Housing Price Prediction Project

Submitted by:

Aamina Ruvaida

ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies that provide me to work on Housing Price Prediction Project. I have research on Housing.com site to know the price of house, how it differs the price based on locality, area size and others features. It gives me an insight that every location, city. The square feet size price is different. I have read the article on Encyclopaedia how inflation affect the house price. I have gone through the articles on how to make the price prediction for house based on analysis of various feature. It helps a lot in completing my project.

Reference:

How Does Inflation Affect House Prices? - Encyclopedia.com

House Price Prediction With Machine Learning in Python | by Nikhil Adithyan | CodeX | Medium

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

CONTENT

INTRODUCTION

Business Problem Framing

Houses are one of the necessary needs 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.

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. House price prediction helps the developer to analyse the data, feature and research on to make the best prediction that help real estates to buy or sell homes at better price.

As a Data Scientist we analyse different features, market trends, range of price. Understanding of people perspective. We provide a better visualization to end user to understand which is in high demand, what people are expecting when they buy or sell their house. We provide the prediction of house price by applying our skill using various machine learning models.

Conceptual Background of the Domain Problem

Real estate agents play crucial roles in terms of determining pricing of house, forecasting property and trends that will attract customer and as well as investors to invest in buying a house. Everyone has dream to make their own house, when they come to invest before they think all criterion and analyse many things. It's going to invest one time. So, providing the better price prediction and demonstrate the things which will be available in nearby their house.

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 data is provided in the CSV file below.

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

The factor that determines the quality of house based on analysis of various features. Real estate is a fast-growing sector, where people are buying and selling the house daily. To give the customer their dream in a better price not with overprice we need to dive into deep analysis of features. Where each feature will give a better insight and meaning. We need to check whether is correlated with other features. What makes this house price is predicted for this range? Does this house price is worth it? We can give answer to this question by using various visualization that used in this project. We used many machines learning to make a good prediction and checked their accuracy using r2 score, mean squared error and mean absolute error.

As it is continuous data, we use linear regression to make prediction. There is an increase in population, as there increase in Information Technology(IT) too. Many IT Sector leverage their field in cities. So, people are moving from one place to another in search of house.

The main objective of a Buyer is to search for their dream house which has all the amenities they need. Furthermore, they look for these houses/Real estates with a price in mind and there is no guarantee that they will get the product for a deserving price and not overpriced. Similarly, A seller looks for a certain number that they can put on the estate as a price tag and this cannot be just a wild guess, lots of research needs to be put to conclude a valuation of a house.

The aim of this project is to use the machine learning technique and statistical method to determine the best house price prediction. To complete the dream of people to get their dream house.

Motivation for the Problem Undertaken

The motivation behind this project is to analyse the independent variables and to predict the house price. Then the report will go to management team, they analyse which variable are giving a high return. So, that they will focus in that particular area. It motivates us to analyse the wholesome data, we process the data cleaning, imputation technique and exploratory data analysis that will provide the better understanding how each feature is playing an important role in determine the prediction.

The prediction of house price is dynamic like a stock price. Its changing on daily basis. We gather the few years data and see how to price is drastically increasing each year. We make the report of each year price high and fall. The economic is increasing, so the price also increasing. We need to find the relationships of housing price and the economic to make the better understanding. That will our motivating factor that influence us to analyse the independent feature and gives a better price prediction.

2.Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

There are two dataset train and test dataset. I have built the machine learning model using the train dataset and predict the SalePrice for test data set. Our target variable is SalePrice, it is a continuous variable. So, it is regression problem. I have used the regression model for making the prediction. In most of the columns null values are present, so I have replaced with mean for numerical data and mode for categorical data. Checked the outliers and removed the outliers using the zscore method and I have checked the skewness of data. In most of the column skewness is present I have removed the skewness using yeo-johnson method. I have removed the column which is having the high variation inflation factor. Used regression plot, distribution pot, scatterplot, strip plot for visualization. Checked the correlation of data. We have built the model and checked the accuracy of each model. Then we have tuned the model to check if we can increase our accuracy or not. Finally saved the model using pickle.

- **Data Sources and their formats**

The data set was provided by Flip Robo Technologies and in the format of .csv file. There are two data set present. One is for training and another for testing. In the train data set, there are 1168 rows and 81 columns. In the test data set, there are 292 rows and 80 columns. I have analysed both the data set separately. In the data set contain integer, float and object data type. Our target variable is SalePrice. It is a continuous, So, I have used the regression method. Then I import those data set in Python and started working on analysis and built the model.

Train Data:

```
# Importing the train data set

df_train=pd.read_csv('train.csv')
df_train.head(10)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Cond
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NPkVill	Norm	
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	Inside	Mod	NAmes	Norm	
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	NoRidge	Norm	
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NWAmes	Norm	
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NWAmes	Norm	
5	1197	60	RL	58.0	14054	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	
6	561	20	RL	NaN	11341	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	
7	1041	20	RL	88.0	13125	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	Sawyer	Norm	
8	503	20	RL	70.0	9170	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	Edwards	Feedr	
9	576	50	RL	80.0	8480	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	

Test Data:

```
# Importing the test data set
df_test=pd.read_csv('test.csv')
df_test.head(10)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Cond
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	Corner	Gtl	StoneBr	Norm	
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	StoneBr	Norm	
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	ColIgCr	Norm	
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	Inside	Gtl	Crawfor	Norm	
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	Somerst	Feedr	
5	650	180	RM	21.0	1936	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	MeadowV	Norm	
6	1453	180	RM	35.0	3675	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Edwards	Norm	
7	152	20	RL	107.0	13891	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	
8	427	80	RL	NaN	12800	Pave	NaN	Reg	Low	AllPub	Inside	Mod	SawyerW	Norm	
9	776	120	RM	32.0	4500	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Mitchel	Norm	

- **Data Pre-processing Done**

✓ I have import the necessary libraries and do the analysis of data.

