HOUSE PRICE PREDICTION

Submitted by:

YASHNA SHAH

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my SME Sapna Verma and Flip Robo Technologies who gave me the opportunity to work on this project – House Price Prediction, which helped me in doing lots of research wherein I came across lots of new things to learn about.

Also, I have referred some external resources that helped me to complete the project. All the external resources used in completing the project are listed below:

INTRODUCTION

• Business Problem Framing

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 modelling, Market mix modelling, 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

• Conceptual Background of the Domain Problem

A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia.

The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

• Which variables are important to predict the price of variable?

• How do these variables describe the price of the house?

• Review of Literature

Based on the sample data provided, we understand that the company is looking at prospective properties to buy houses to enter the market. The data given, explains that it is a regression problem where we need to build a machine learning model in order to predict the actual prices of the prospective properties and then decide whether to invest in them or not. Also, we have other independent variables which would help us in understanding which features are important to predict the values and how do these variables describe the prices of the house.

- ## Motivation for the Problem Undertaken

The main objective of doing this project is to build a machine learning model to predict the prices of the house based on the supporting features. Will be predicting the same with the help of machine learning algorithms. The data is provided to us from client database. We need to do some predictions that could help the clients in the further investments and also in improvement in selection of customers.

To explore various impacts of features on prediction methods, this project will apply machine learning approaches to investigate difference between several models. This project will also validate multiple techniques in model implementation on regression and provide an optimistic result for predicting the house prices.

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

We are building a machine learning model to predict the actual value of the prospective properties and to decide whether to invest in them or not. This model will help us to determine which variables are important to predict the price of variables and also how do these variables describe the price of the house. With the help of the independent variables, will be able to determine the price of the houses.

Regression analysis is a form of predictive modelling technique which investigates the relationship between dependent and independent variables. This technique is used for forecasting, time series modelling, and also find the relationship between the variables. The most common form of regression analysis is linear regression, in which one finds the line that most closely fits the data according to a mathematical criterion.

- ## Data Sources and their formats

Dataset provided by Flip Robo Technologies is in the CSV format. There are 2 datasets given to us. One is training data and the other is testing data.

1. Train file will be used for training the model, where the model will learn from this file. The training data contains all the independent variables and the target variable. The size of the train data is 1168 records
2. Test file contains all the independent variables, but not the target variable. Here we will be applying the model to predict the target variable. The size of the test data is 292 records.

- # Data Preprocessing Done

Data Pre-processing refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training machine learning models. Whenever the data is gathered from various sources, it is collected in raw format. Data pre-processing is considered as an integral part in Machine Learning. Its extremely important to pre-process the data before feeding it to our model. Hence, it is considered to be the first and crucial step while creating a machine learning model. Data pre-processing steps used in this project are as below:

1. Loading the dataset
2. Checked the number of rows and columns in the dataset using df.shape().
3. Checked the null values in the dataset and filled them using mean, median mode.
4. Dropped the unrequired columns from the dataset.
5. Checked the unique values.
6. Converted all the data to numeric data type using Label Encoding technique.

- # Data Inputs- Logic- Output Relationships

The steps include:

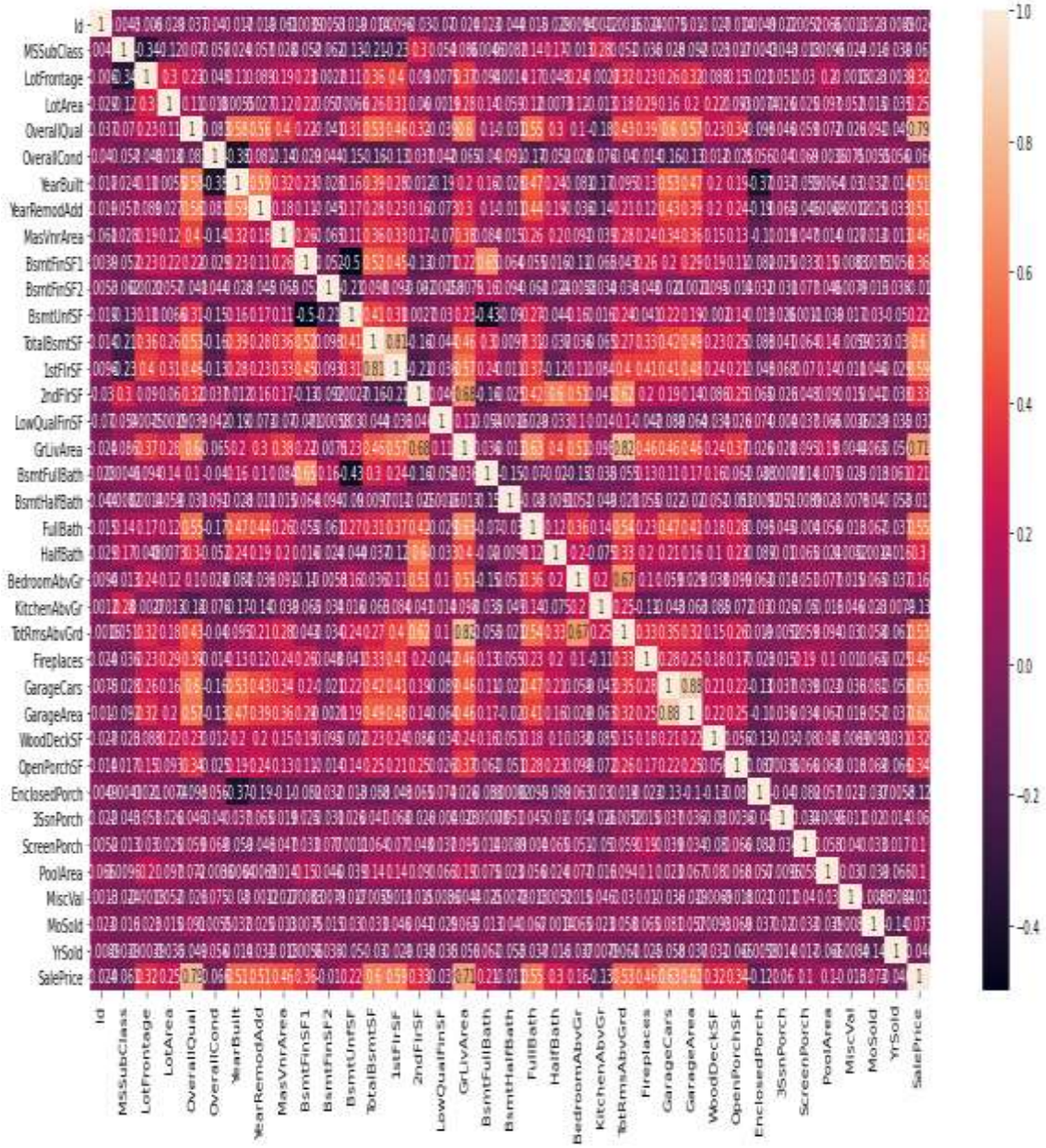
- Data Pre- processing and feature Engineering- which includes converting the data type using Label Encoder.

```
# Using Label Encoding to convert non numeric datato numeric data
LE = LabelEncoder()
for column in df_train.columns:
    df_train[column] = LE.fit_transform(df_train[column])
```

```
df_train.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2
0	84	11	3	41	80	1	0	3	0	4	0	13	2	2
1	720	0	3	66	808	1	0	3	0	4	1	12	2	2
2	642	5	3	63	449	1	0	3	0	1	0	15	2	2
3	78	0	3	76	632	1	0	3	0	4	0	14	2	2
4	341	0	3	41	821	1	0	3	0	2	0	14	2	2

- Checking the correlation among the variables and plotting a correlation matrix for graphical representation.



- Correlation helps in feature selection and making the feature ready for the model.

- from the above plot we observe that SalesPrice is highly positively correlated with the columns OverallQual, YearBuilt, YearRemodAdd, TotalBsmtSF, 1stFlrSF, GrLivArea, FullBath, TotRmsAbvGrd, GarageCars, GarageArea. whereas SalePrice is negatively correlated with OverallCond, KitchenAbvGr, Encloseporch, YrSold. There is multicollinearity in between columns so we will be using Principal Component Analysis(PCA). No correlation has been observed between the column Id and other columns so we will be dropping this column.

- State the set of assumptions (if any) related to the problem under consideration

The assumption part for me was relying strictly on the data given and taking into consideration that the separate datasets are provided, training and the testing dataset which were obtained from people and survey was also done on it depending on their preferences.

- Hardware and Software Requirements and Tools Used

Hardware:

Intel i5

Software :

Jupyter Notebook (Anaconda 3)

Language : Python

Libraries :

1. Pandas - Used for loading the data and doing analysis

2. Numpy - Used for loading the data and doing analysis.

3. Matplotlib - Used for data visualization.

4. Seaborn - Also used for plotting library and is more advanced than matplotlib.

5. SkLearn - Used for data Pre processing

6. Scipy

7. Statsmodel - Provides functions for calculating mathematical statistics of numeric data.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

- We had null values in our dataset, which we imputed using mean, median and mode methods.
- Dropped unrequired columns.
- Converted all the categorical data into numeric data using Label Encoder.
- Identified skewness and outliers in the dataset and removed the outliers with the help of z-score method.
- Splitting the dataset and scaling the data with Standard Scaler method
- Used PCA for multicollinearity.

```
pca =PCA(n_components=65)
pca.fit(x)
```

```
PCA(n_components=65)
```

```
pca.explained_variance_ratio_
```

```
array([1.78843244e-01, 7.72501498e-02, 6.73717578e-02, 5.86791678e-02,
       3.53997805e-02, 3.28175474e-02, 3.17513807e-02, 2.81995641e-02,
       2.55983013e-02, 2.34216786e-02, 2.29305558e-02, 2.21073192e-02,
       2.10635799e-02, 2.05420653e-02, 1.98814398e-02, 1.90481053e-02,
       1.81986657e-02, 1.63198172e-02, 1.61454812e-02, 1.52841927e-02,
       1.42021279e-02, 1.40684623e-02, 1.35941286e-02, 1.31181255e-02,
       1.26736088e-02, 1.23375765e-02, 1.16851079e-02, 1.10770623e-02,
       1.07546564e-02, 1.05863688e-02, 9.64121946e-03, 9.30431674e-03,
       9.12832024e-03, 8.78698956e-03, 8.16709849e-03, 7.63938742e-03,
       7.43274669e-03, 7.19943409e-03, 6.56752031e-03, 6.43274812e-03,
       5.25873259e-03, 4.95123758e-03, 4.70363685e-03, 4.65497713e-03,
       4.07656243e-03, 3.78755623e-03, 3.16588777e-03, 3.05230180e-03,
       2.33040846e-03, 2.27057683e-03, 1.85350501e-03, 1.67744193e-03,
       1.33109273e-03, 1.09889217e-03, 3.15273831e-04, 2.21115886e-04,
       3.52706109e-32, 1.19509175e-33, 7.87189652e-34, 7.87189652e-34,
       7.87189652e-34, 7.87189652e-34, 7.87189652e-34, 7.87189652e-34,
       7.87189652e-34])
```

