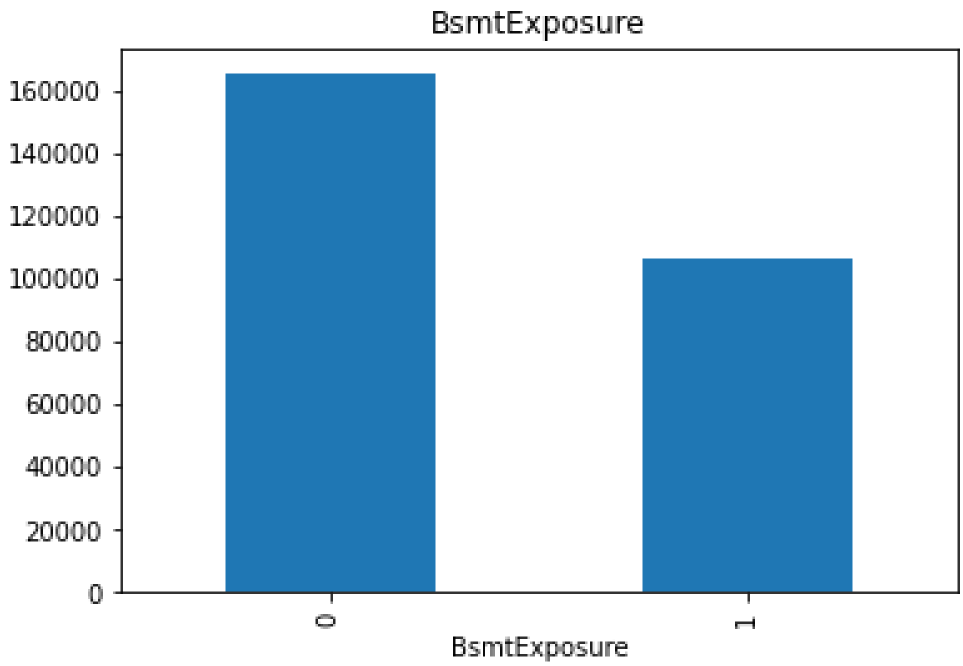
Representation of Missing values vs Sales price