✓ Checked the unique value, data info, data statistical summary of data, shape of data.

✓ I have found in most of the columns have null value. I have replaced those null value with mean if it is numerical data and replace with mode if it is categorical data.

✓ There are skewness present in data. We have removed the skewness using yeo-johnson method.

✓ Upon checked find that there are outliers are present in both train and test data. I have removed the outliers using zscore.

- ## Data Inputs- Logic- Output Relationships

- I see that there is linear relationship between features and target variables.
- In box plot we see the relationship between the feature. Median value compares with median.
- In regression plot we saw the relationship of feature for train data.
- In box plot it show the relationship of SalePrice for each sub categories.
- In Scatterplot it shows the relations of each feature and with the target variables.

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

I have the two data set for training and testing separately. I can combine both the data and make the prediction. But I have not done that because of data leakage issue.

In the year column I have separated into three column day, month and year. Dropped the year column. I assume that age will give a better understanding and insight then year.

- ## Hardware and Software Requirements and Tools Used

Here are the list of software, tools and package used to complete this project.

Hardware:

Processor: core i3

RAM Size: 12GB

Software:

Jupyter Notebook/Anaconda

Libraries:

```
# Importing Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

For building the model I have used below libraries.

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import BaggingRegressor
from xgboost import XGBRegressor
from sklearn.linear_model import SGDRegressor
from sklearn.ensemble import AdaBoostRegressor
```

I have used various model for making the prediction. The reason for using the various model is to check the which model giving me the good score and what's the different output comes from each model.

After building the model I have done the cross-validation score whether it is giving a good score for that model that I am supposed to go ahead to make the final model for prediction.

Before building the model, I have checked the variation inflation factor for my data set. By importing

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Then I done the hyper parameter tuning for our final model.

Model/s Development and Evaluation

- ## Identification of possible problem-solving approaches (methods)

There are null values in data I have replace them with mean for numerical value and replace them with mode for categorical value. Removed the skewness using 'yeo-johnson' method. Removed the outliers using the zscore method. Removed the multicollinearity from data set using variation inflation factor. I have encoded the categorical data to numerical data using Label Encoder. Then I have scaled the data using Standard Scaler. Then I have started building our model.

- ## Testing of Identified Approaches (Algorithms)

As our target column is SalePrice, it is a continuous column. So, I have used the regression model. Here is the list of algorithms used in project.

- LinearRegression
- DecisionTreeRegressor
- RandomForestRegressor
- GradientDescentRegressor
- KNeighborsRegressor
- BaggingRegressor
- ExtraTreeRegressor
- AdaBoostRegressor
- SupportVectorRegressor
- SGDRegressor
- XGBRegressor

After built the model I have done the cross validation to avoid the over fitting problem. I have checked the r2 score, mean squared error and mean absolute error for each model. I choose the Gradient Boosting Regressor as it gives a best score and good cross validation score. I have done the hyper parameter tuning to check the accuracy if it increases after tuning the parameter.

- **Run and Evaluate selected models**

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import BaggingRegressor
from xgboost import XGBRegressor
from sklearn.linear_model import SGDRegressor
from sklearn.ensemble import AdaBoostRegressor
```

```
max_accu=0
max_RS=0
for i in range(1,200):
    x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=i,test_size=.30)
    rf=RandomForestRegressor()
    rf.fit(x_train,y_train)
    mod=rf.predict(x_test)
    score=r2_score(y_test,mod)
    if score>max_accu:
        max_accu=score
        max_RS=i
print("Best Accuracy score is:",max_accu,"On Random State:",max_RS)
```

```
Best Accuracy score is: 0.9008140205013233 On Random State: 135
```

Checked the random state, On 135 random state it the accuracy score 90%. With this I have built the model.

Linear Regression Model:

```
lr=LinearRegression()
lr.fit(x_train,y_train)
```

```
LinearRegression()
```

```
lr_pred=lr.predict(x_test)
print("Predicted value:\n",lr_pred)
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

print("R2 Score value is:",r2_score(y_test,lr_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,lr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,lr_pred))
```

```
R2 Score value is: 0.7798773656305659
Mean Squared Error value is: 1421966718.20338
Mean Absolute Error value is: 24124.668990447943
```

In Linear Regression it gives the R2 Score: 77%.

Random Forest Regressor:

```
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
```

```
RandomForestRegressor()
```

```
rfr_pred=rfr.predict(x_test)
print("Predicted value:\n",rfr_pred)
```

```
print("R2 Score value is:",r2_score(y_test,rfr_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,rfr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,rfr_pred))
```

```
R2 Score value is: 0.8657195622421776
Mean Squared Error value is: 867436072.3711108
Mean Absolute Error value is: 19138.336153846154
```

In Random Forest Regressor it gives the r2 score value is 86%.

KNeighbors Regressor

```
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
```

```
KNeighborsRegressor()
```

```
knn_pred=knn.predict(x_test)
print("Predicted value:\n",knn_pred)
```

```
print("R2 Score value is:",r2_score(y_test,knn_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,knn_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,knn_pred))
```

```
R2 Score value is: 0.7789691007255608
Mean Squared Error value is: 1427834004.2729344
Mean Absolute Error value is: 24478.465527065528
```

In KNeighbors Regressor gives a r2 score value is 78%.

Gradient Boosting Regressor