```
x_returned_pca=pca.transform(x)
```

```
x_returned_pca.shape
```

```
(490, 65)
```

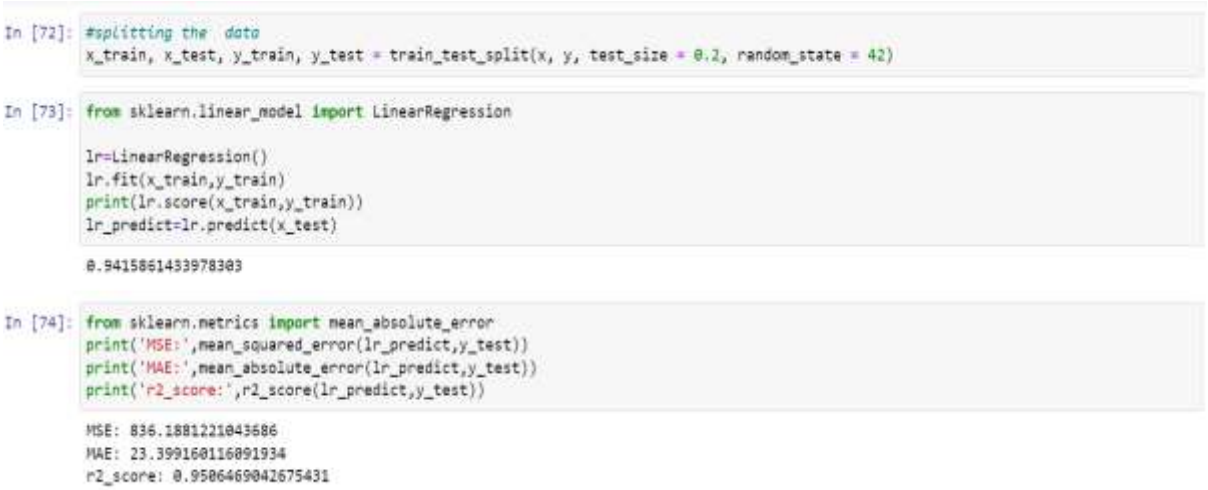
- # Testing of Identified Approaches (Algorithms)

The algorithms used for model building and predictions are listed below:

1. Linear Regression Model
2. RIDGE Regularization Regression Model
3. LASSO Regularization Regression Model
4. SGB Regressor
5. Random Forest Regressor.

These algorithms have been used for both training and testing the model.

- Run and Evaluate selected models



```
In [72]: #splitting the  data
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)

In [73]: from sklearn.linear_model import LinearRegression

         lr=LinearRegression()
         lr.fit(x_train,y_train)
         print(lr.score(x_train,y_train))
         lr_predict=lr.predict(x_test)

         0.9415861433978303

In [74]: from sklearn.metrics import mean_absolute_error
         print('MSE:',mean_squared_error(lr_predict,y_test))
         print('MAE:',mean_absolute_error(lr_predict,y_test))
         print('r2_score:',r2_score(lr_predict,y_test))

         MSE: 836.1881221043686
         MAE: 23.399160116091934
         r2_score: 0.9506469042675431
```

```
# Using SGD regressor
from sklearn.linear_model import SGDRegressor
from sklearn import metrics
sgd=SGDRegressor()
sgd.fit(x_train,y_train)
pred=sgd.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
print('MSE:',metrics.mean_squared_error(y_test,pred))
print('MAE:',metrics.mean_absolute_error(y_test,pred))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.955279117299597
MSE: 875.6734436268313
MAE: 23.916090515521987
RMSE: 29.59178000098729
```

```
# Using LASSO
from sklearn.linear_model import Lasso,Ridge
parameters={'alpha':[0.0001,0.001,0.01,0.1,1,10],'random_state':list(range(0,100))}
ls=Lasso()
Z=GridSearchCV(ls,parameters)
Z.fit(x_train,y_train)
print(Z.best_params_)
```

```
{'alpha': 1, 'random_state': 0}
```

```
ls=Lasso(alpha=1,random_state=0)
ls.fit(x_train,y_train)
pred=ls.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
print('MSE:',metrics.mean_squared_error(y_test,pred))
print('MAE:',metrics.mean_absolute_error(y_test,pred))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.9576780448362539
MSE: 828.7003739960622
MAE: 23.211610834353923
RMSE: 28.78715640691283
```

```
# Using RIDGE
parameters={'alpha':[0.001,0.01,0.1,1,10],'solver':['auto','svd','cholesky','lsqr','sparse_cg','sag','saga']}
rd=Ridge()
Z1=GridSearchCV(rd,parameters)
Z1.fit(x_train,y_train)
print(Z1.best_params_)
```

```
{'alpha': 1, 'solver': 'sparse_cg'}
```

```
rd=Ridge(alpha=10,solver='lsqr')
rd.fit(x_train,y_train)
pred=rd.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
print('MSE:',metrics.mean_squared_error(y_test,pred))
print('MAE:',metrics.mean_absolute_error(y_test,pred))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.9556656826077627
MSE: 868.104161578981
MAE: 23.760853747221127
RMSE: 29.463607409463307
```

```
]: #Fitting the RandomForest Model
   from sklearn.ensemble import RandomForestRegressor
   reg_rf = RandomForestRegressor()
   reg_rf.fit(x_train, y_train)

]: RandomForestRegressor()

]: #predicting the value on X_test

   y_pred = reg_rf.predict(x_test)

]: reg_rf.score(x_train, y_train)

]: 0.9843960261574494

]: reg_rf.score(x_test, y_test)

]: 0.9164725706509037

]: metrics.r2_score(y_test, y_pred)

]: 0.9164725706509037
```

- Key Metrics for success in solving problem under consideration

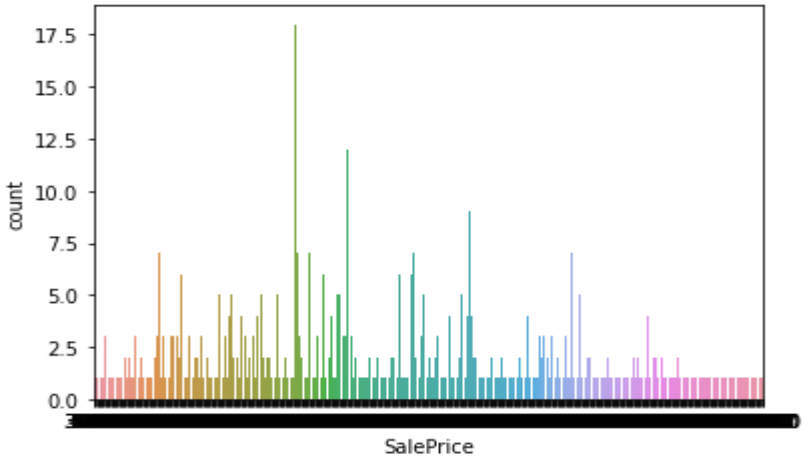
Will go with Random Forest Regressor

Reasons
1. Random Forest reduces overfitting in decision tree and helps to improve accuracy.
2. It is flexible for both classification and regression tasks.
3. It also works well with both categorical and continuous values.
4. It is a rule based approach.
5. It automates missing values present in the data

The key metrics used here were R2_Score, MSE, MAE, RMSE and Cross Validation Score. We also tried to find out the best parameters by using Hyper Parameter Tuning and Random Grid Search to increase the accuracy score.

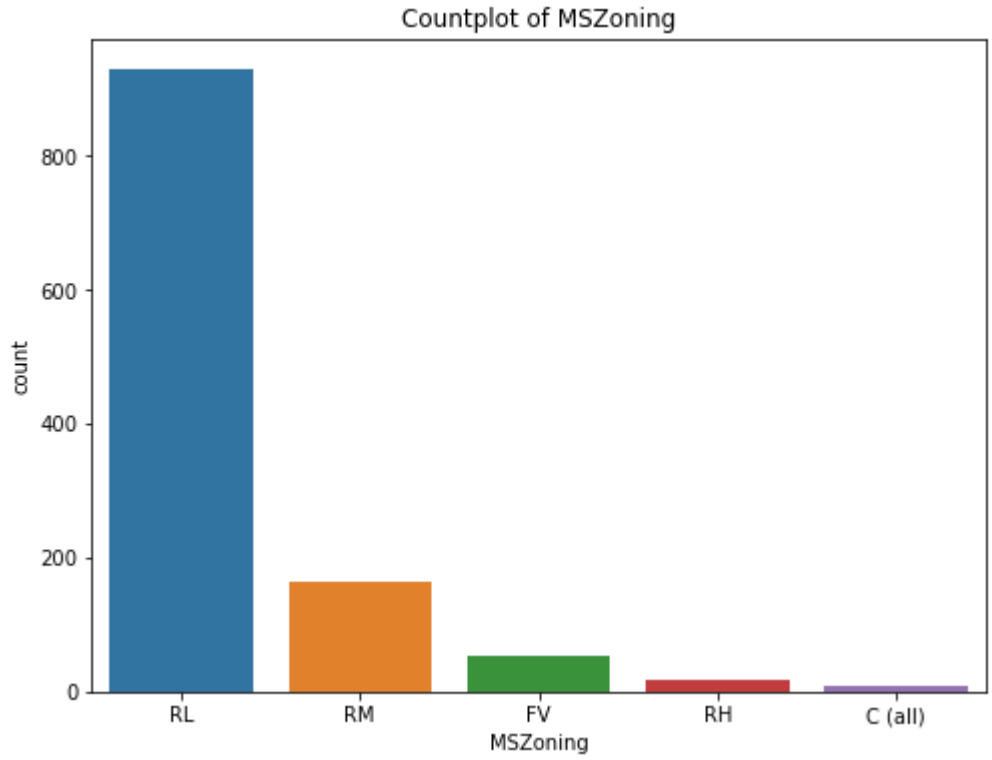
- Visualizations

1. Countplot of Target Variable (SalePrice) : we can observe that the maximum number of SalePrice lies between 140000 and 230000



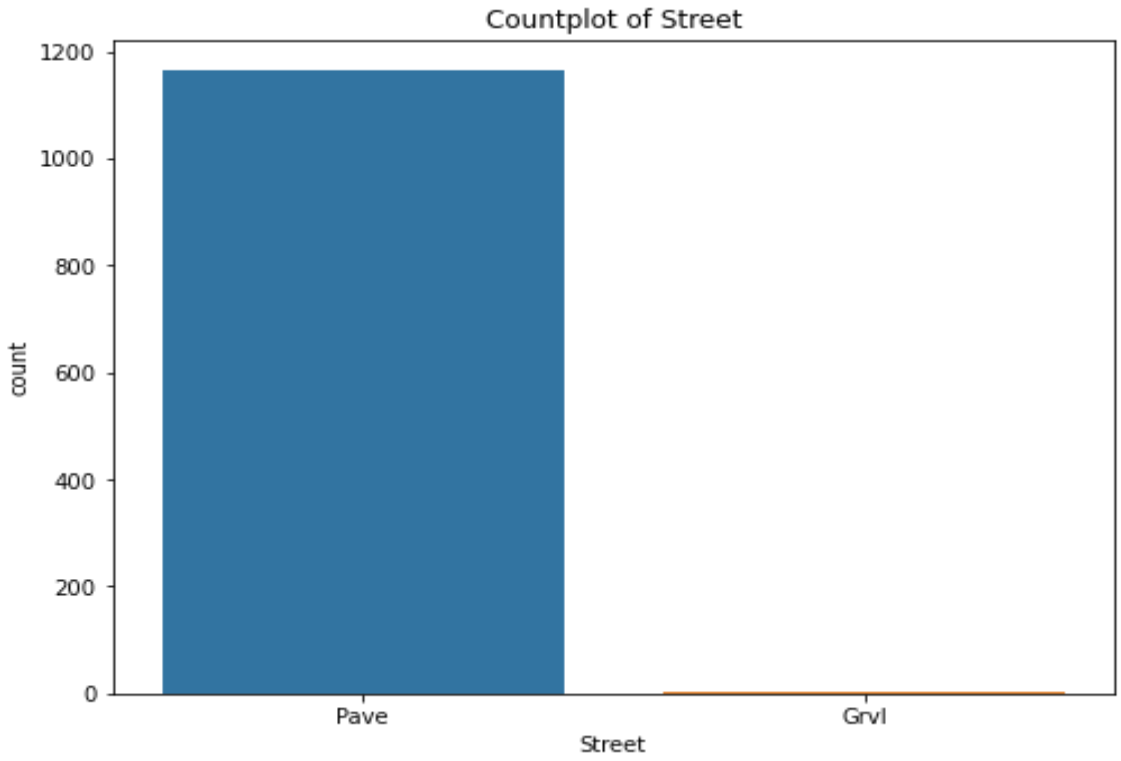
2. Countplot of column MSZoning: here, we observe that the maximum(928)number of MSZoning are RL.