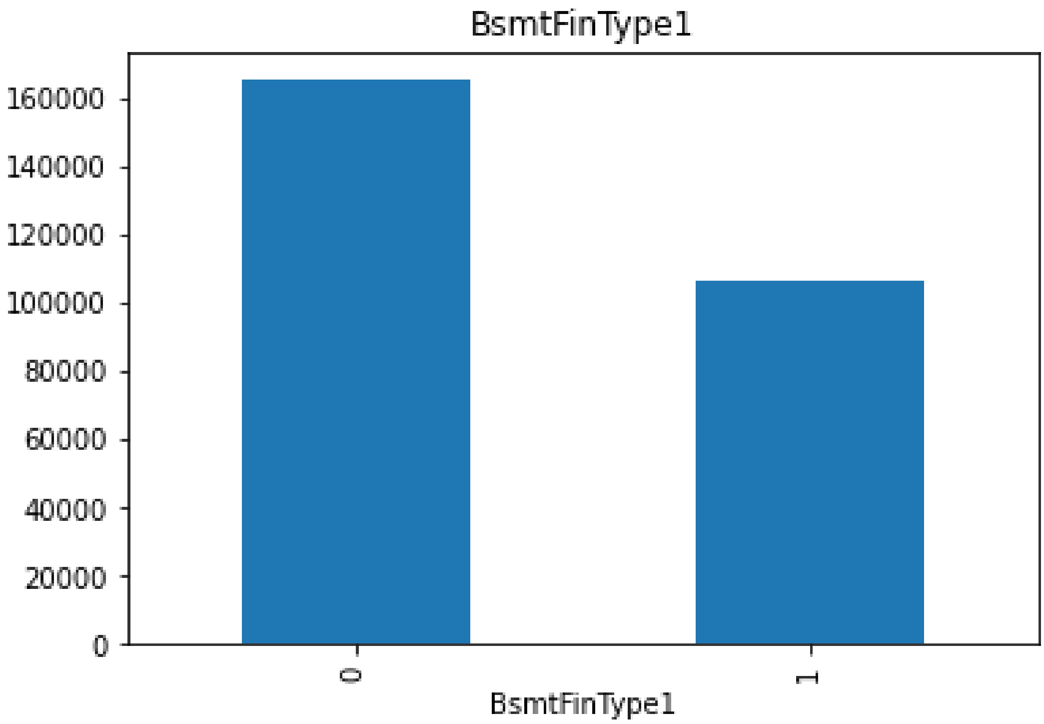
Missing value vs alley



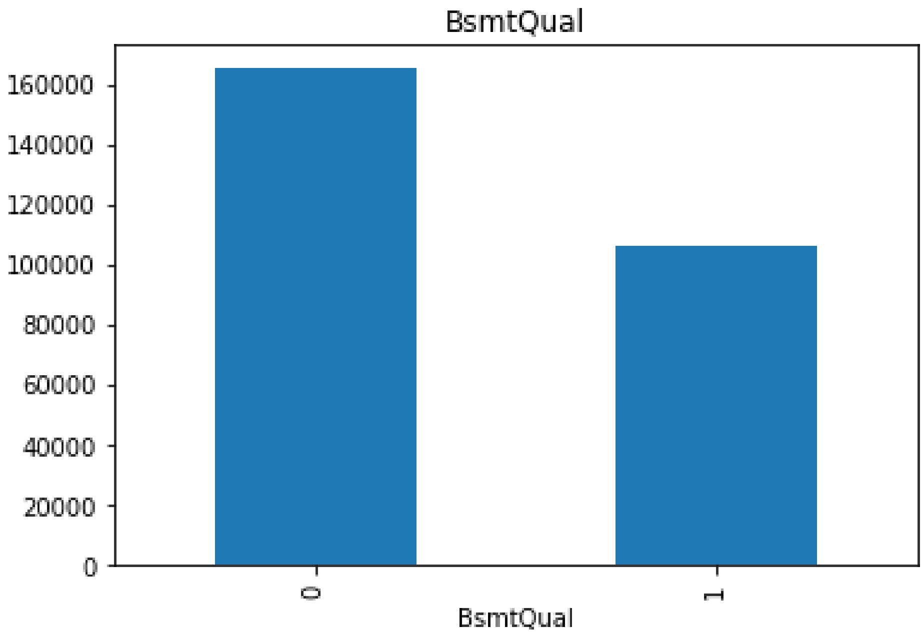
Missing value vs Bsmtcond



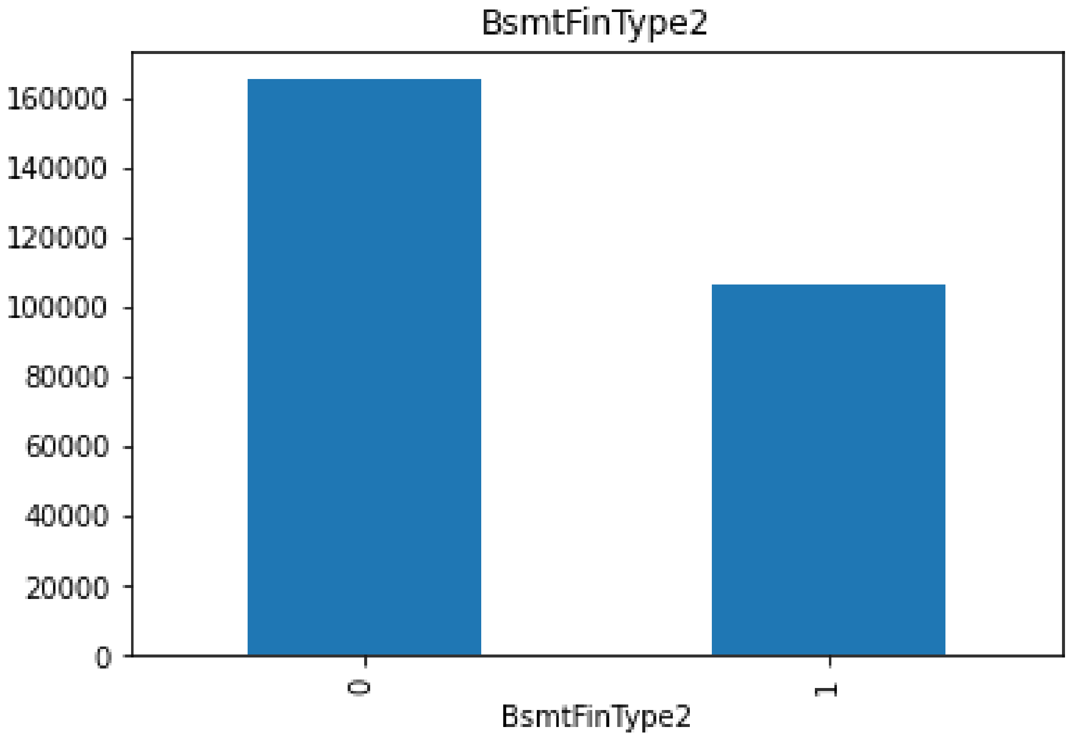
Missing value vs bsmtExposure



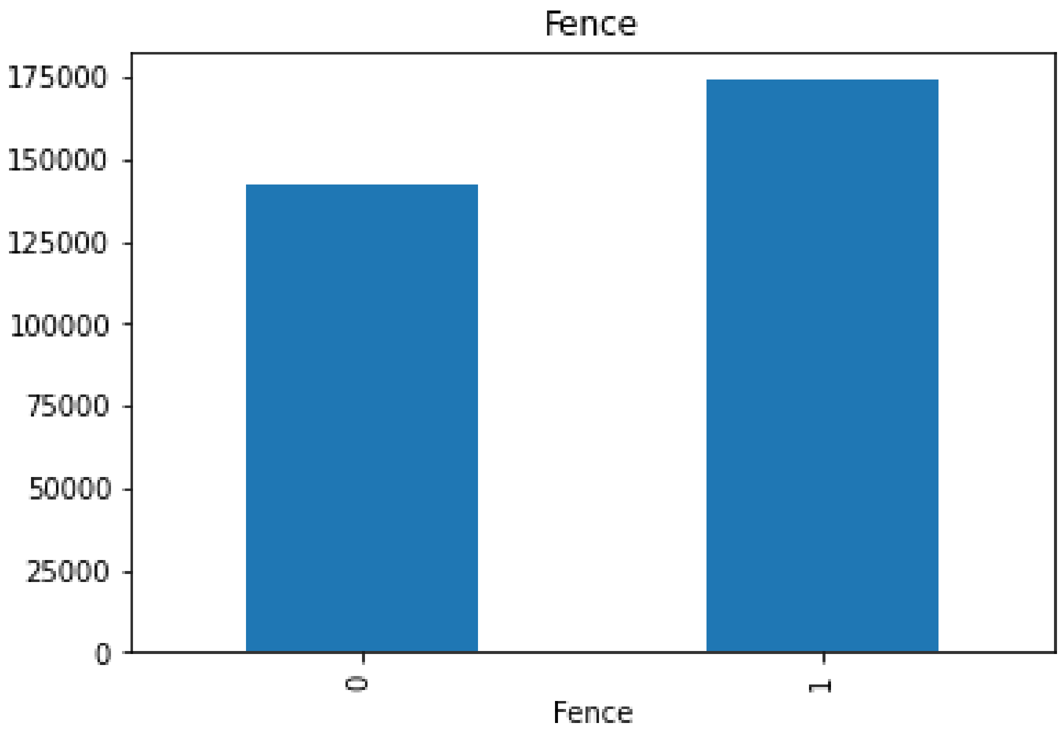
Missing value vs BsmtFintype1



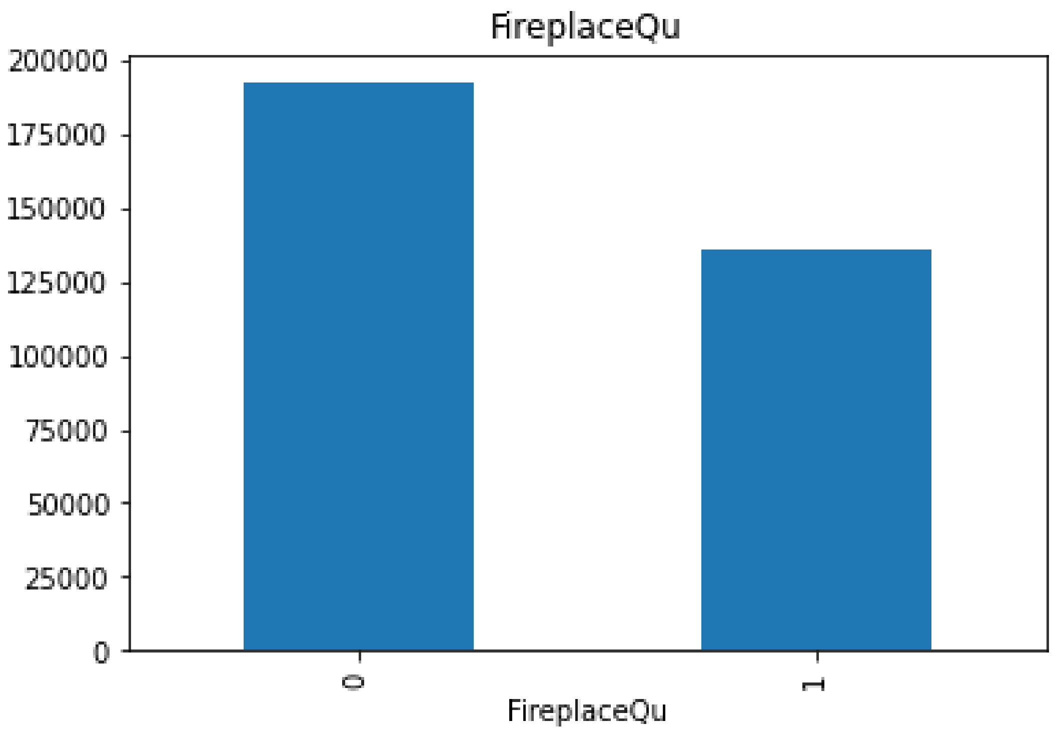
Missing value vs Bsmtqual



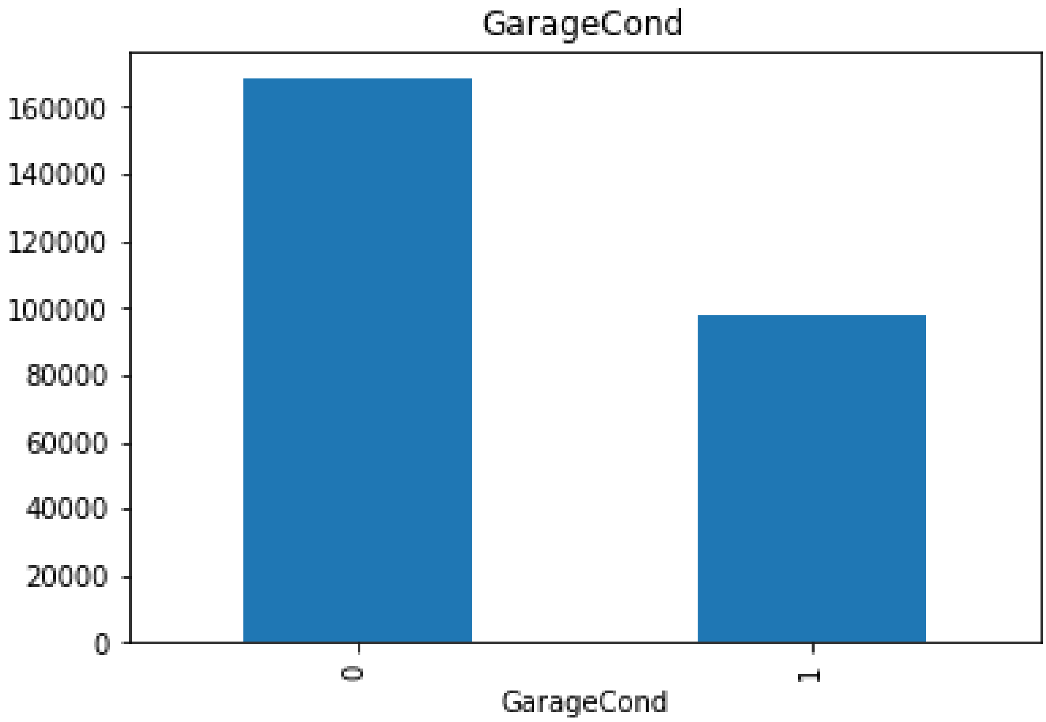
Missing value vs Bsmttype2



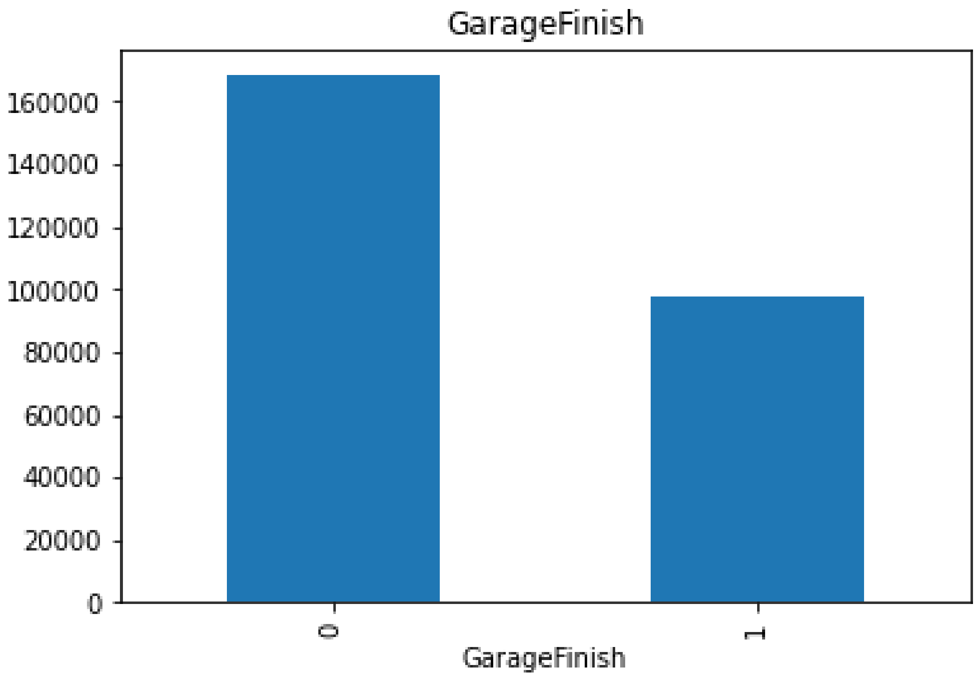
Missing value vs Fence



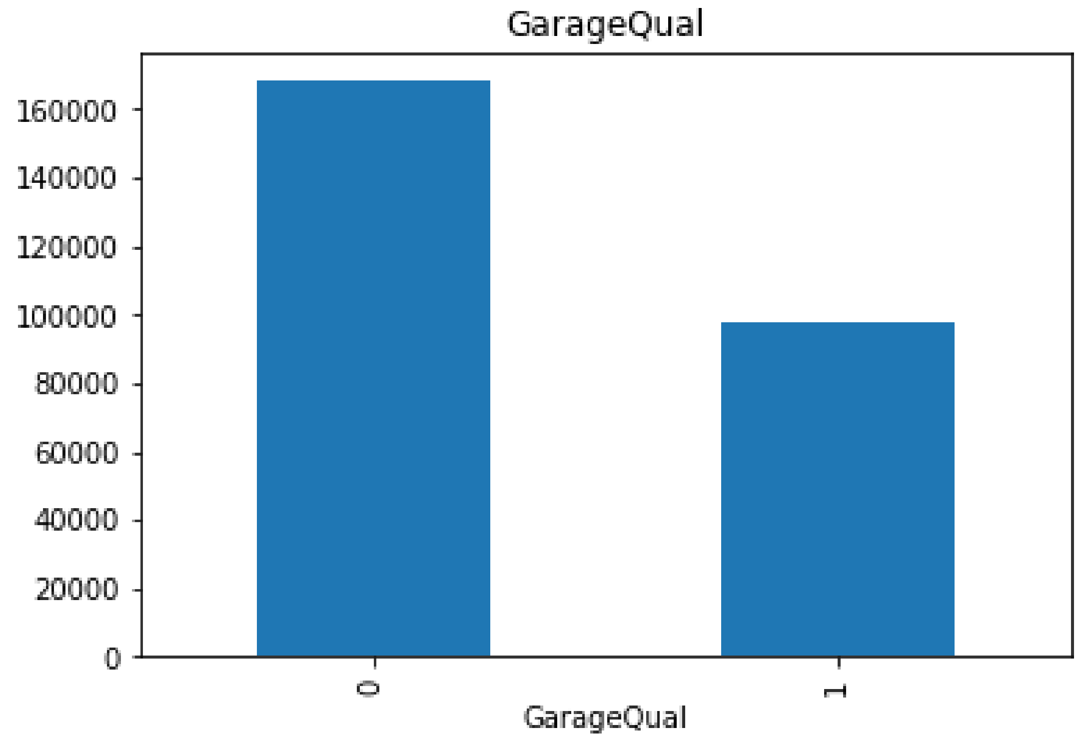
Missing value vs FireplaceQu



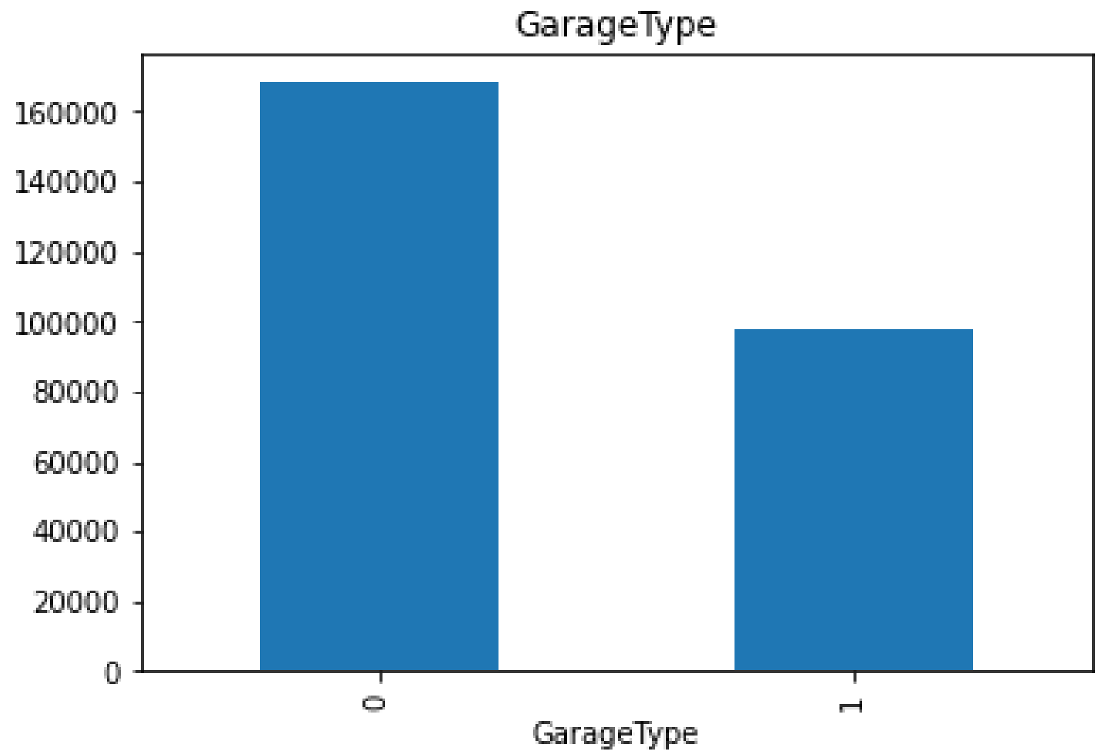
Missing value vs GarageCond



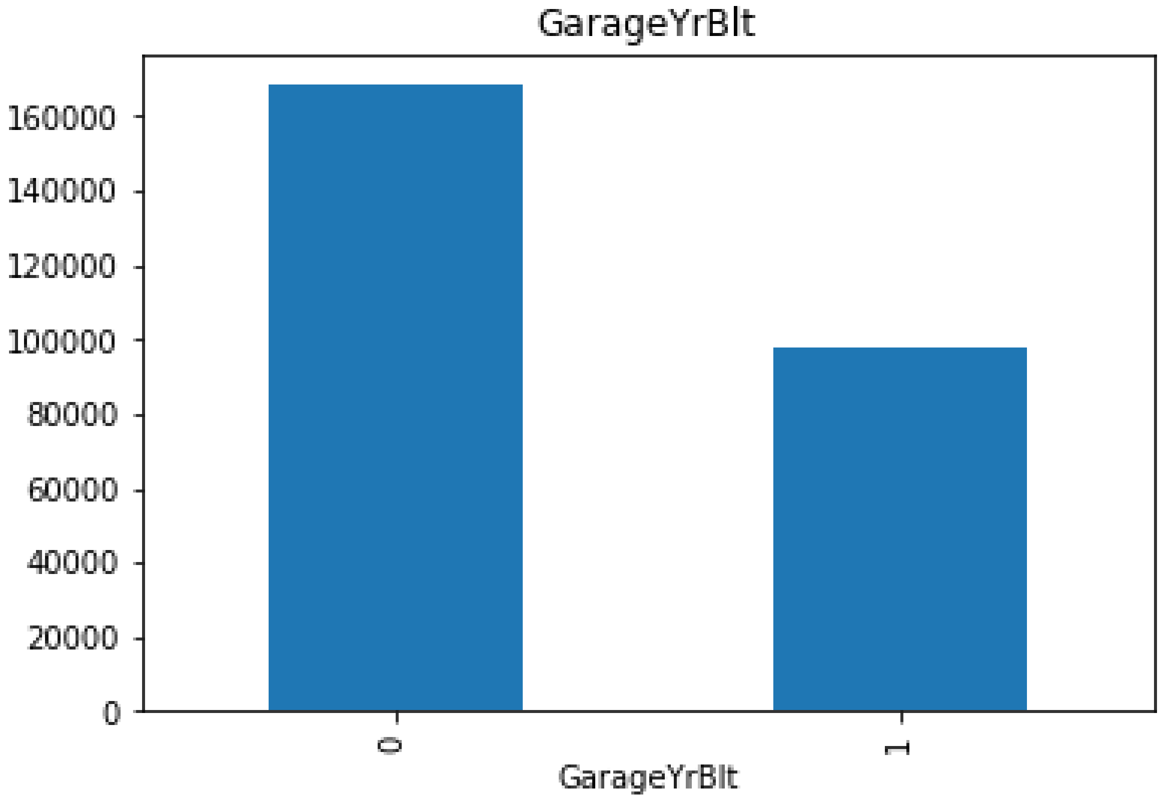
Missing value vs GarageFinish



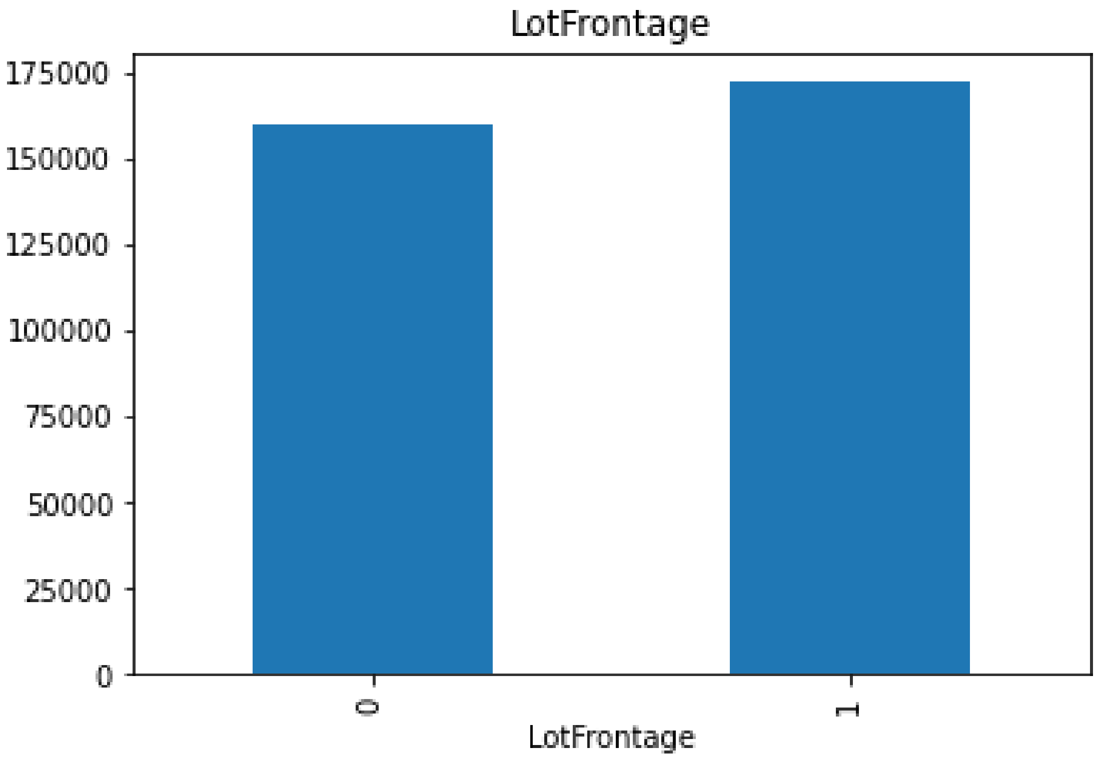
Missing value vs GarageQual



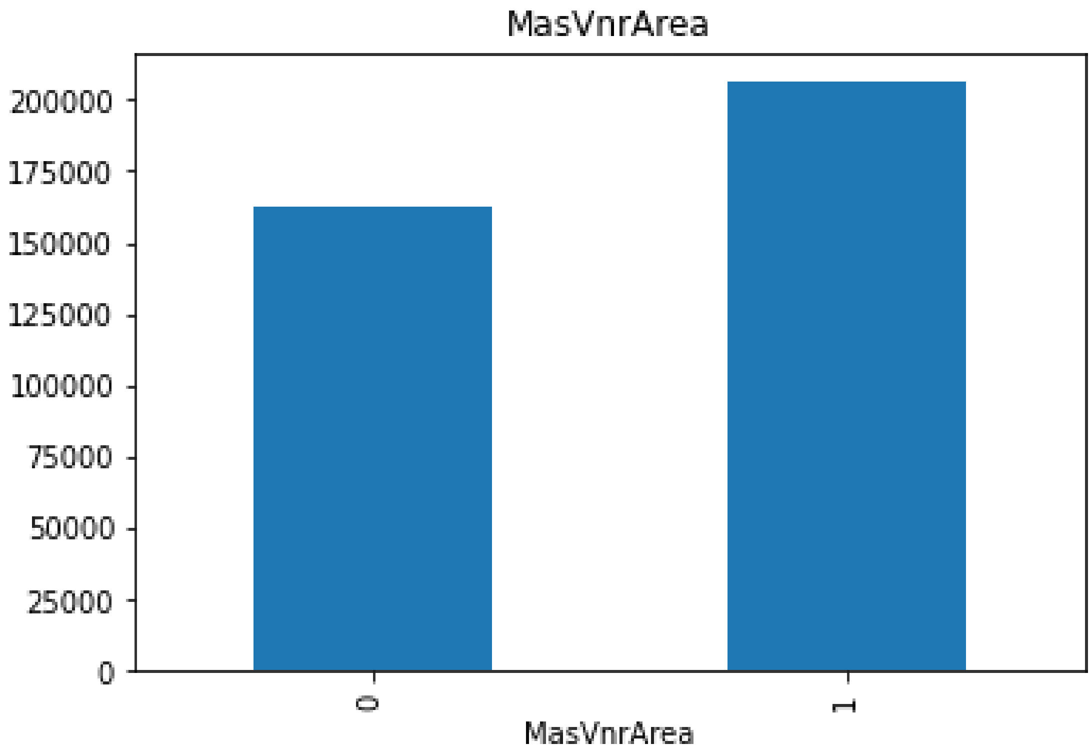
Missing value vs GarageType



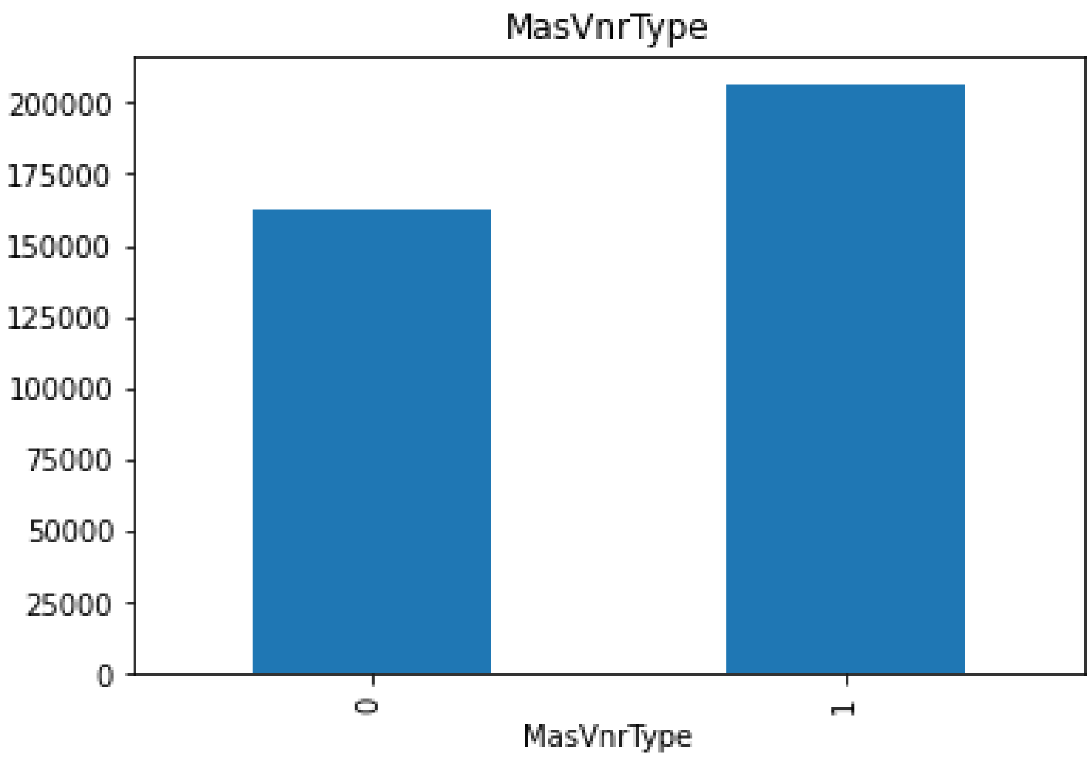
Missing value vs GarageYrBuilt



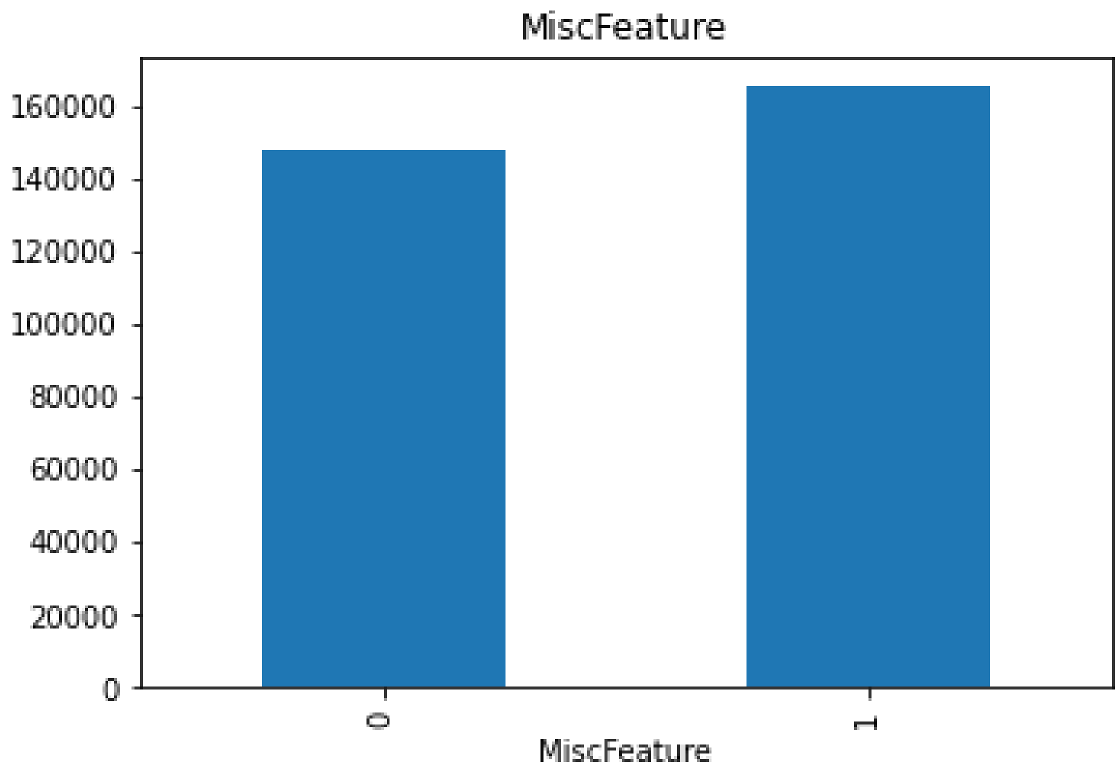
Missing value vs Lotfrontage



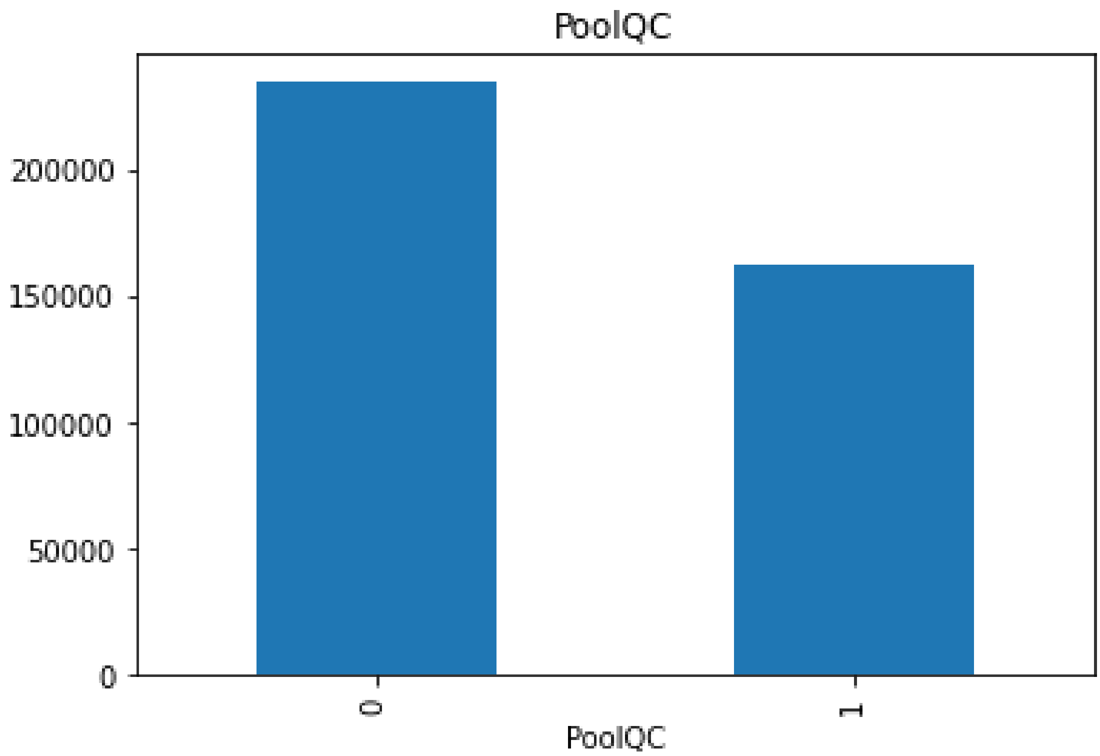
Missing value vs Mass VnArea



Missing value vs MassVnrType



Missing value vs MiscFeature



Missing value vs PoolQC

3.Extracting all the numerical feature

Extracting all the numerical values using python code:-

numerical_features=[xforxindf.columnsifdf[x].dtypes!="O"]

print("The number of the numerical columns in the dataset:",len(numerical_features))

print("Numerical columns in the dataset:\n",

numerical_features)print("-"*125)

df[numerical_features].head()

Observation:-

1. 'YearBuilt','YearRemodAdd','GarageYrBlt','YrSold' are date columns wehaveinthisdataset.

2. From the datatime column we usually extract the no of days, years, hours,minutes etc.hencethiscan bederivedfromthecolumns.

4. Extracttheyearcolumn fromthedataset:

Extract the year column from the dataset using the python code:-

year_feature=[x for x in df.columns if 'Yr' in x or 'Year' in

x]print("The number of Year column in the dataset

:",len(year_feature))print("Yearcolumnsin the dataset

:\n",year_feature)

print("-

"*125)df[year_feature]

.head()

5.Checkingtheuniqueitemsindate timecolumns

Checking the unique items in datetime columns using the python code:-

checkingtheuniqueitemsinthedatetimecolumns

forfeatureinyear_feature:

print("Theuniqueitemsinthecolunmn",feature,":\n",df[feature].unique())

Relationship between feature vs Saleprice

relationship between year variables and SalePrice can be done using thepython code

for feature in

year_feature:plt.figure(figsize=(8,6))df.groupb

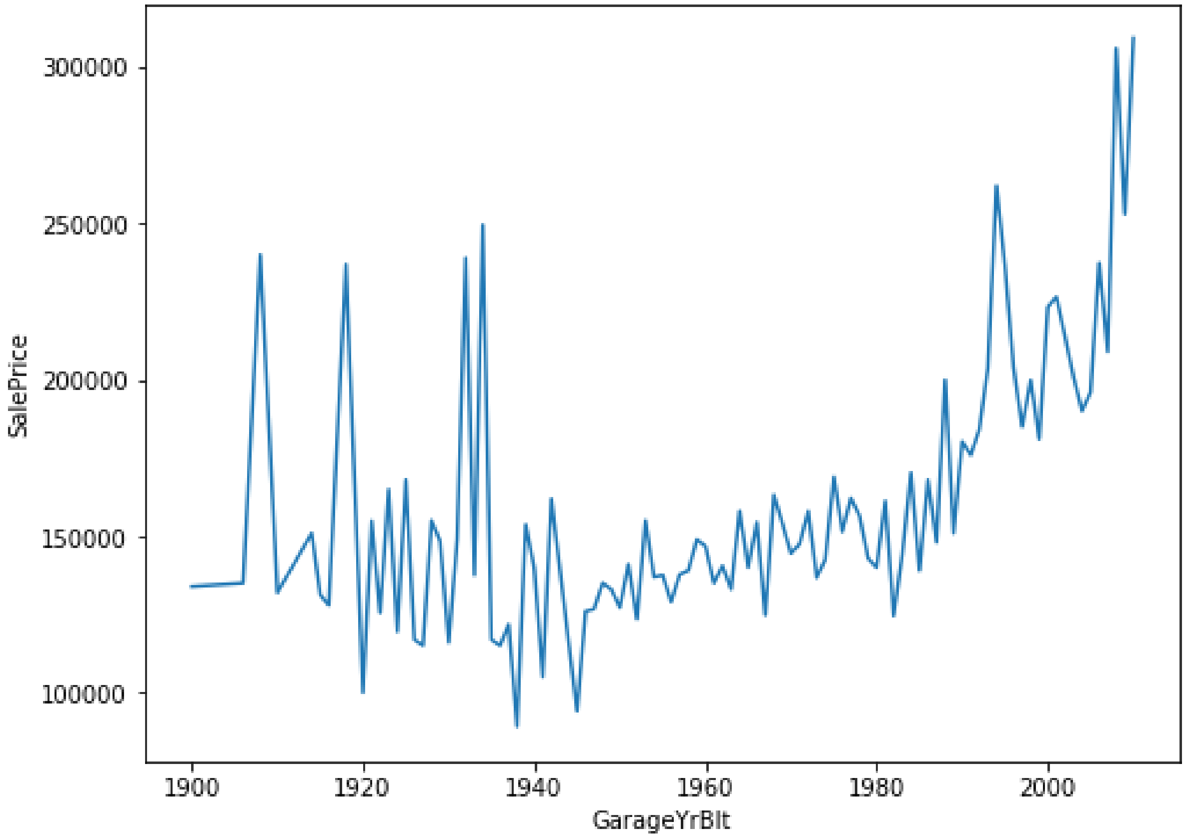
y(feature)['SalePrice'].median().plot()plt.xlabel

(feature)

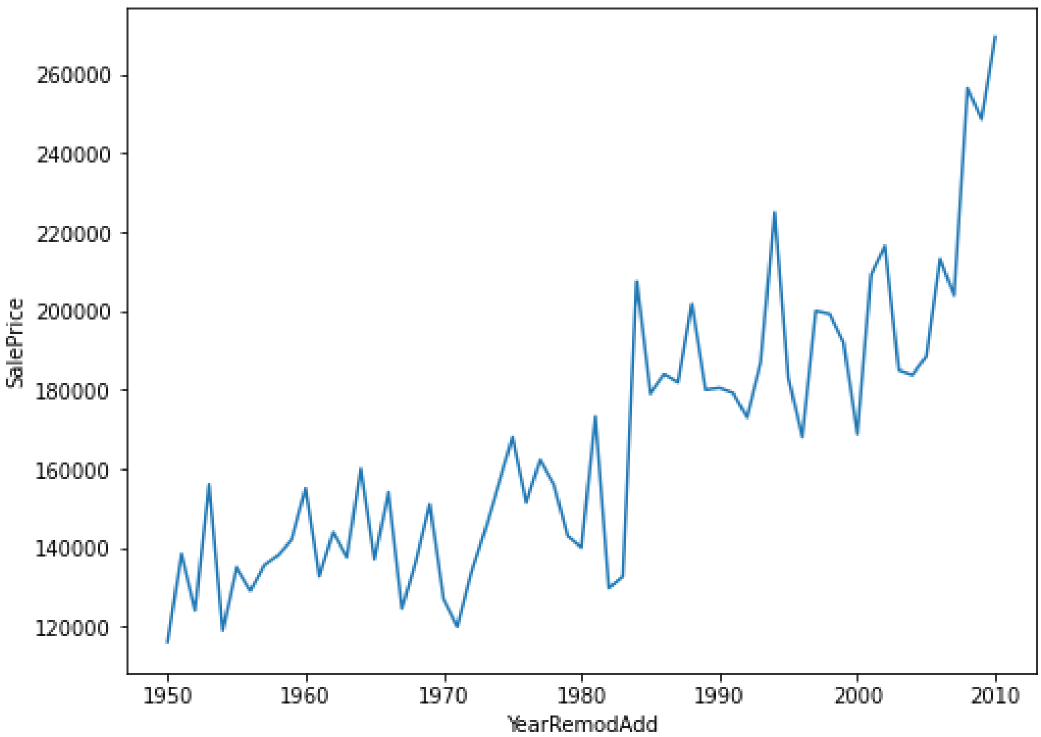
plt.ylabel('SalePrice')

plt.show()

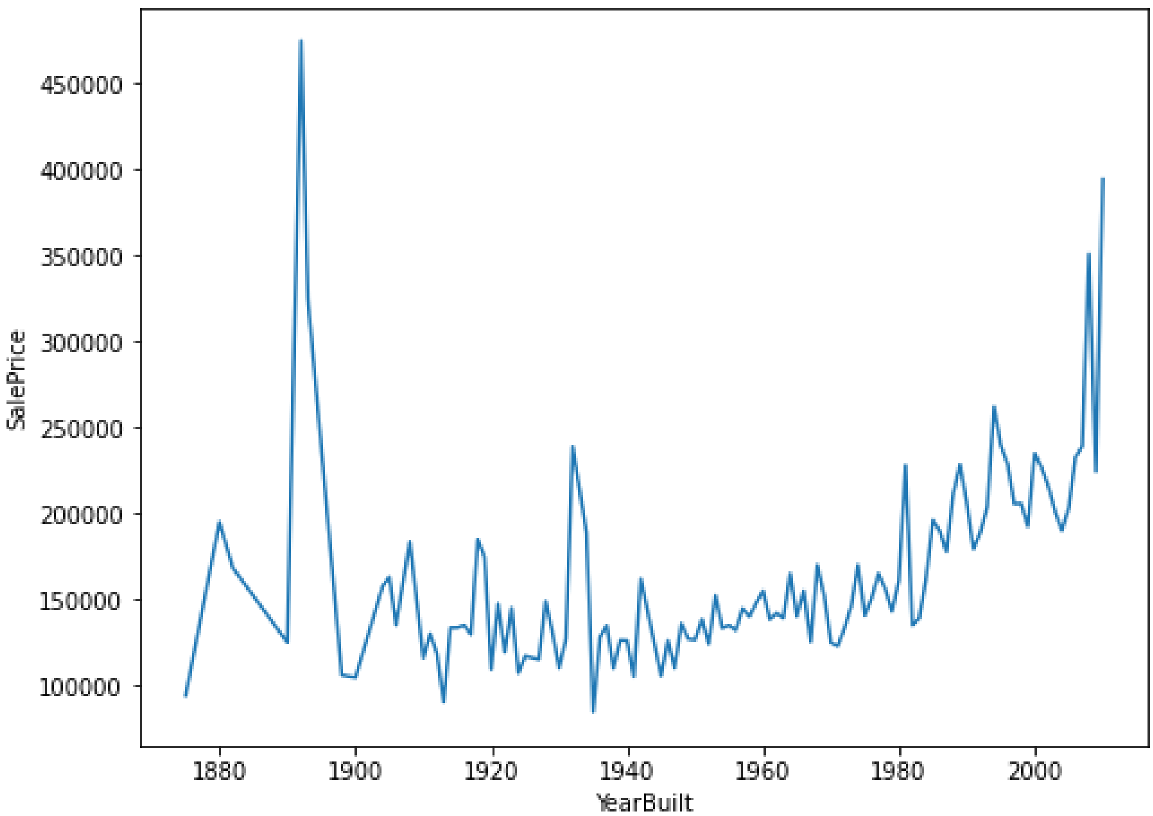
DataVisualization:-



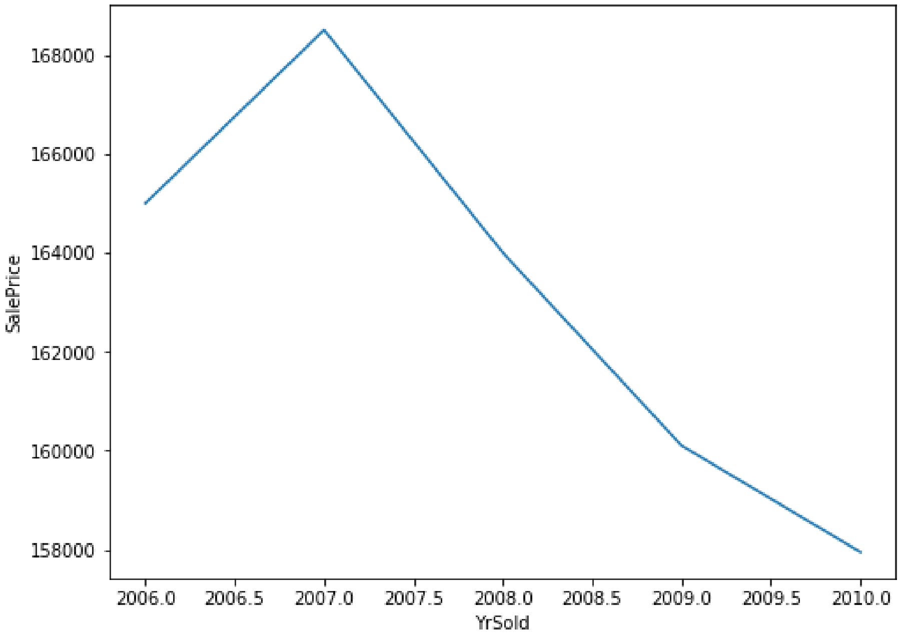
SalePricevsGarageBelt



YearRemodAddvsSalePrice



SalePricevsYearBuilt



YrSoldvsSalePrice

Extracting the discrete and continous variable using the python code:-

discrete_feature=[x for x in numerical_features if len(df[x].unique())<25 and xnot inyear_feature+['Id']]

print("The number of discrete column in the dataset:",

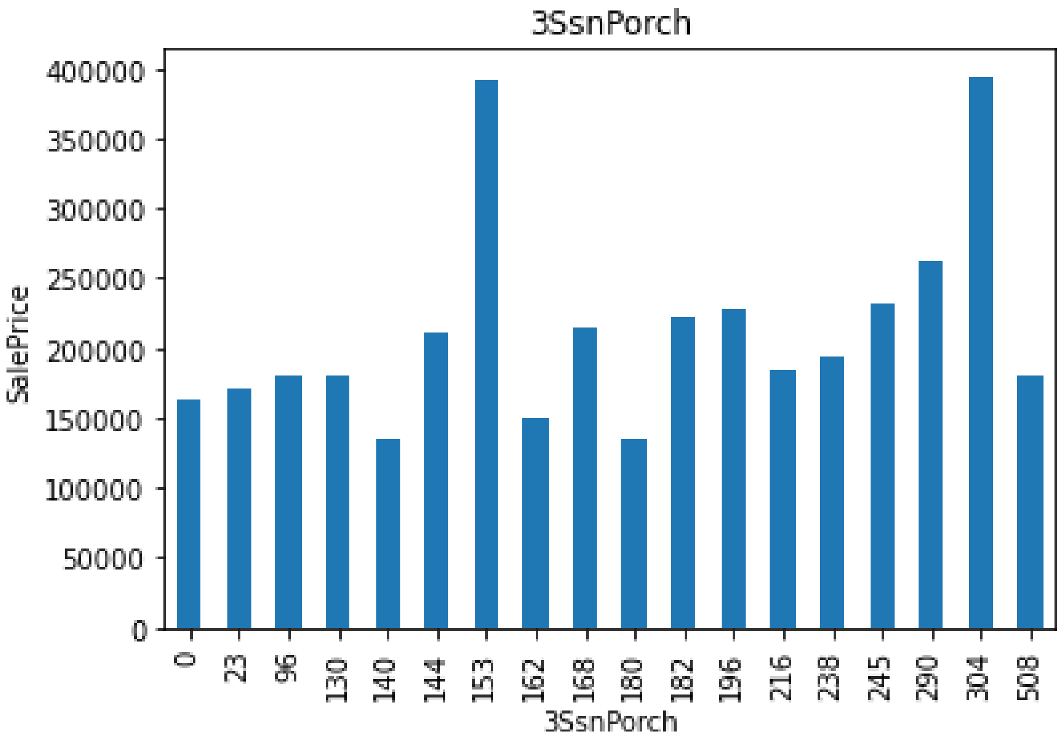
len(discrete_feature))print("Discretecolumnsinthedatset:\n",discrete_feature

)

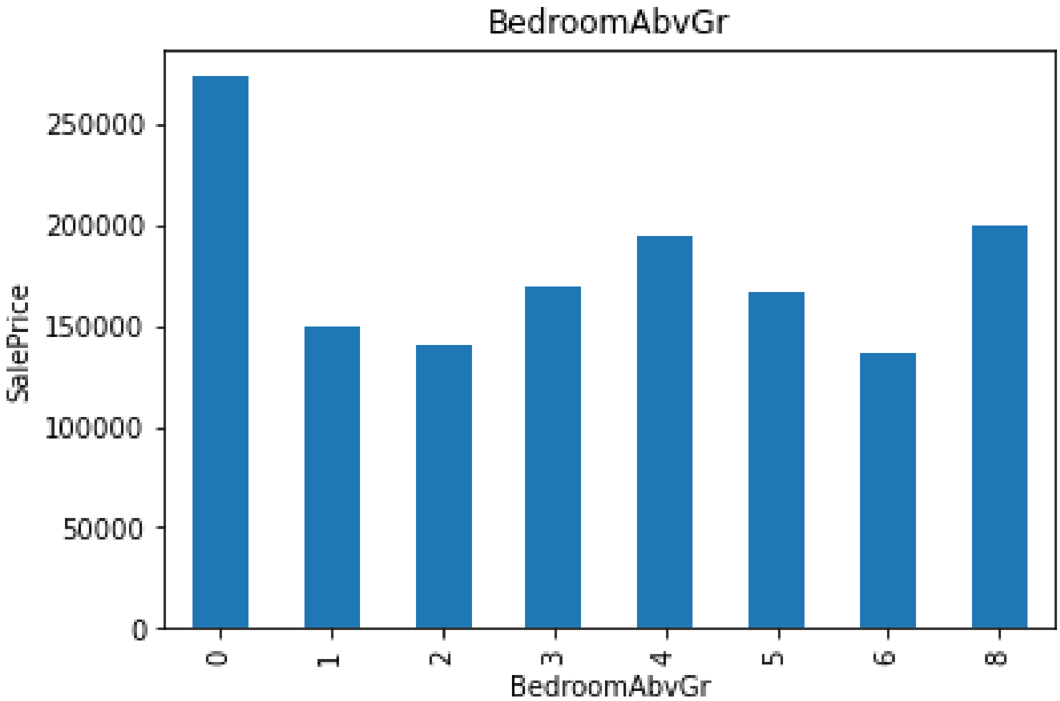
print("-

"*125)df[discrete_feature]

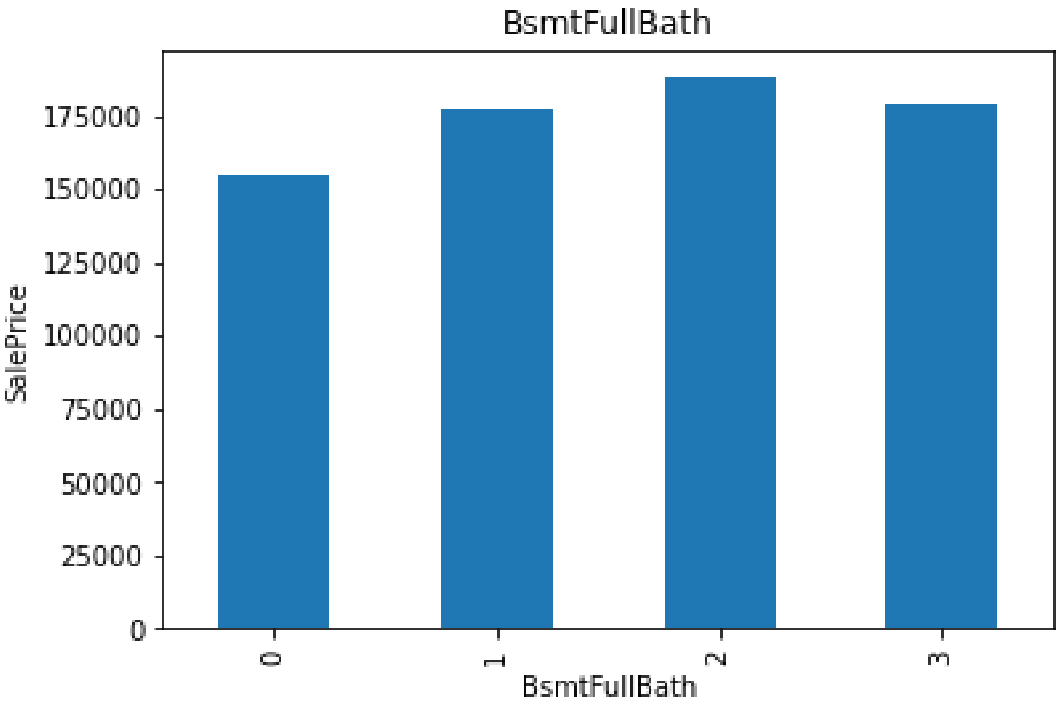
.head()