```
gbr=GradientBoostingRegressor()
gbr.fit(x_train,y_train)
```

```
GradientBoostingRegressor()
```

```
gbr_pred=gbr.predict(x_test)
print("Predicted value:\n",gbr_pred)
```

```
print("R2 Score value is:",r2_score(y_test,gbr_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,gbr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,gbr_pred))
```

```
R2 Score value is: 0.8568908538535339
Mean Squared Error value is: 924468505.8113929
Mean Absolute Error value is: 18584.143961828326
```

In Gradient Boosting Regressor it gives r2 score value is 86%.

Ada Boost Regressor

```
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
```

```
AdaBoostRegressor()
```

```
ada_pred=ada.predict(x_test)
print("Predicted values:\n",ada_pred)
```

```
print("R2 Score value is:",r2_score(y_test,ada_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,ada_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,ada_pred))
```

```
R2 Score value is: 0.7803799398530608
Mean Squared Error value is: 1418720146.946127
Mean Absolute Error value is: 27886.92022642047
```

In Ada Boost Regressor it gives a r2 score value 78%

Decision Tree Regressor

```
dtc=DecisionTreeRegressor()
dtc.fit(x_train,y_train)
```

```
DecisionTreeRegressor()
```

```
dtc_pred=dtc.predict(x_test)
print("Predicted value:\n",dtc_pred)
```

```
print("R2 Score value is:",r2_score(y_test,dtc_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,dtc_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,dtc_pred))
```

```
R2 Score value is: 0.7218954002902283
Mean Squared Error value is: 1796523497.4558403
Mean Absolute Error value is: 28408.344729344728
```

In Decision Tree Regressor it gives the r2 score value is 72%

Support Vector Regressor

```
svr=SVR(kernel='linear')
svr.fit(x_train,y_train)
```

```
SVR(kernel='linear')
```

```
svr_pred=svr.predict(x_test)
print("Predicted values:\n",svr_pred)
```

```
print("R2 Score value is:",r2_score(y_test,svr_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,svr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,svr_pred))
```

```
R2 Score value is: 0.08078556628743161
Mean Squared Error value is: 5938018756.570634
Mean Absolute Error value is: 51766.304609052444
```

In Support Vector Regressor it gives a r2 score value is 80%

Bagging Regressor

```
br=BaggingRegressor()
br.fit(x_train,y_train)
```

```
BaggingRegressor()
```

```
br_pred=br.predict(x_test)
print("Predicted values:\n",br_pred)
```

```
print("R2 Score value is:",r2_score(y_test,br_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,br_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,br_pred))
```

```
R2 Score value is: 0.8122389024038115
Mean Squared Error value is: 1212914939.528775
Mean Absolute Error value is: 22090.98803418803
```

In Bagging Regressor it gives the r2 score is 81%

XGB Regressor

```
xgb=XGBRegressor()
xgb.fit(x_train,y_train)
```

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
             gamma=0, gpu_id=-1, importance_type=None,
             interaction_constraints='', learning_rate=0.300000012,
             max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=100, n_jobs=4,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method='exact',
             validate_parameters=1, verbosity=None)
```

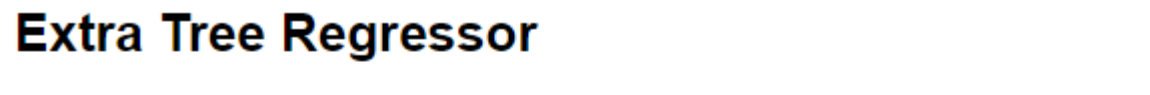
```
xgb_pred=xgb.predict(x_test)
print("Predicted value:\n",xgb_pred)
```

```
print("R2 Score value is:",r2_score(y_test,xgb_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,xgb_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,xgb_pred))
```

```
R2 Score value is: 0.8384264332020065
Mean Squared Error value is: 1043746524.233223
Mean Absolute Error value is: 19911.45108840812
```

In XGB Regressor it gives the r2 score 83%

Extra Tree Regressor




```
from sklearn.tree import ExtraTreeRegressor

etr=ExtraTreeRegressor()
etr.fit(x_train,y_train)
```

```
ExtraTreeRegressor()
```

```
etr_pred=etr.predict(x_test)
print("Predicted values:\n",etr_pred)
```

```
print("R2 Score value is:",r2_score(y_test,etr_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,etr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,etr_pred))
```

```
R2 Score value is: 0.7080055438541986
Mean Squared Error value is: 1886250360.980057
Mean Absolute Error value is: 30604.222222222223
```

In Extra Tree Regressor it gives the r2 score is 70%

Cross Validation Score:

```
print("Cross Validation Score for Linear Regression is:",cross_val_score(lr,x,y,cv=10).mean()*100)
print("Cross Validation Score for Decision Tree Regressor is:",cross_val_score(dtc,x,y,cv=10).mean()*100)
print("Cross Validation Score for Random Forest Regressor is:",cross_val_score(rfr,x,y,cv=10).mean()*100)
print("Cross Validation Score for Gradient Boosting Regressor is:",cross_val_score(gbr,x,y,cv=10).mean()*100)
print("Cross Validation Score for Ada Boost Regressor is:",cross_val_score(ada,x,y,cv=10).mean()*100)
print("Cross Validation Score for KNeighbors Regressor is:",cross_val_score(knn,x,y,cv=10).mean()*100)
print("Cross Validation Score for Support Vector Regressor is:",cross_val_score(svr,x,y,cv=10).mean()*100)
print("Cross Validation Score for XGB Regressor is:",cross_val_score(xgb,x,y,cv=10).mean()*100)
print("Cross Validation Score for SGD Regressor is:",cross_val_score(sgd,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for Linear Regression is: 77.14564376165815
Cross Validation Score for Decision Tree Regressor is: 65.82174005922823
Cross Validation Score for Random Forest Regressor is: 83.81729275921984
Cross Validation Score for Gradient Boosting Regressor is: 83.87671632260538
Cross Validation Score for Ada Boost Regressor is: 77.05790718695312
Cross Validation Score for KNeighbors Regressor is: 74.17921365976
Cross Validation Score for Support Vector Regressor is: 10.190583912029023
Cross Validation Score for XGB Regressor is: 82.23367062787655
Cross Validation Score for SGD Regressor is: 77.13040463440848
```

