PROJECT REPORT ON

“Housing Project”

SUBMITTED BY

Adi Rajit Mahesh

ACKNOWLEDGMENT

I would like to thank FlipRobo Technologies who have given me this opportunity to work on this amazing project. I would also like to thank “Srishti Mann” ma’am who helped me out during my project.

Contents

1. INTRODUCTION

1.1 Business Problem Framing

This is a real estate issue where a US based real estate company called Surprise Housing decided to invest in the Australian market. Their plan is to buy homes in Australia at prices below their actual market value and sell them at high prices for a profit. To do this, this company uses data analysis to decide which property to invest in.

The company has collected data from homes already sold in Australia and using this data they want to know the value of potential properties to decide whether it will be appropriate to invest in the properties or not.

To know the value of properties, the company provided us with data to perform data analysis and extract information on important attributes to predict house prices. They want a machine learning model that can predict house prices as well as the meaning of each important attribute in house prediction - how and with what intensity each variable affects house prices.

1.2 Conceptual Background of the Domain Problem

In real estate the value of property usually increases with time as seen in many countries. One of the causes for this is due to rising population.

The value of property also depends on the proximity of the property, its size its neighborhood and audience for which the property is subjected to be sold. For example, if audience is mainly concerned of commercial purpose. Then the property which is located in densely populated area will be sold very fast and at high prices compared to the one located at remote place. Similarly, if audience is concerned only on living place, then property with less dense area having large area with all services will be sold at higher prices.

The company is looking at prospective properties to buy houses to enter the market. We 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.

1.3 Review of Literature

Houses are one of the necessary needs of each 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.

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.

We 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.

1.4 Motivation for the Problem Undertaken

Houses are a basic necessity for every human being all around the globe and it is one of the largest markets that contributes to the world's economy. This project motivates me because it helps us predict the price of the use looking at the features the house provides us and the prices of properties that have already been sold. We can get to know if the real estate agent is being fraudulent or not by looking at the prices that our model has predicted v/s what the agent has told us.

2. ANALYTICAL PROBLEM FRAMING

2.1 Mathematical/ Analytical Modelling of the Problem

This particular problem contains two datasets train dataset and test dataset. Model are built using train dataset. This model is then used to predict the Sale Price for test dataset. By analyzing into the target column. After analysis it was concluded that the data entries of sale Price column contains data points of continuous nature, it is a Regression problem, and hence all regression algorithms were used while building the model. While checking for the null values in the datasets, many columns with Nan values were found and null values were replaced with suitable entries like mean for numerical columns and specific value for categorical columns. For further analyses graph plot like distribution plot, bar plot, and scatter plot were used. With these plots, the relation between the feature columns and target column was visualized. Upon analyzing outliers and skewness were found in the dataset and were removed. Outliers were removed using percentile method and Skewness using Yeo-Johnson method. All the regression models were iterated to find the best model and then further Hyper-tune the best model and save the best model. Finally, Sale Price was predicted for test dataset using the saved model built from train dataset.

2.2 Data Sources and their Formats

There were 2 data sets that were provided by Fliprobo Technologies, one was the train dataset and the other was the test dataset. The train dataset contained 1168 rows and 81 columns. The test dataset contained 292 rows and 80 columns. The dataset was in a CSV format.

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1163	289	20	RL	NaN	9819	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1164	554	20	RL	67.0	8777	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1165	196	160	RL	24.0	2280	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1166	31	70	C (all)	50.0	8500	Pave	Pave	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1167	617	60	RL	NaN	7861	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0

1168 rows × 81 columns

2.3 Data Pre-processing

The dataset I received was raw data which needed to be processed. After looking at the data I dropped a few columns which I felt were unnecessary. In ID column the unique counts were 1168, which concludes that all the data entries in ID column is unique. ID is the unique identity number given to every asset. Utilities column contained only one data point in whole dataset, hence I dropped both the columns.

I then visualized the relationship between different features and plotted different kinds of graphs. I then checked the correlation using .corr() and checked for multicolinearity problem which did not exist.

2.4 Data Inputs-Logic-Output Relationships

Sale Price is our target variable i.e., output column. Rest all columns are to be used as feature column i.e., input column. We need to find which feature columns have positive correlation and which have negative correlation, accordingly to train our model.

The dataset is not clean, i.e., consists of missing values as well.

2.5 Hardware and Software Requirements and Tools Used

Hardware required:

- Processor — i5 and above
- RAM — 8 GB or above
- SSD — 250GB or above

Software/s required: Anaconda

LIBRARIES:

The tools, libraries, and packages we used for accomplishing this project are pandas, numpy, matplotlib, seaborn, scipy, sklearn, xgboost, pickle.

Through pandas library we loaded our csv file ‘Data file’ into dataframe and performed data manipulation and analysis. With the help of numpy we worked with arrays. With the help of matplotlib and seaborn we did plotted various graphs.

Train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

With Sklearn’s standardscaler package we scaled all the feature variables onto single scale. As these columns are different in scale, they are standardized to have common scale while building machine learning model. This is useful when you want to compare data that correspond to different units.

With Sklearn’s package we imported many regression models, we could obtain cross_val_score which is an accuracy metric used to evaluate model, we could obtain best parameters of a model using GridSearchCV, we could reduce skewness using power transform library of sklearn.

3. DATA ANALYSIS AND VISUALIZATION

3.1 Identification of possible problem-solving approaches

Null values of numerical columns can be filled by replacing null values with mean of respective column. Null values can of categorical column can be either replaced by using mode value or if it's for a column having ordinal datapoint, we can perform ordinal encoding.

If outliers are present, we shall remove them using Z-Score method. If skewness exists, we shall remove them using Yeo-Johnson method.

If models have low accuracy, we shall fine tune them to improve accuracy but if accuracy is still low then we shall stack up our top performing models to boost accuracy by combining models.