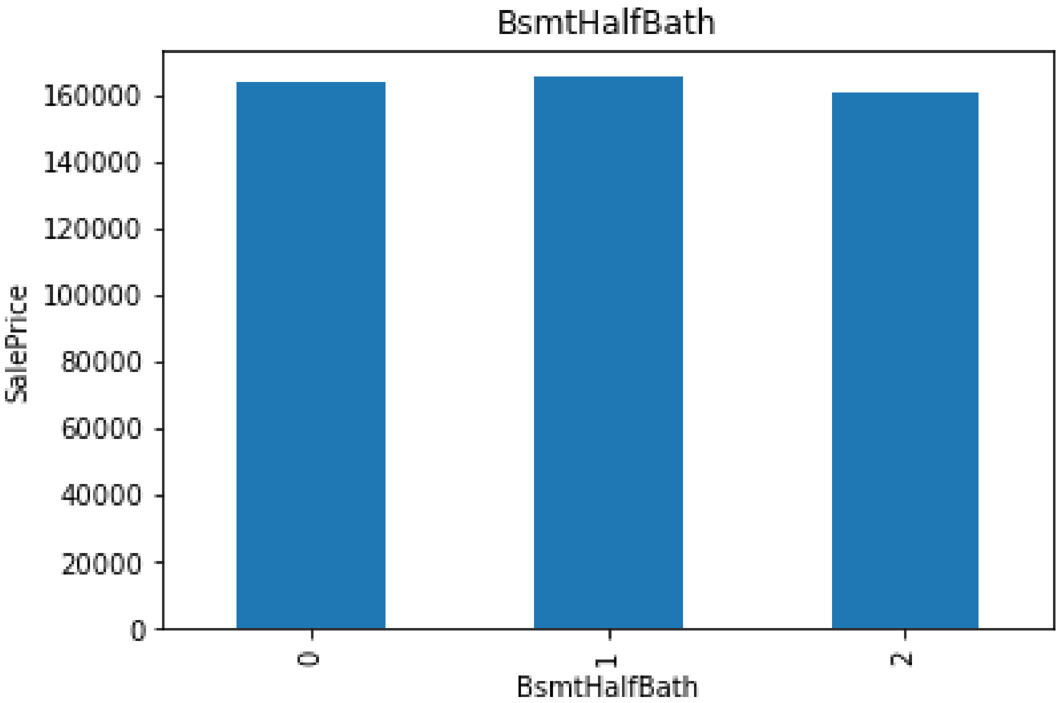
Sale Price vs 3SnPorch



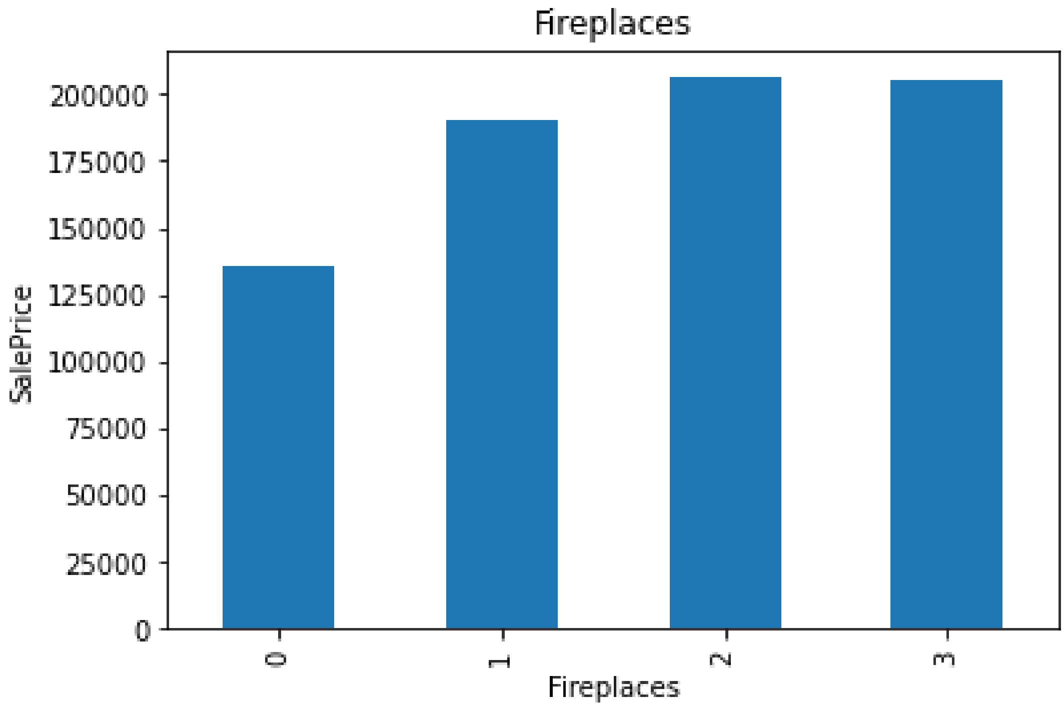
Sale Price vs Bedroom AbvGr



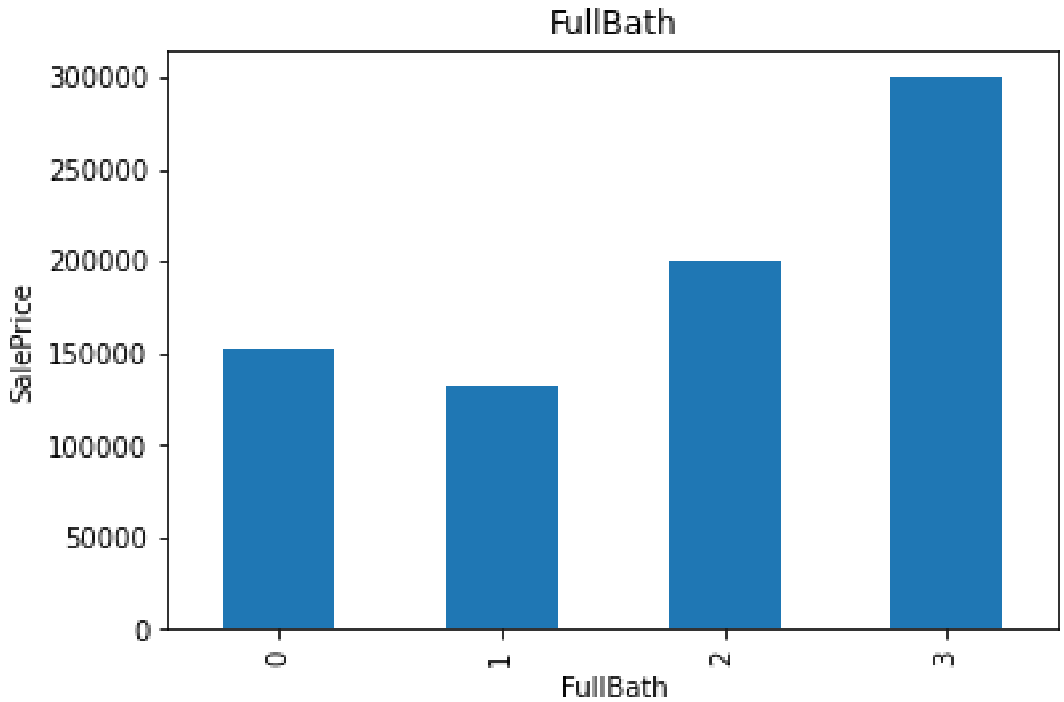
Sale price vs bsmtFullBath



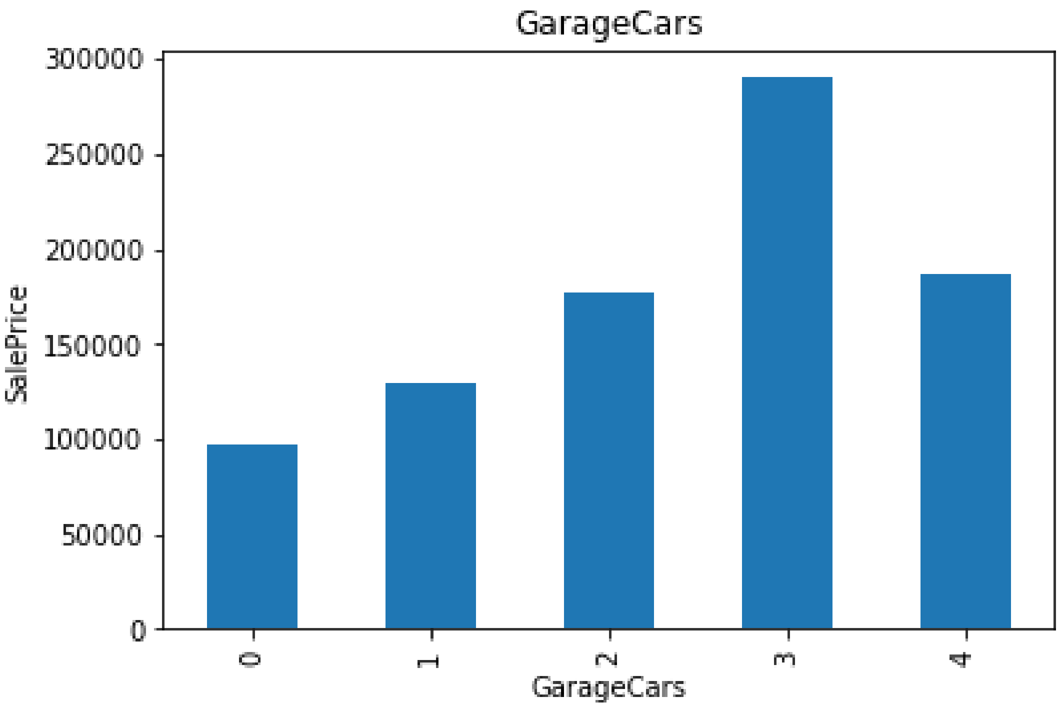
Sale Price vs BsmtHalfBath



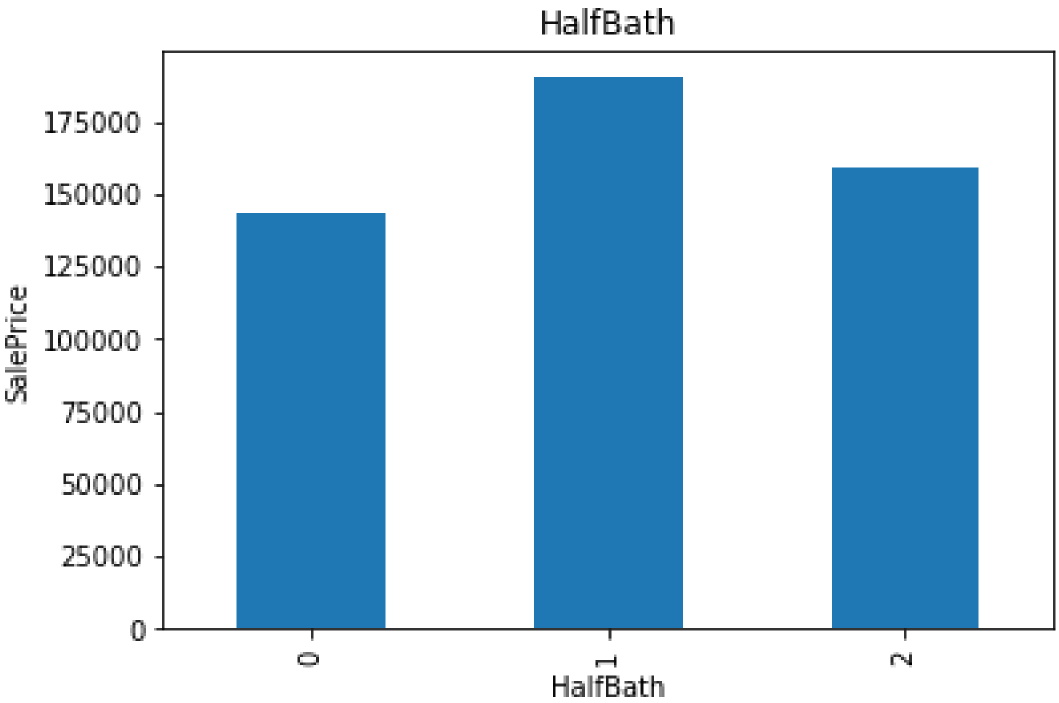
Salepricevs Fireplaces



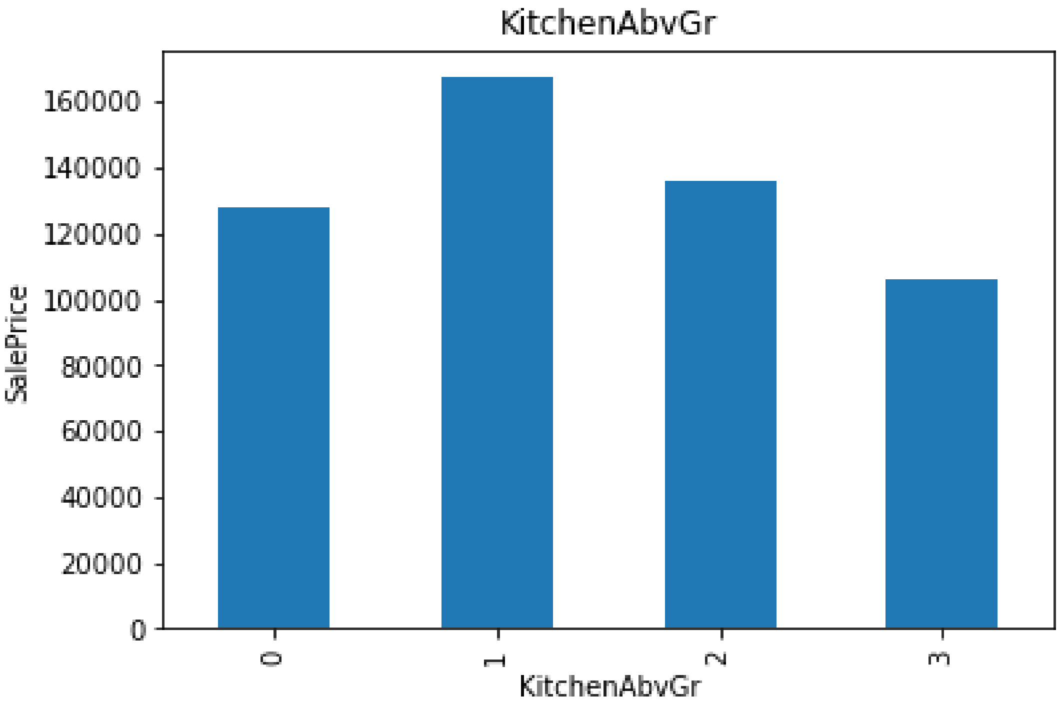
Salepricevs FullBath



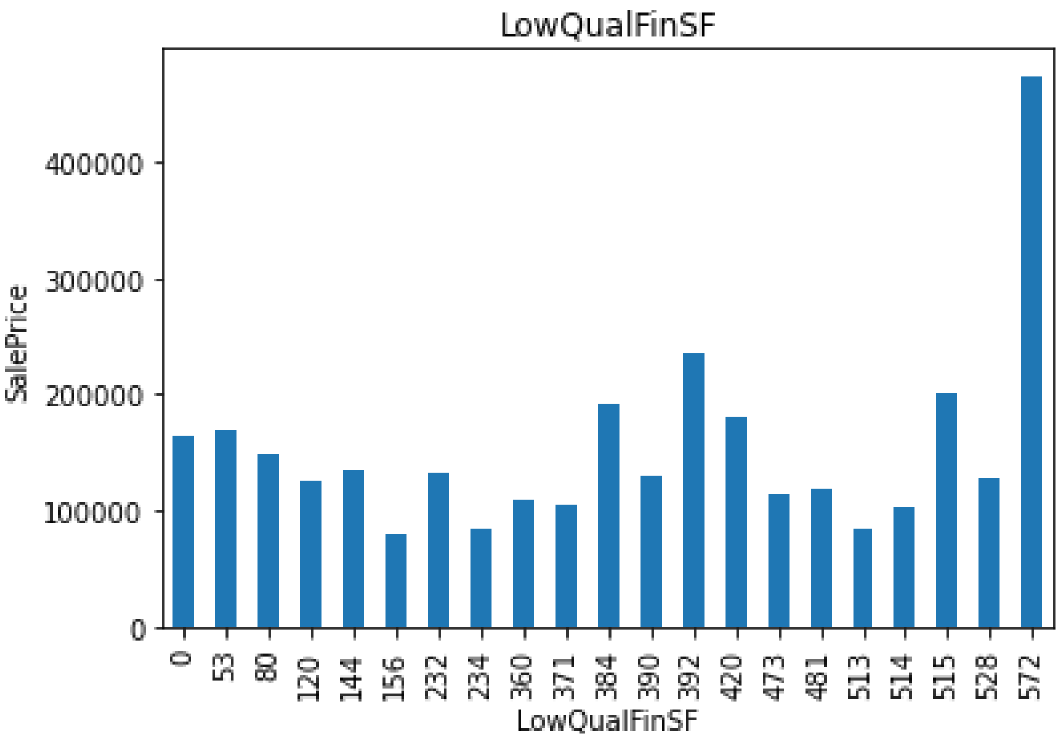
Sales price vs GarageCars



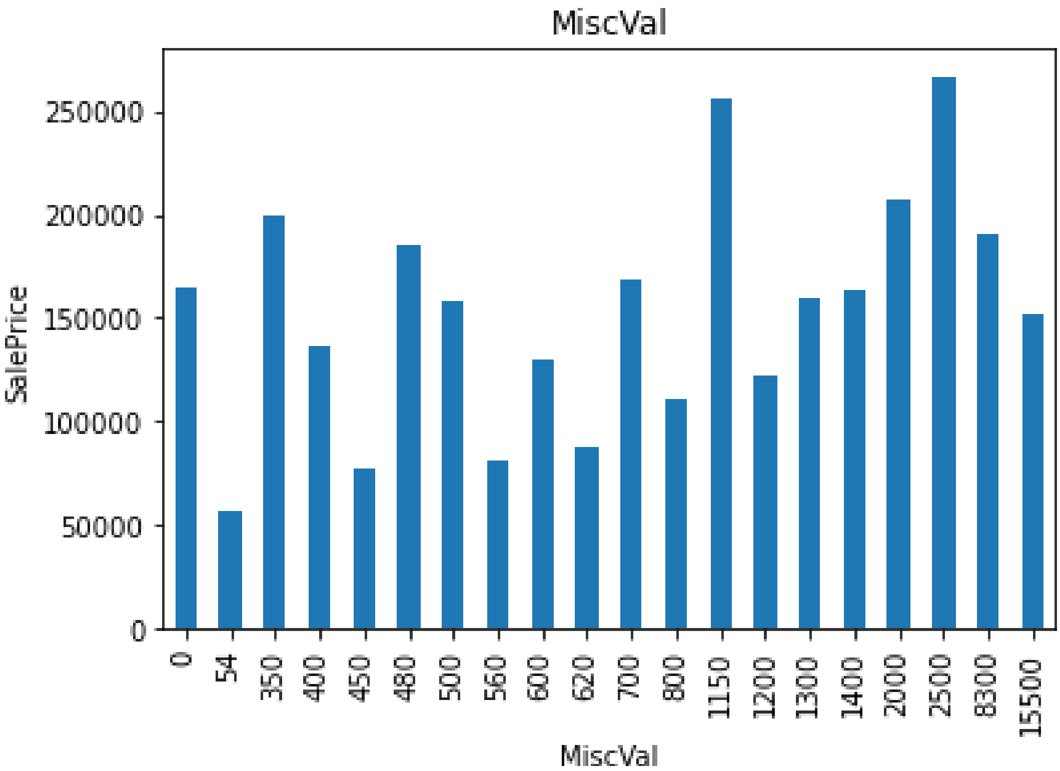
Sales price vs HalfBath



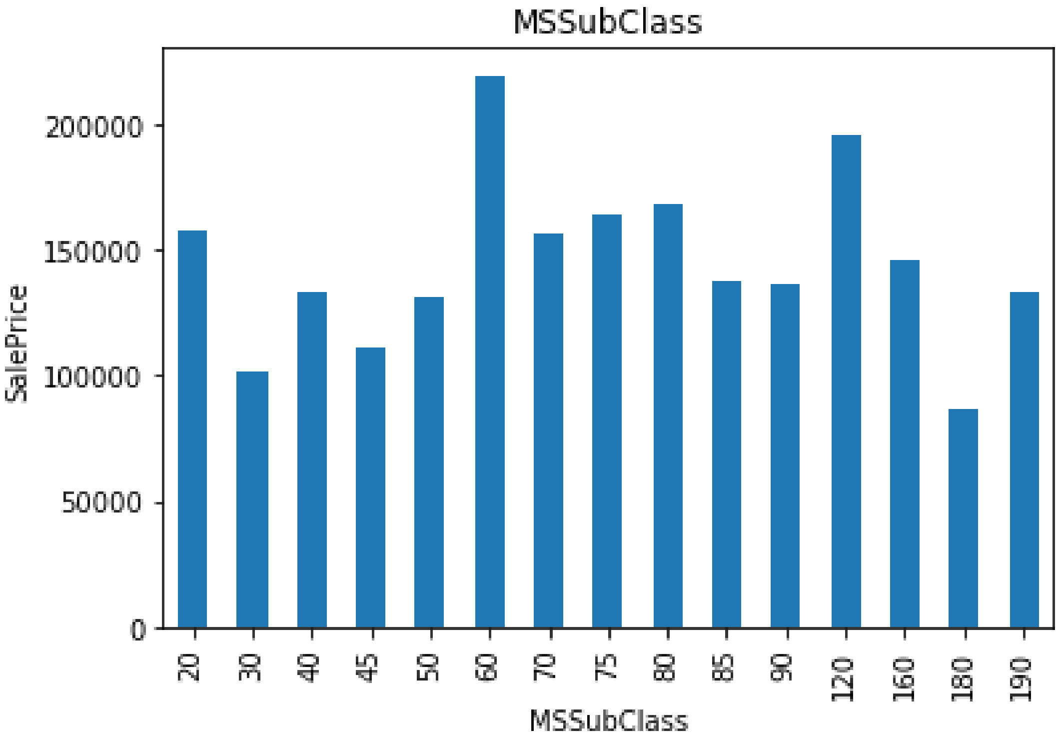
Sales price vs KitchenAbvGr



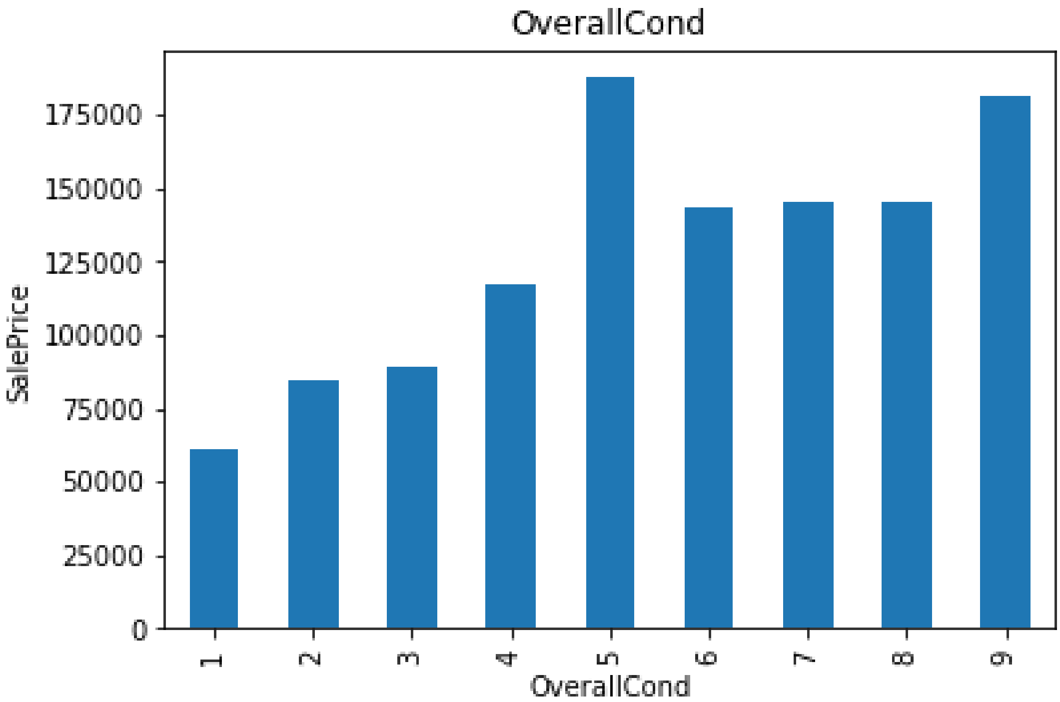
Sale price Vs LowQualinSf



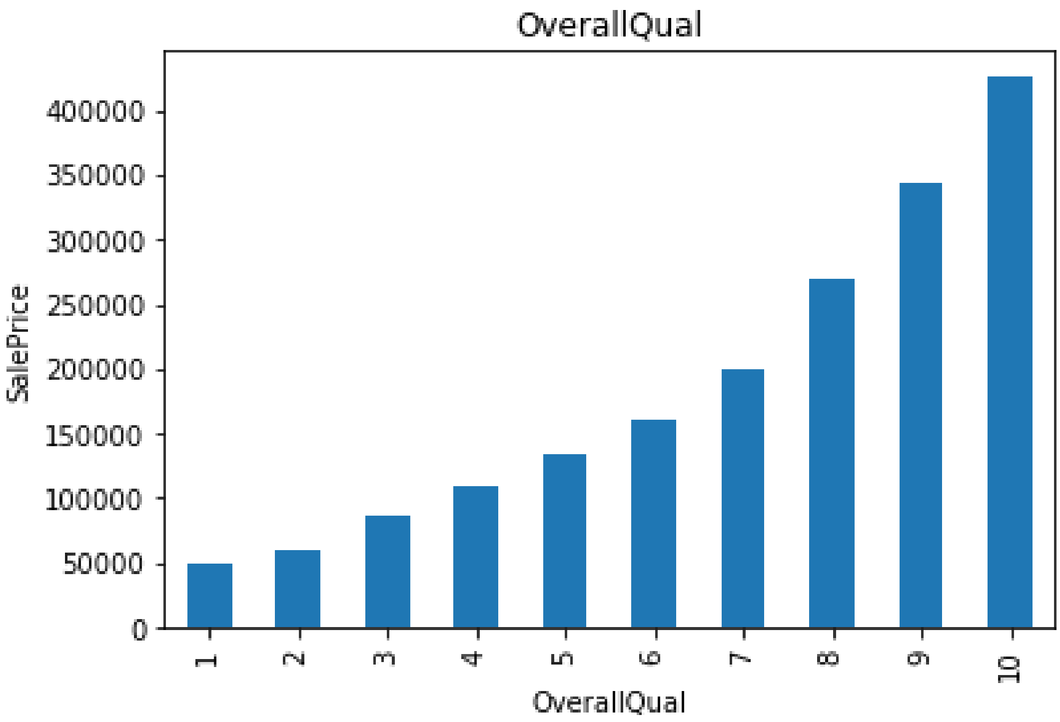
SalepricevsMiscVal



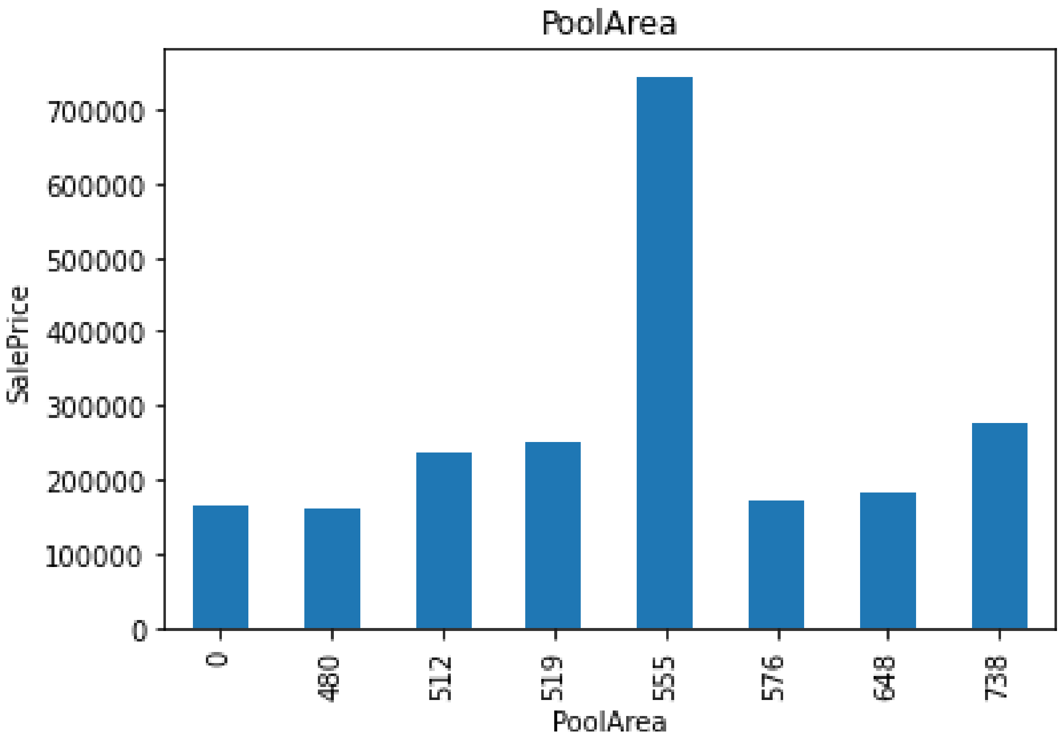
SalepricevsMSsubclass



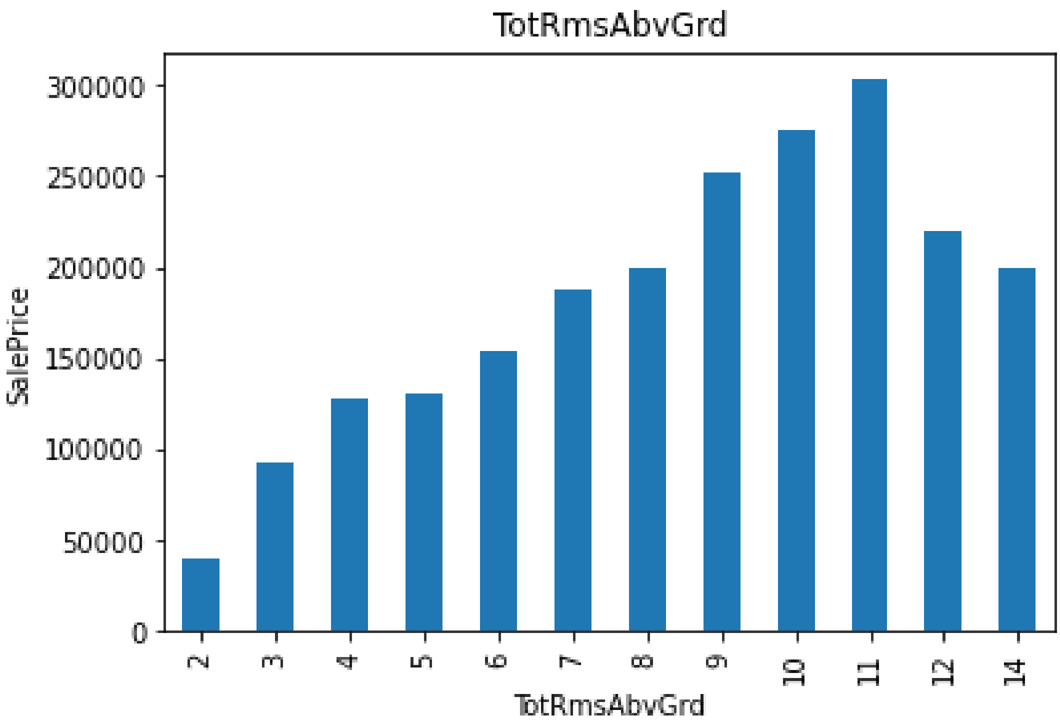
Salepricevsoverallcond



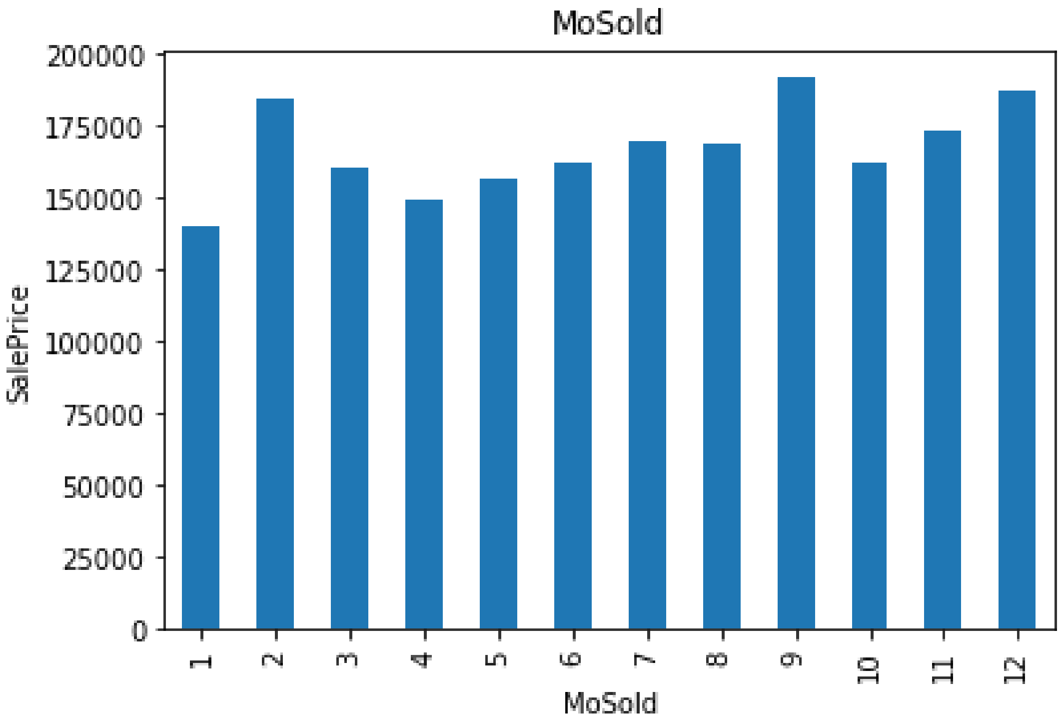
SalepricevsOverallQaul



SalePricevsPoolArea



SalePriceVsTotRmsAbvGrd



SalepricevsMOsold

Extractingthecontinousvariable

continous_feature=[x for x in numerical_features if x not indiscrete_feature+year_feature+['Id']]

print("The number of continous feature column in the dataset :",len(continous_feature))