Cross Validation Score of all the model used in this project. Upon analysis I see the Gradient Boosting Regression gives a good score 84% and in model it gives a r2 score is 86%. In my project Gradient Boosting Regressor model gives a good score as compare to another model. So, I choose Gradient Boosting Regressor as my final model.

Hyper Parameter Tuning:

```
params={'criterion':['mse','mae'],
        'n_estimators':[100,150],
        'learning_rate':[0.001,0.01,1.0],
        'random_state':[19],
        'max_depth':[5,10]}
```

```
from sklearn.model_selection import GridSearchCV
```

```
grid_param=GridSearchCV(estimator=gbr,cv=3,param_grid=params)
```

```
grid_param.fit(x_train,y_train)
```

```
GridSearchCV(cv=3, estimator=GradientBoostingRegressor(),
             param_grid={'criterion': ['mse', 'mae'],
                         'learning_rate': [0.001, 0.01, 1.0],
                         'max_depth': [5, 10], 'n_estimators': [100, 150],
                         'random_state': [19]})
```

```
grid_param.best_params_
```

```
{'criterion': 'mse',
 'learning_rate': 0.01,
 'max_depth': 5,
 'n_estimators': 150,
 'random_state': 19}
```

```
grid_param.best_params_
```

```
{'criterion': 'mse',
 'learning_rate': 0.01,
 'max_depth': 5,
 'n_estimators': 150,
 'random_state': 19}
```

```
grid_param.best_estimator_
```

```
GradientBoostingRegressor(criterion='mse', learning_rate=0.01, max_depth=5,
                          n_estimators=150, random_state=19)
```

```
grid_param.best_score_
```

```
0.769957334352636
```

Final Model

```
final_model=GradientBoostingRegressor(criterion='mse', learning_rate=0.01, max_depth=5,n_estimators=150, random_state=190)
```

```
final_model.fit(x_train,y_train)
```

```
GradientBoostingRegressor(criterion='mse', learning_rate=0.01, max_depth=5,
                          n_estimators=150, random_state=190)
```

```
final_model_pred=final_model.predict(x_test)
print("Predicted values:\n",final_model_pred)
```

```
print("R2 Score value is:",r2_score(y_test,final_model_pred))
print("Mean Squared Error value is:",mean_squared_error(y_test,final_model_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,final_model_pred))
```

```
R2 Score value is: 0.7950755785962432
Mean Squared Error value is: 1323788023.0625176
Mean Absolute Error value is: 24163.878115537846
```

After tuning the parameter, the score didn't increase.

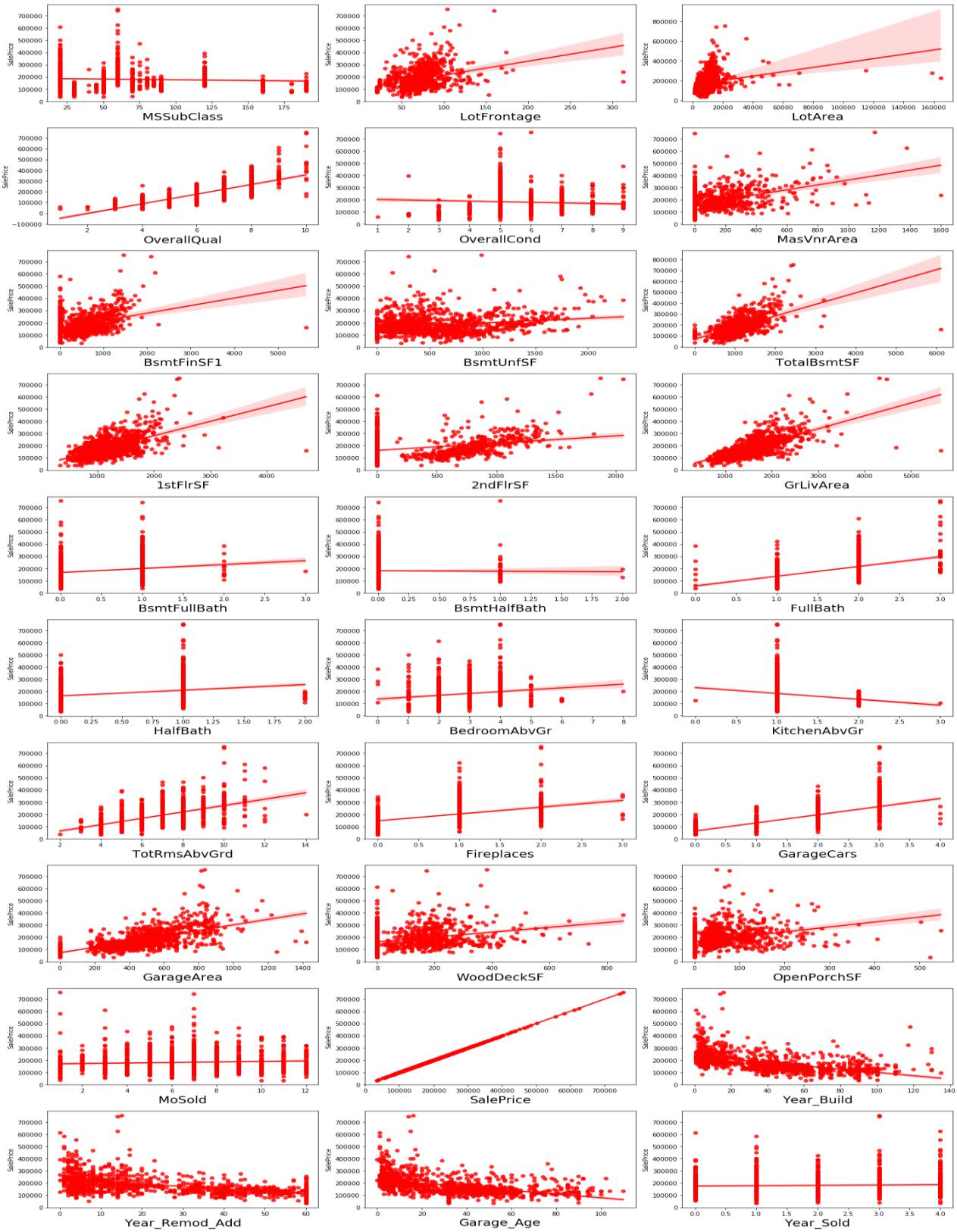
- **Key Metrics for success in solving problem under consideration**