3.2 Testing of Identified Approaches (Algorithms)

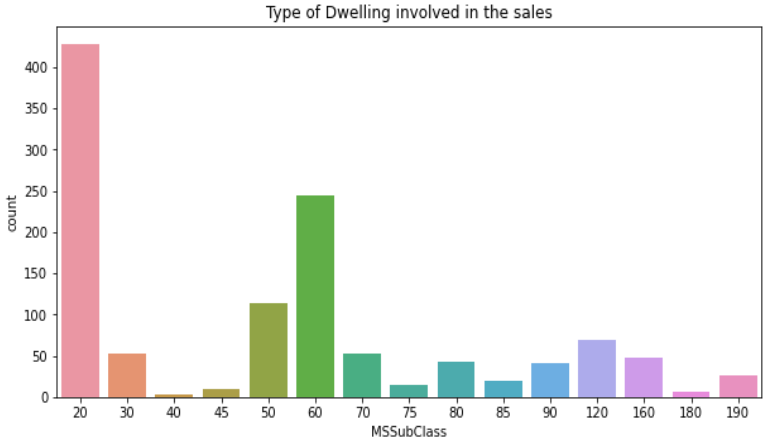
We can check null values using info function. Outliers can be detected using Boxplot. Skewness can be detected using skew function. Our target variable is SalePrice which has datapoints of continous in nature, hence it is a regression problem. For that we shall use all regression algorithms to find & build the optimized model. By looking into the difference of r2 score and cross validation score of each model we can find our best model with least difference. To get the best model we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation.

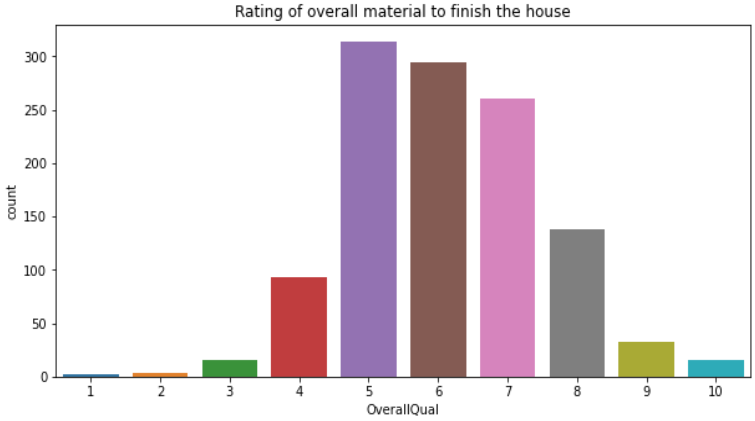
3.3 Key Metrics for success in solving problem under consideration

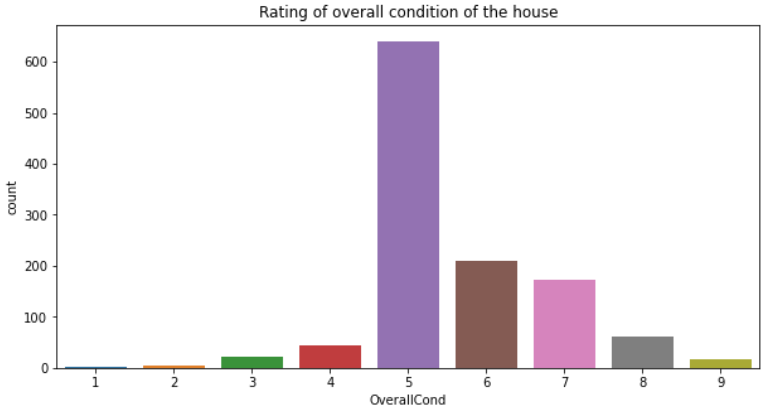
Following metrics were used to evaluate our model:

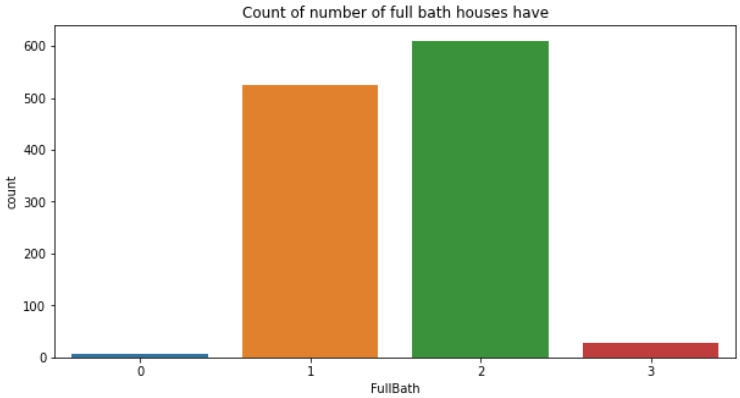
- Cross Val Score
- R2 Score
- Mean absolute error
- Mean squared error
- Standard deviation error