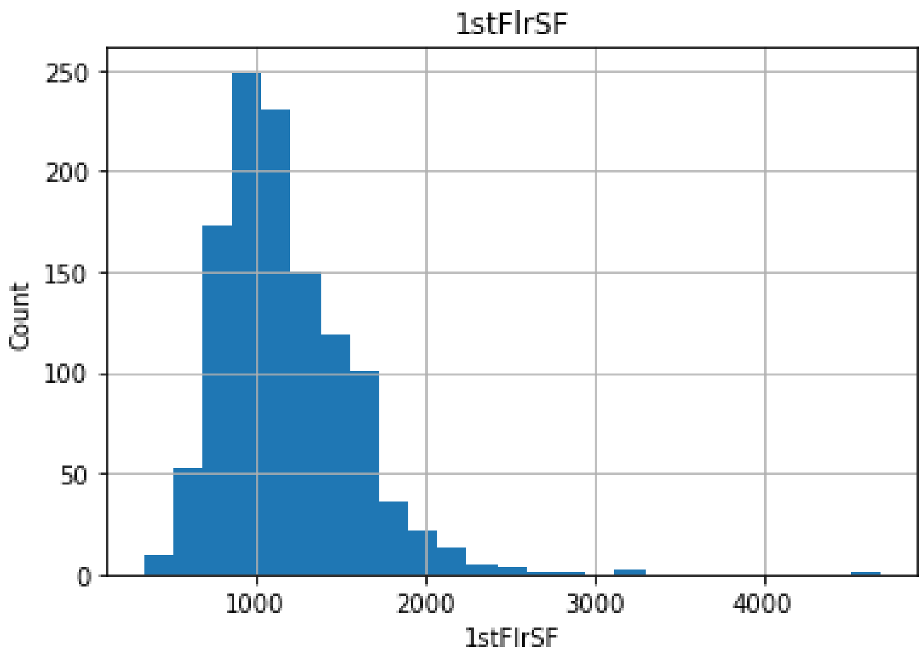
print("Continous feature columns in the dataset

:\n",continous_feature)print("-"*125)

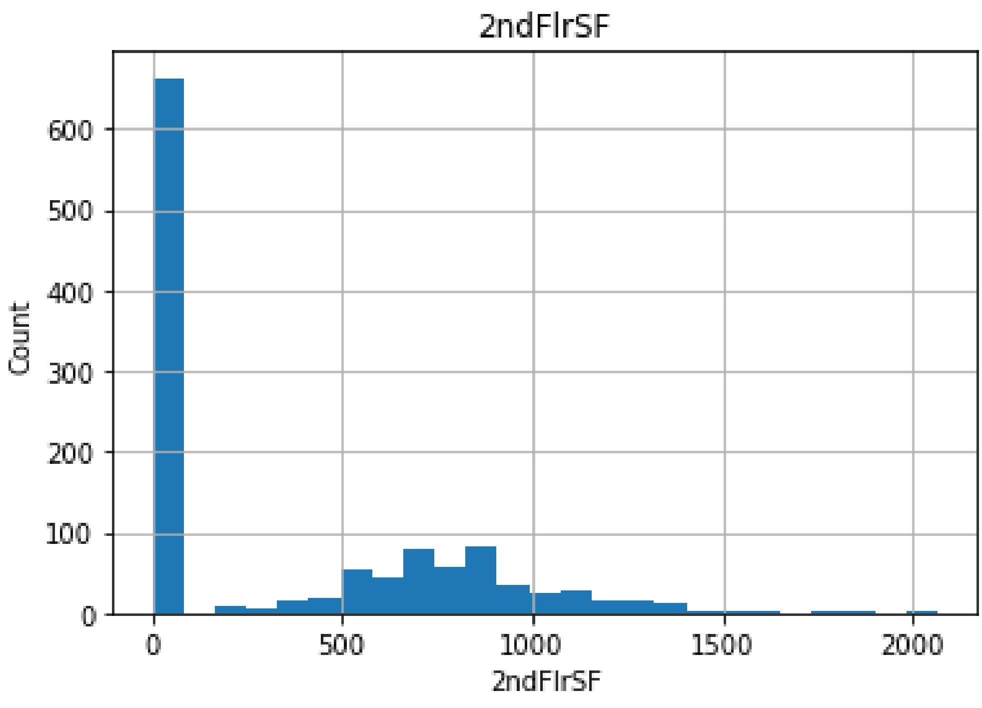
df[continous_feature].head()

observation:-

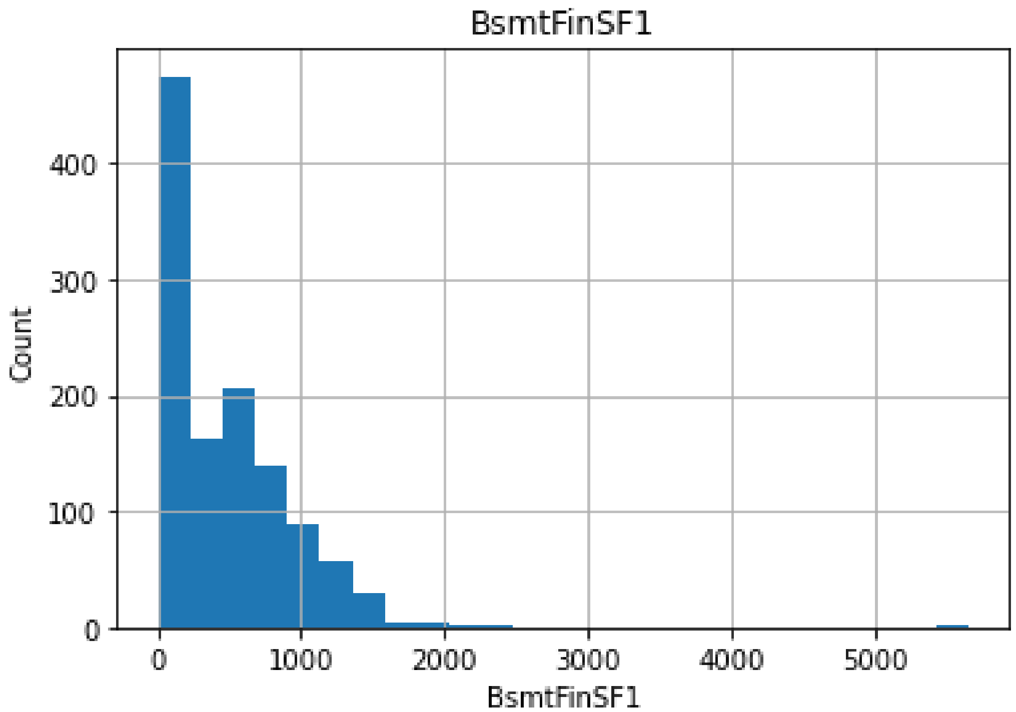
Thenumberofcontinuousfeaturecolumninthedataset:16