```
RL          928
RM          163
FV           52
RH           16
C (all)       9
Name: MSZoning, dtype: int64
```



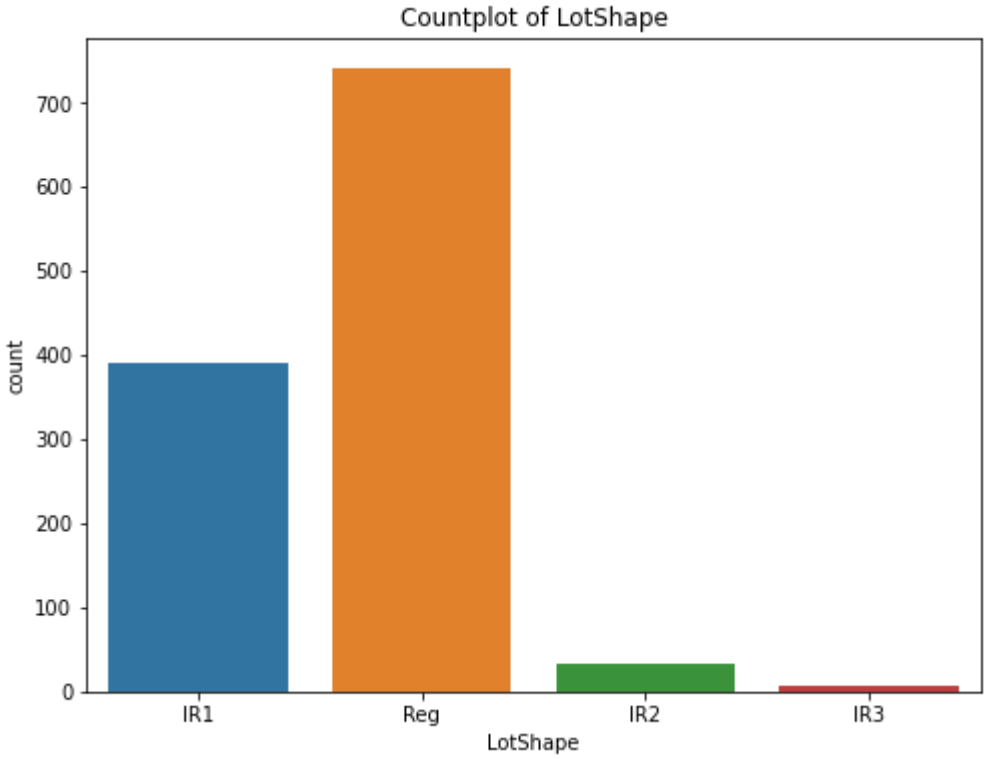
3. Countplot of Column Street: the maximum number of Street are Pave i.e 1164 where as only 4 are Grvl.

```
Pave    1164
Grvl       4
Name: Street, dtype: int64
```



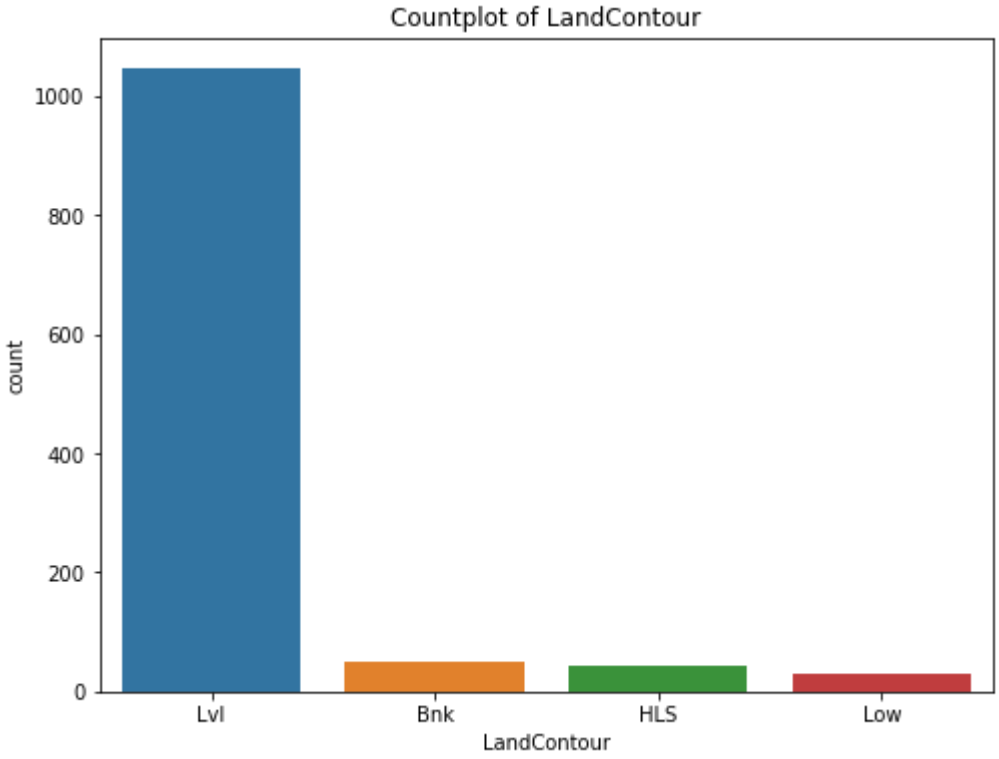
4. Countplot of column LotShape: we can say that the maximum number of LotShape are Reg i.e 740

```
Reg    740
IR1    390
IR2     32
IR3      6
Name: LotShape, dtype: int64
```



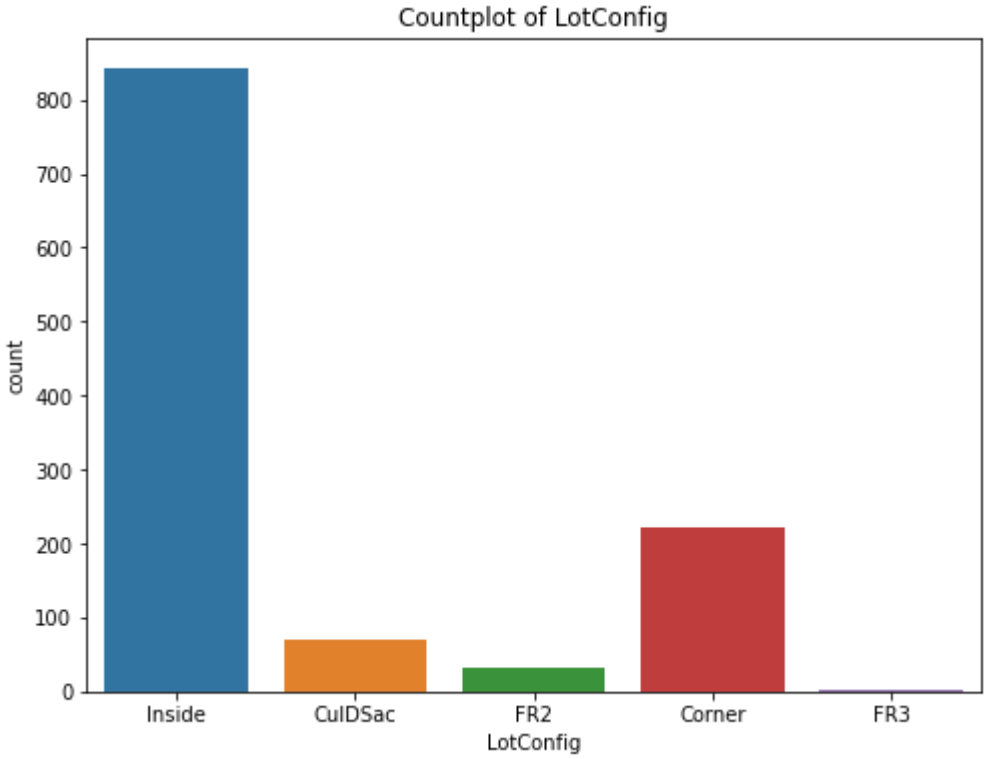
5. Checking the countplot of column LandContour: we can observe that the maximum number of LandContour are Lvl i.e 1046.

```
Lvl    1046
Bnk      50
HLS      42
Low      30
Name: LandContour, dtype: int64
```



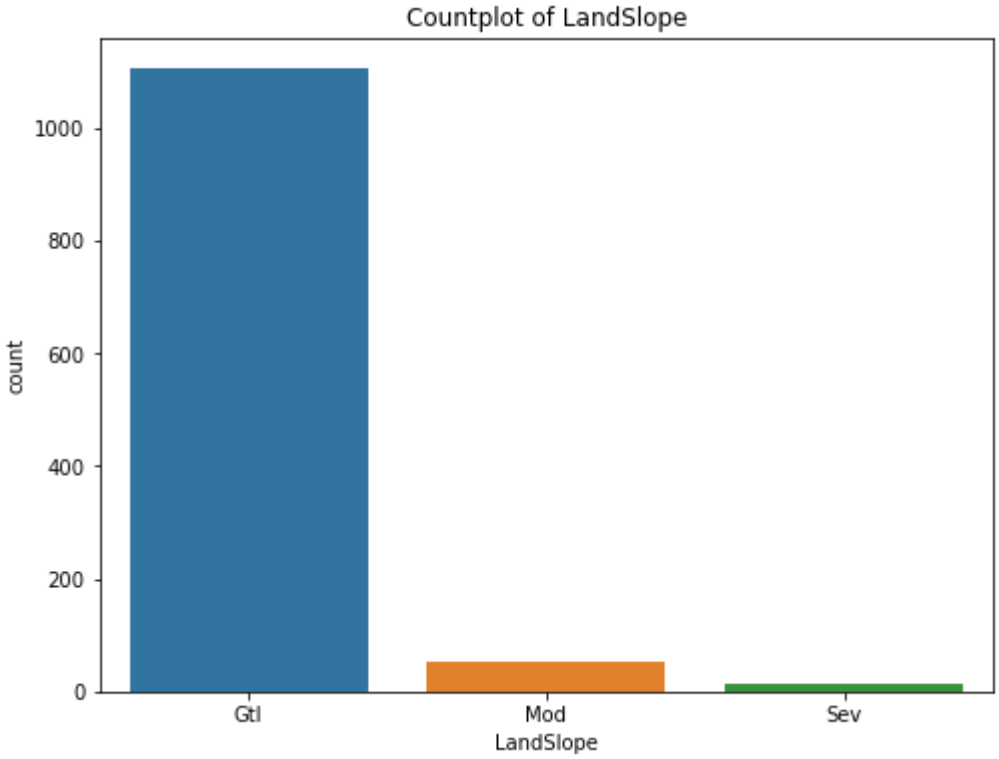
6. Countplot of column LotConfig : we can observe that the maximum number of L otConfig are Inside i.e 842.

```
Inside     842
Corner     222
CulDSac     69
FR2         33
FR3          2
Name: LotConfig, dtype: int64
```

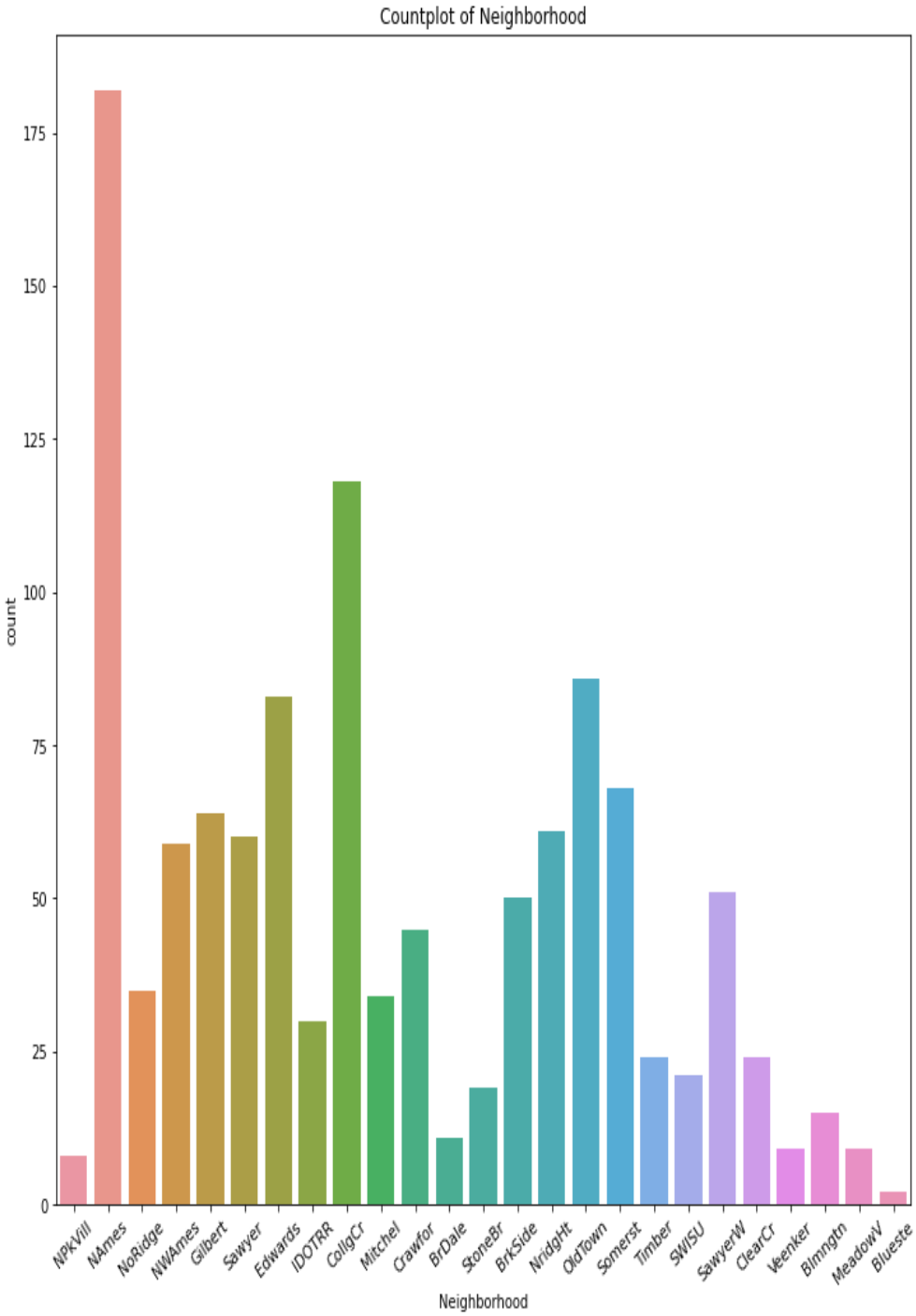


7. Countplot of column LandSlope: we can say that the maximum number of LandSl ope are Gtl 1105.