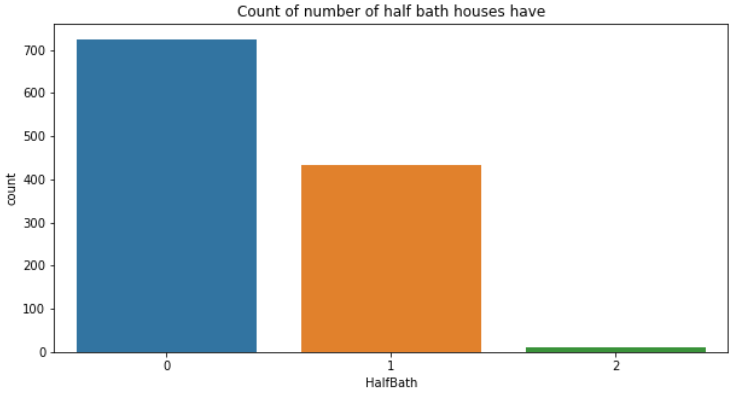
3.4 Visualization

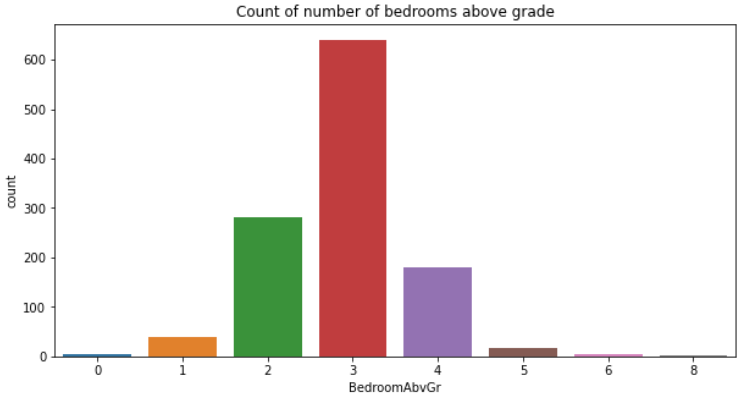


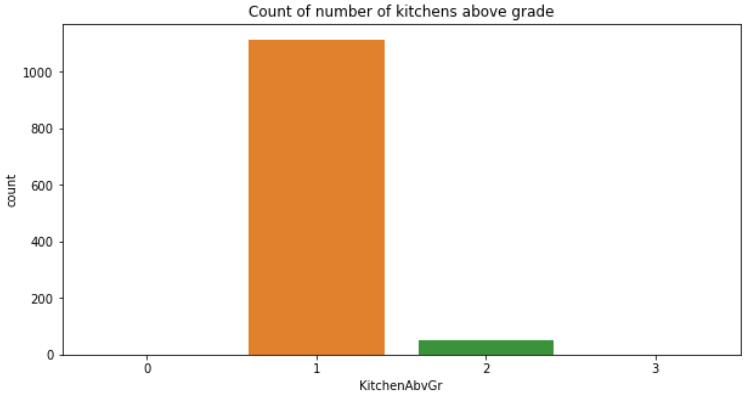


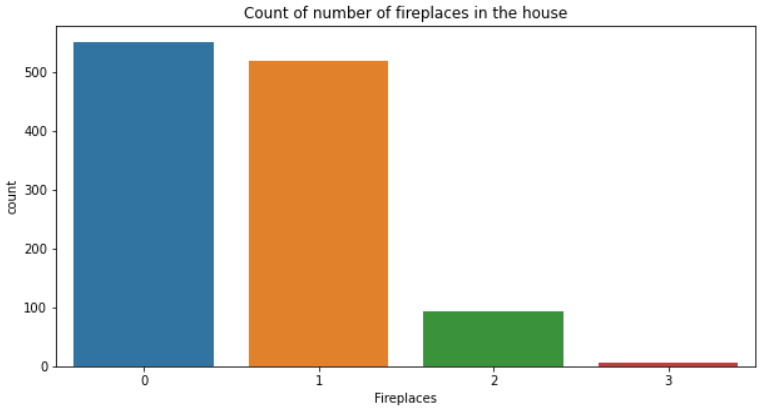


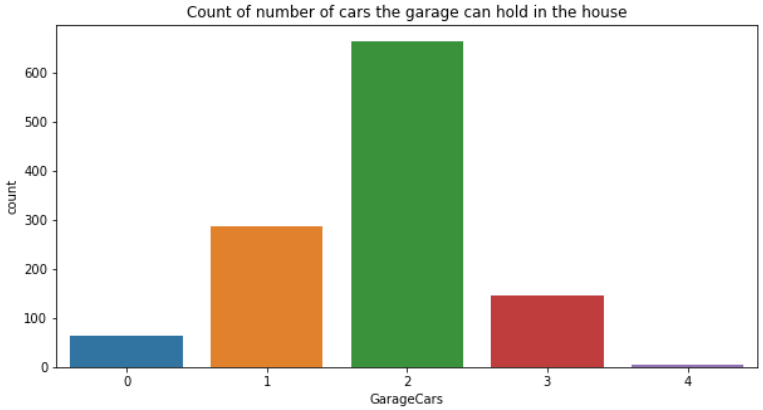

Count of number of full bath houses have
count
FullBath
0
1
2
3
0
100
200
300
400
500
600

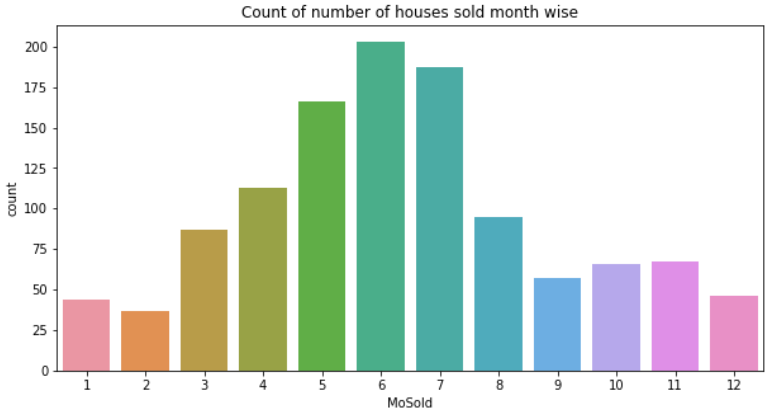

Count of number of half bath houses have
count
HalfBath
0
1
2
0
100
200
300
400
500
600
700

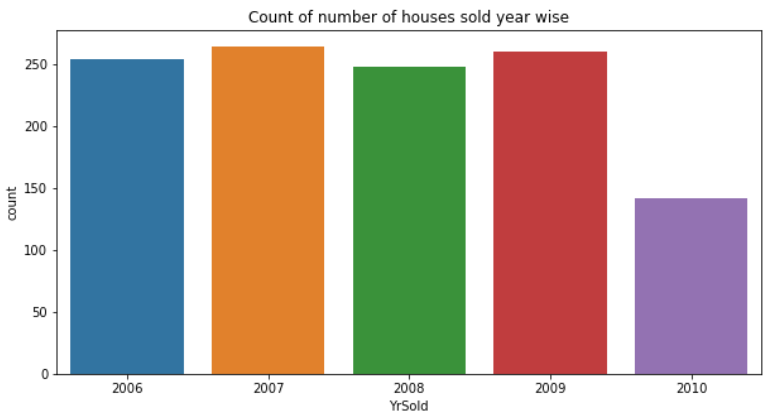

Count of number of bedrooms above grade
count
BedroomAbvGr
0
1
2
3
4
5
6
8
0
100
200
300
400
500
600

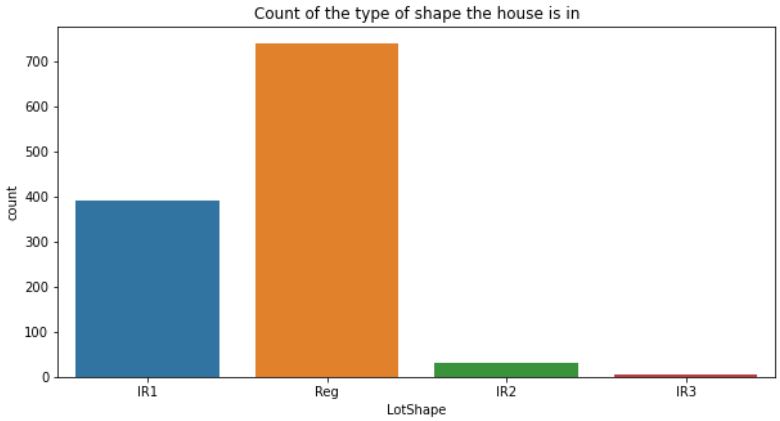

Count of number of kitchens above grade
count
KitchenAbvGr
0
1
2
3
0
200
400
600
800
1000