I have used the below Metrix in project.

- I used the r2 score which gives score of models, it tells how accurate our model is.
- I have used the mean absolute error which gives the difference between the predicted and actual value.
- I have used the mean squared error is to check the average of the square difference between the actual and predicted value.

Visualizations

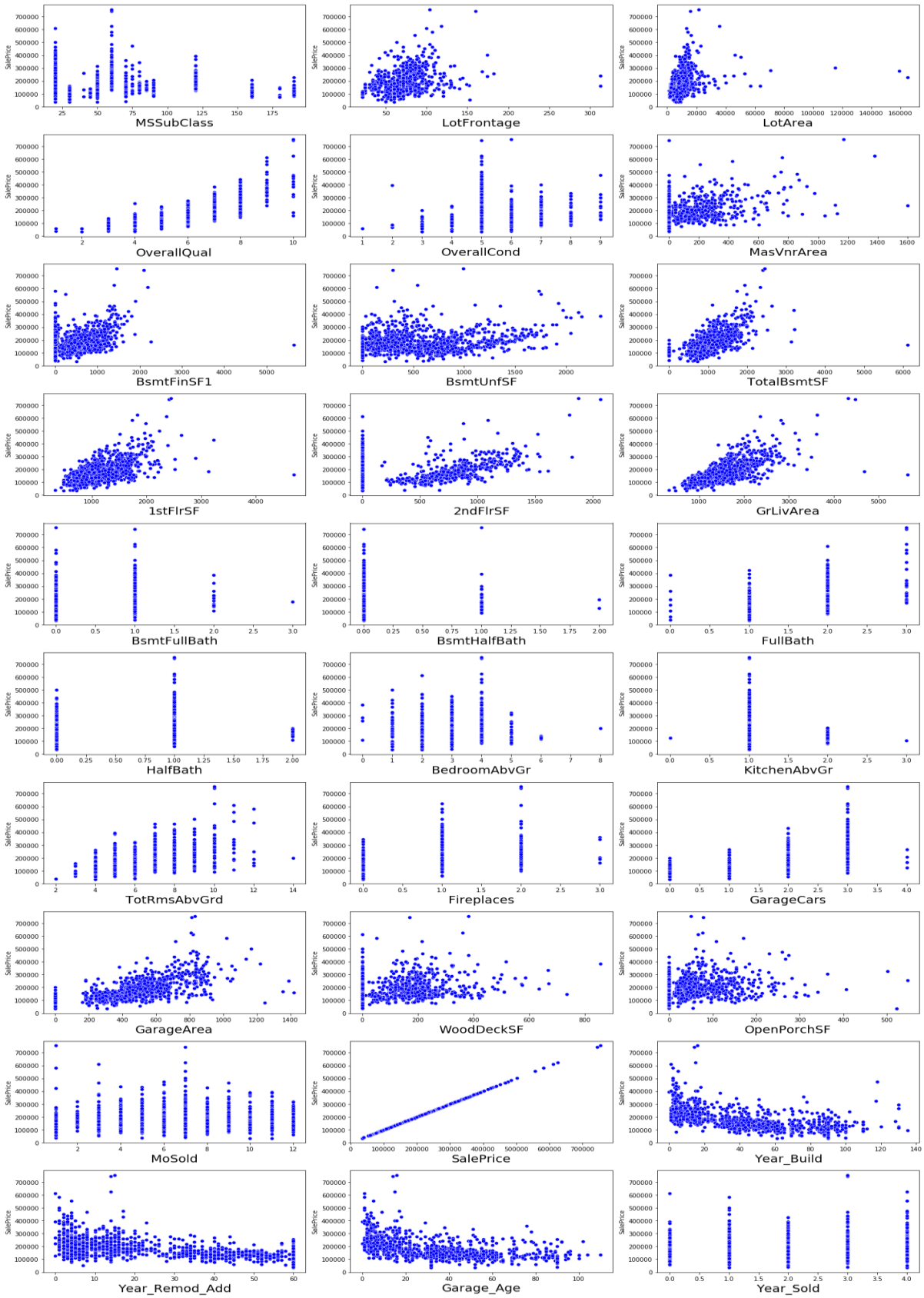
Visualization of regression plot of each features with respect to target variables.