```
Gtl    1105
Mod      51
Sev      12
Name: LandSlope, dtype: int64
```

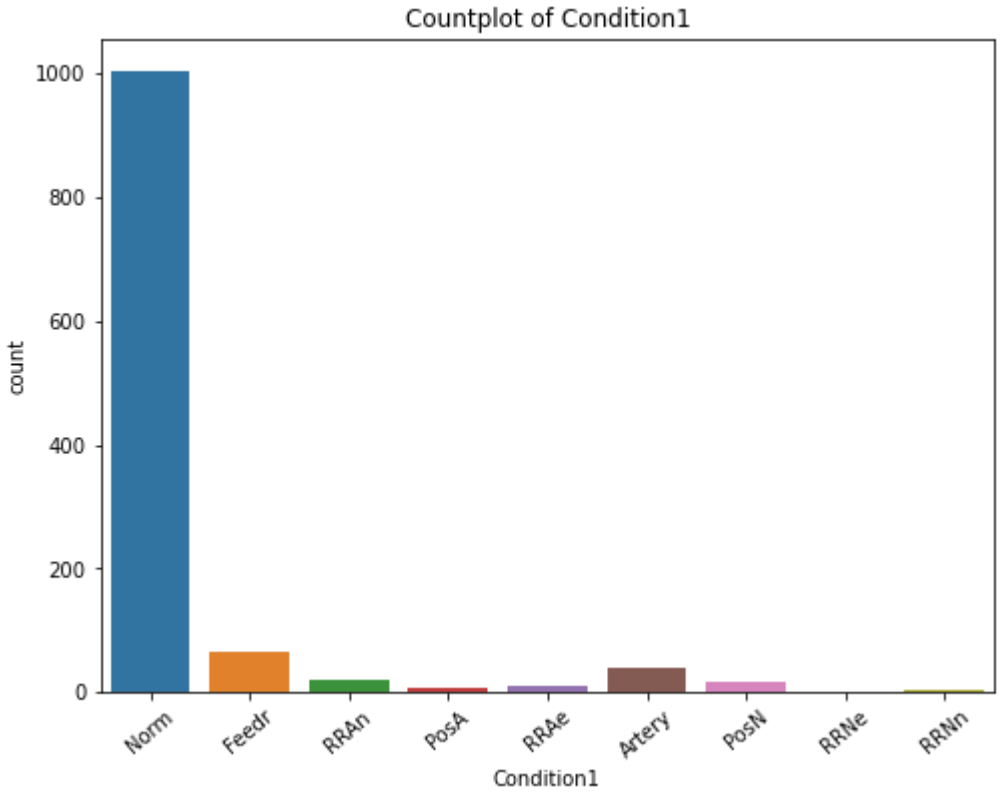


8. Countplot of column Neighbourhood: we can say that the maximum number of Neighborhood are Names i.e 182.

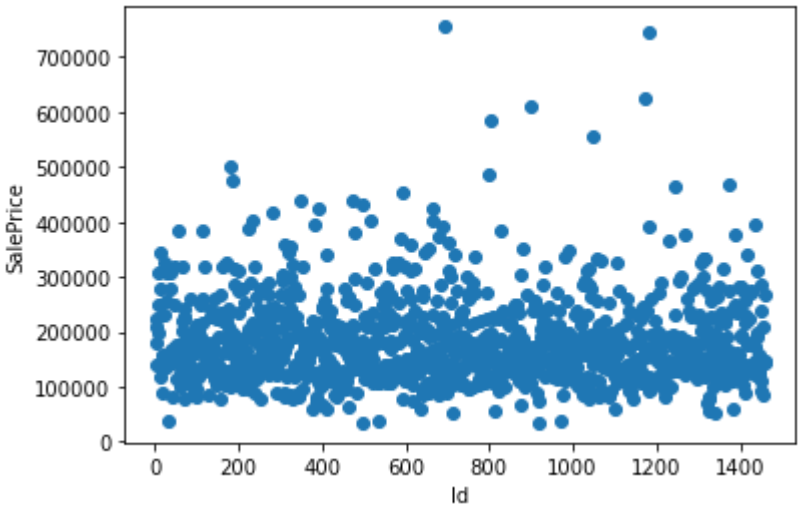


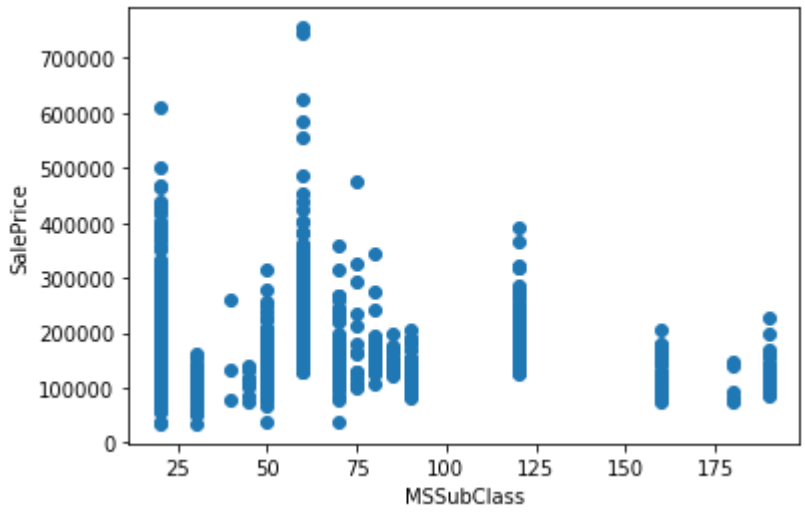
9. Countplot of column Condition1: we can say that the maximum number of Condition1 is Norm i.e 1005.

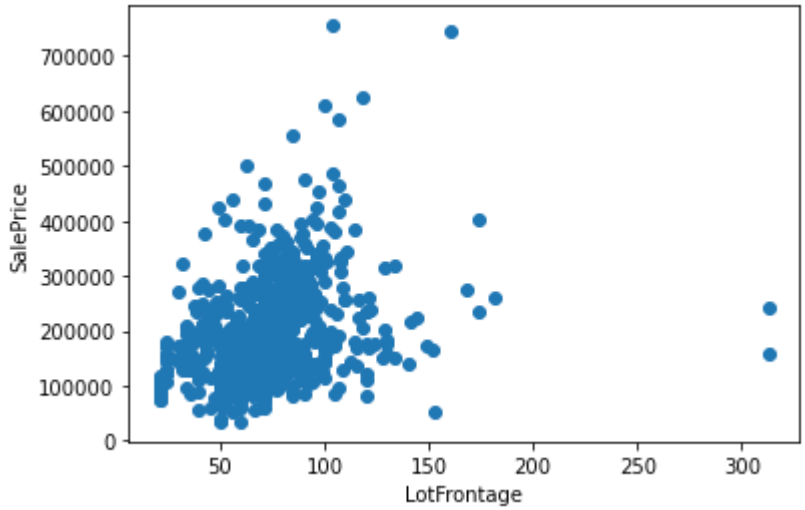
```
Norm      1005
Feedr       67
Artery      38
RRAn        20
PosN        17
RRAe         9
PosA         6
RRNn         4
RRNe         2
Name: Condition1, dtype: int64
```

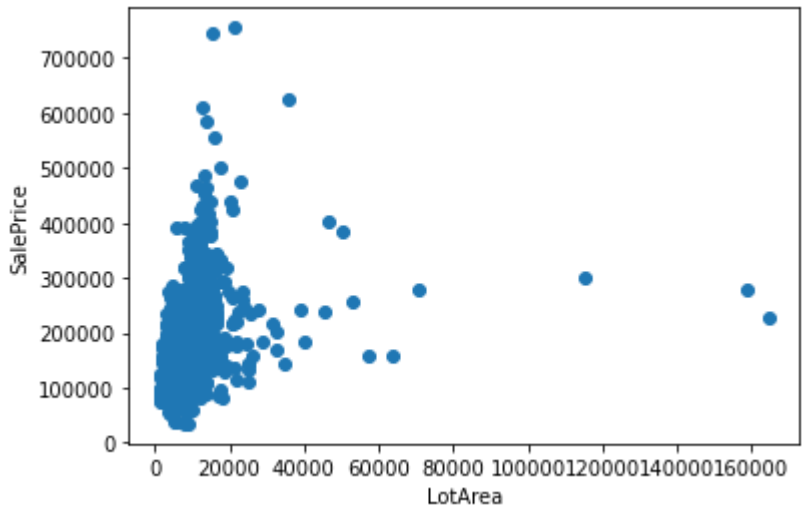


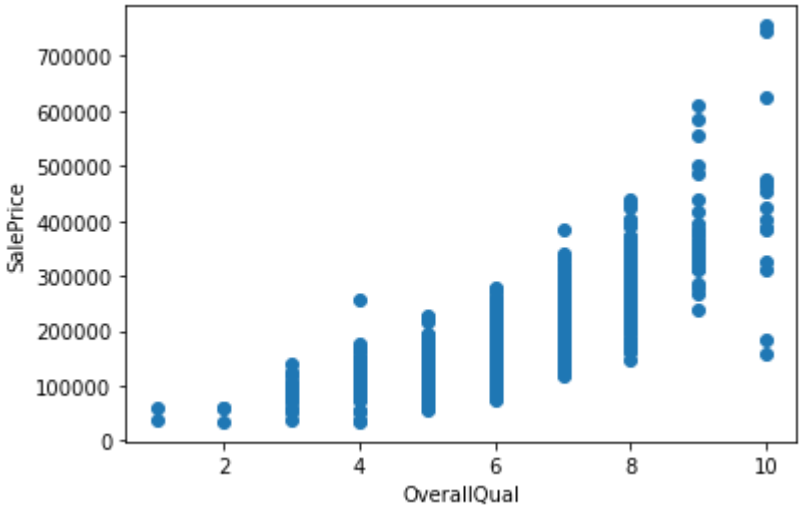
10. Scatterplot of all feature variables and target variable

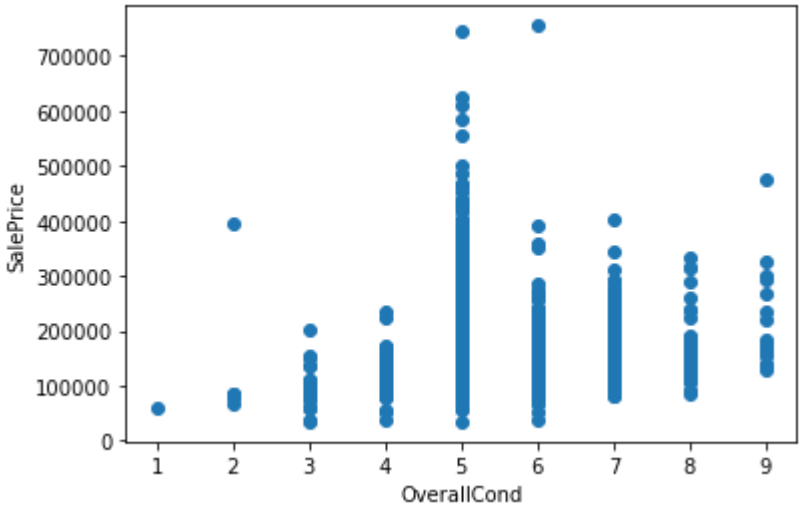


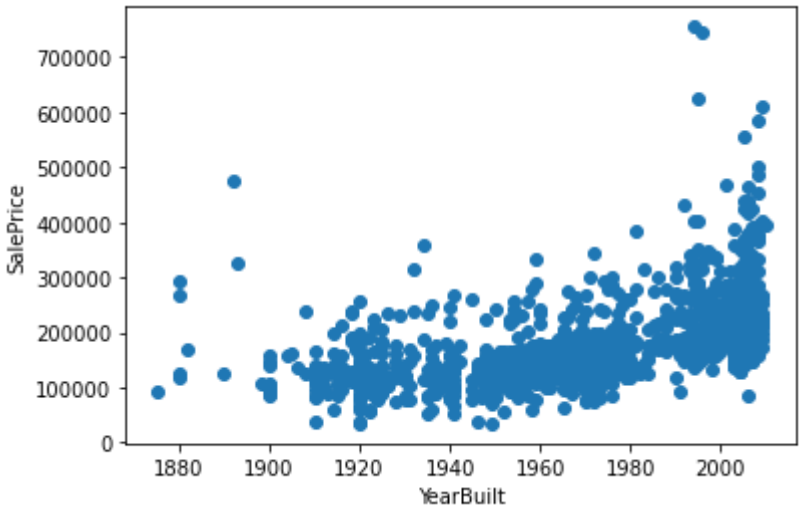

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
25
50
75
100
125
150
175
MSSubClass

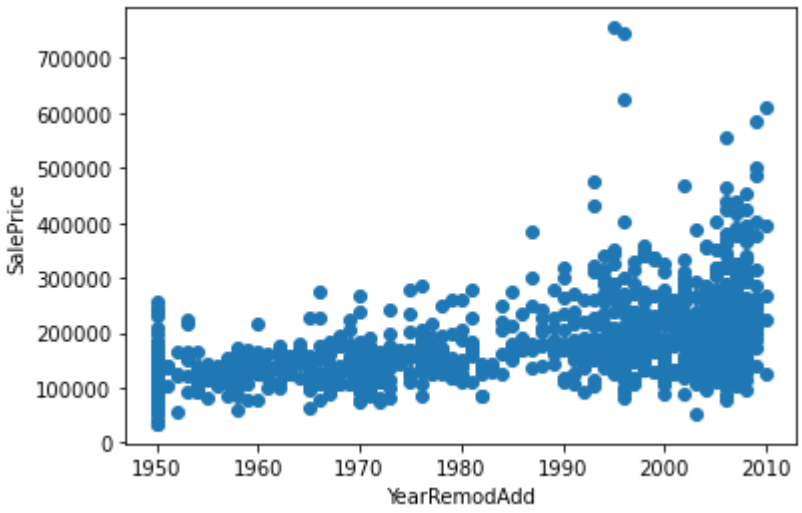

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
50
100
150
200
250
300
LotFrontage

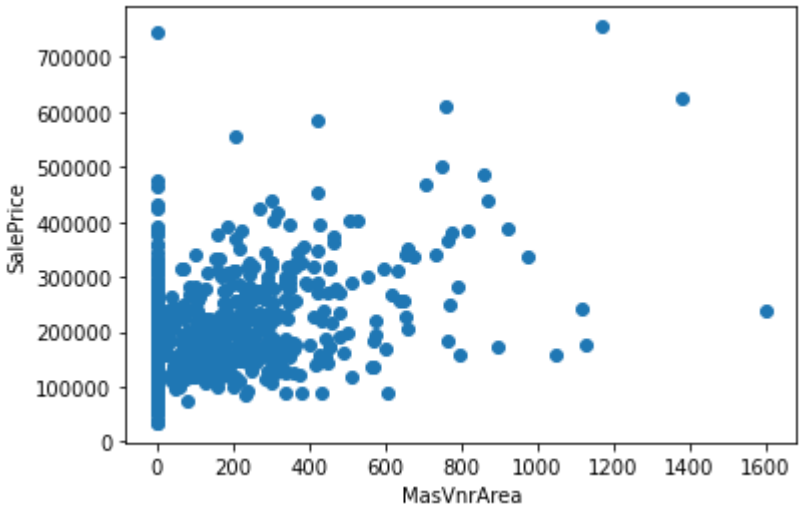

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
20000
40000
60000
80000
100000
120000
140000
160000
LotArea

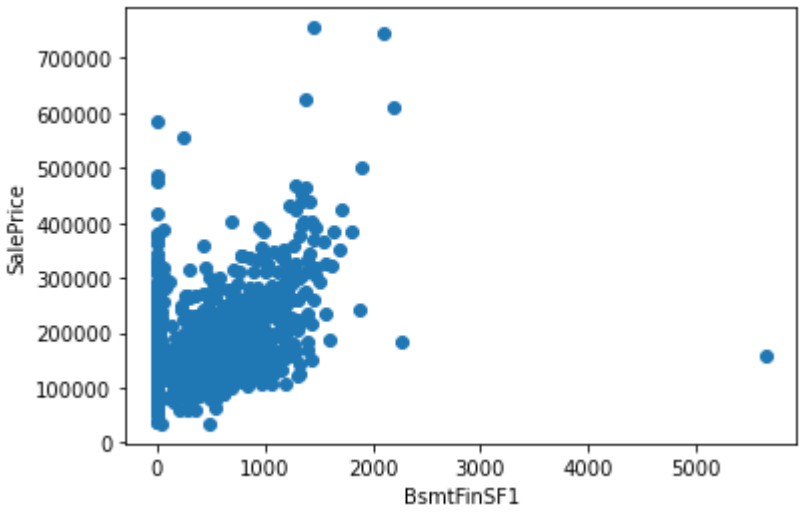

SalePrice
OverallQual
0
100000
200000
300000
400000
500000
600000