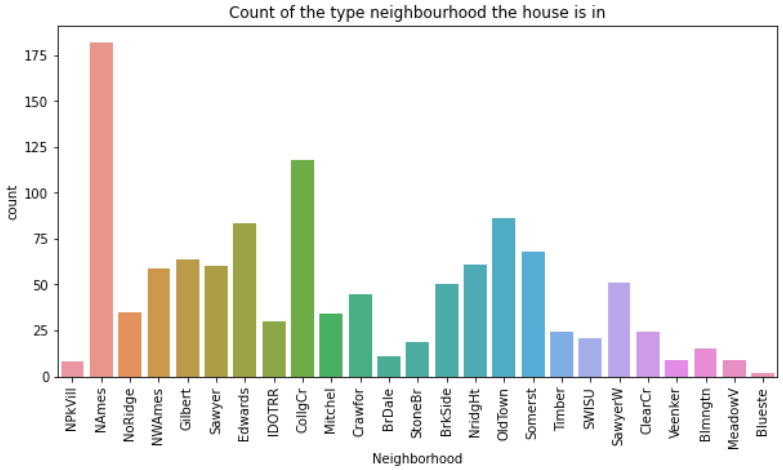

Count of number of fireplaces in the house
count
Fireplaces
0
1
2
3
0
100
200
300
400
500

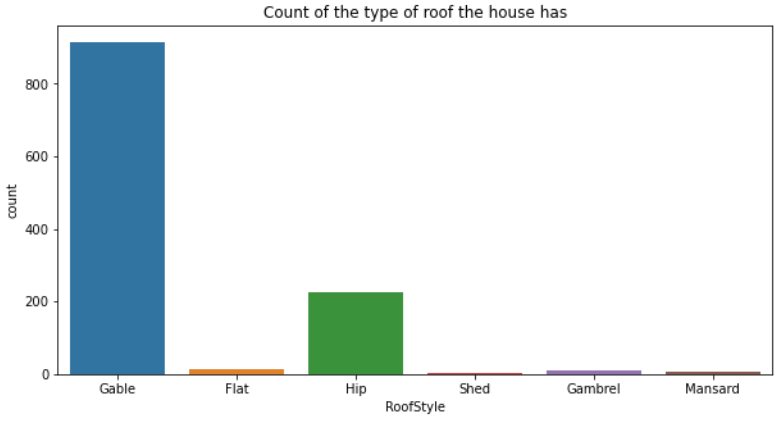

Count of number of cars the garage can hold in the house
count
GarageCars
0
1
2
3
4
0
100
200
300
400
500
600

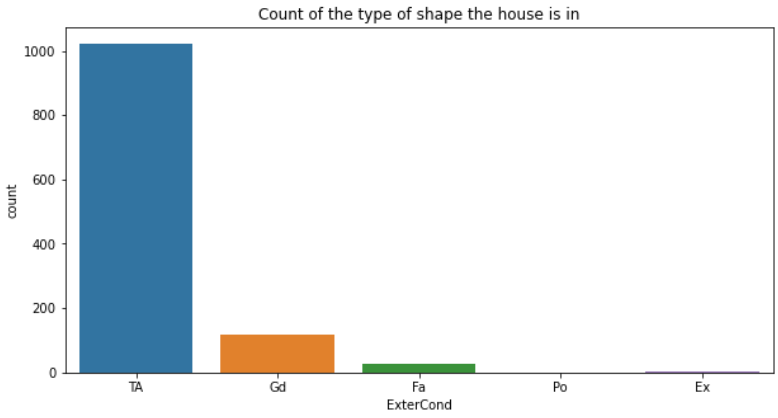

Count of number of houses sold month wise
count
MoSold
1
2
3
4
5
6
7
8
9
10
11
12
0
25
50
75
100
125
150
175
200

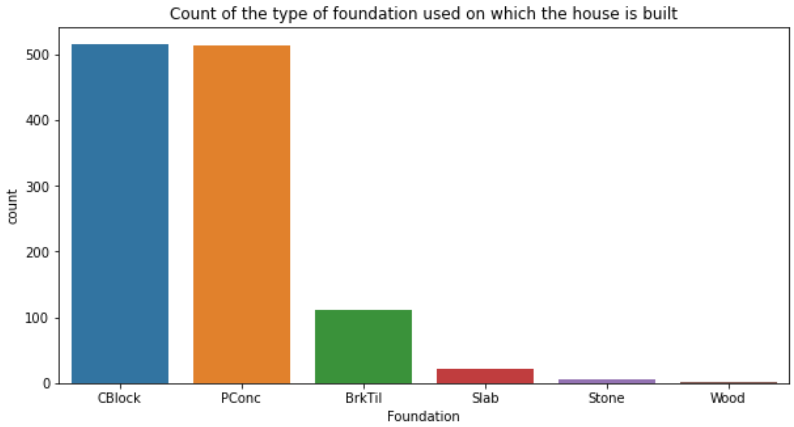

Count of number of houses sold year wise
count
YrSold
2006
2007
2008
2009
2010
0
50
100
150
200
250