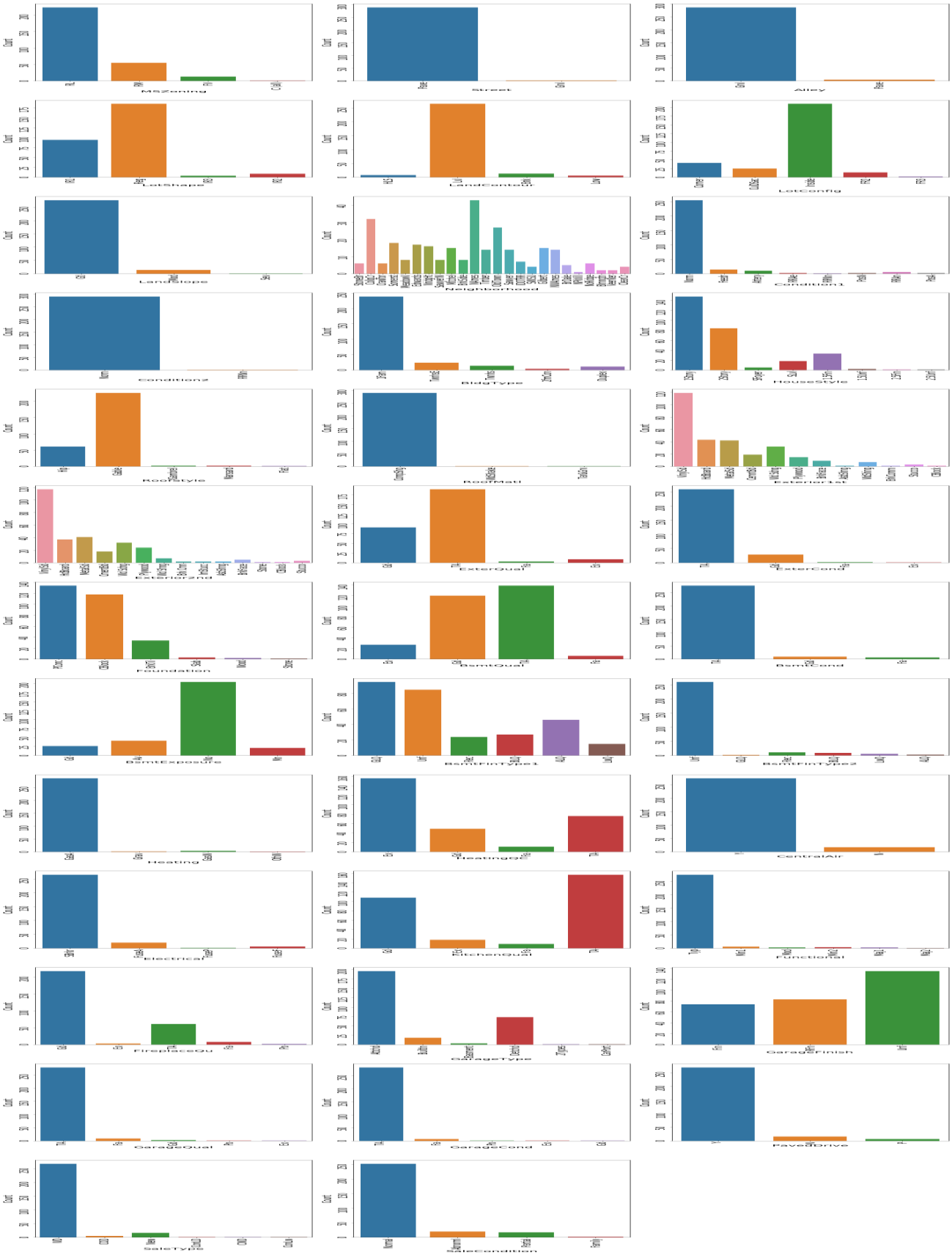
Observation of features with target in the Regression plot:

- 1.As Linear feet of street connected to property(LotFrontage) is increasing sales is decreasing and the SalePrice is ranging between 0-3 lakhs.
- 2.As Lot size in square feet(LotArea) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 3.As Masonry veneer area in square feet (MasVnrArea) is increasing sales is decreasing and SalePrice is ranging between 0-4 lakhs.
- 4.As Type 1 finished square feet(BsmtFinSF1) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 5.As Unfinished square feet of basement area (BsmtUnfSF) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs. There are some outliers also.
- 6.As Total square feet of basement area (TotalBsmtSF) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 7.As First Floor square feet(1stFlrSF) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 8.As Second floor square feet(2ndFlrSF) is increasing sales is increasing in the range 500-1000 and the SalePrice is in between 0-4 lakhs.
- 9.As Above grade (ground) living area square feet (GrLivArea) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 10.As Size of garage in square feet(GarageArea) is increasing sales is increasing and the SalePrice is in between 0-4 lakhs.
- 11.As Wood deck area in square feet(WoodDeckSF) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 12.As Open porch area in square feet (OpenPorchSF) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 13.As Year_SinceBuilt is increasing sales is decreasing and the SalePrice is high for newly built building and the sales price is in between 0-4 lakhs.
- 14.As Since Remodel date (same as construction date if no remodeling or additions)(Year_SinceRemodAdded) is increasing sales is decreasing and the SalePrice is in between 1-4 lakhs.
- 15.As Since Year garage was built(GarageAge) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
- 15.As Since Year garage was built(GarageAge) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.

Visualizing the scatterplot of each feature and the target variable SalePrice. It will to which feature the target price is high under which categories.



Visualizing the count plot of each features for categorical data in the test data.