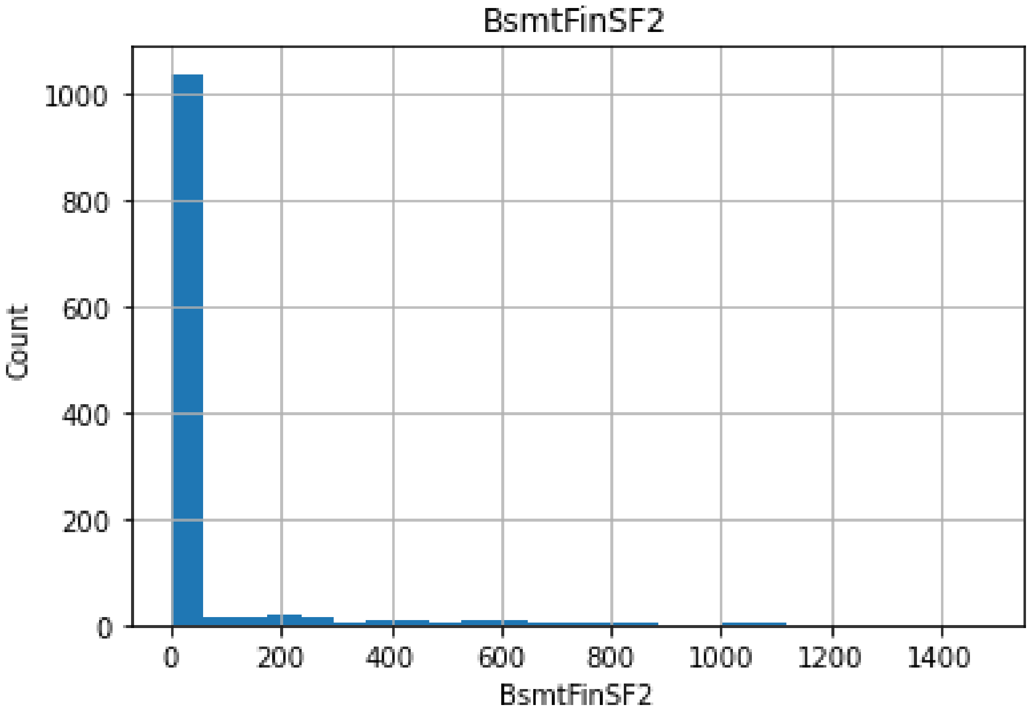
Histogramplotof1stF1r



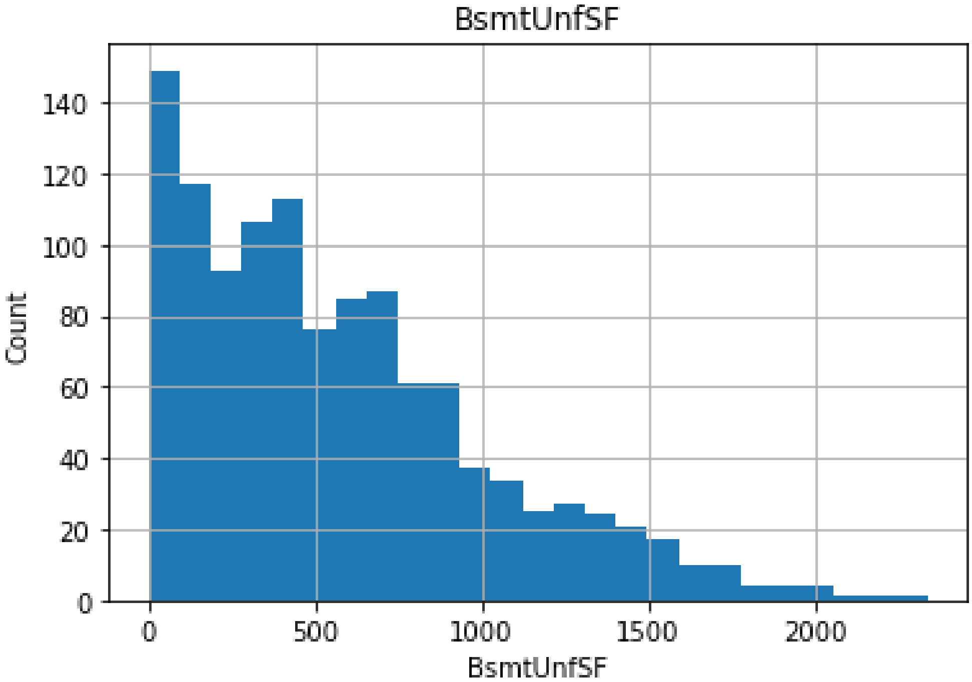
Histogramof2ndF1Srf



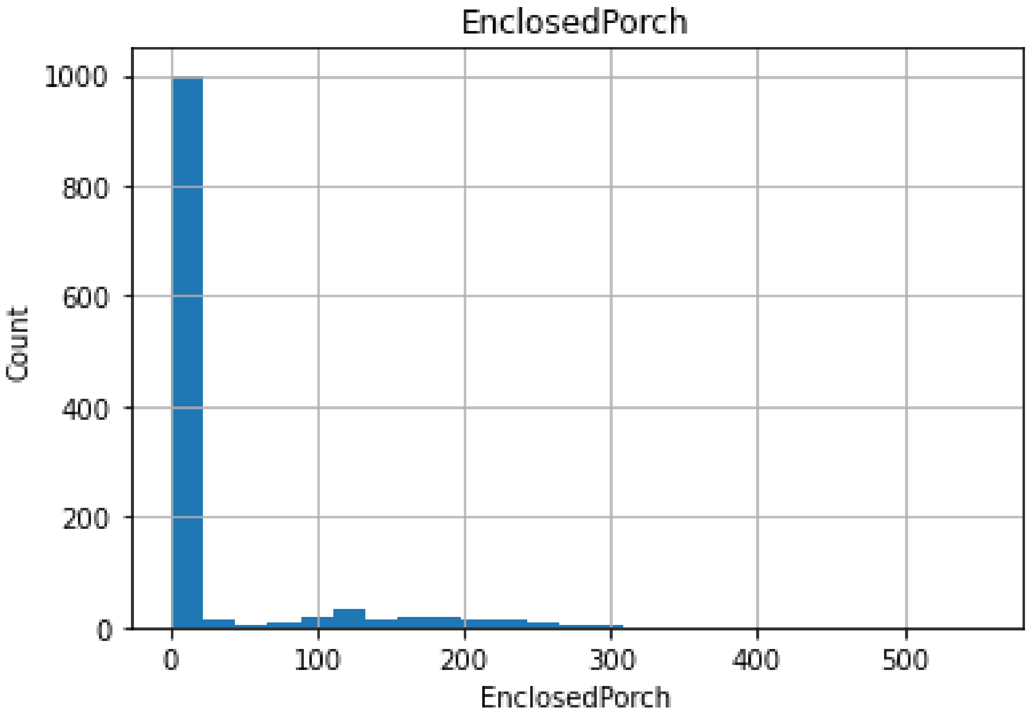
HistogramofBsmtFinSF1



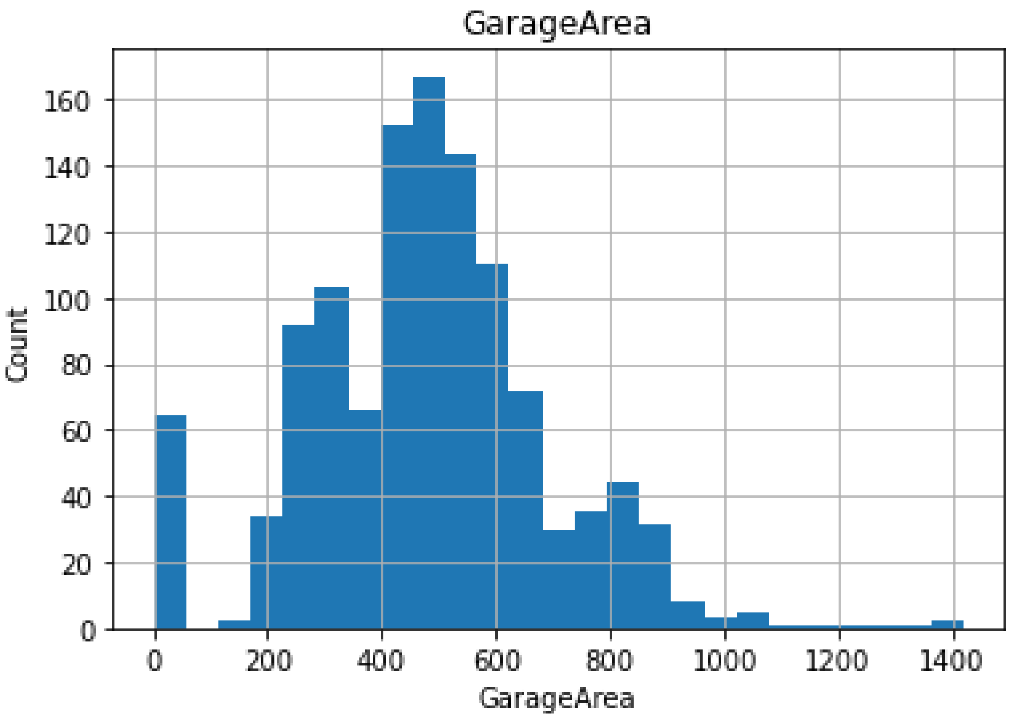
HistogramofBsmtfinSF2



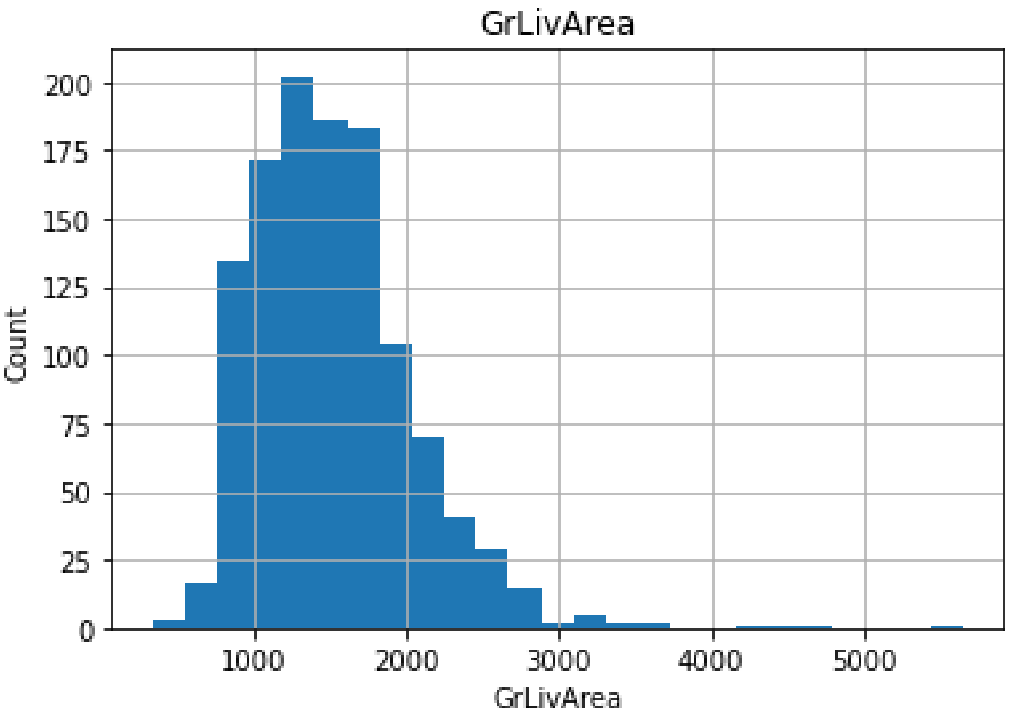
Histogramof BSmmtunfsf



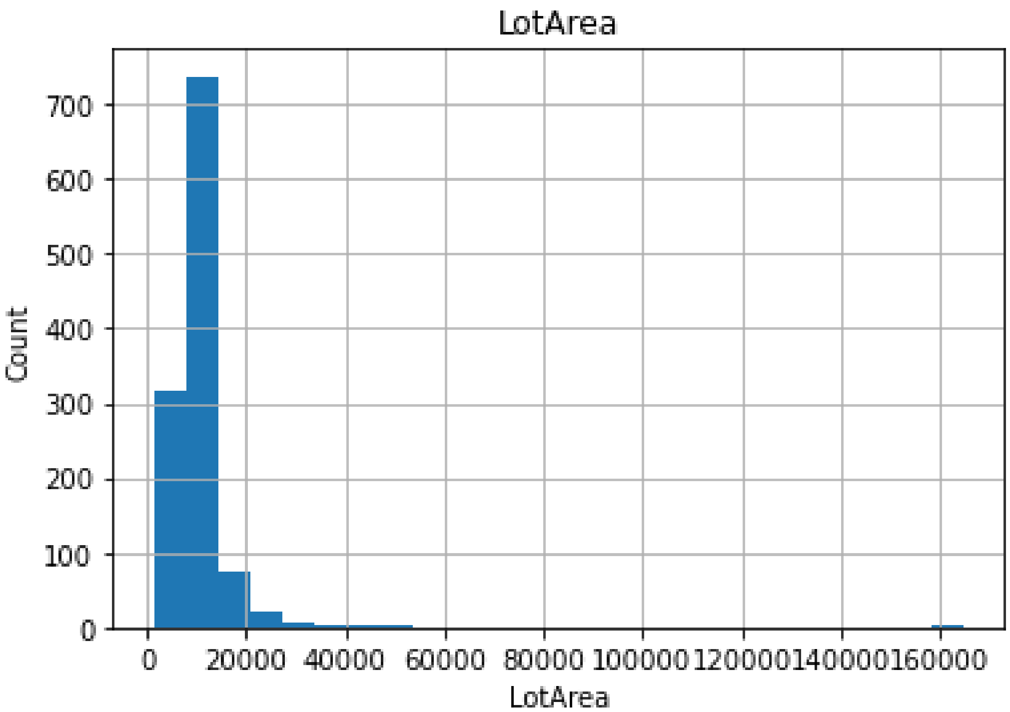
HistogramofEnclosedPorch



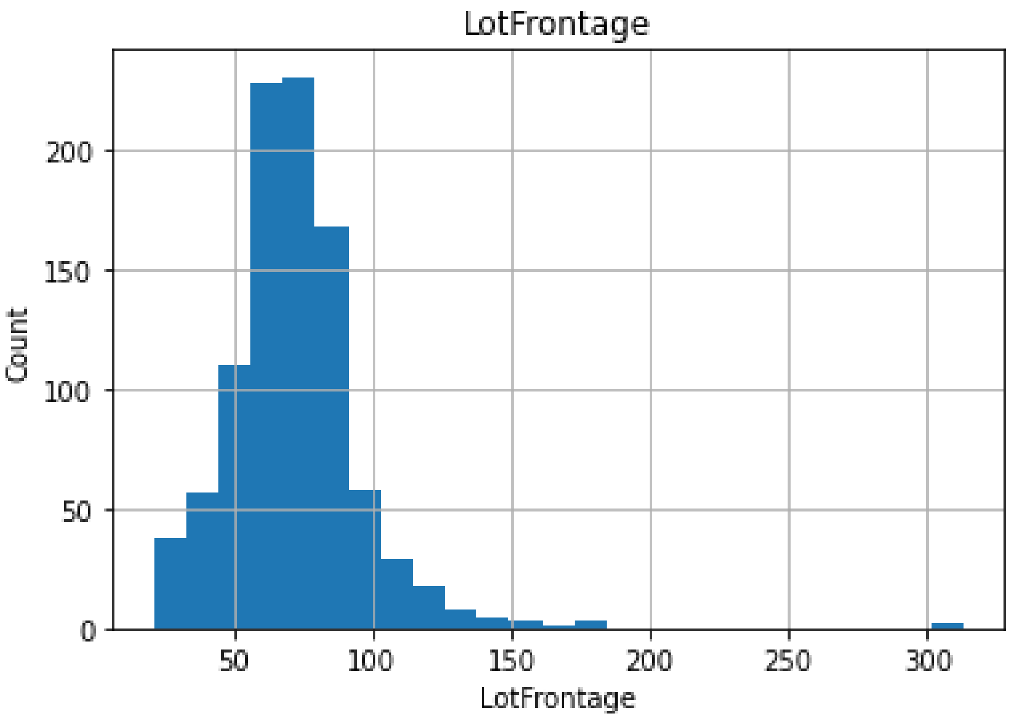
HistogramofGarageArea



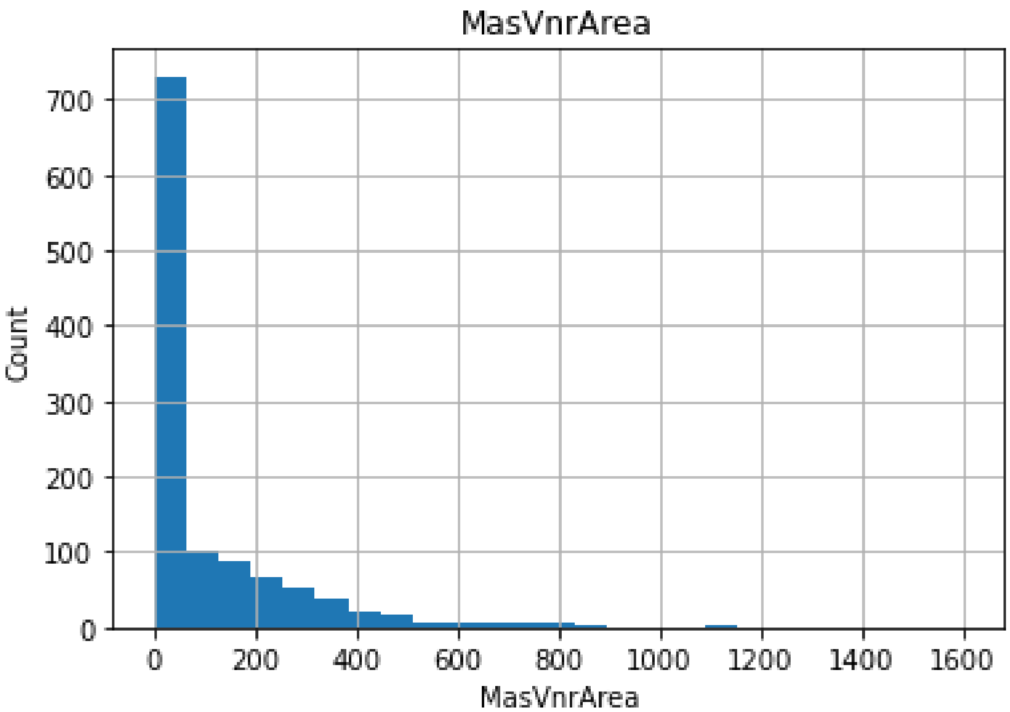
HistogramofGrlivArea



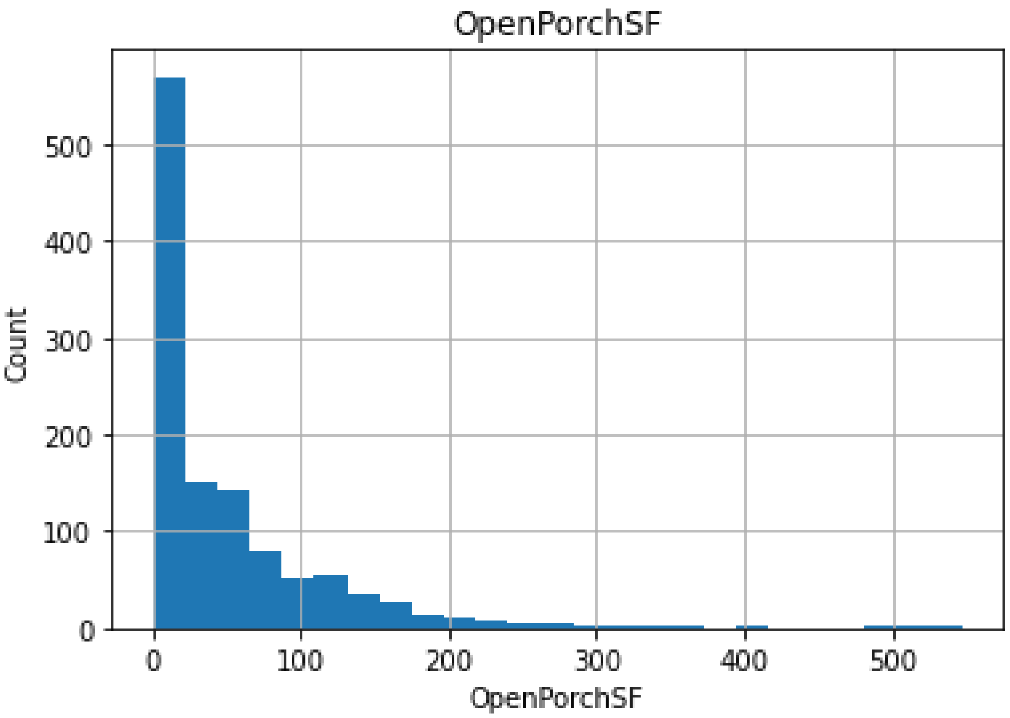
HistogramofLotArea



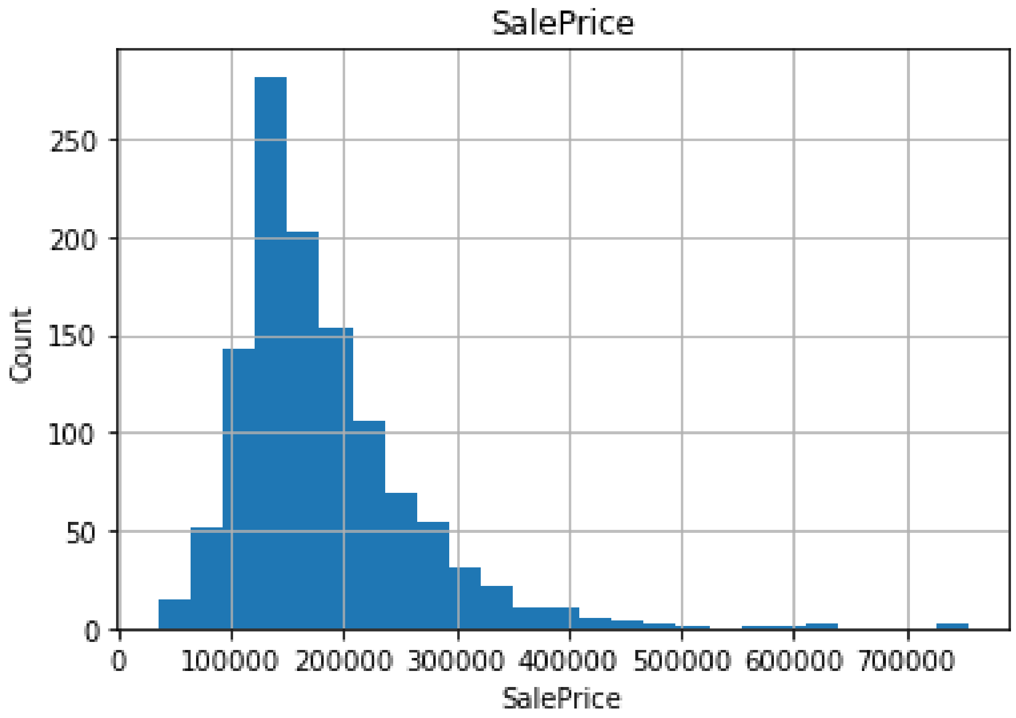
HistogramofLotFrontage



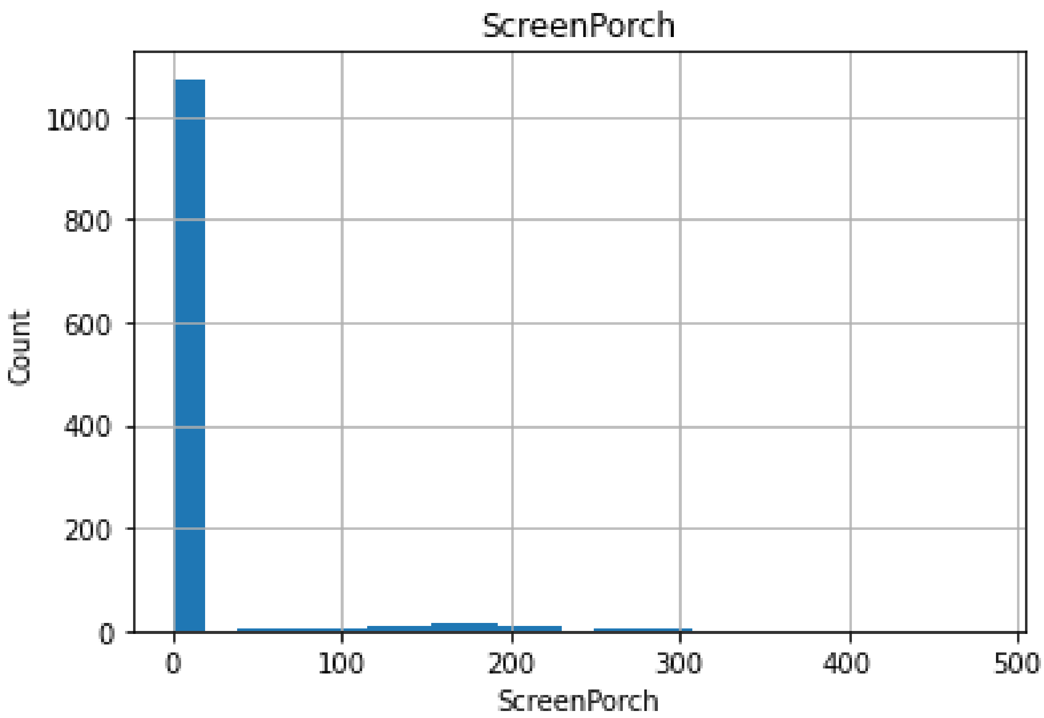
Histogramof MassvnArea



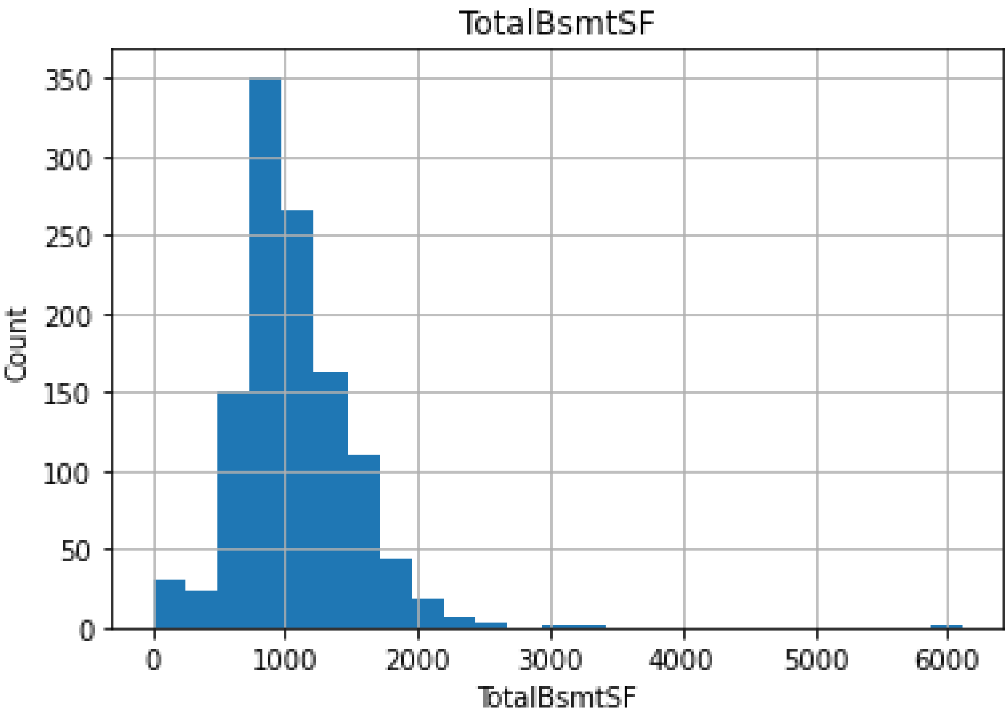
HistogramofOpenPorchSF



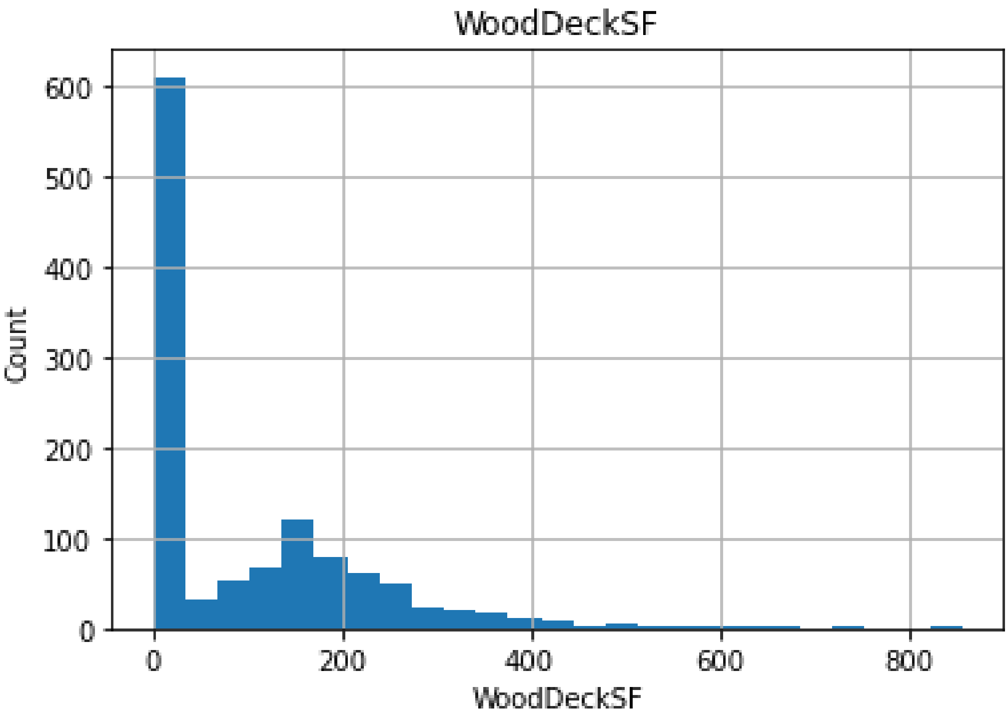
HistogramofSalePrice



HistogramofScreenPorch



HistogramofTotalBsmtSF



HistogramofWoodDeckSF

Observation:-1.Mostofthefeaturesarerightskewed

2.Needtogototransformation

Log transformation can be done using the following python code:-

forfeature incontinous_feature:

data=df.copy()

if0indata[feature].unique():

pass

else:

data[feature]=np.log(data[feature])data['S

alePrice']=np.log(data['SalePrice'])plt.sca

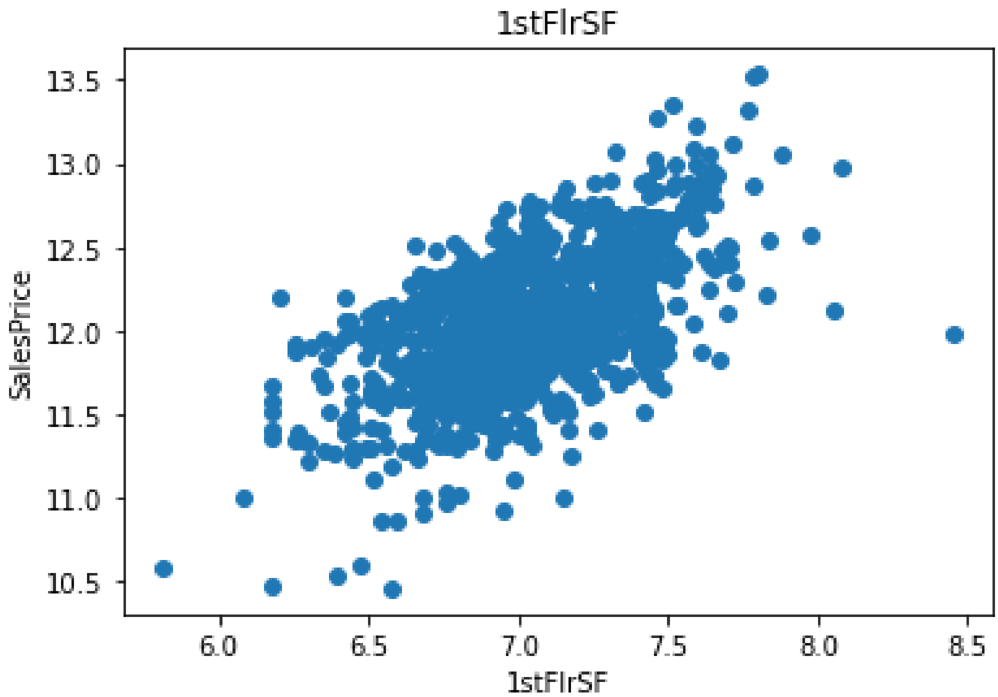
tter(data[feature],data['SalePrice'])plt.xla

bel(feature)

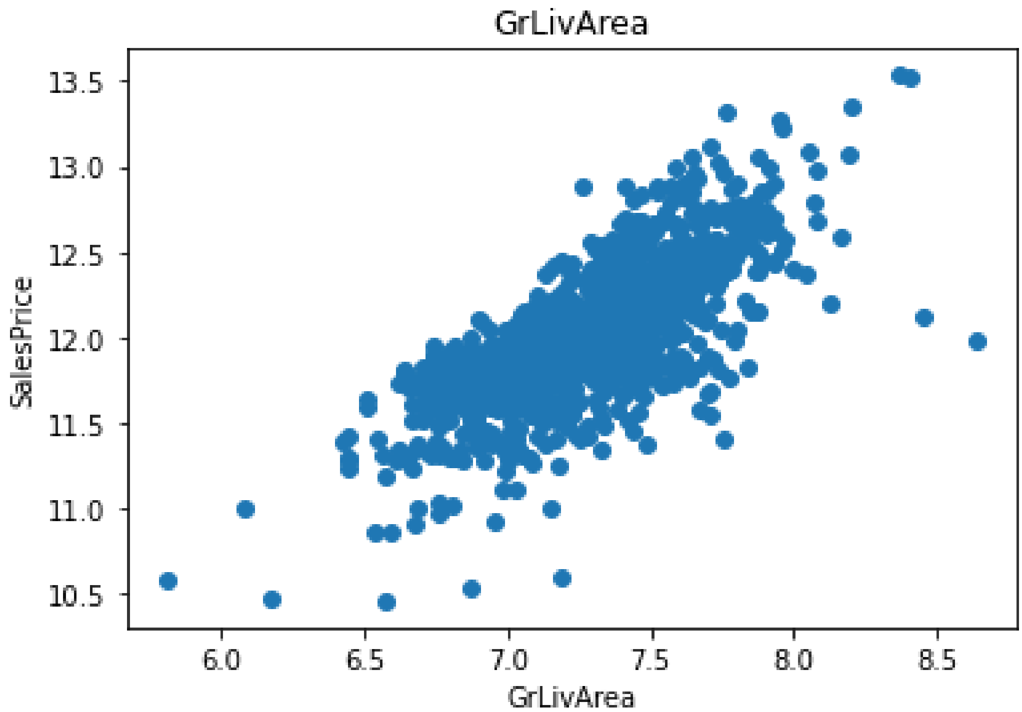
plt.ylabel('SalesPrice')

plt.title(feature)plt.sho

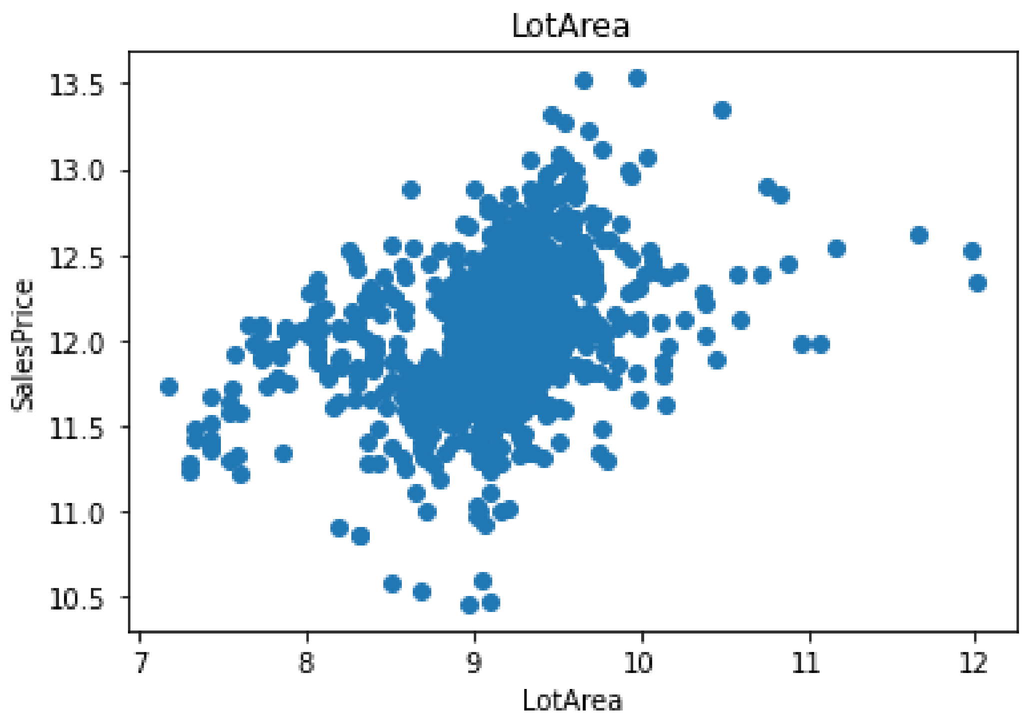
w()



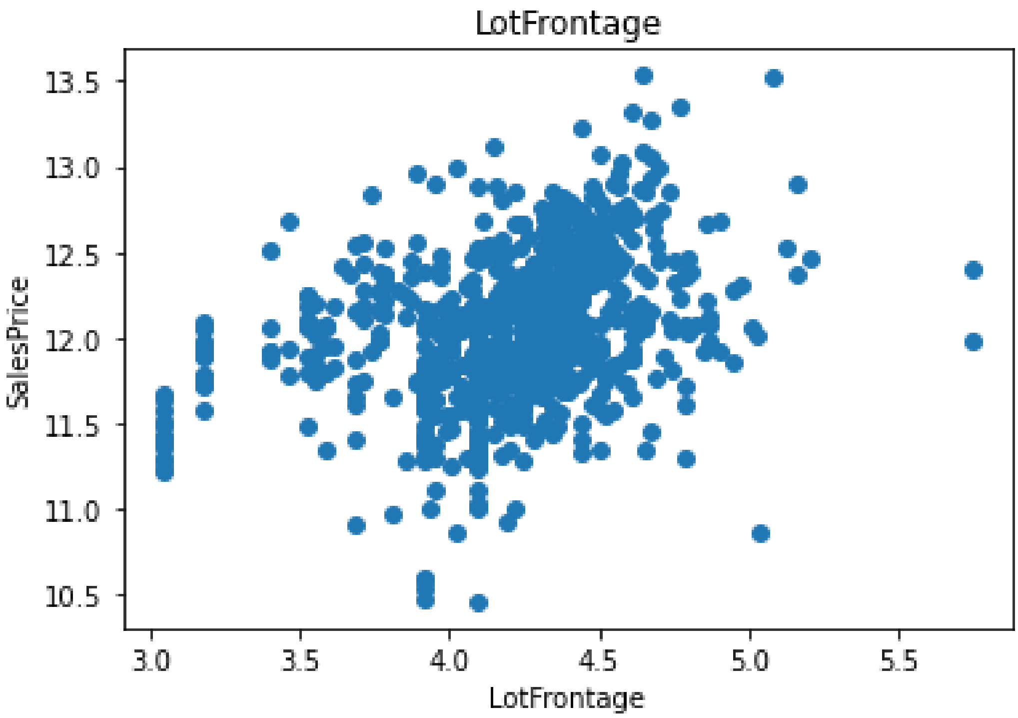
Salepricevs1StFlrSF



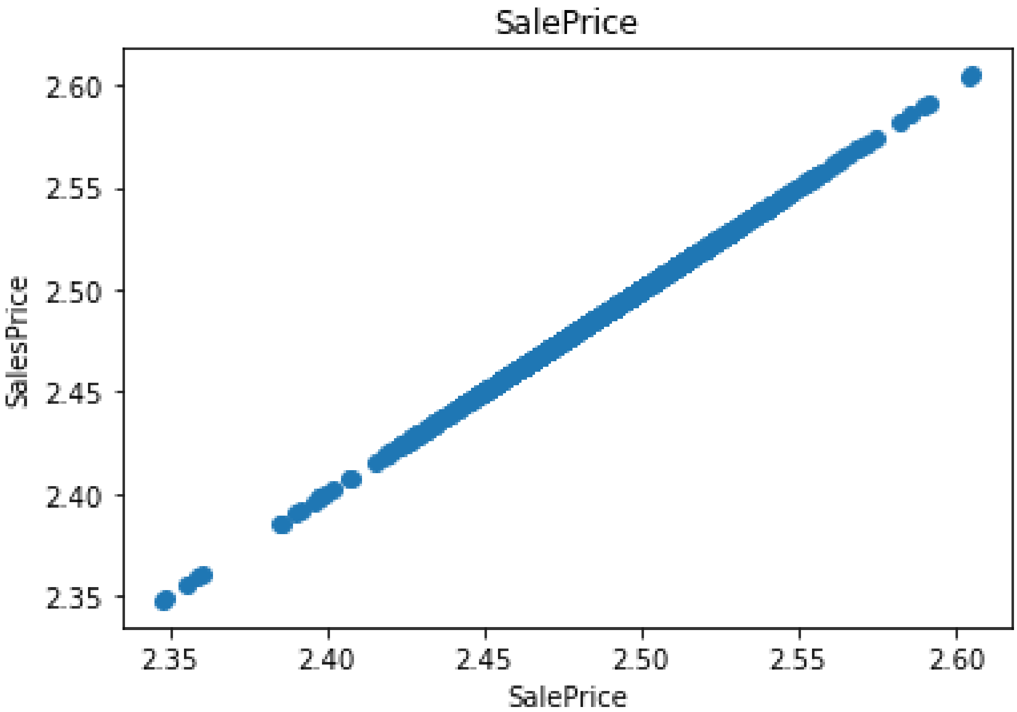
SalepricevsGLivArea



SalepricevsLotArea



SalePriceVsLotFrontage



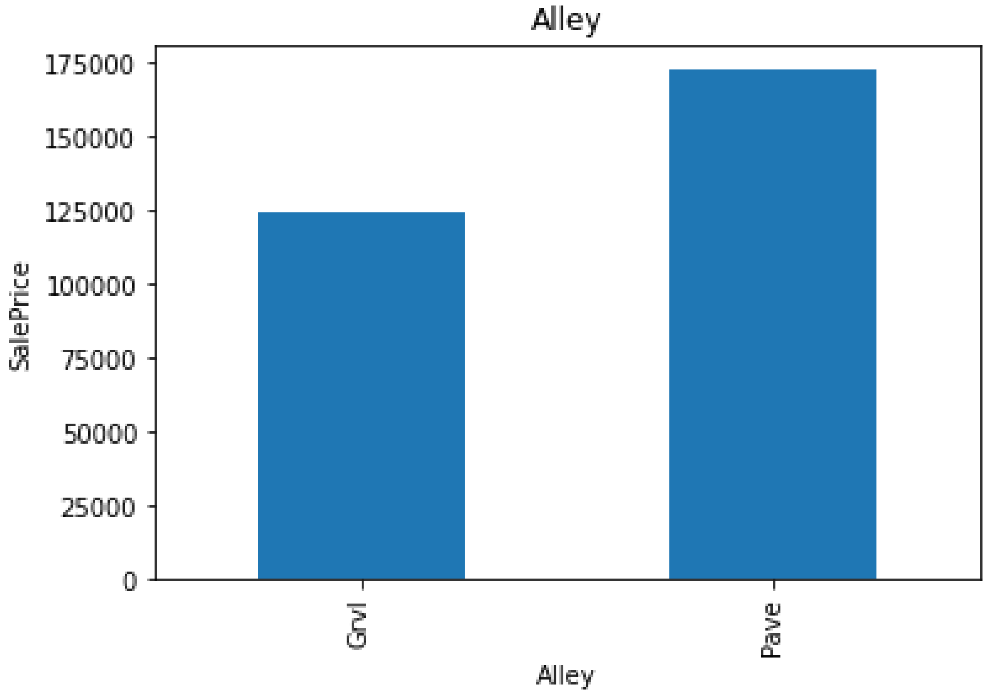
SalePricevsSalePrice

To check the outliers we are using the box

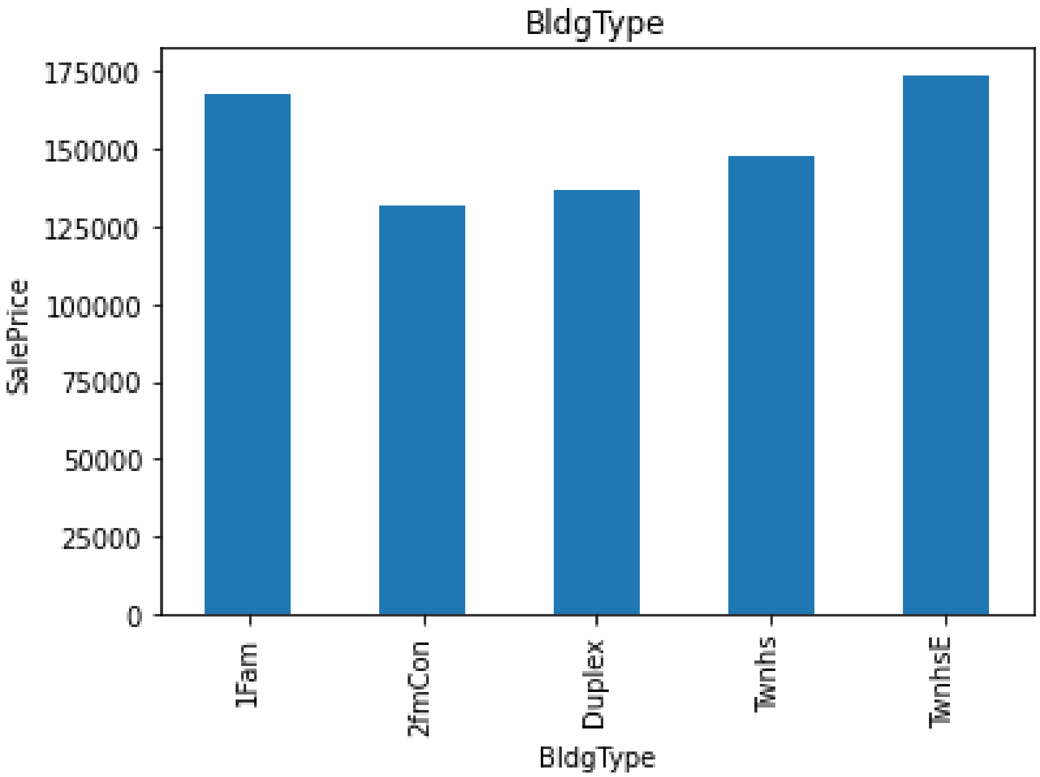
plot.forfeatureincontinous_feature:

```
    data=df.copy()
    if 0 in
        data[feature].unique():pas
        s
    else:
        data[feature]=np.log(data[feature])
        data.boxplot(feature)plt.ylabel(feat
        ure)
        plt.title(feature)
        plt.show()
```

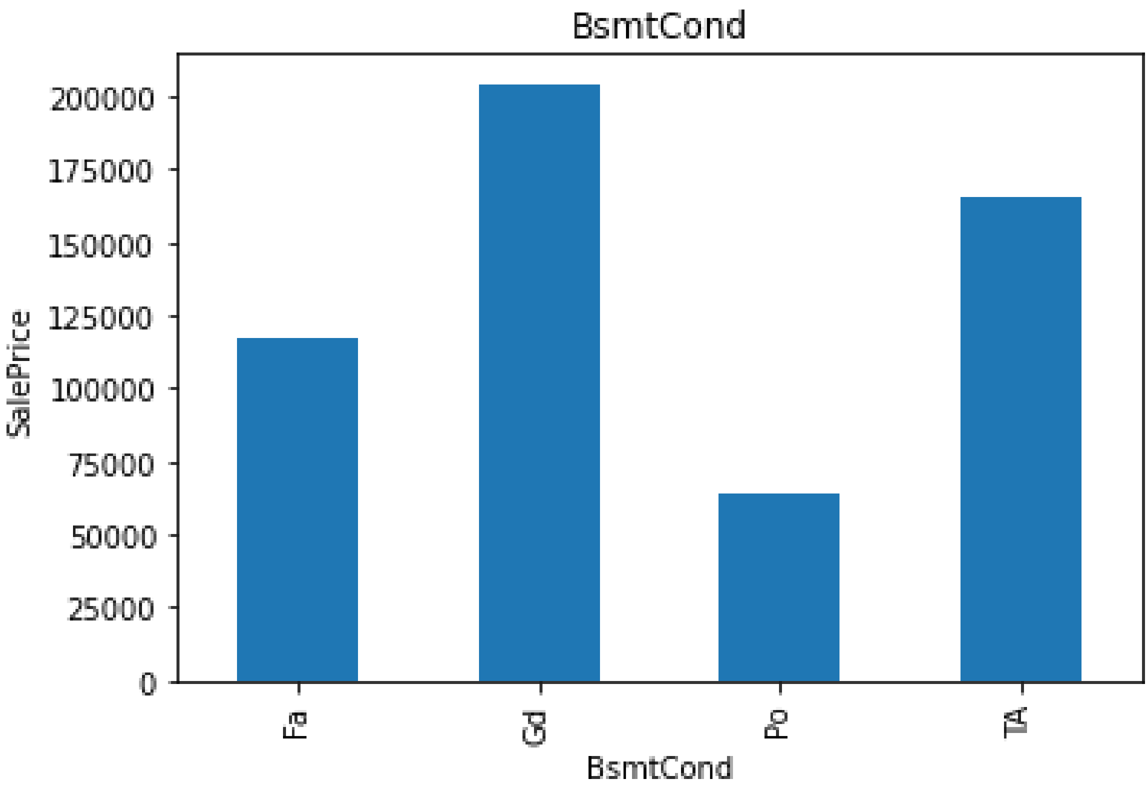
Observation:-There are lot of outliers therefore outlier treatment is required.Realation Betweencategorical feature andSalePrice