700000
2
4
6
8
10

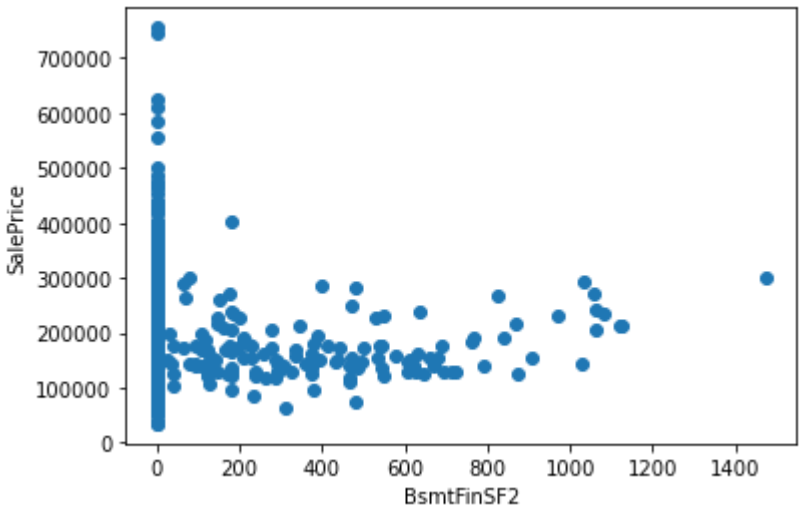

SalePrice
OverallCond
0
100000
200000
300000
400000
500000
600000
700000
1
2
3
4
5
6
7
8
9

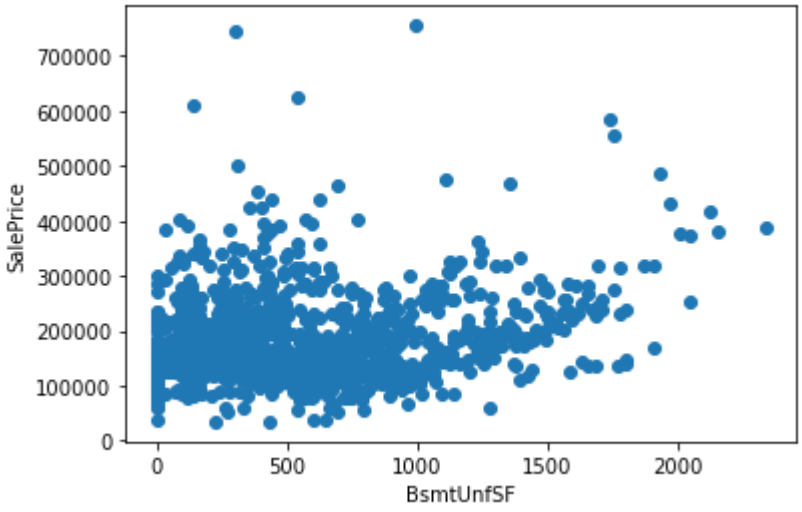

SalePrice
YearBuilt
0
100000
200000
300000
400000
500000
600000
700000
1880
1900
1920
1940
1960
1980
2000

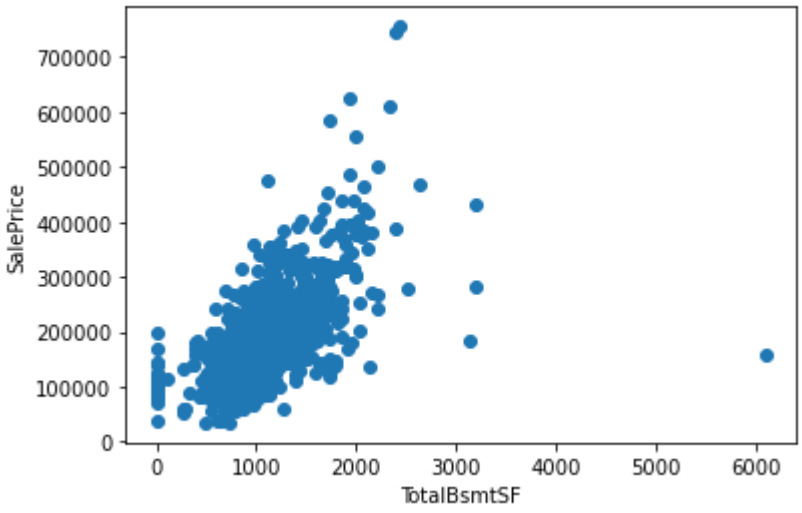

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1950
1960
1970
1980
1990
2000
2010
YearRemodAdd

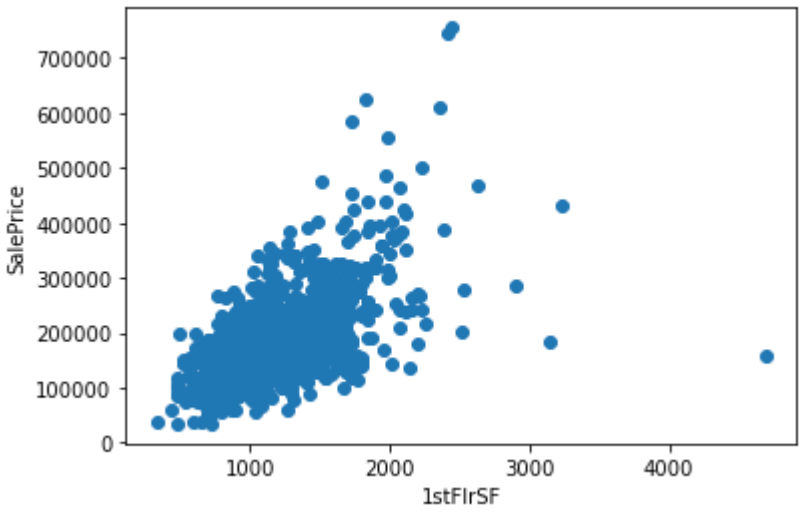

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
200
400
600
800
1000
1200
1400
1600
MasVnrArea

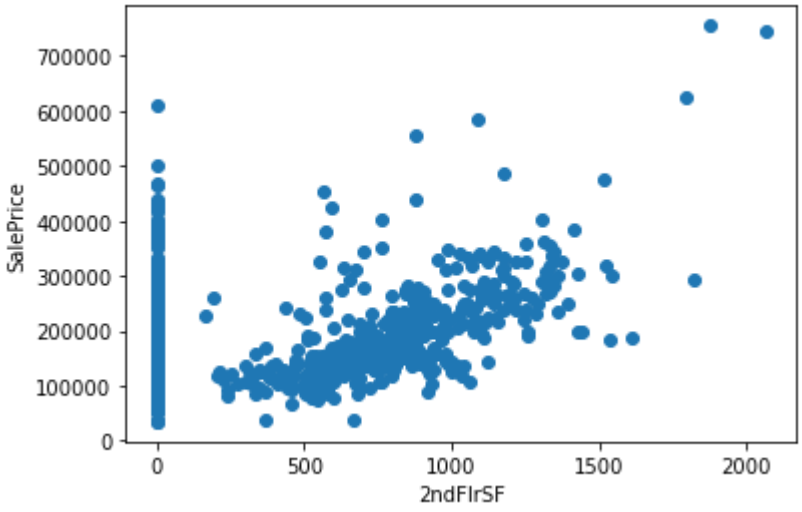

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
BsmtFinSF1

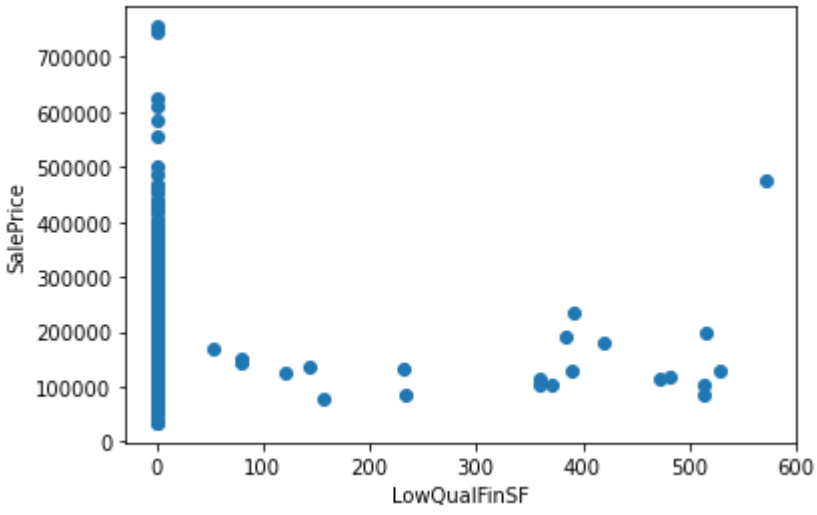

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
200
400
600
800
1000
1200
1400
BsmtFinSF2

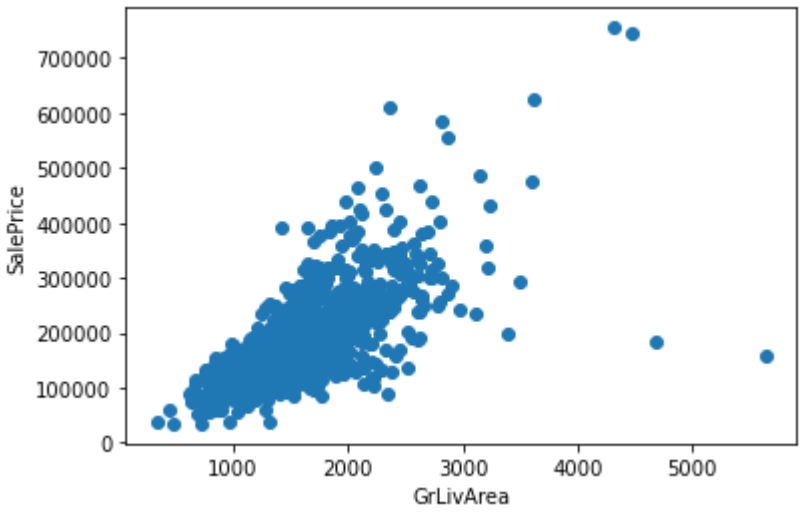

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
500
1000
1500
2000
BsmtUnfSF

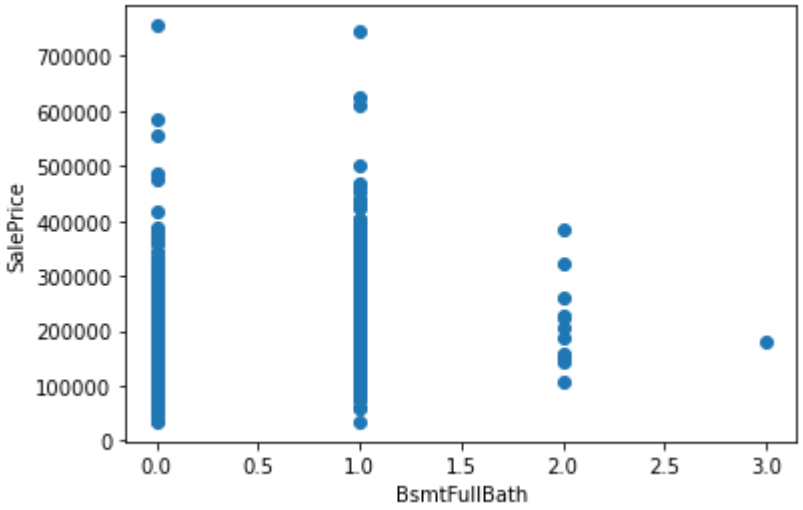

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000
TotalBsmtSF

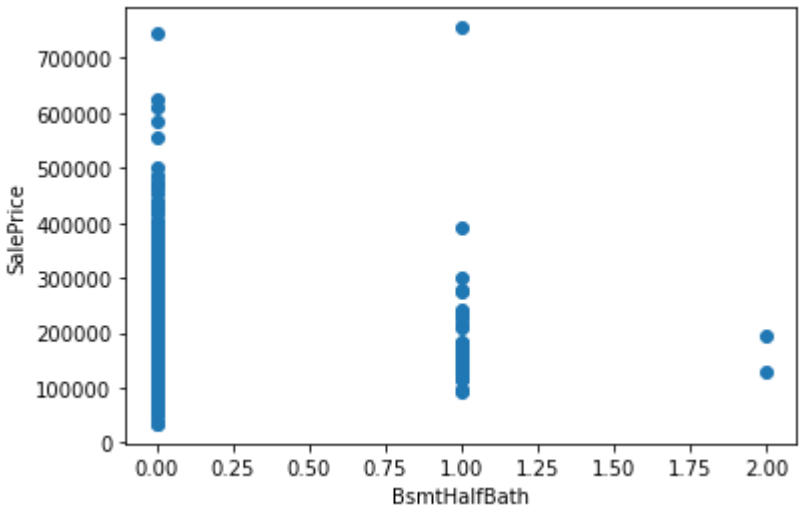

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
1000
2000
3000
4000
1stFlrSF

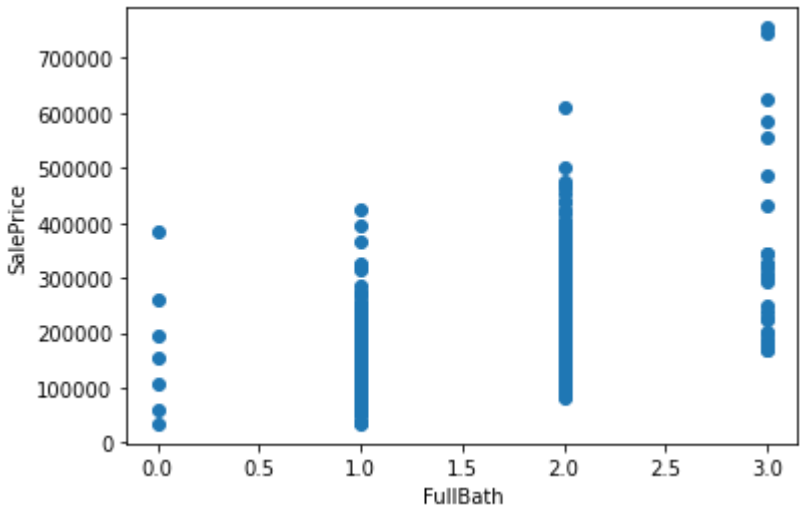

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
0
500
1000
1500
2000
2ndFlrSF