Observation:

Residential Low Density count is high as compare to other MSZoning

* In the Stree Pave count is high, in increase of sales it also depends on the type of street

* In the general shape of the property, the regural shape has the maximum count, Single Regular count is also little high next to regular shape.

* If we see the flatness of the property, the property which is having the Near Flat level has maximum count. In the Depression flatness of propery has very less count.

* In the lot configuration, inside lot configuration has the maximum count.

* In the slope of propety, the gentle slope has the maximum count.

* In the Neighborhood: Physical locations within Ames city limits, NAMES has the maximum count as compare to others.

* In Condition1, proximity to various condition the normal condition has the maximum count. Many people to have normal proximity condition. So, its count is high.

* In Condition2, Proximity to various conditions (if more than one is present) also people prefer the normal condition. So, it has the maximum count.

* In BldgType, the type of dwelling, 1 Farm-Single Family Detached has the maximum count.

For Vinyl Siding exterior-2 covering on house has maximum counts for the feature Exterior covering on house (if more than one material)(Exterior2nd).

* For Masonry veneer type(MasVnrType) None has maximum count.

* For Typical/Average(TA) quality of the material on the exterior has maximum count, for the feature Evaluates the quality of the material on the exterior (ExterQual).

* For Typical/Average(TA) condition of the material on the exterior has maximum count for the feature Evaluates the present condition of the material on the exterior(ExterCond).

* For Cinder Block and Poured Contrete foundations the count is maximum for the feature Type of foundation(Foundation).

* For good and average quality heights of the basement the count is high for the feature Evaluates the height of the basement(BsmtQual).

* For Typical/Average(TA) general condition of the basement the count is high for the feature Evaluates the general condition of the basement(BsmtCond)

* For No Exposure garden level walls the count is maximum for the feature Refers to walkout or garden level walls(BsmtExposure).

* For unfinished Rating of basement finished area-1 the count is maximum for the feature Rating of basement finished area(BsmtFinType1).

* For unfinished Rating of basement finished area-2 the count is maximum for the feature Rating of basement finished area (if multiple types)(BsmtFinType2).

* For Gas forced warm air furnace type of heating the count is maximum for the feature Type of heating(Heating).

* For Excellent Heating quality and condition the count is high for the feature Heating quality and condition(HeatingQC).

* For Central air conditioning-yes has maximum count for the feature Central air conditioning(CentralAir).

* For Standard Circuit Breakers & Romex Electrical system the count is high for the feature Electrical system(Electrical).

* For Typical/Average(TA) and good Kitchen quality the count is maximum for the feature Kitchen quality(KitchenQual).

* Typical Functionality has highest count for Home functionality (Assume typical unless deductions are warranted)(Functional).

* For good Fireplace quality the count is high for the feature Fireplace quality(FireplaceQu).

* If Garage location Attached to home then the count is high, for the feature Garage location(GarageType).

* For Unfinished Interior of the garage the count is maximum, for the feature Interior finish of the garage(GarageFinish).

* For Typical/Average(TA) Garage quality the count is high, for the feature Garage quality(GarageQual).

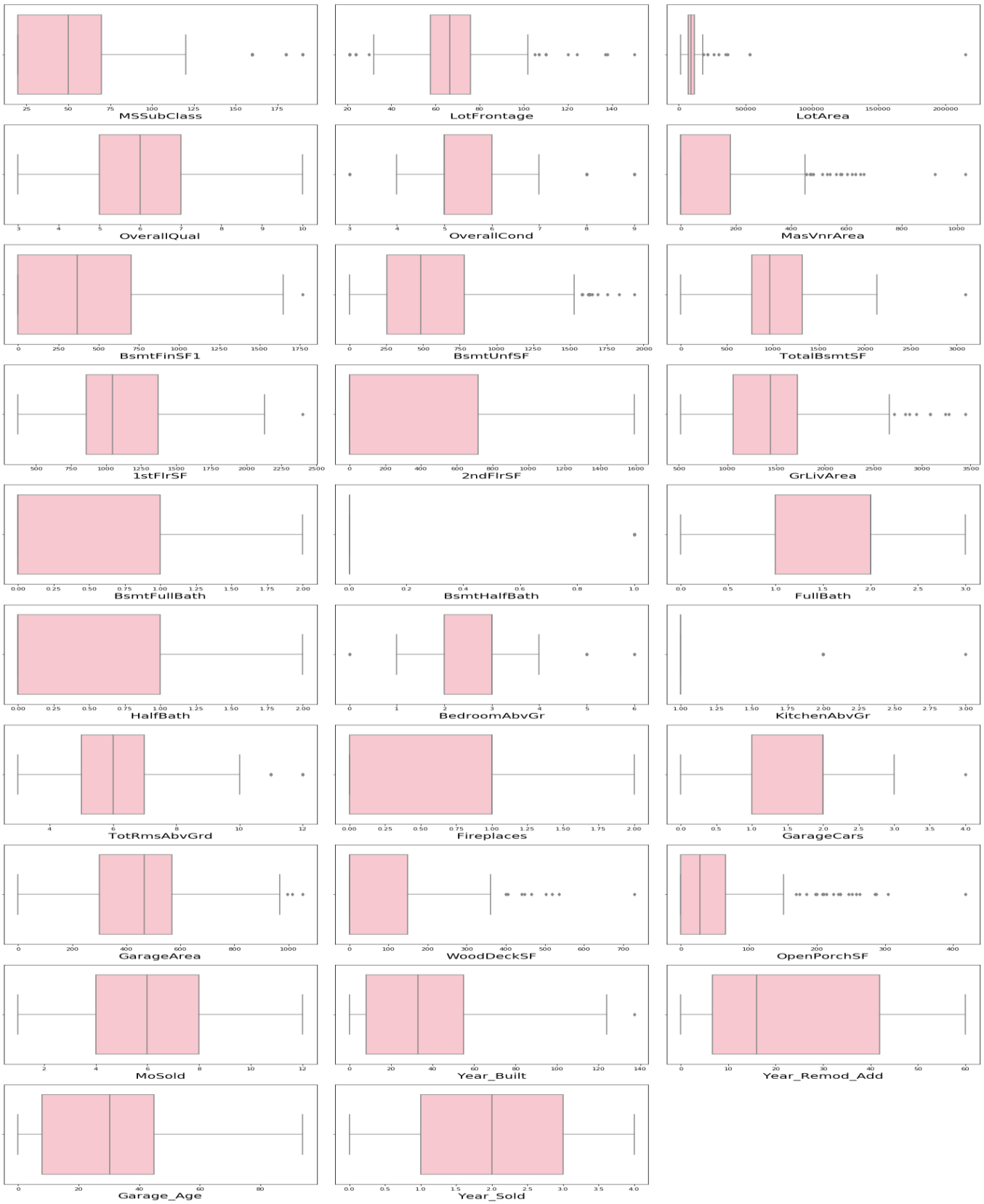
* For Typical/Average(TA) Garage condition the count is high, for the feature Garage condition(GarageCond).