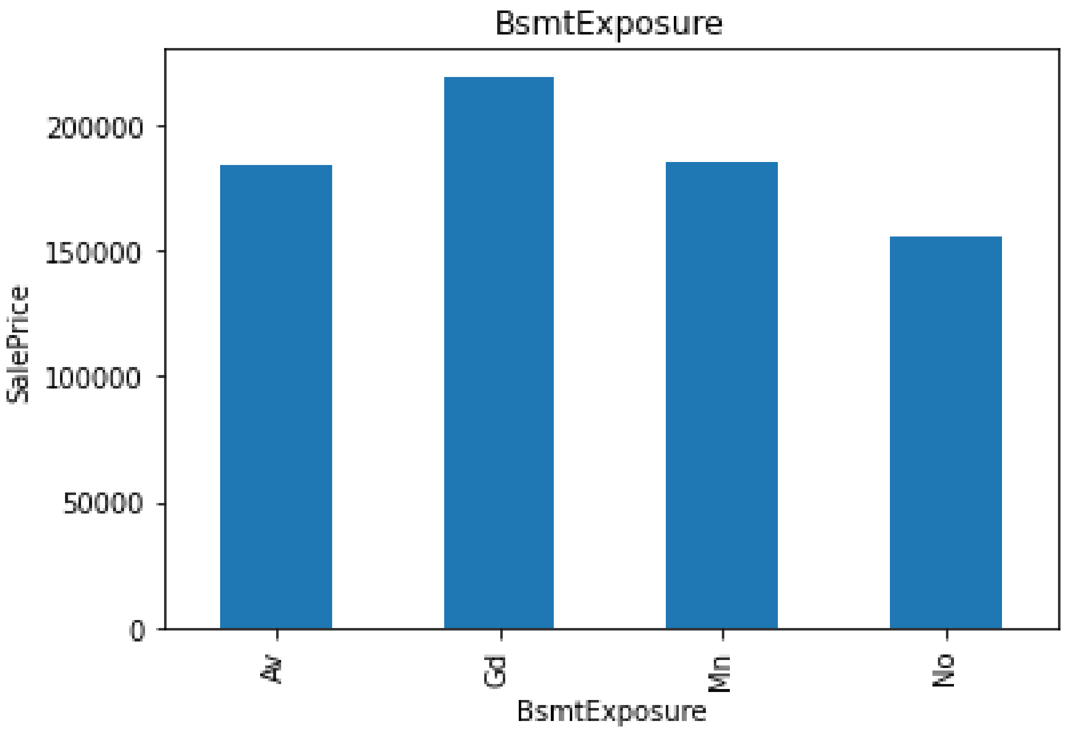
SalepiceVsAlley



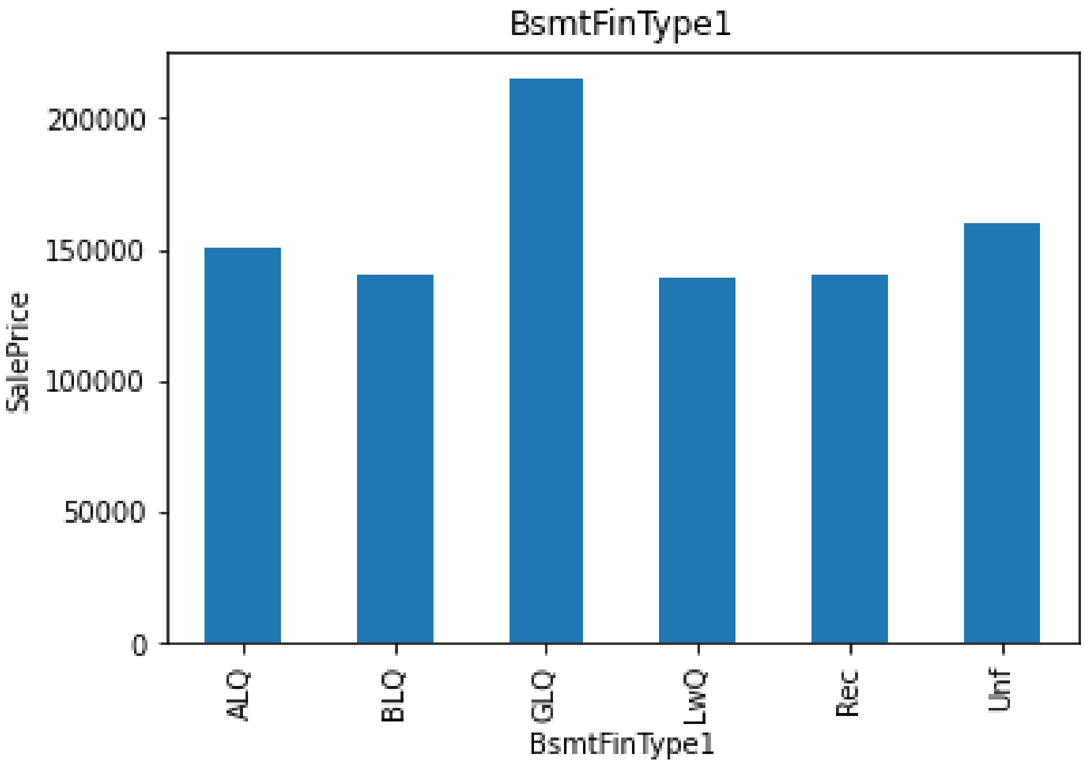
SalePriceVsBldgType



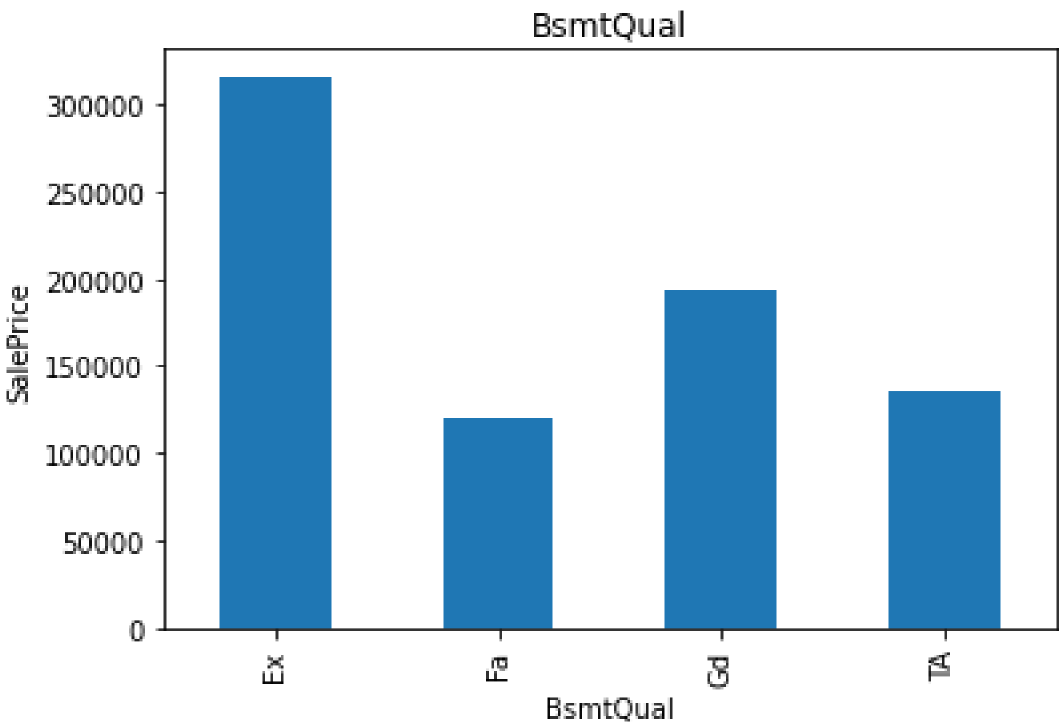
SalePriceVs BSmt Cond



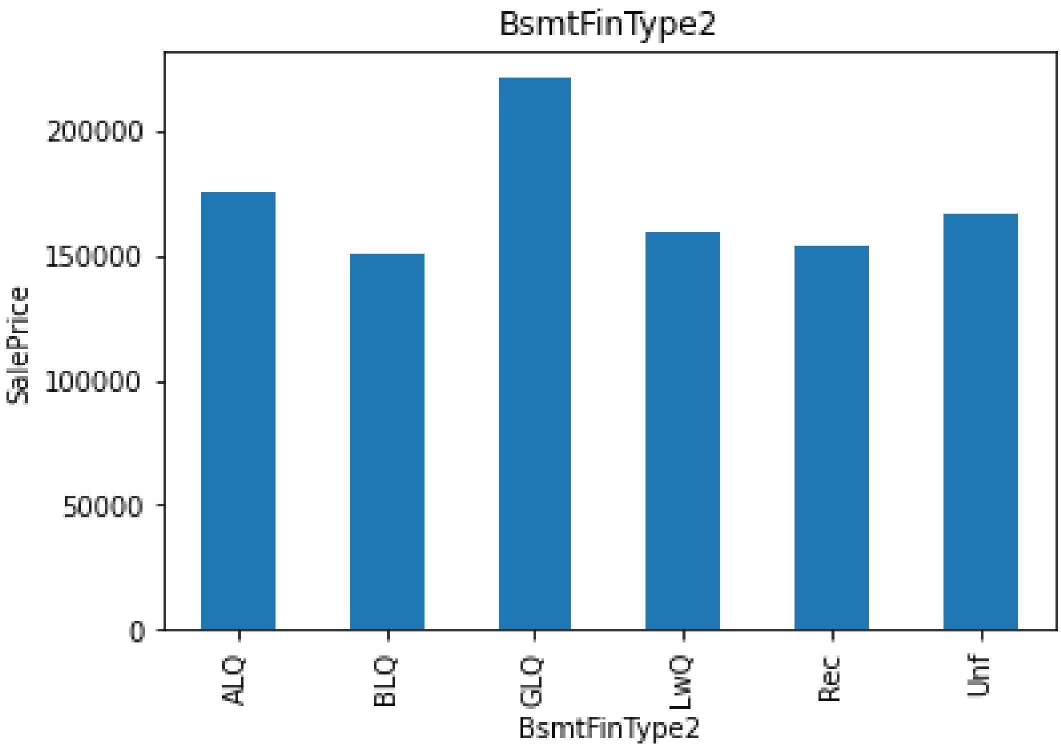
SalePriceVsBsmtExposure



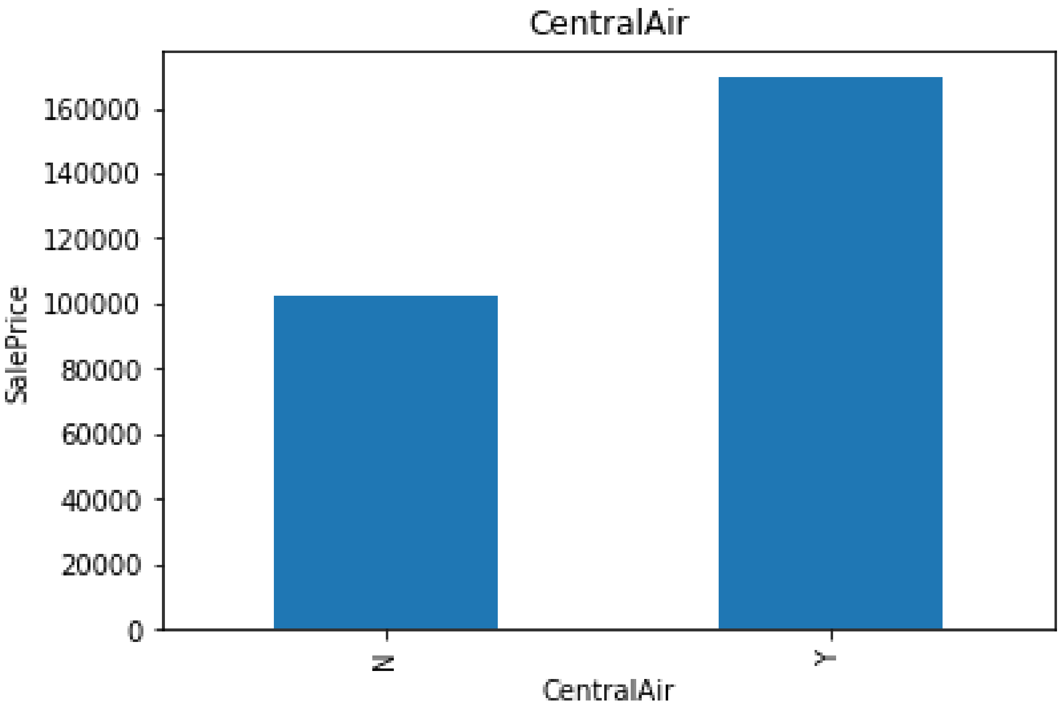
SalePriceVSBsmtFinType1



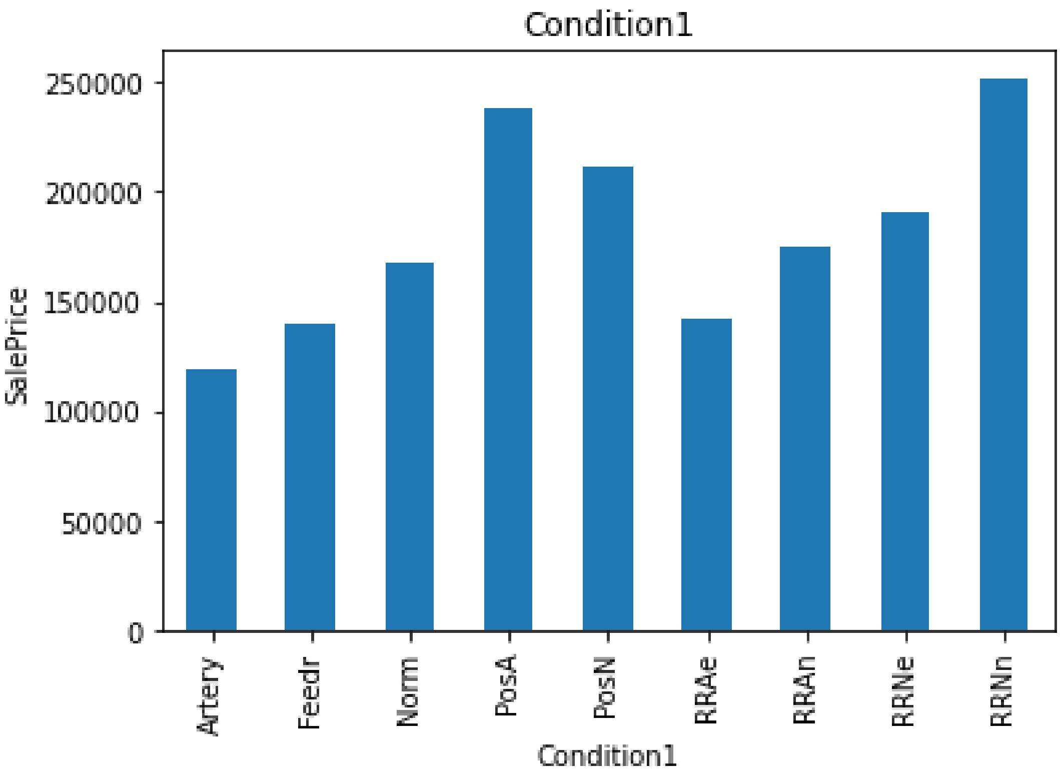
SalePriceVSBsmtQual



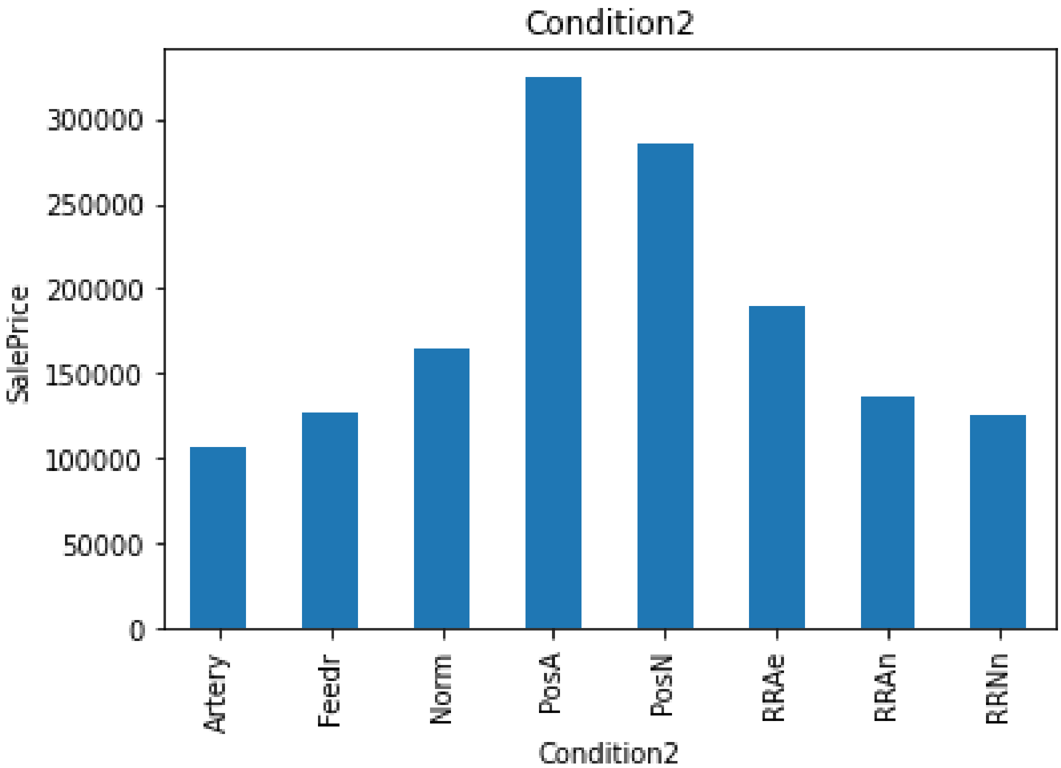
SalepricevsBsmtFinType2



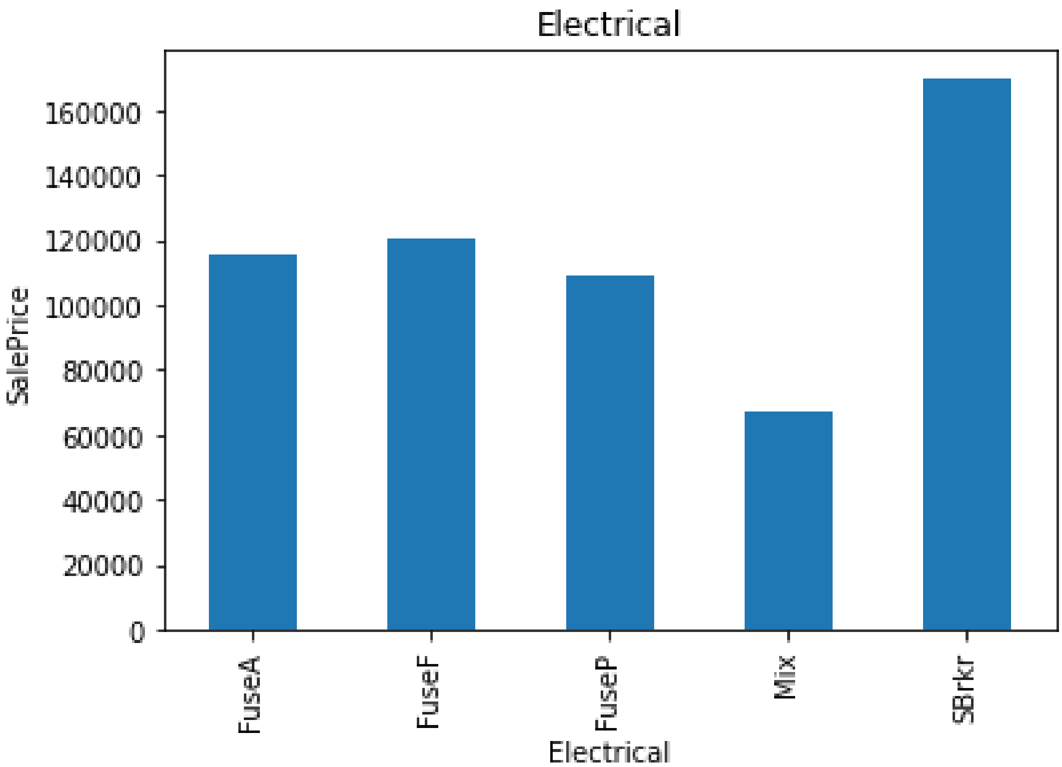
SalePriceVsCentralAir



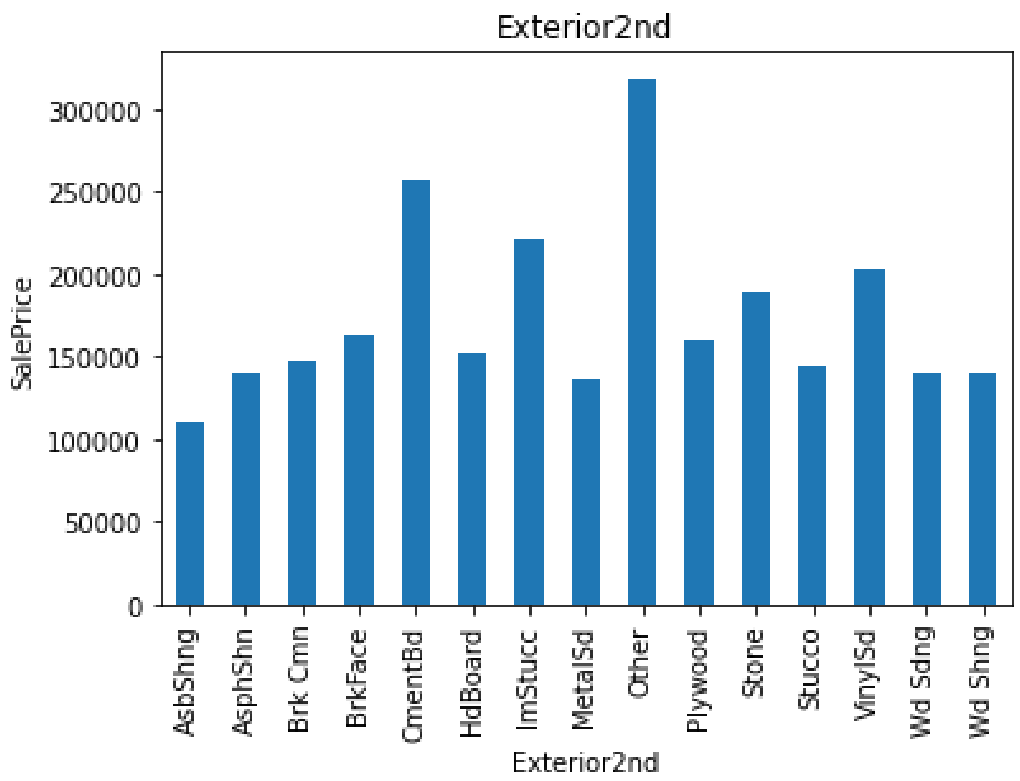
SalePriceVsCondition1



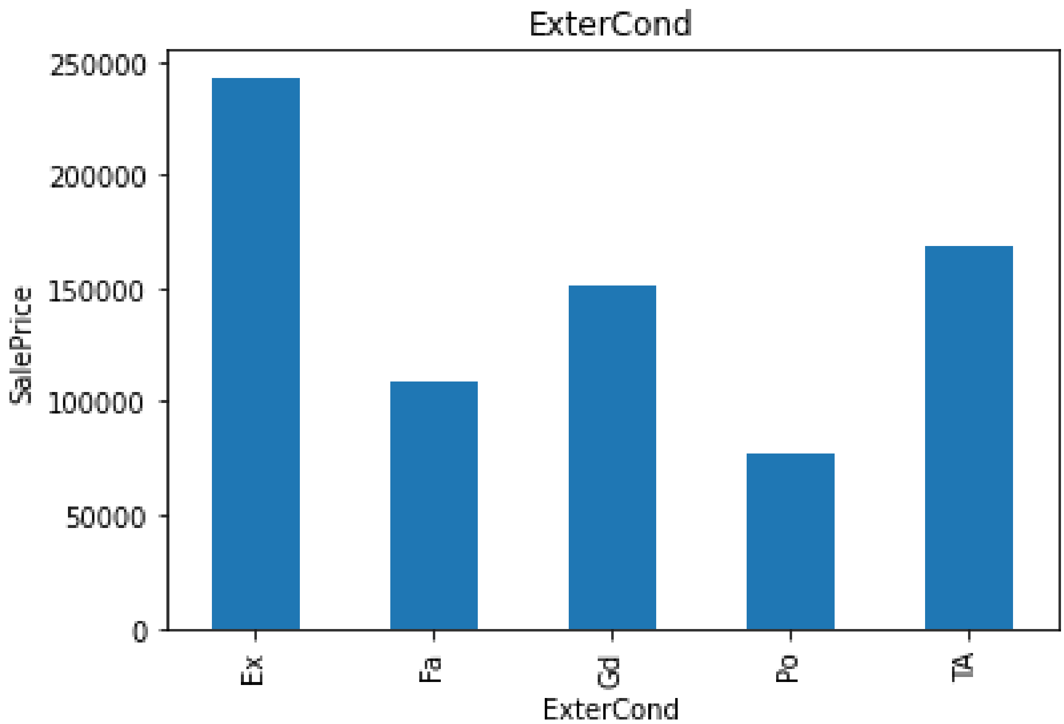
SalePriceVscondition2



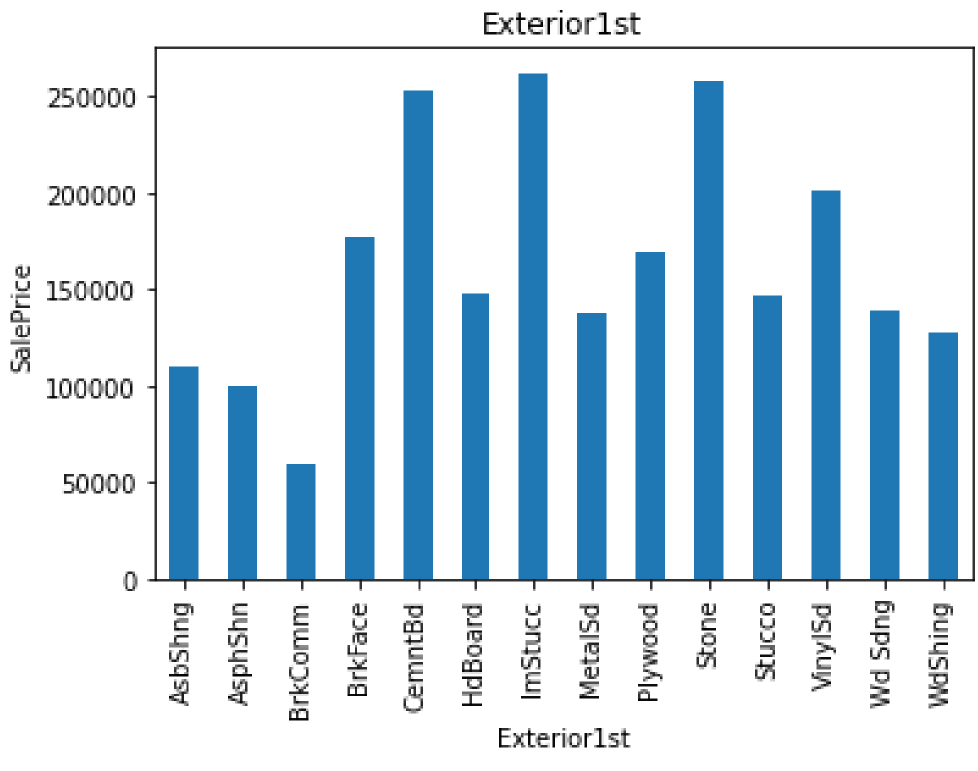
SalepriceVsElectrical



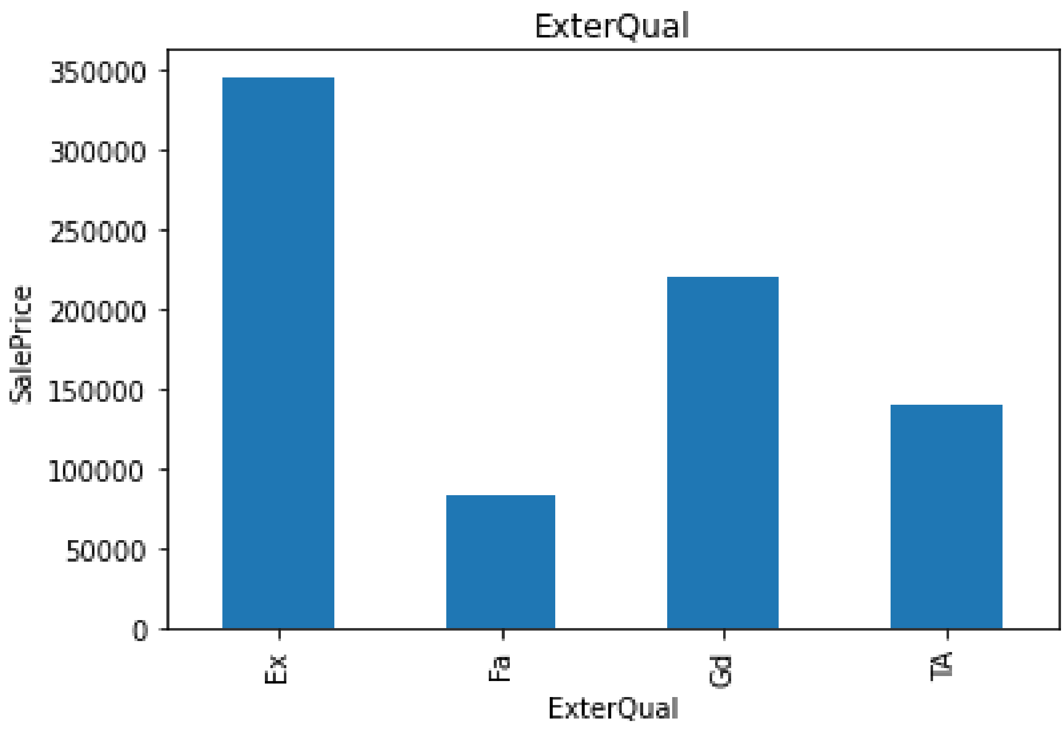
SalepriceVsExterior2nd



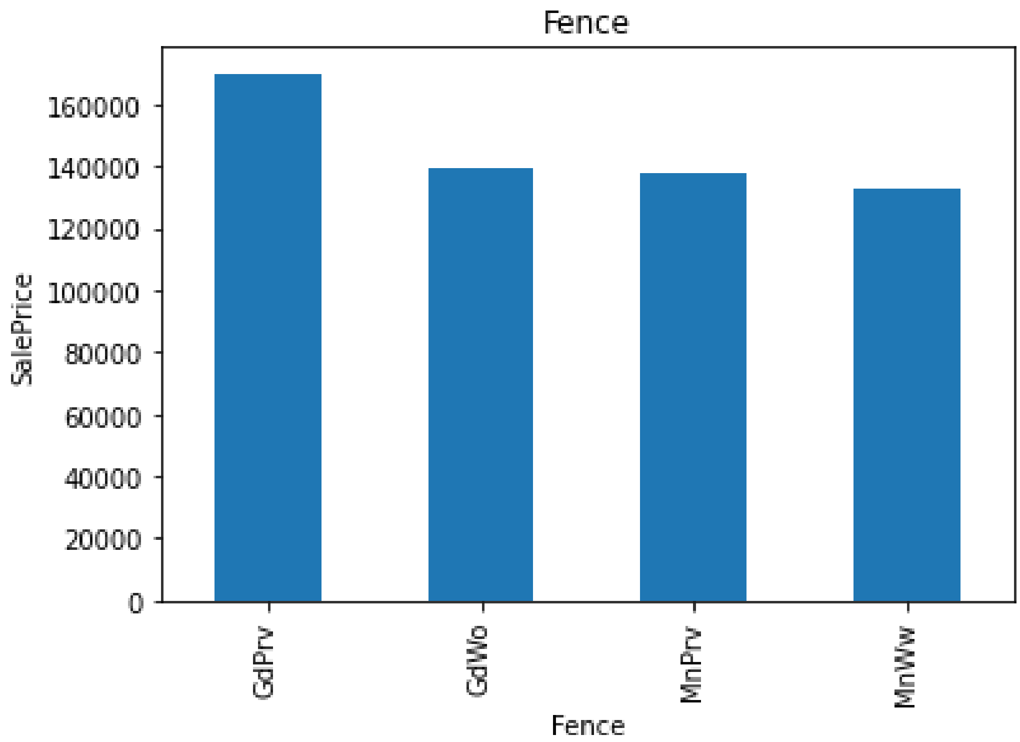
SalepriceVsExterCond



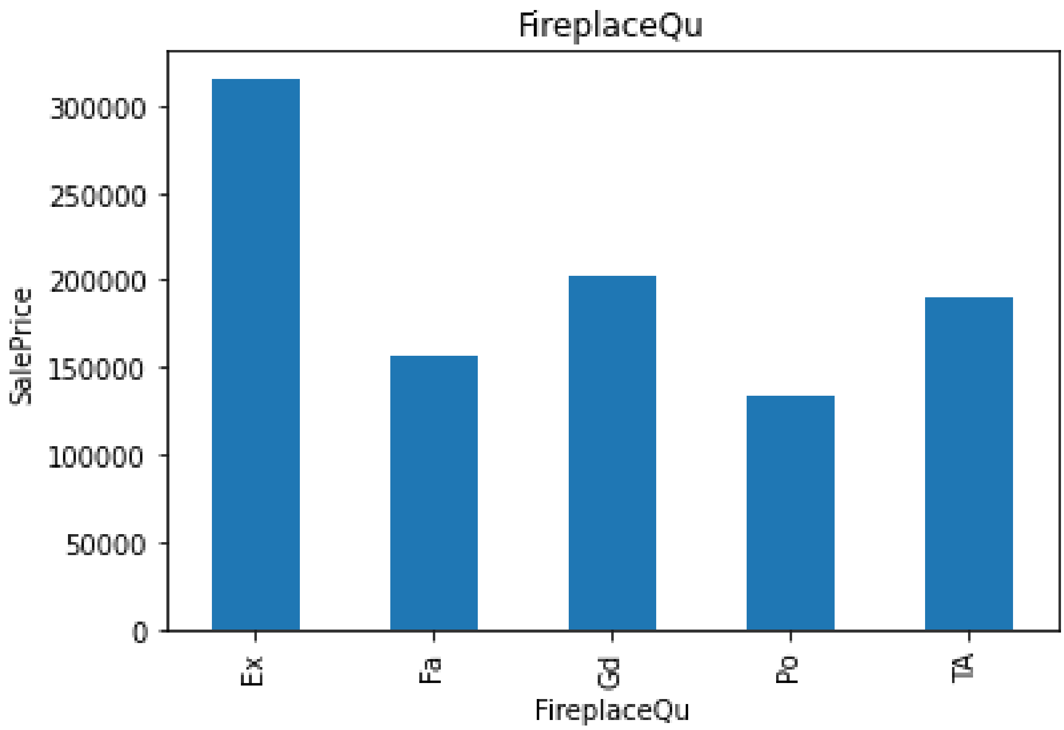
SalePriceVsExterior1st



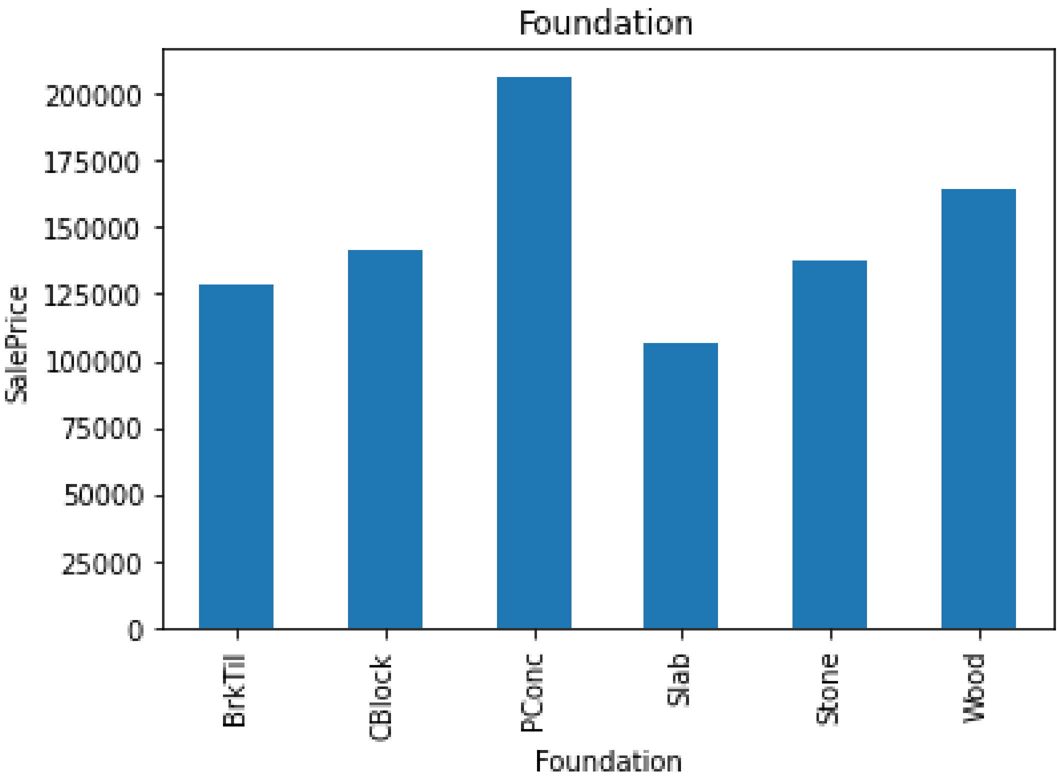
SalePriceVSExteriorQual



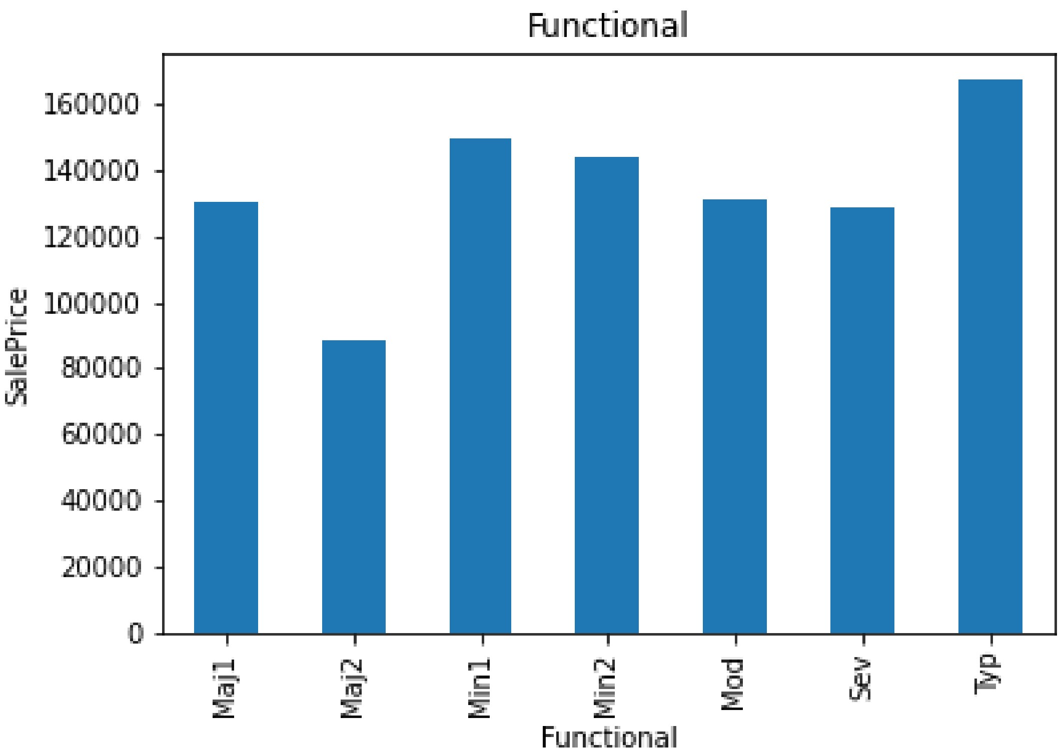
SalePriceVsFence



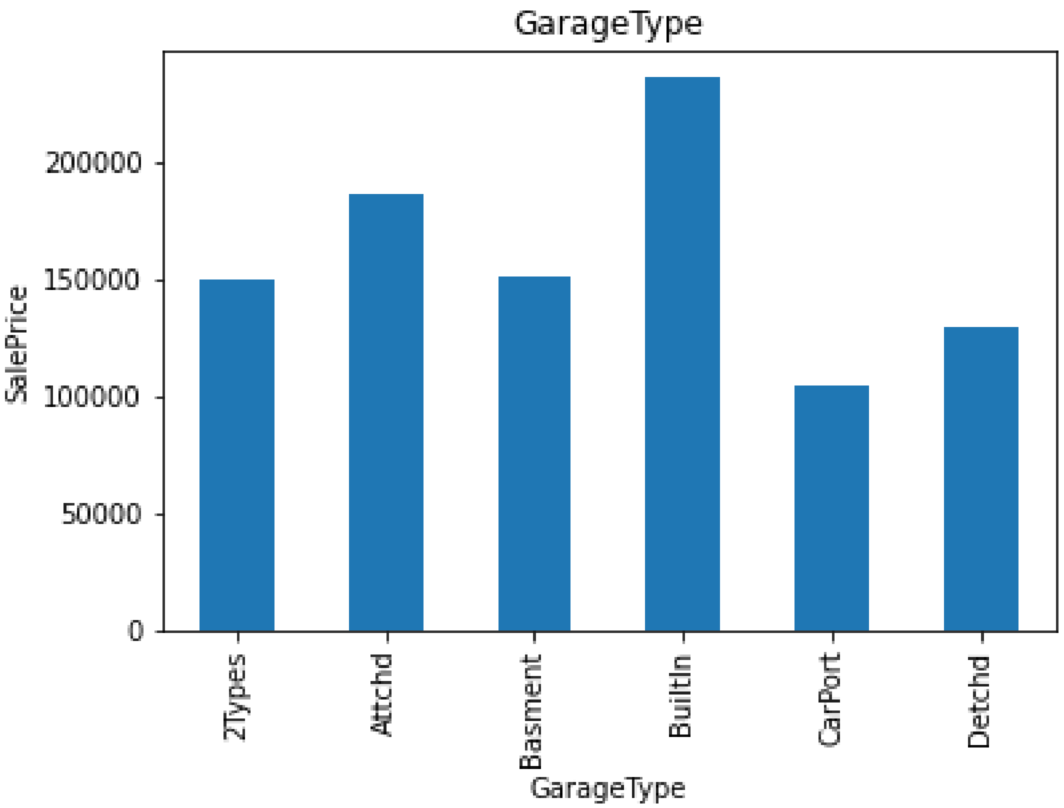
SalePriceVsFirePlaceQU



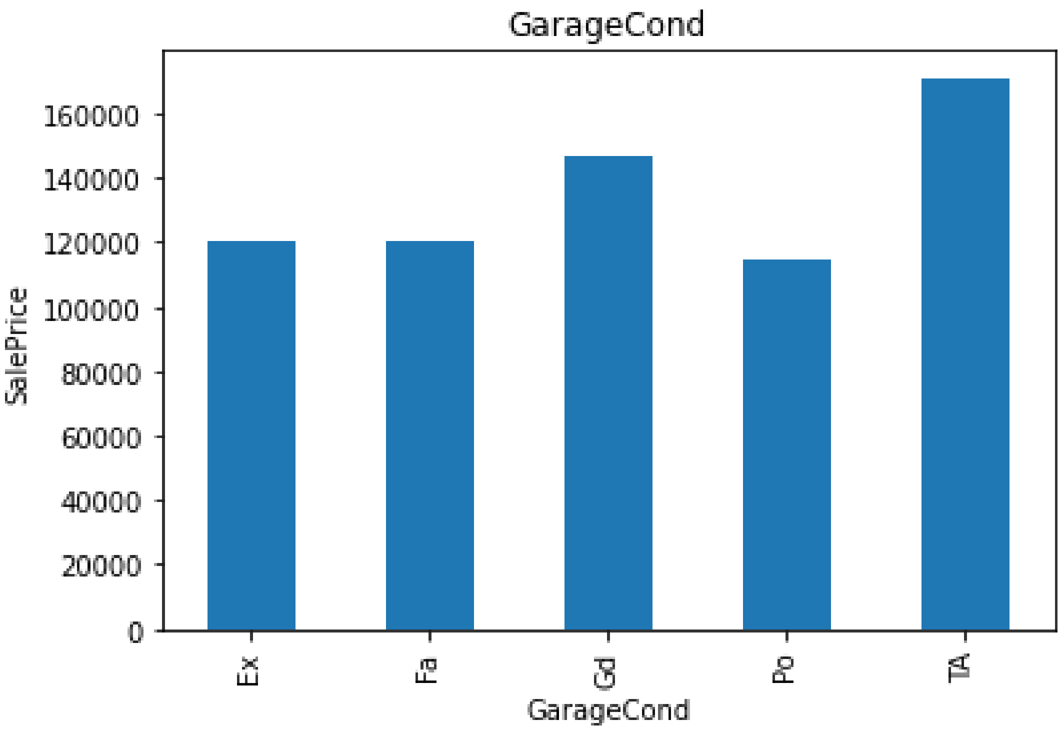
SalePriceVsFoundation



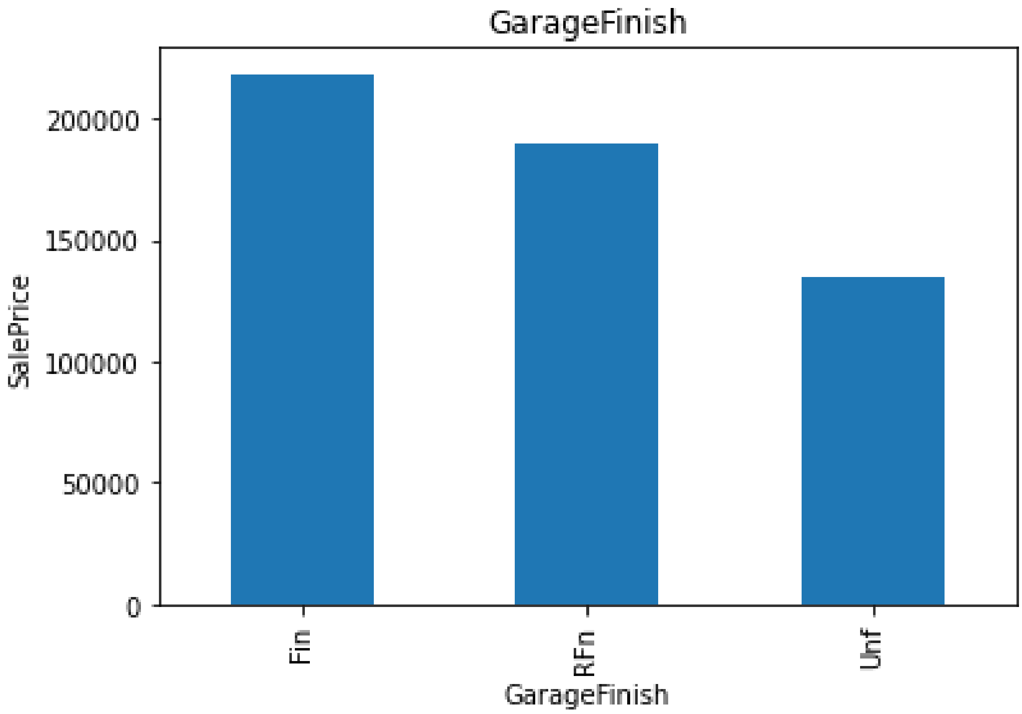
SalePriceVsFunctional



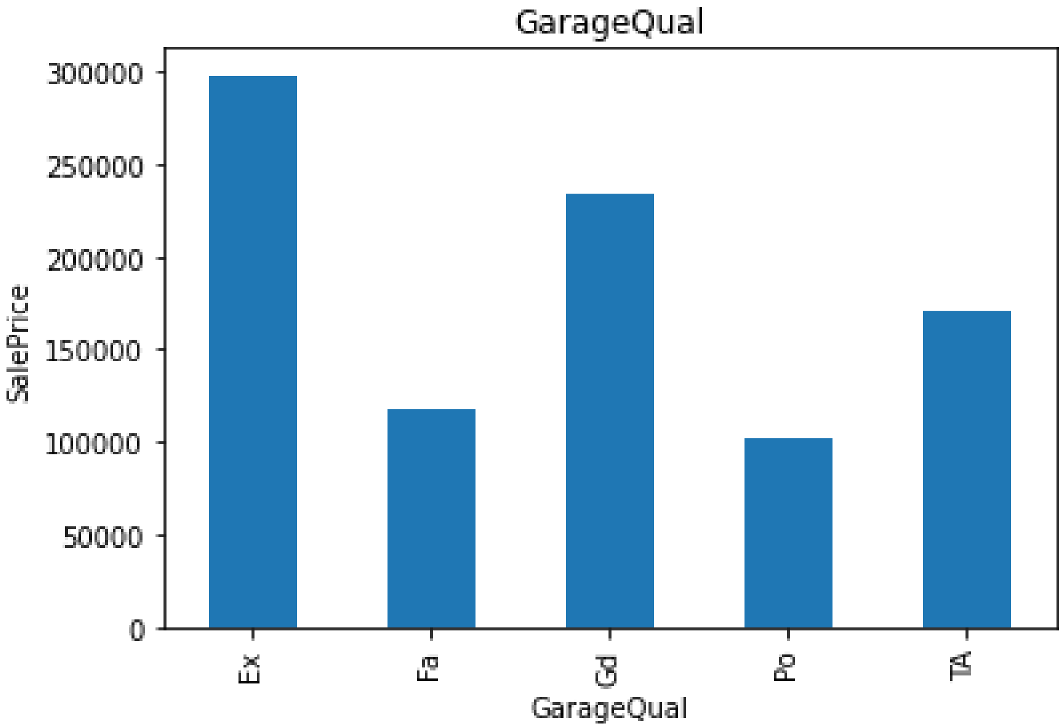
SalePriceVsGarageType



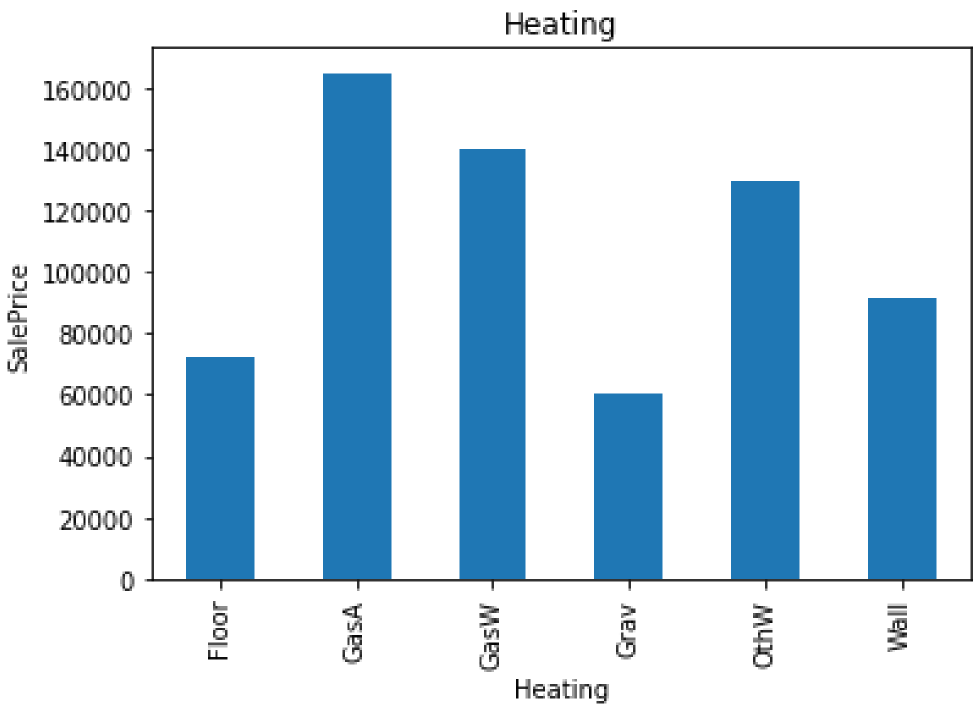
SalaryPriceVsGaragecond



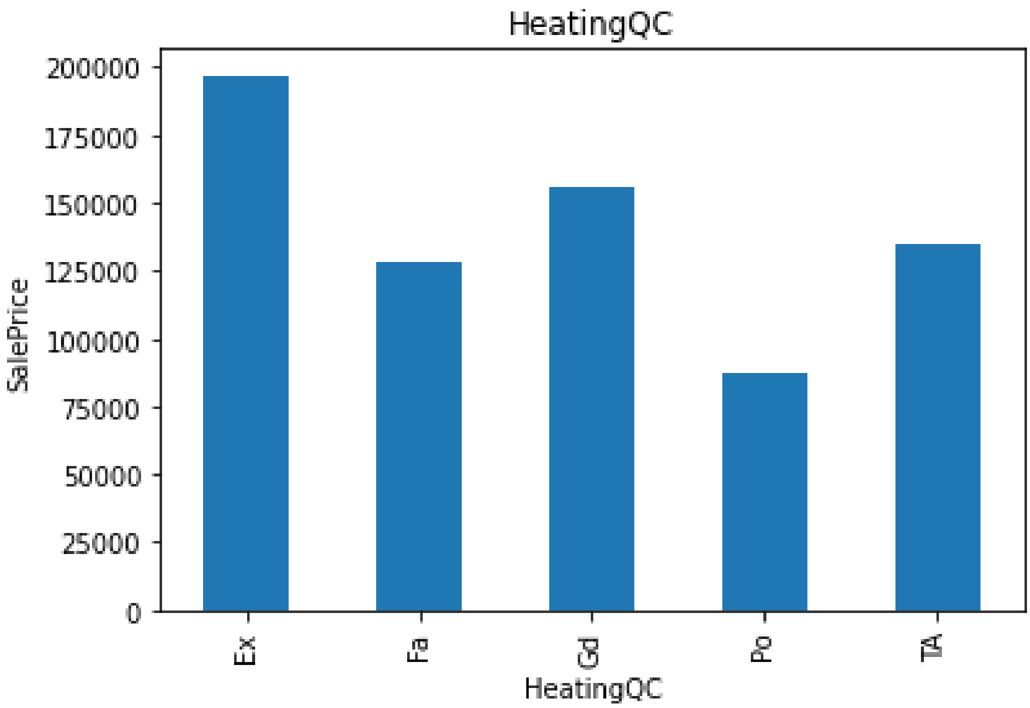
SalePriceVsGargeFinish



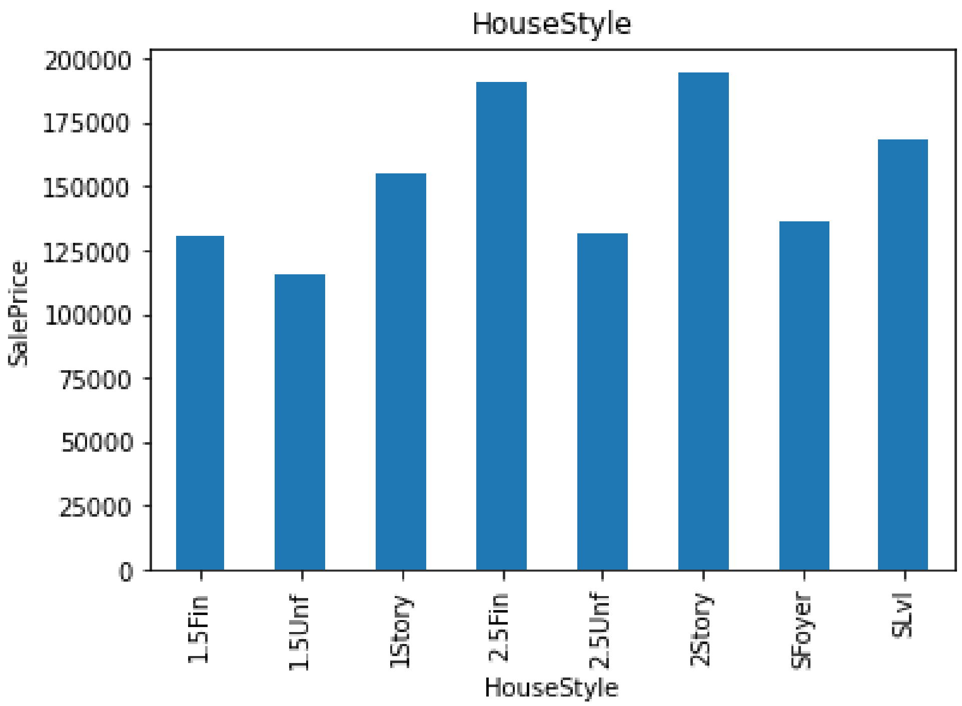
SalePriceVsGarageQual



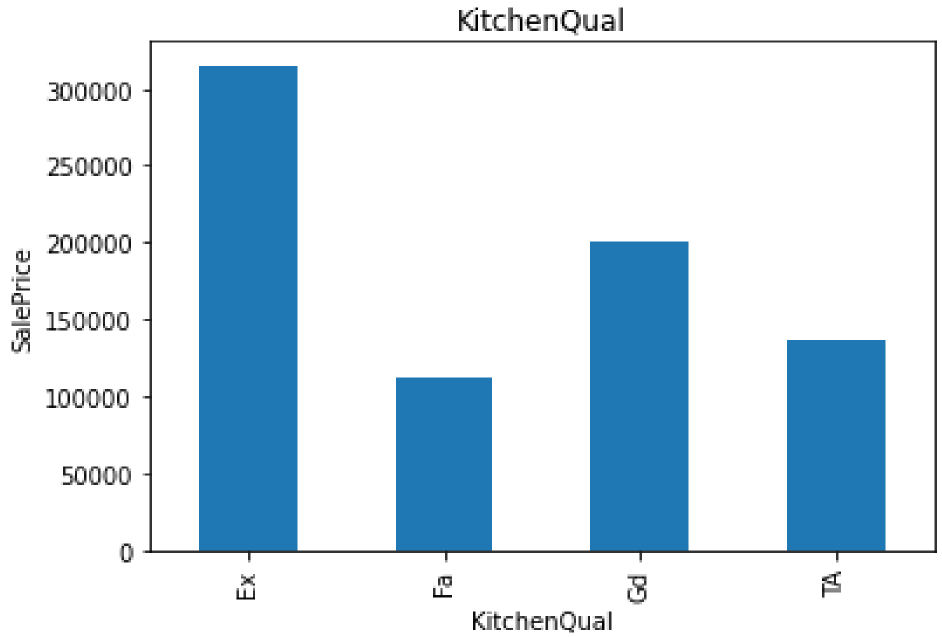
SalePriceVsHeating



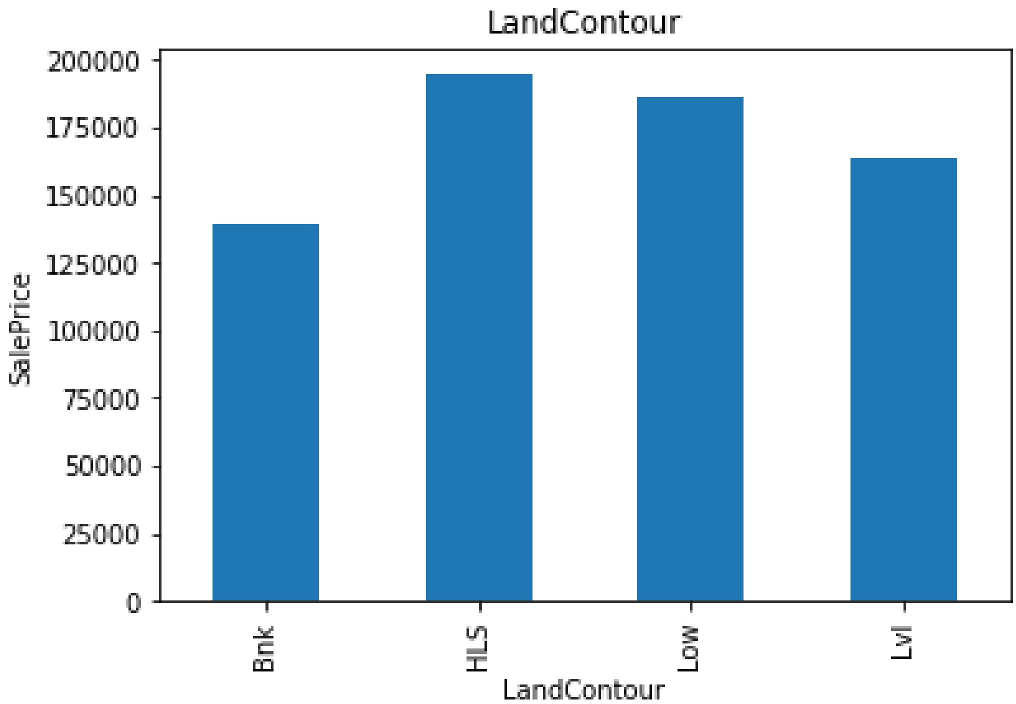
SalePriceVsHeatingQC



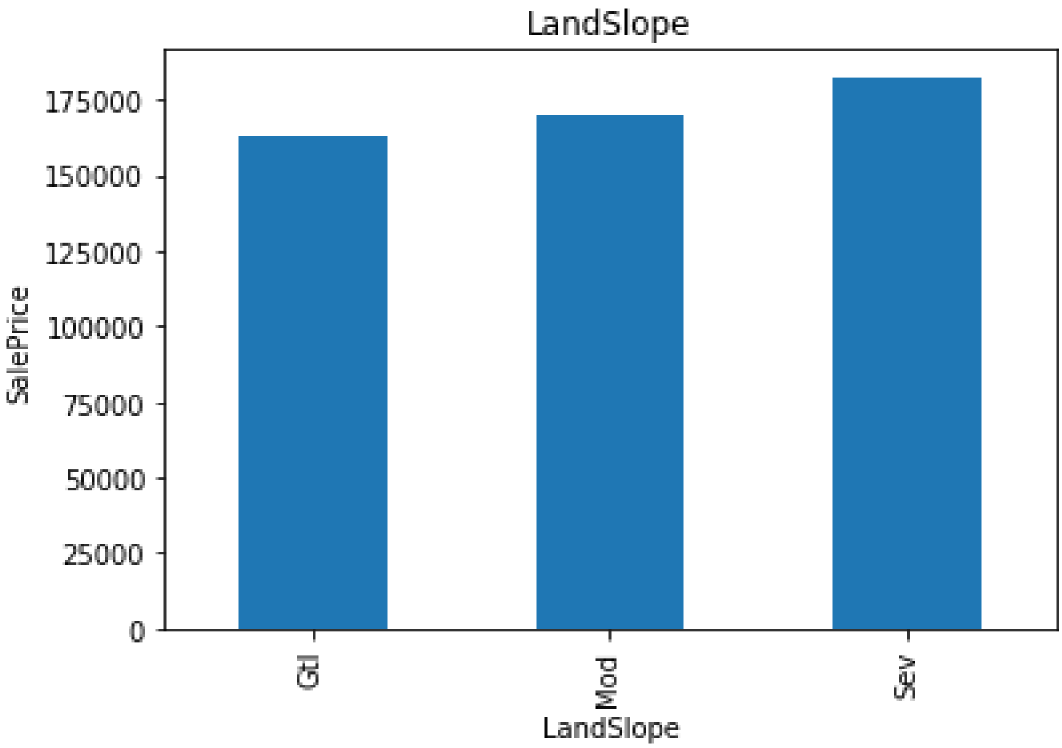
SalePriceVsHouseStyle



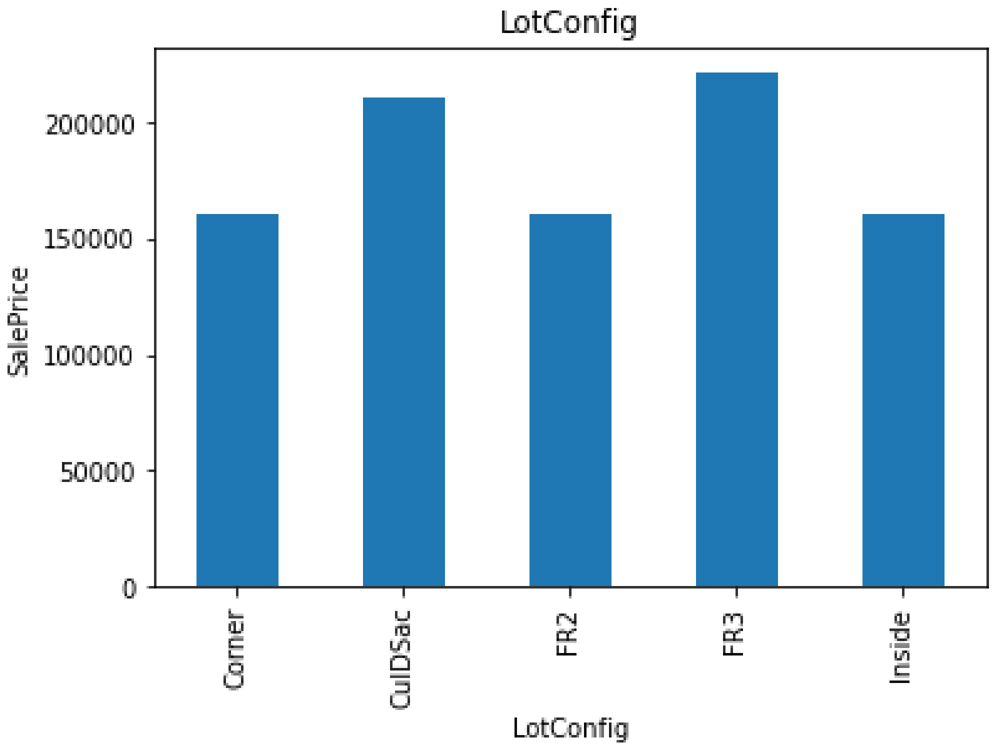
SalePriceVSKitchenQual



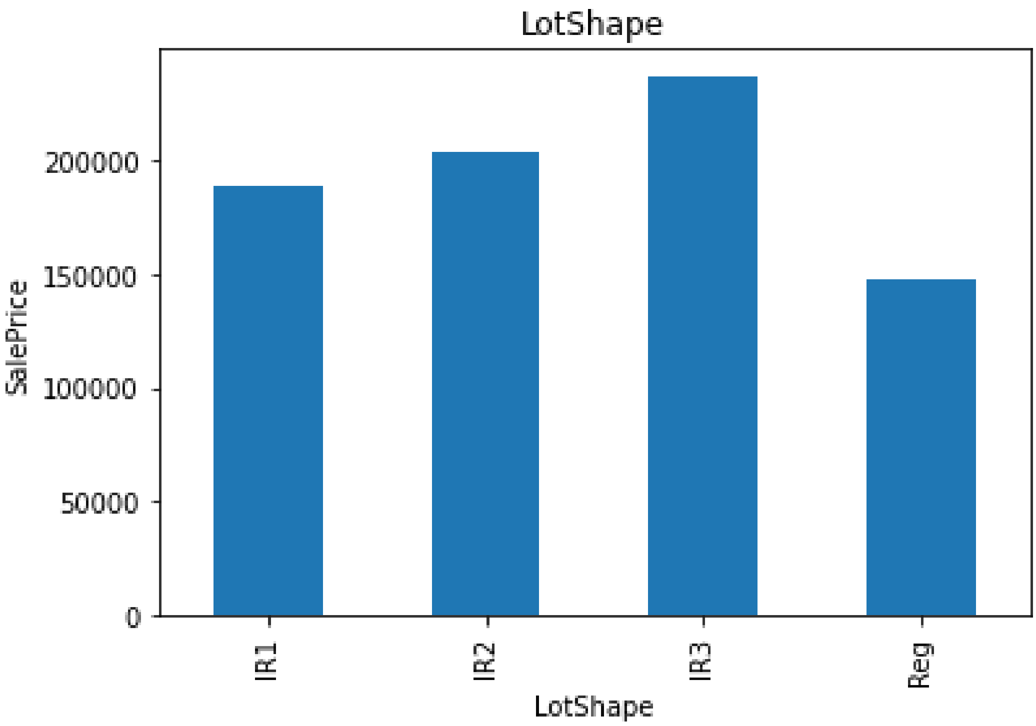
SalePriceVsLandContour



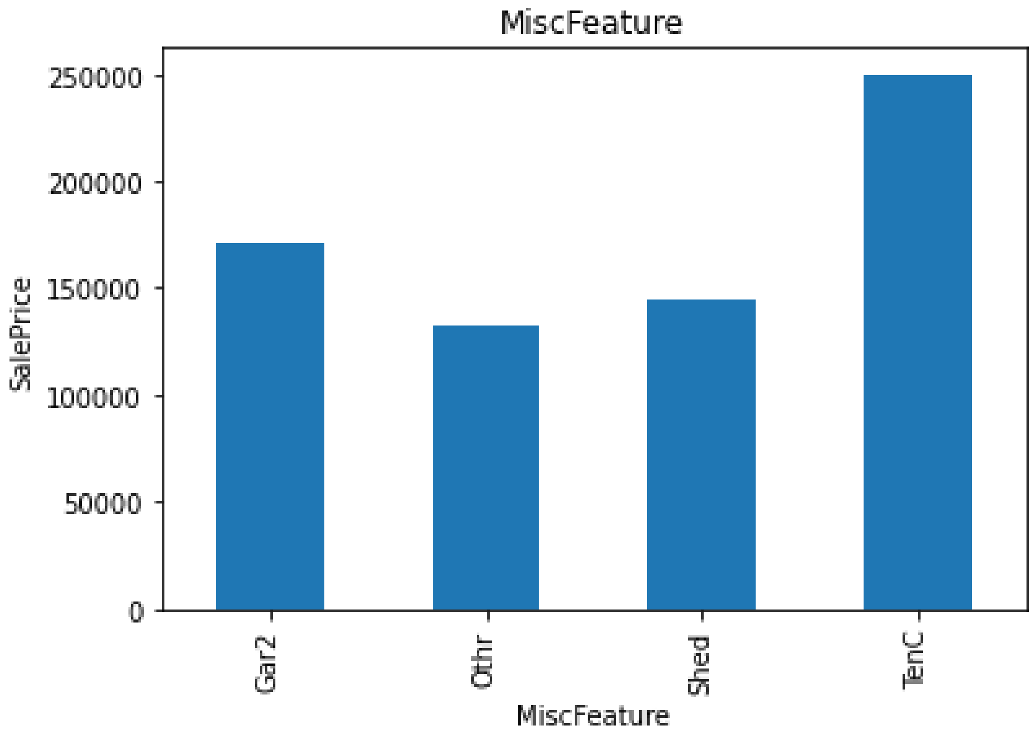
SalePriceVS LandSlope



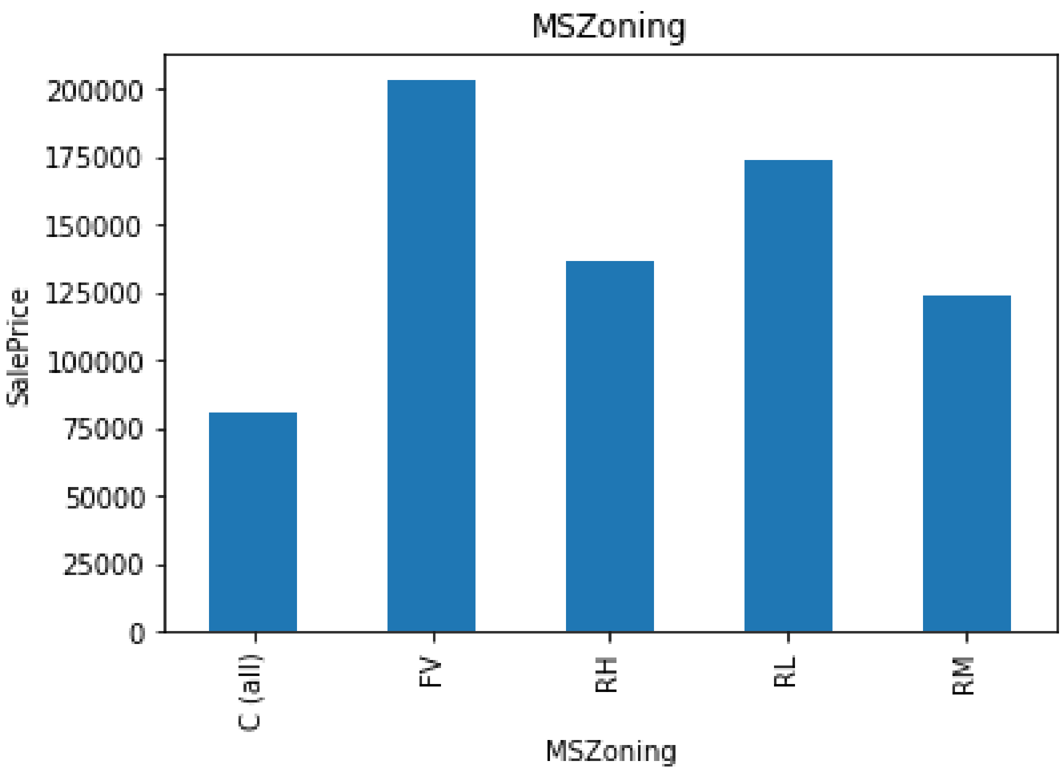
SalePriceVsLotConfig



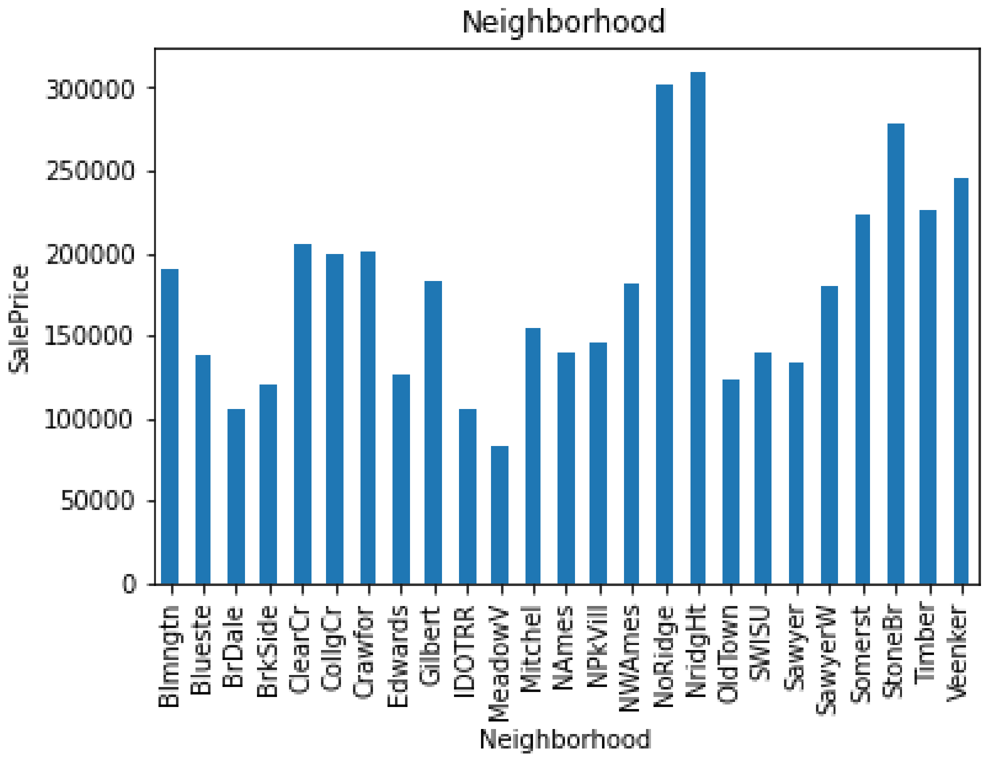
SalePriceVsLotshape