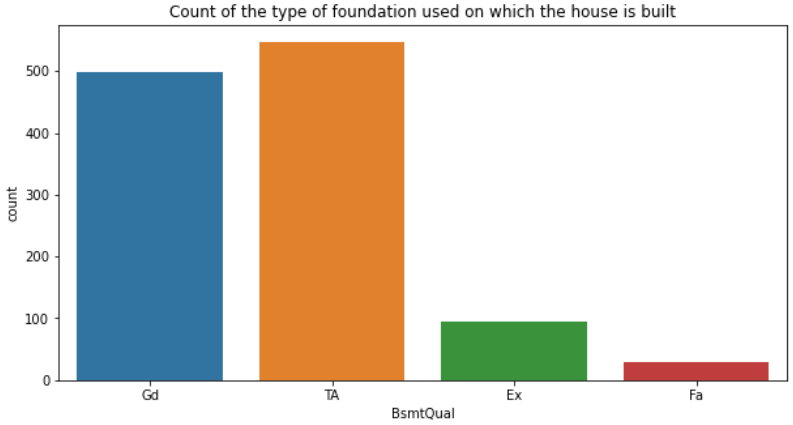

Count of the type of shape the house is in
count
LotShape
IR1
Reg
IR2
IR3

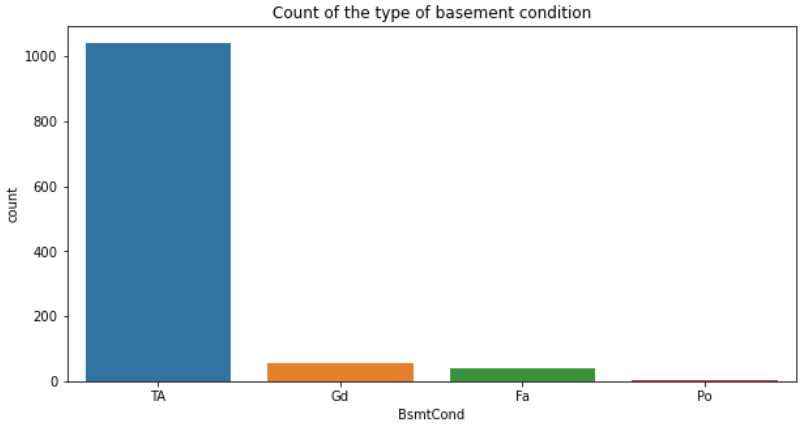

Count of the type neighbourhood the house is in
count
Neighborhood
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
Sawyer
Edwards
IDOTRR
CollgCr
Mitchel
Crawfor
BrDale
StoneBr
BrkSide
NridgHt
OldTown
Somerst
Timber
SWISU
SawyerW
ClearCr
Veenker
Bimngtn
MeadowV
Blueste

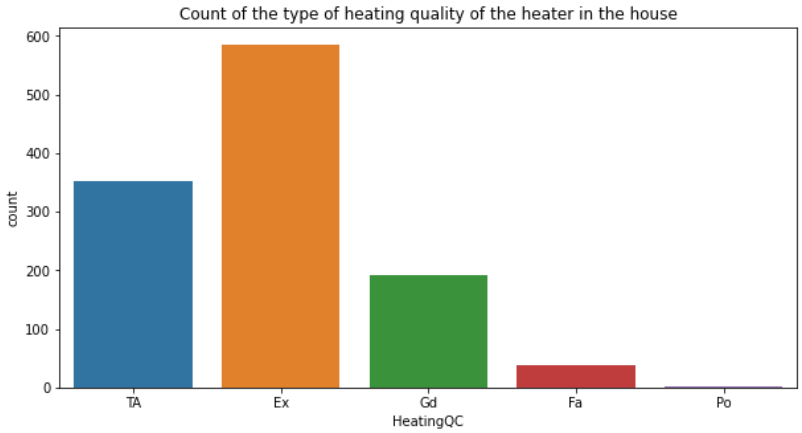

Count of the type of roof the house has
count
RoofStyle
Gable
Flat
Hip
Shed
Gambrel
Mansard

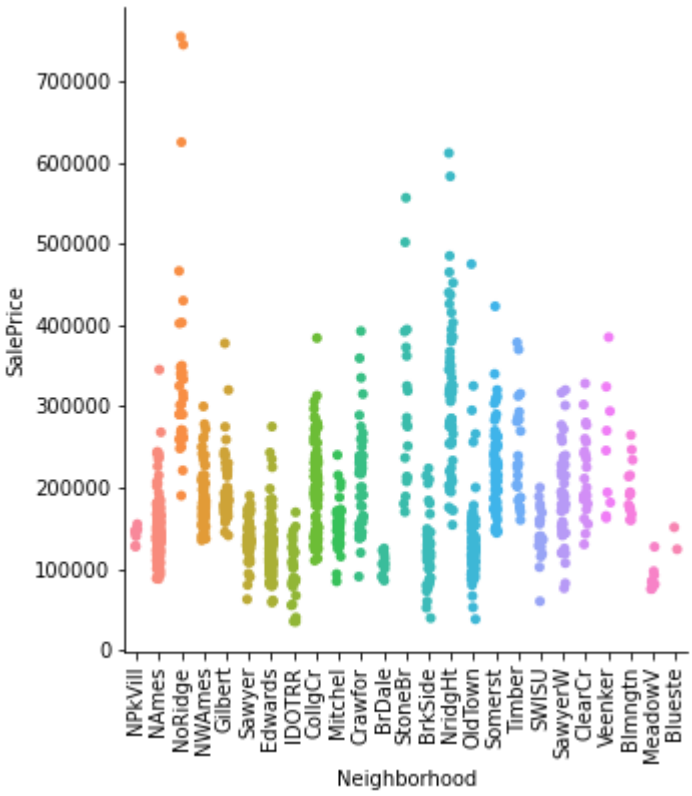

Count of the type of shape the house is in
count
ExterCond
TA
Gd
Fa
Po
Ex

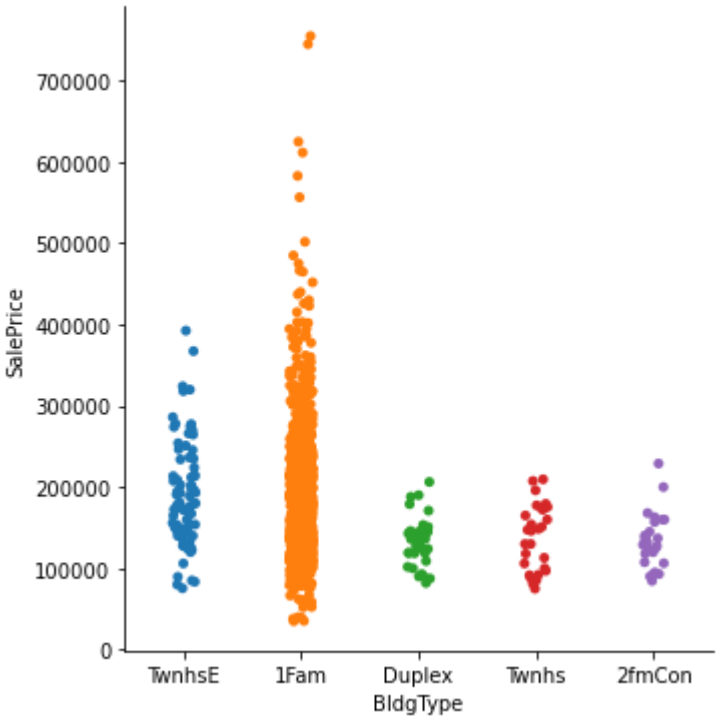

Count of the type of foundation used on which the house is built
count
0
100
200
300
400
500
CBlock
PConc
BrkTil
Slab
Stone
Wood
Foundation