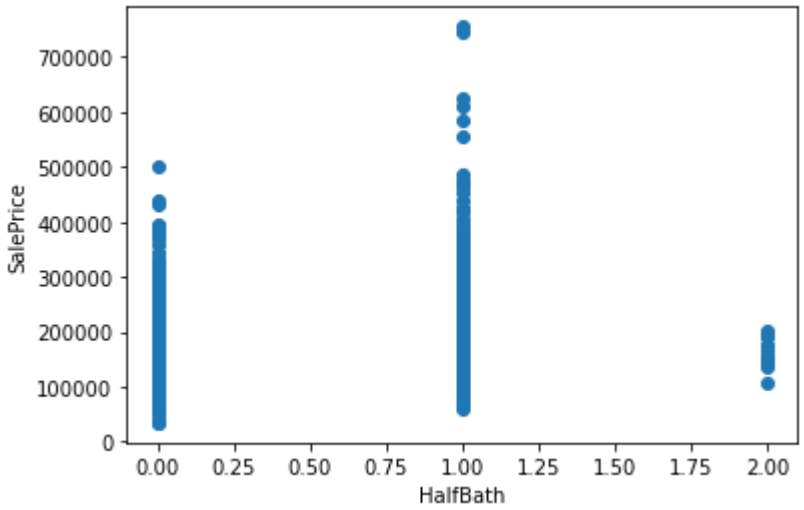

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
0
100
200
300
400
500
600
LowQualFinSF

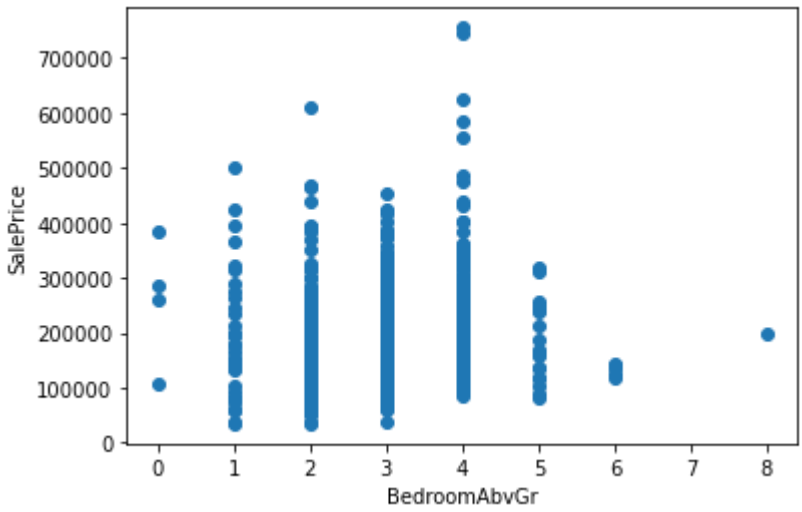

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1000
2000
3000
4000
5000
GrLivArea

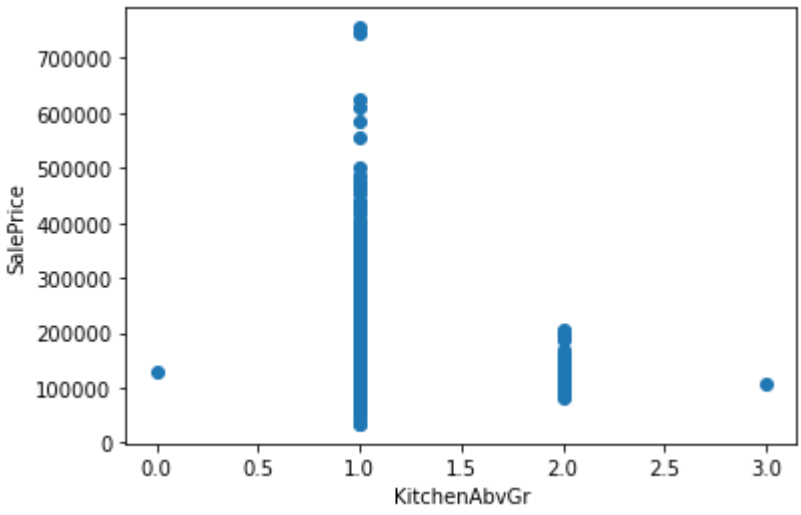

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
BsmtFullBath

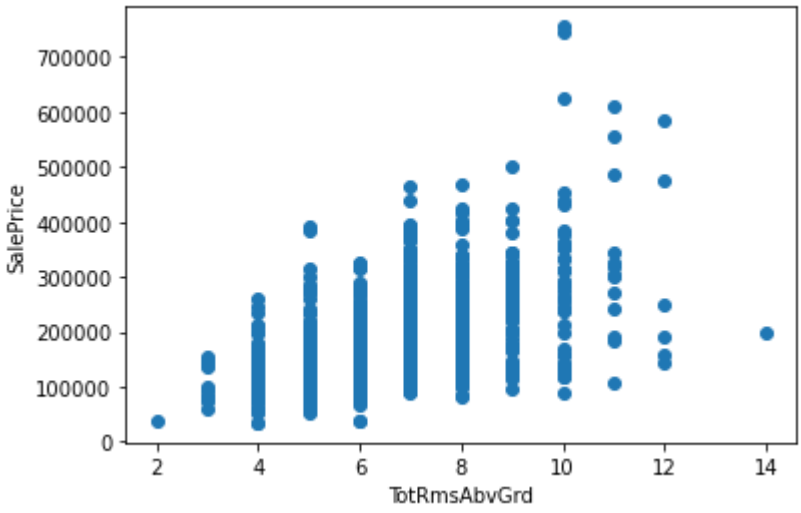

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
BsmtHalfBath

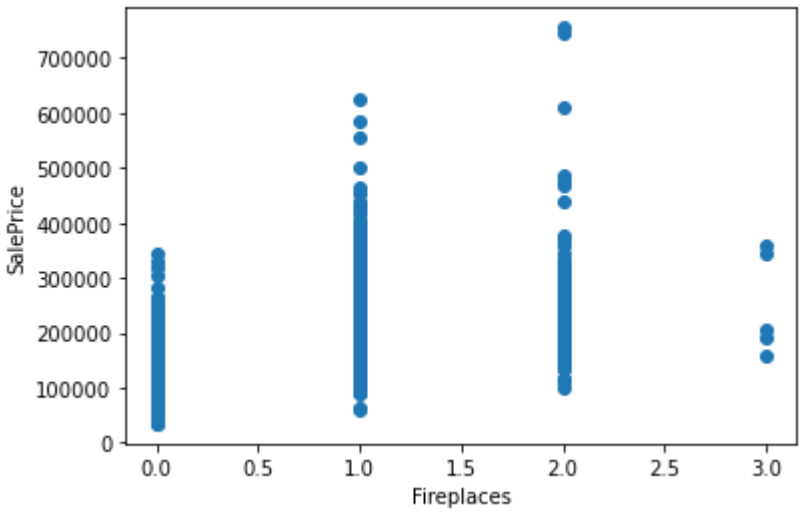

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
FullBath

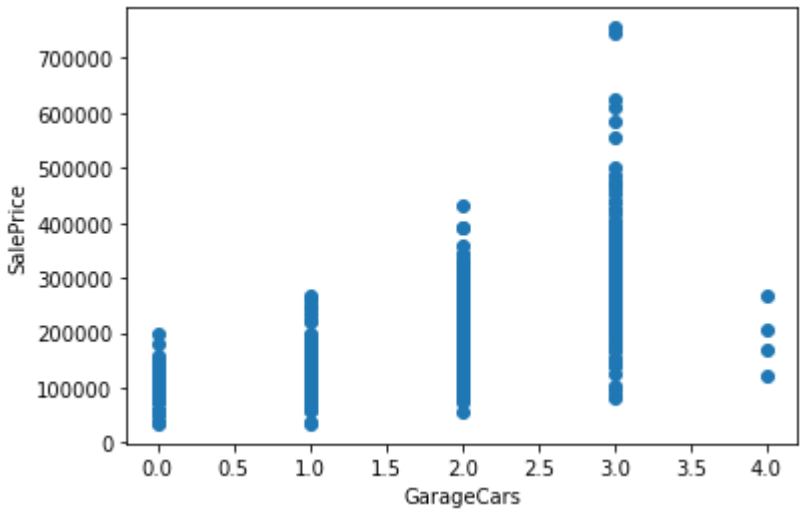

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
HalfBath

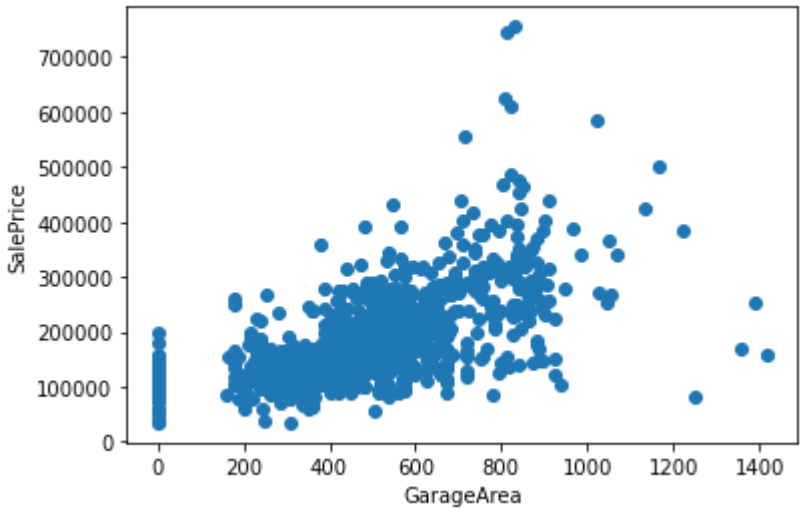

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
5
6
7
8
BedroomAbvGr

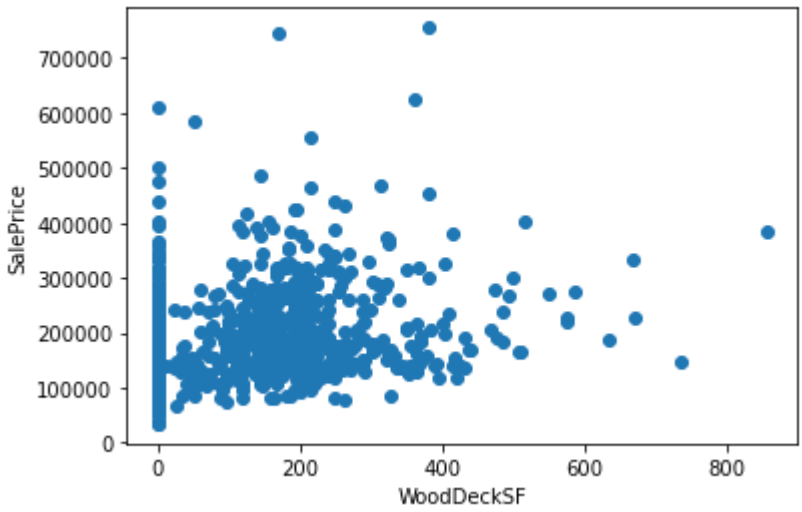

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr

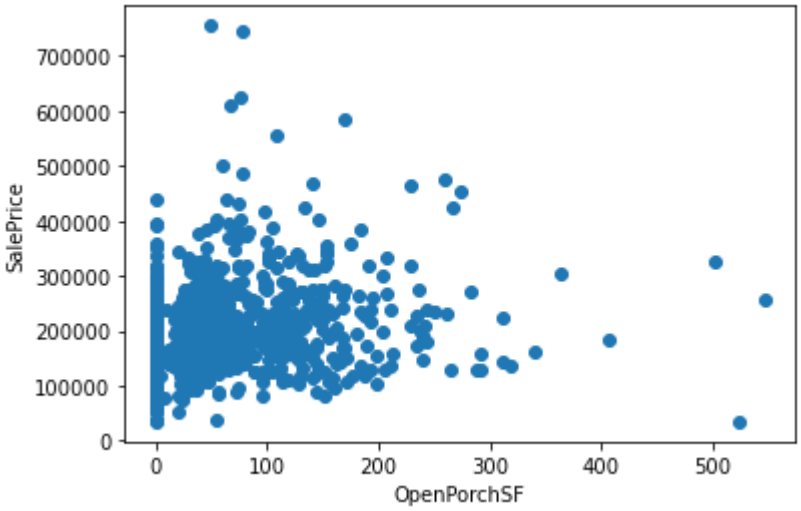

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
2
4
6
8
10
12
14
TotRmsAbvGrd

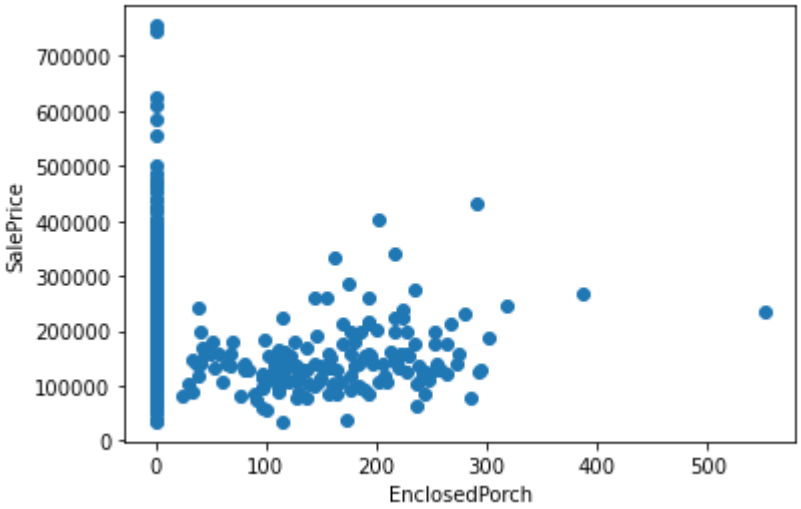

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Fireplaces

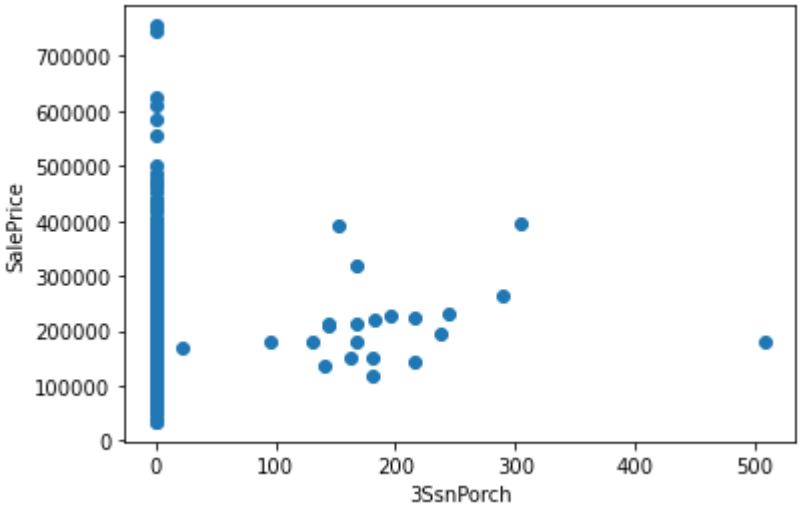

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
GarageCars

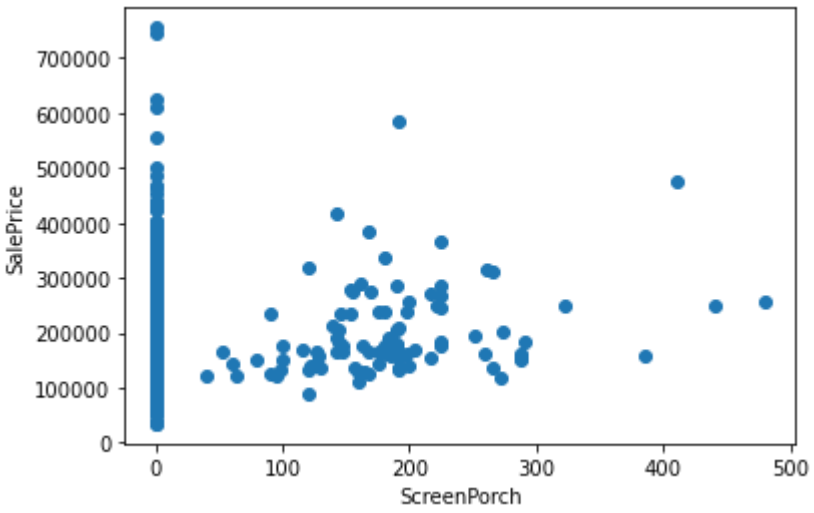

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
200
400
600