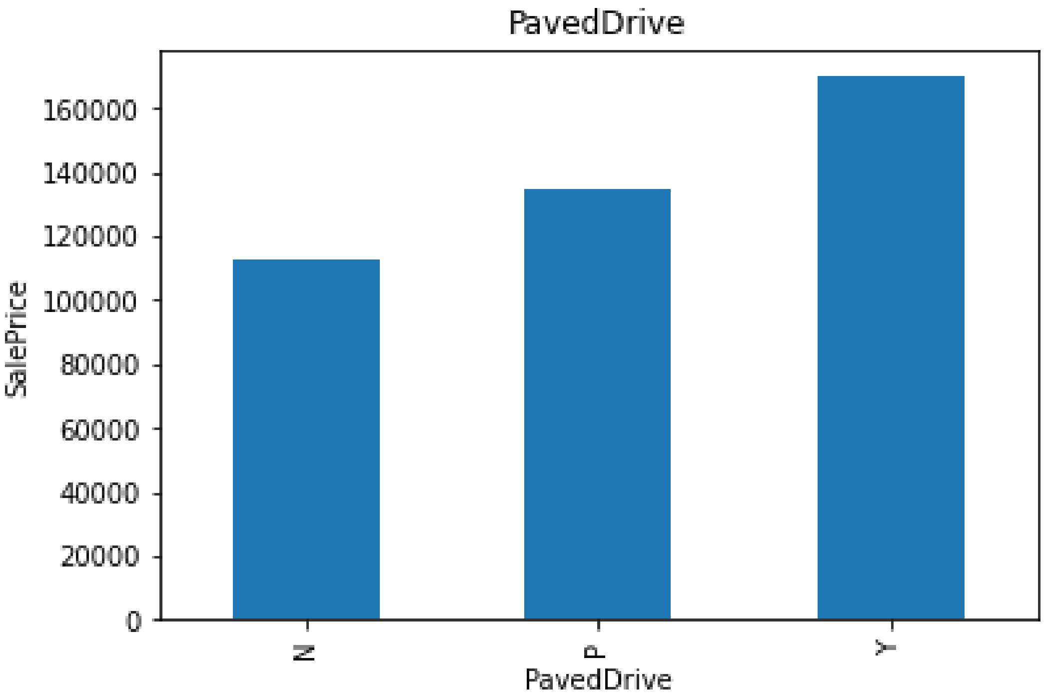
SalePriceVsMiscFeature



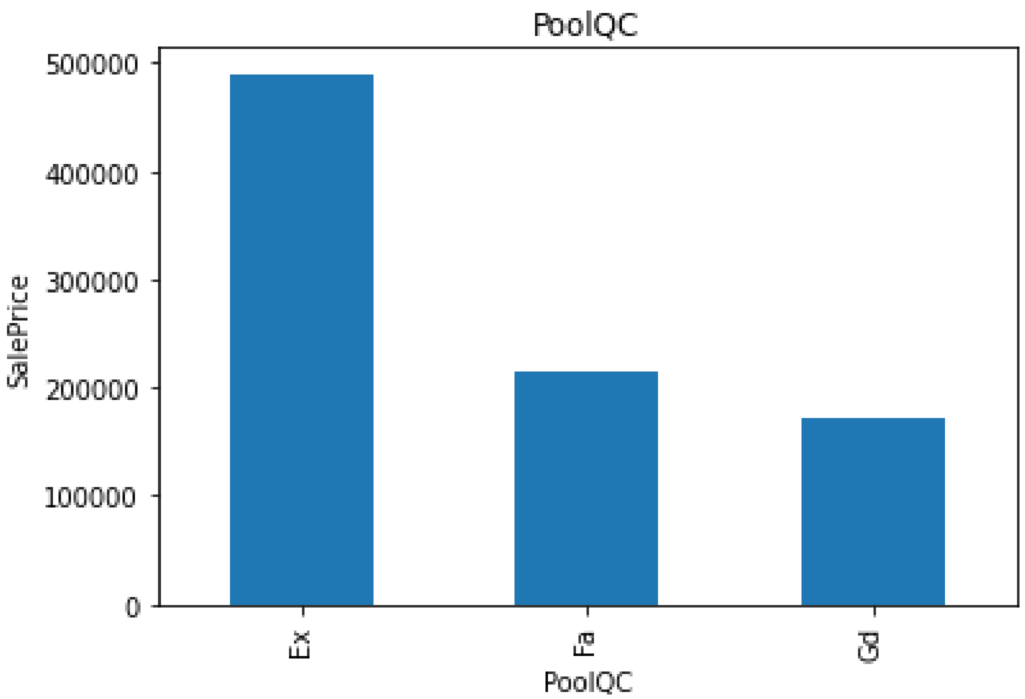
SalePriceVS MSZoning



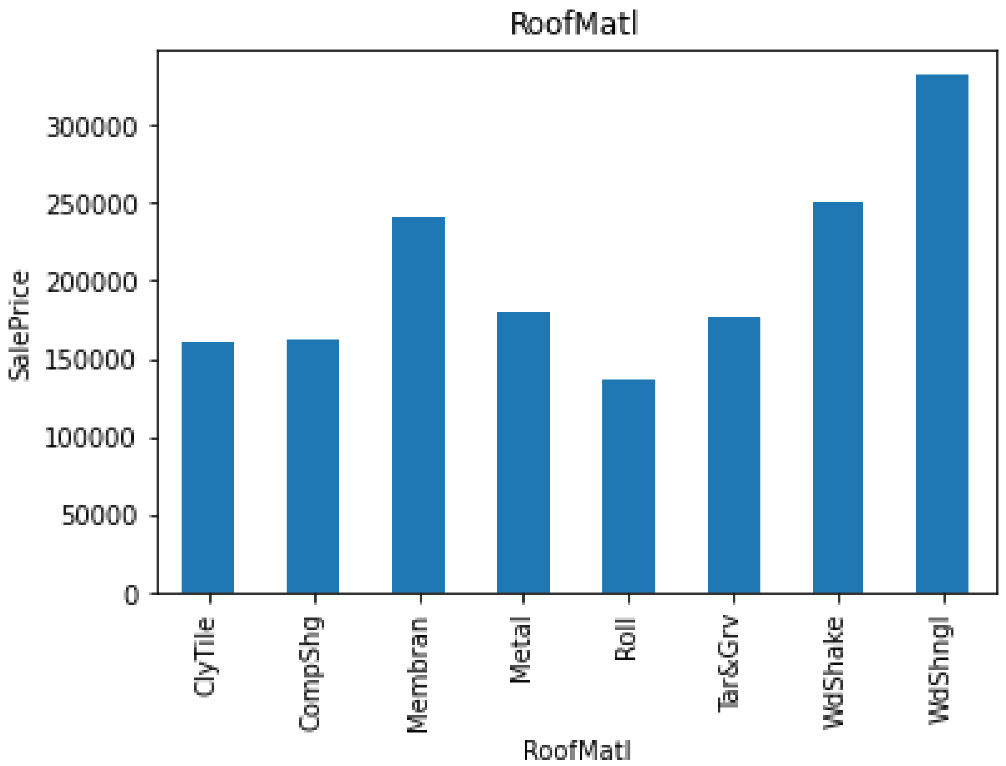
SalePriceVs Neighborhood



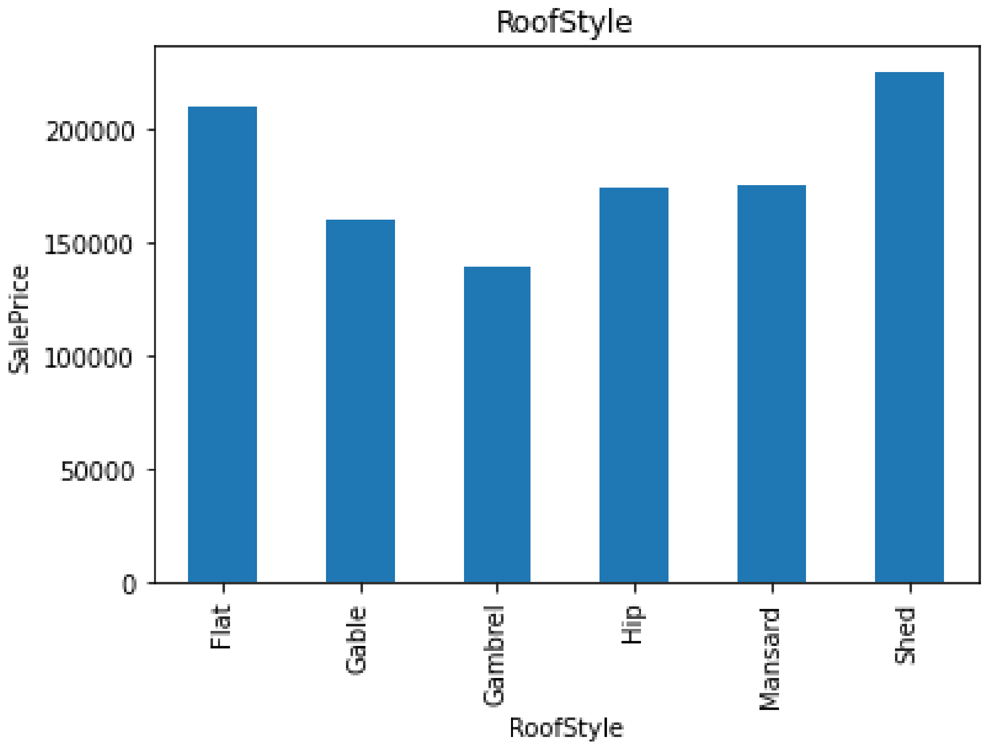
SalePriceVsPavedDrive



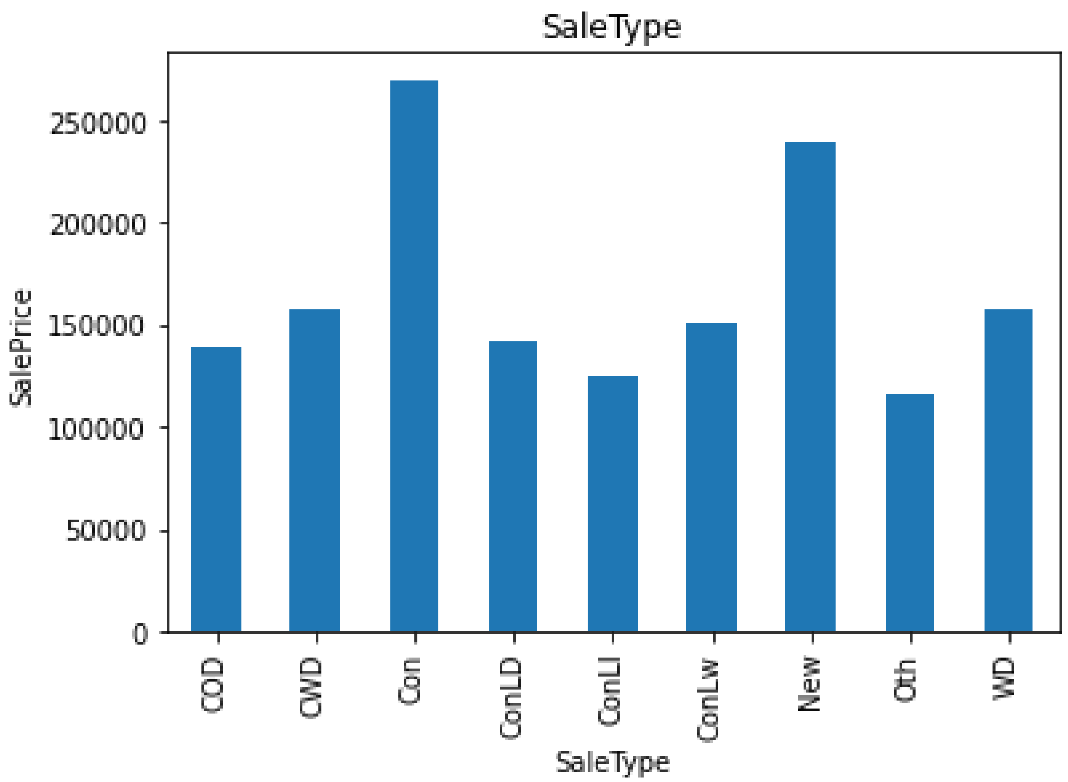
SalePriceVS PoolQC



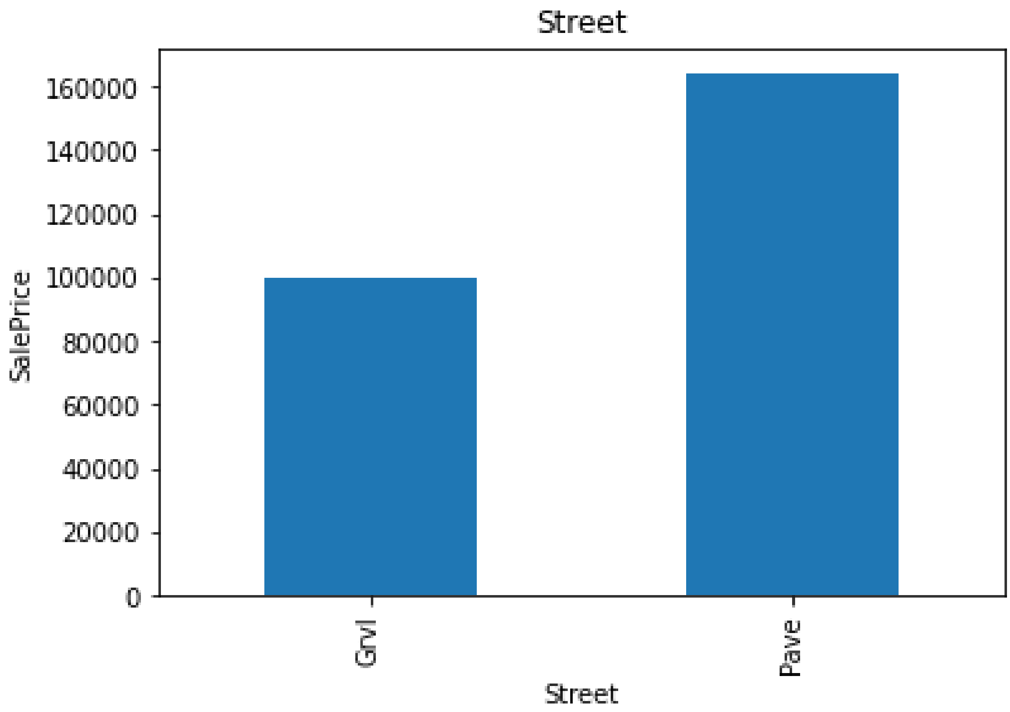
SalePriceVSRoofMati



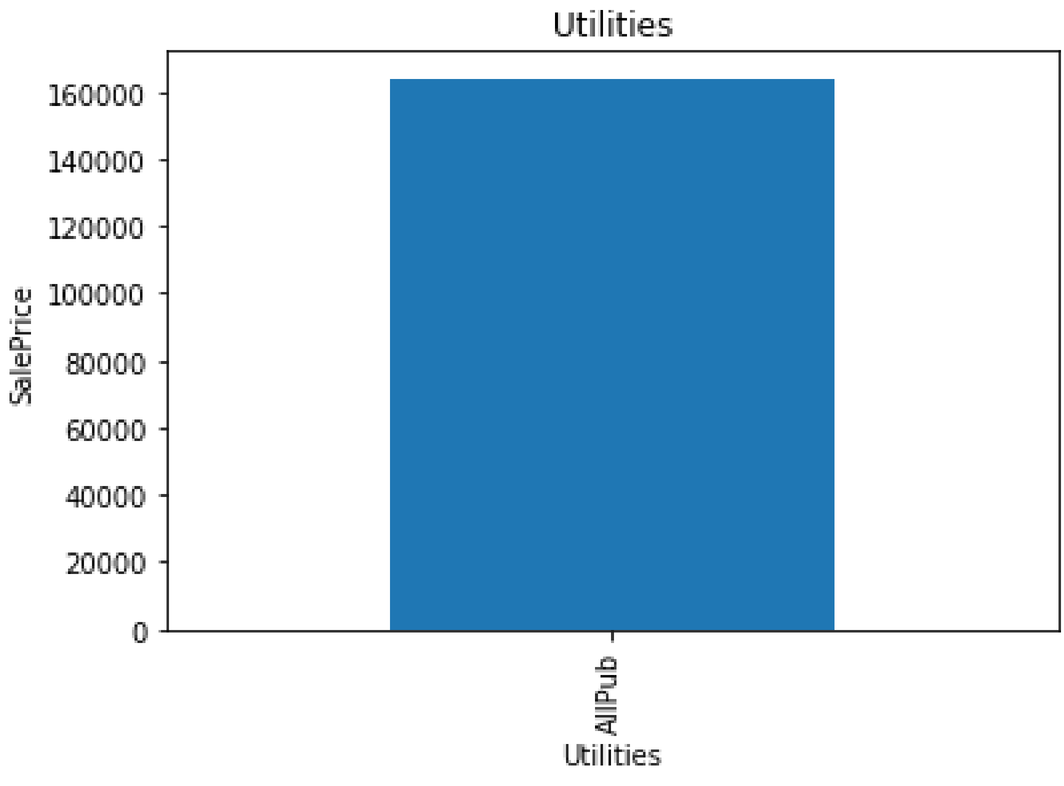
SalePriceVSRoofStyle



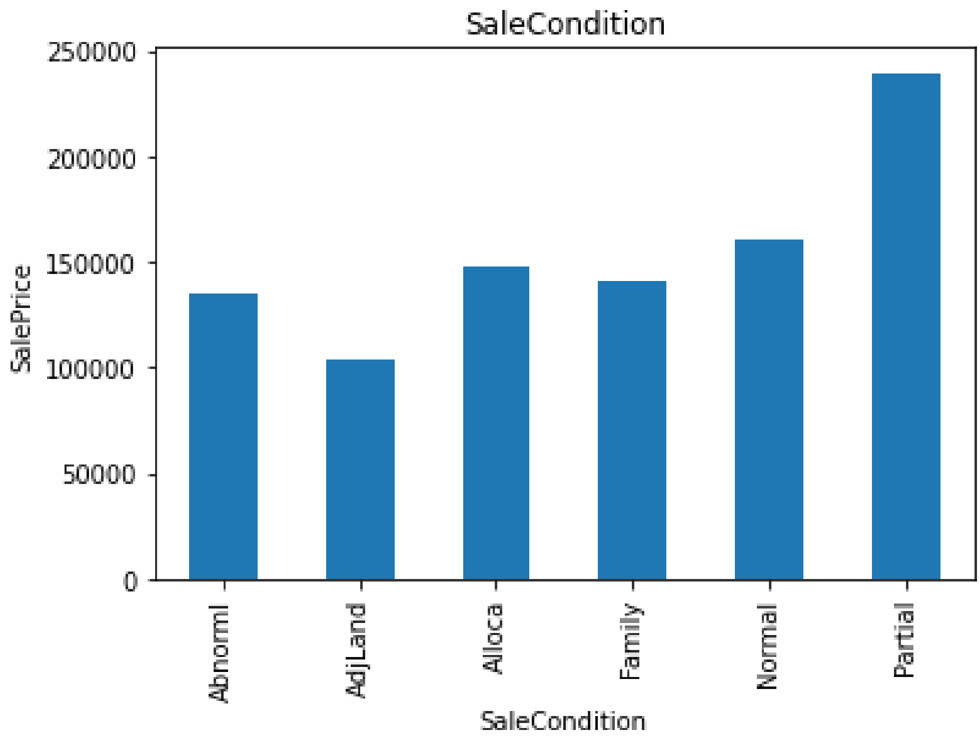
SalePriceVSSaleType



SalePriceVsStreet



SalePriceVsutilities



SalePriceVsSaleCondition

Missing value can be replaced by the word Missing in Feature Engineeringusing Pythoncode

def replace_cat_feature(df,

features_nan):data=df.copy()

data[features_nan]=data[features_nan].fillna('Missing')re

turndata

df=replace_cat_feature(df,

features_nan)df[features_nan].isnull().su

m()

Missingvalues present in Numerical Variables can replaced by the wordMissing usingthefollowingpythoncode:-

for feature in

numerical_with_nan:median_value=df[feature].me

dian()df[feature+'nan']=np.where(df[feature].isnull

(),1,0)df[feature].fillna(median_value,inplace=Tru

e)

df[numerical_with_nan].isnull().sum()

Extracting the new Feature from Date time Variable using the following Pythoncode:-

for feature in

['YearBuilt','YearRemodAdd','GarageYrBlt']:df[featur

e]=df['YrSold']-df[feature]

Make the logTransformation to remove the Right skewness in the histogramusingthefollowingpythoncode:- inthefeatures'LotFrontage','LotArea','1stFlrSF','GrLivArea', 'SalePrice',outliersarepresent

num_features=['LotFrontage','LotArea','1stFlrSF','GrLivArea','SalePrice']

forfeatureinnum_features:df[feat

ure]=np.log(df[feature])

Categorical Encoding:- after outliers, skewness is removed using boxplot, logtransformation,WeareusingLabelEncodertolabelfromcategoricaltonumerical usingthe followingcode:-

from sklearn.preprocessing import

LabelEncoderlabelencoder=LabelEncoder()

for feature in

categorical_features:df[feature]=labelencod