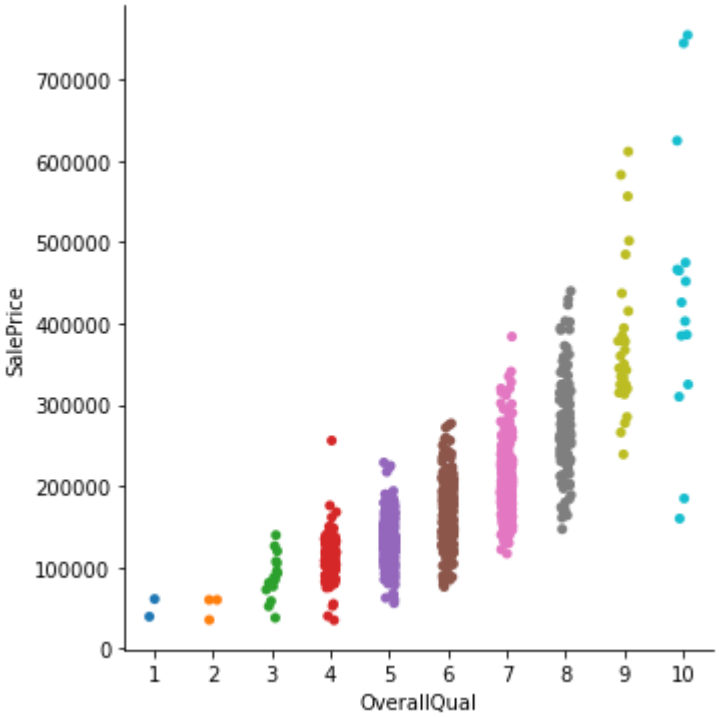

Count of the type of foundation used on which the house is built
count
0
100
200
300
400
500
Gd
TA
Ex
Fa
BsmtQual

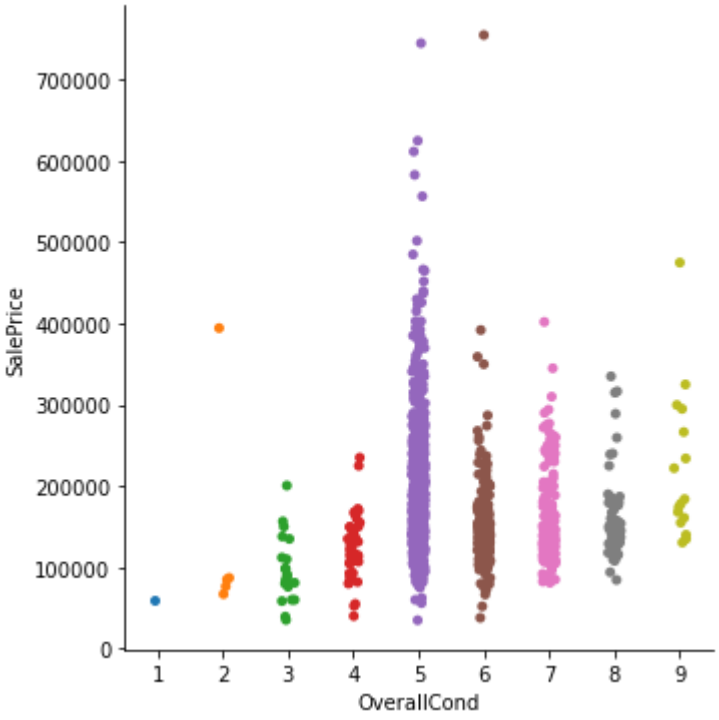

Count of the type of basement condition
count
0
200
400
600
800
1000
TA
Gd
Fa
Po
BsmtCond

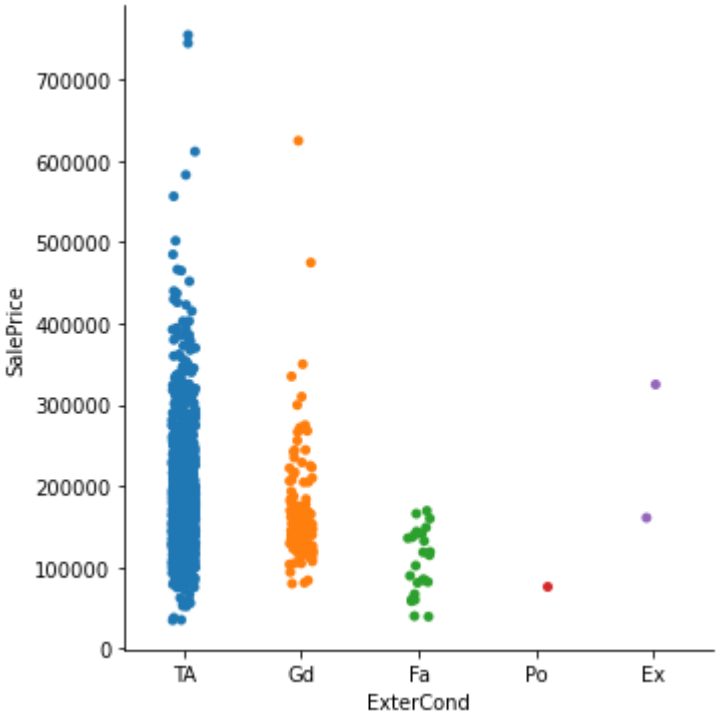

Count of the type of heating quality of the heater in the house
count
0
100
200
300
400
500
600
TA
Ex
Gd
Fa
Po
HeatingQC