800
1000
1200
1400
GarageArea

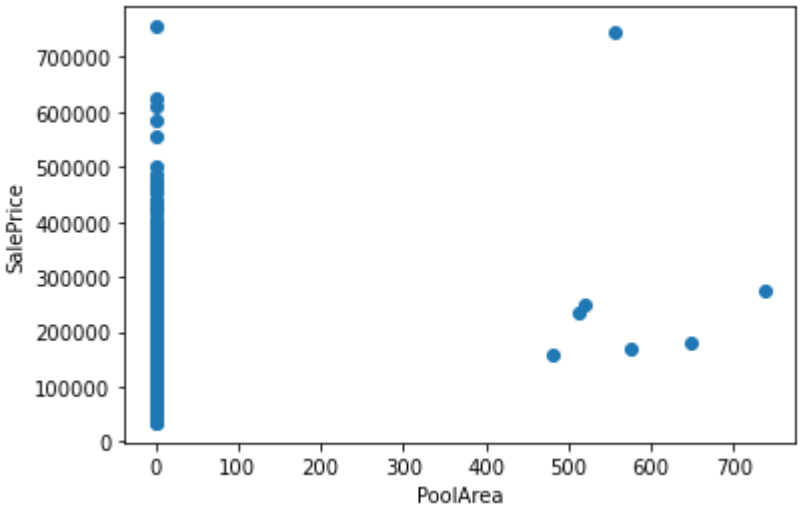

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
200
400
600
800
WoodDeckSF

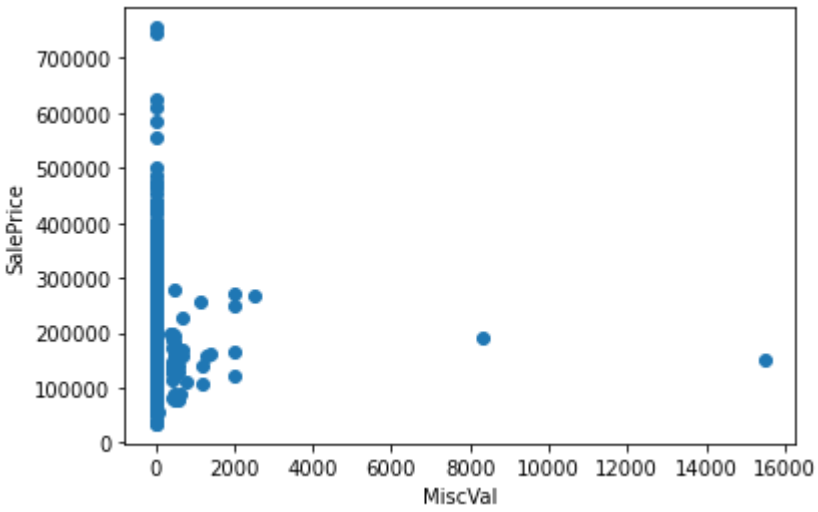

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
OpenPorchSF

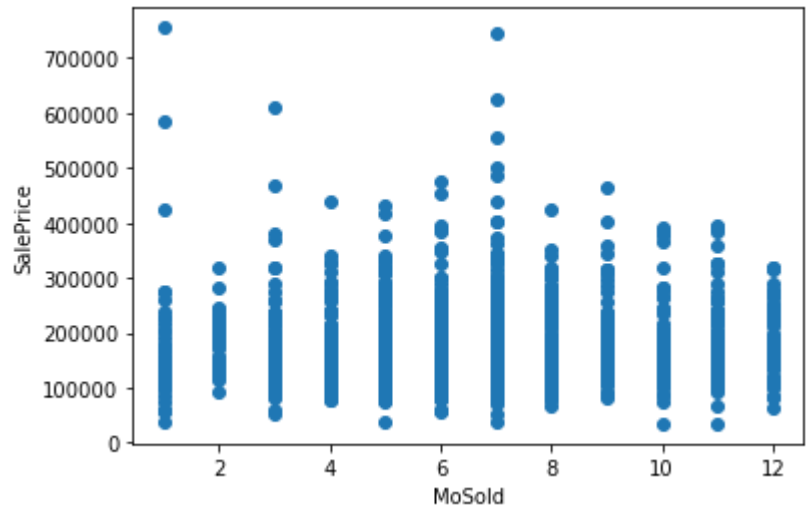

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
EnclosedPorch

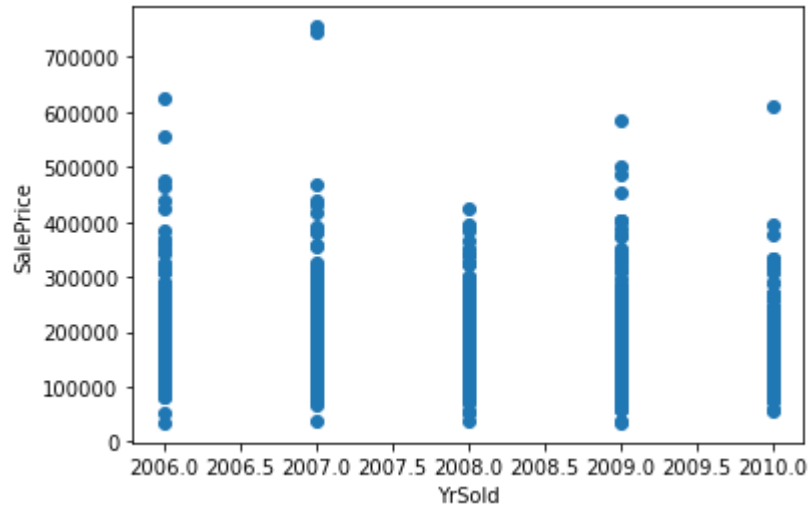

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
3SsnPorch

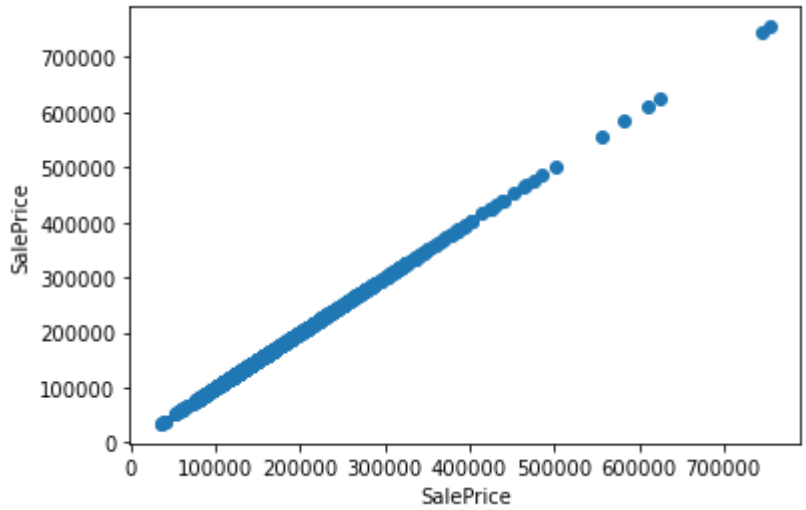

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
ScreenPorch


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


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

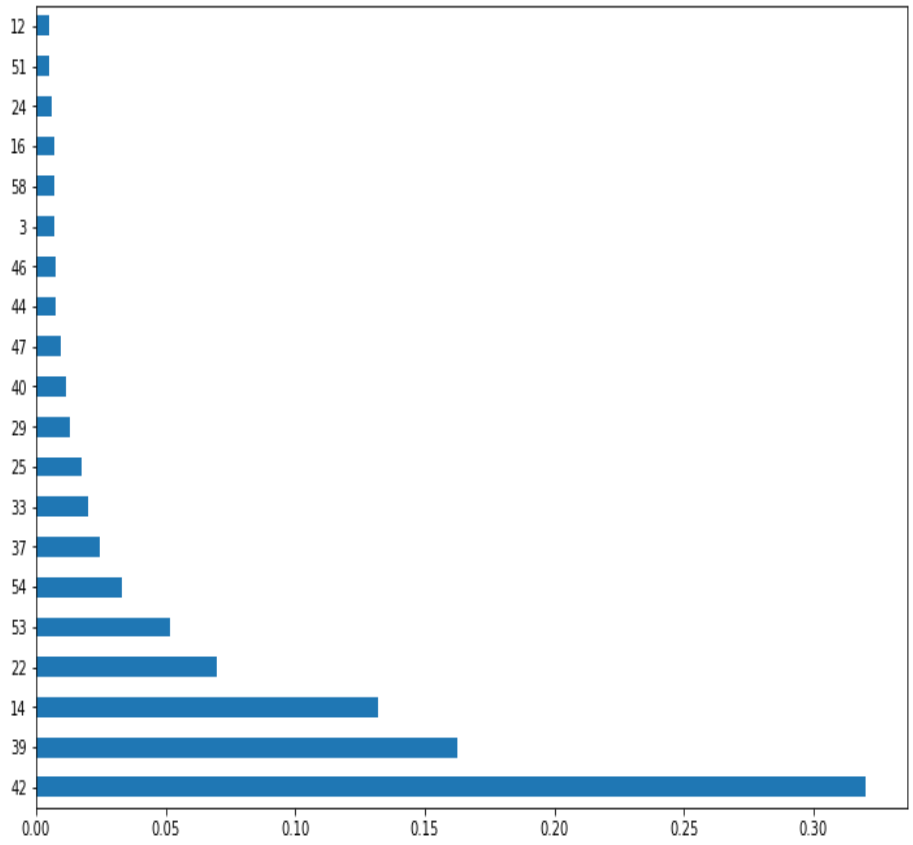






Here, from the above plots, we can observe that SalePrice is maximum with FV MSZoning.

11. Plotting the graph for Feature Importance using Extra Trees Regressor



INTERPRETATION

- The purpose of this article was twofold: to understand the pattern of Australian real estate market and make predictive model, which is able to effectively predict the price of houses in Australia.
- We use many algorithms to find best model and best result were observed of the catboost regressor with 84% r2 score accuracy.
- There are many variables important to predict the price of houses. Like quality of houses, exterior quality, basement area, kitchen quality, total rooms above grade and many more.
- In order to increase profit of surprise housing company, the company should start using of machine learning model.
- By using machine learning model company can decide whether to invest in properties or not.

CONCLUSION

- ## Key Findings and Conclusions of the Study

MS Subclass seems to have the biggest impact on the House Prices followed by Basement Full Bath and Basement Half bath

Other than the Basement related features, Condition2 , Exterior Quality and Lot Area are some of the other important features.

- ## Learning Outcomes of the Study in respect of Data Science

The above study helps one to understand the business of real estate i.e how the price changes across the properties depending upon the various amenities like swimming pool, garage, pavement, lawn size and type of the building raise. All these features do affect the cost. With the help of this analysis, we can sketch the needs of the property buyer and accordingly can predict the price of the property.

- ## Limitations of this work and Scope for Future Work

The main issue in this study, was there were many null values in our dataset which we need to fill in the correct manner. We can still improve our model accuracy by doing some extensive hyper parameter tuning.