er.fit_transform(df[feature])

SimilarlytheMissingdatacanbehandledintestdata usingthefollowingpython code:-

##replcemissingvaluewithnewvalue

def replace_cat_feature_test(df1,

features_nan_test):data=df1.copy()data[features_nan_test]=data[f

eatures_nan_test].fillna('Missing') returndata

df1=replace_cat_feature_test(df1,features_nan_test)df1[features_nan_test].isnull(

).sum()

Missing value present in the test data can be removed using the python code:-

forfeature innumerical_with_nan_test:

median_value=df1[feature].median()df1[feature+'nan

']=np.where(df1[feature].isnull(),1,0)df1[feature].filln

a(median_value,inplace=True)

df1[numerical_with_nan_test].isnull().sum()

Similarly as in the train data, Extract the Date Time Variable using followingthepythoncode:-

##DateTimeVariables

for feature in

['YearBuilt','YearRemodAdd','GarageYrBlt']:df1[featu

re]=df1['YrSold']-df1[feature]

Feature 'LotFrontage', 'LotArea', '1stFlrSF', 'GrLivArea' have the missing value.It canbehandled byusinglog transformation

num_features_test=['LotFrontage', 'LotArea', '1stFlrSF',

'GrLivArea']forfeature innum_features_test:

df1[feature]=np.log(df1[feature])

similarly after removing the skewness, we are using the Label Encoder to convert categorical to numerical using the following python code:-

for feature in

categorical_features_test:df1[feature]=labelencoder.f

it_transform(df1[feature])

FeatureScaling:-

We are using the Min Max scaler for Scaling purpose:-

from sklearn.preprocessing import Min Max

Scalerscaler=MinMaxScaler()

after applying the MinMax Scaler, we are dividing the train and test data using the follow python code:-

y_train=df[['SalePrice']]

x=df.drop(['Id', 'SalePrice'], axis=

1)Regression Techniques used:-

1.Linear Regression

2.Lasso Regression

3.Ridge Regression

4.DecisionTree Regression

5.Random Forest Regression

Conclusion

Lasso regression model is considered as the best model among 5 because of less error 0.20 followed by ridge (0.22)

Submitted by

Taufiq Ahmed Chandshah