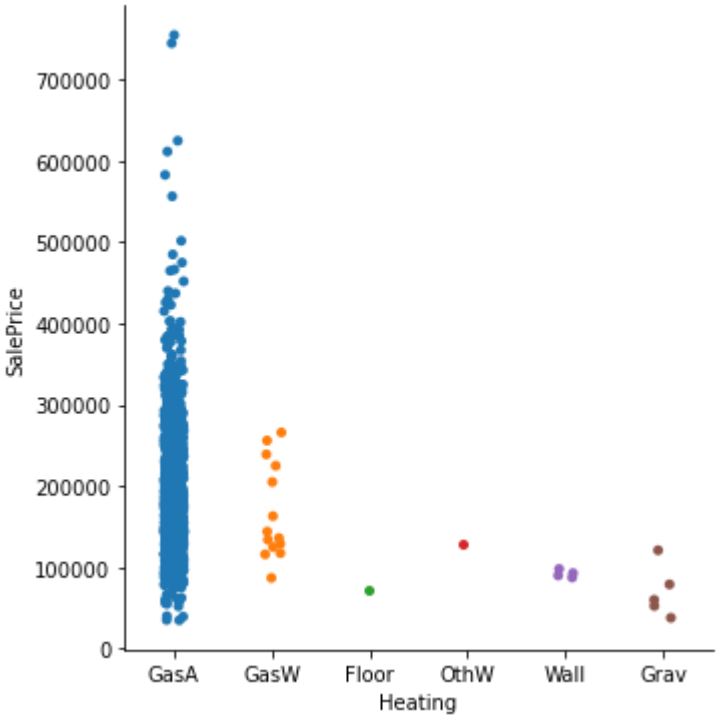

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
Sawyer
Edwards
IDOTRR
CollgCr
Mitchel
Crawfor
BrDale
StoneBr
BrkSide
NridgHt
OldTown
Somerst
Timber
SWISU
SawyerW
ClearCr
Veenker
Blmngtn
MeadowV
Blueste
Neighborhood

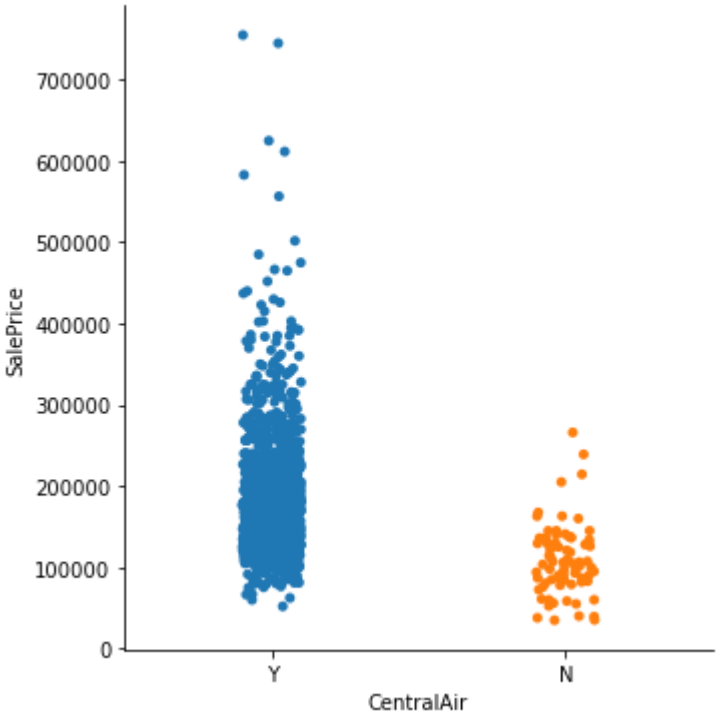

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TwnhsE
1Fam
Duplex
Twnhs
2fmCon
BldgType

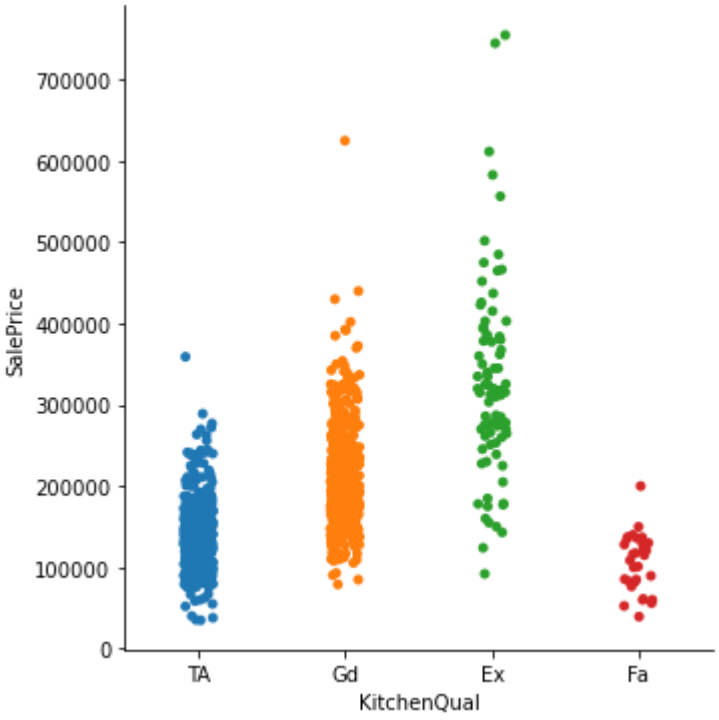

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
1
2
3
4
5
6
7
8
9
10
OverallQual

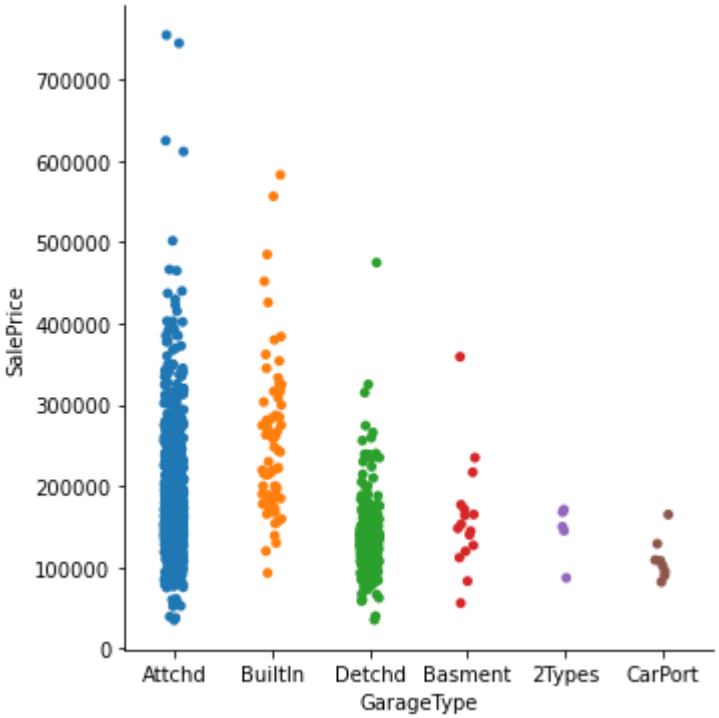

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
1
2
3
4
5
6
7
8
9
OverallCond

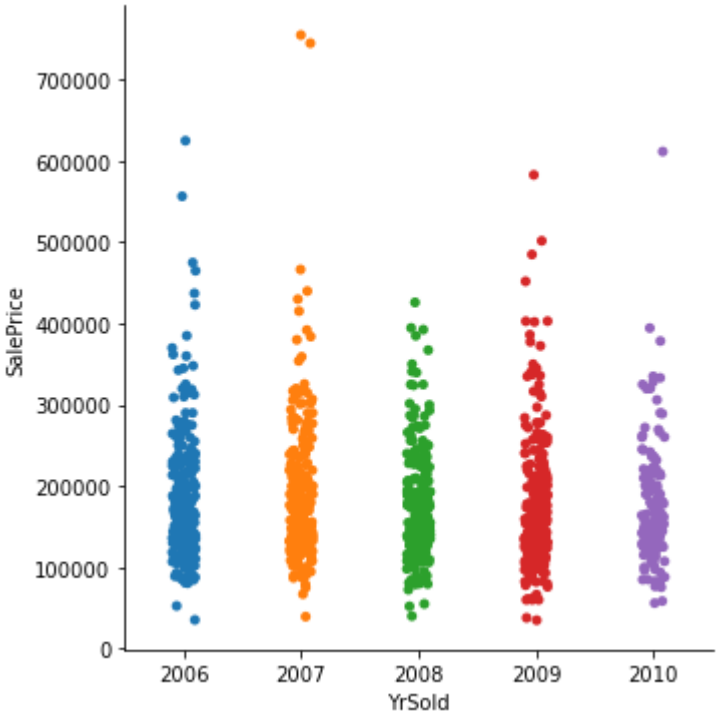

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Fa
Po
Ex
ExterCond


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


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


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


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


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
2006
2007
2008
2009
2010
YrSold

3.5 Run and evaluate selected models

Linear Regression

```
LR = LinearRegression()
LR.fit(X_train,y_train)
pred = LR.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(LR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 87.08335970242894
mean_squared_error: 573027546.9969226
mean_absolute_error: 18350.626482045045
root_mean_squared_error: 23937.993796409144

Cross validation score : 85.85045786191188
```

Random Forest Regression

```
RFR = RandomForestRegressor()
RFR.fit(X_train,y_train)
pred = RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(RFR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 90.66327030043934
mean_squared_error: 414210134.636841
mean_absolute_error: 15118.615318471337
root_mean_squared_error: 20352.153071280714

Cross validation score : 86.62897838723815
```

Decision Tree Regression

```
DTR = DecisionTreeRegressor()
DTR.fit(X_train,y_train)
pred = DTR.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(DTR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 73.00729421612736
mean_squared_error: 1197491269.0764332
mean_absolute_error: 24618.04458598726
root_mean_squared_error: 34604.7867942635

Cross validation score : 70.32476725769848
```

AdaBoost Regression

```
ABR = AdaBoostRegressor()
ABR.fit(X_train,y_train)
pred = ABR.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(ABR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 85.49339158159569
mean_squared_error: 643564119.2862126
mean_absolute_error: 19619.928979858174
root_mean_squared_error: 25368.565574076365

Cross validation score : 81.16459811608661
```

Extra Trees Regression

```
ETR = ExtraTreesRegressor()
ETR.fit(X_train,y_train)
pred = ETR.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(ETR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 88.11403833153348
mean_squared_error: 527303021.658163
mean_absolute_error: 16476.64662420382
root_mean_squared_error: 22963.079533419794

Cross validation score : 86.18486049445772
```

Gradient Boosting Regression

```
GBR = GradientBoostingRegressor()
GBR.fit(X_train,y_train)
pred = GBR.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(GBR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 90.637818008869
mean_squared_error: 415339287.71906674
mean_absolute_error: 14914.80589401157
root_mean_squared_error: 20379.87457564611

Cross validation score : 88.22534721995237
```

XG Boost Regression

```
XGB = XGBRegressor()
XGB.fit(X_train,y_train)
pred = XGB.predict(X_test)
R2_score = r2_score(y_test,pred)*100

print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#Cross Validation Score
scores = cross_val_score(XGB, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)
```

```
R2_score: 89.695267999843
mean_squared_error: 457154117.8259066
mean_absolute_error: 16153.81460240844
root_mean_squared_error: 21381.162686484255

Cross validation score : 86.62178222039195
```

After building the various models and looking at the difference between the r2 score and cross validation score, Linear Regression gave us the least difference and hence, I selected Linear Regression is the best model.

Saving the best model

```
import pickle
filename = "House_price.pkl"
pickle.dump(LR,open(filename,'wb'))
```

3.6 Interpretation of Results

The feature “OverallQual” has the highest positive correlation with the target variable “SalePrice” and the feature “KitchenAbGr” has the highest negative correlation with our target variable “SalePrice”. After building the regression models, I found that the Linear Regression model gave us the least difference between the r2 score and accuracy score, so I selected Linear Regression as the best model, saved this model using pickle (.pkl) and predicted the results using this model.

4. CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this project, I have used machine learning algorithms to predict whether a customer is a defaulter or not based on some features. I have mentioned the step by step process that I have done to get the best accuracy score of my model. After performing all the steps I have found that “Random Forest Classifier” gave me the best accuracy score of 94.96% when compared to the other model that I built.

4.2 Learning Outcomes of the Study In Respect of Data Science

- It was a really interesting project to work on as the dataset told me the risk involved in giving a customer money without knowing anything about him, this model would help us in analyzing whether you should give that particular customer a loan or not
- Visualization is such a powerful tool, which helps us in interpreting the dataset in a very easy manner
- This project helped me in breaking a task into sub-tasks and working on each part individually to easily complete the problem without any hassle

4.3 Limitations of this work and Scope for Future work

- As Linear Regression was the model and we know that Linear Regression does not have hyperparameters that can be tuned
- The scope of this project would help buyers to get to know an approximate value of the house and then go ahead with finding out whether they can afford the property or not before going ahead and asking for details from the owner.