Visualizing the box of each features for numerical data set in the train data to check if any outliers are present.

Outliers are present in below columns. Sales price is our target variable, so we can ignore it.

* MSSubClass
* LotFrontage
* LotArea
* OverallQual
* OverallCond
* MasVnrArea
* BsmtFinSF1
* BsmtUnfSF
* TotalBsmtSF
* 1stFlrSF
* 2ndFlrSF
* GrLivArea
* BsmtFullBath
* BsmtHalfBath
* BedroomAbvGr
* KitchenAbvGr
* TotRmsAbvGrd
* Fireplaces
* GarageCars
* GarageArea
* WoodDeckSF
* OpenPorchSF
* Year_Build
* Garage_Age
* SalePrice

Visualizing the box of each features for numerical data set in the test data set to check if any outliers are present.



Outliers are present in below Columns for test data set

* MSSubClass
* LotFrontage
* LotArea
* OverallCond
* MasVnrArea
* BsmtFinSF1
* BsmtUnfSF
* TotalBsmtSF
* 1stFlrSF
* GrLivArea
* BsmtHalfBath
* BedroomAbvGr
* KitchenAbvGr
* TotRmsAbvGrd
* GarageCars
* GarageArea
* WoodDeckSF
* OpenPorchSF
* Year_Built

- **Interpretation of the Results**

  Saving the final model and predict the Sale Price for test data

```
import pickle
```

```
file_name="house_price_prediction.pickle"
```

```
pickle.dump(final_model,open(file_name,'wb'))
```

```
loaded_model=pickle.load(open(file_name,'rb'))
loaded_model_pred=loaded_model.predict(x_test)
loaded_model_pred
```

```
array([178074.70959869, 206002.386154  , 168672.38783045, 315196.64112572,
       140715.24784453, 129596.07699918, 263226.96724117, 146916.21740122,
       314980.31946146, 135185.25451054, 155815.4717295 , 194001.60413737,
       148378.10984087, 280271.8227247 , 192181.36720497, 157726.8057918 ,
       121464.80268386, 117259.83705646, 271063.72092675, 439706.80947184,
       126634.07904198, 137186.15712701, 245995.55828063, 273805.23638646,
       186112.71032102, 196925.62934033, 191917.39202865, 141659.45294871,
       177036.31642894, 187821.62647433, 137184.95946153, 192961.07278115,
       191408.16589977, 260475.10369801, 213278.31642167, 149994.97533838,
       273811.65877487, 212831.12359211, 146969.2783476 , 135870.25814633,
       262315.23253603, 135996.21435584, 190943.03298436, 154131.30908511,
       188540.73565883, 191018.27921128, 189886.49787448, 218285.10878812,
       135539.66498052, 182658.6836234 , 167345.05196745, 123071.59637362,
       138519.46098124, 161153.76371043, 351813.13852585, 193059.15625316,
       168555.78228176, 169568.4494542 , 145722.90265643, 128497.65459636,
       139351.85309873, 140267.52868197, 143299.78973132, 146310.05264874,
       162238.24673191, 165332.902674  , 155397.6680874 , 192398.77990091,
       151951.08056717, 141514.65676336, 210267.22862791, 249930.98989133,
       155310.88121362, 209798.34725061, 170939.11237205, 169781.22535447,
        94096.52501021, 217591.50425628, 297529.35559179, 120277.89657973,
       141029.39770995, 117232.59214163, 156643.06160938, 142299.3639128 ,
       171455.06179888, 358557.53021609, 190312.37339062, 191306.44840439,
       152014.08824134, 249063.90817845,  94439.27334494, 185778.92706645,
       132765.08906694, 152001.62203943, 195195.92793606, 132598.5665095 ,
       153558.83245435, 151842.96549732, 226611.75079178, 153542.29771843,
       231310.90743516, 117379.67652785, 225131.75735383, 210512.71598274,
       158896.19787864, 230645.53675997, 244916.32712706, 158623.76057315,
       175213.57177752, 173834.25088217, 176235.27394296, 172255.46699438,
       105395.34154423, 140701.06872739, 163679.51606301, 135978.04782786,
       262424.23648655, 132546.01397891, 144221.84679767, 129205.42661591,
       259387.43041883, 190117.75646014, 153144.89906475, 238870.68872101,
```

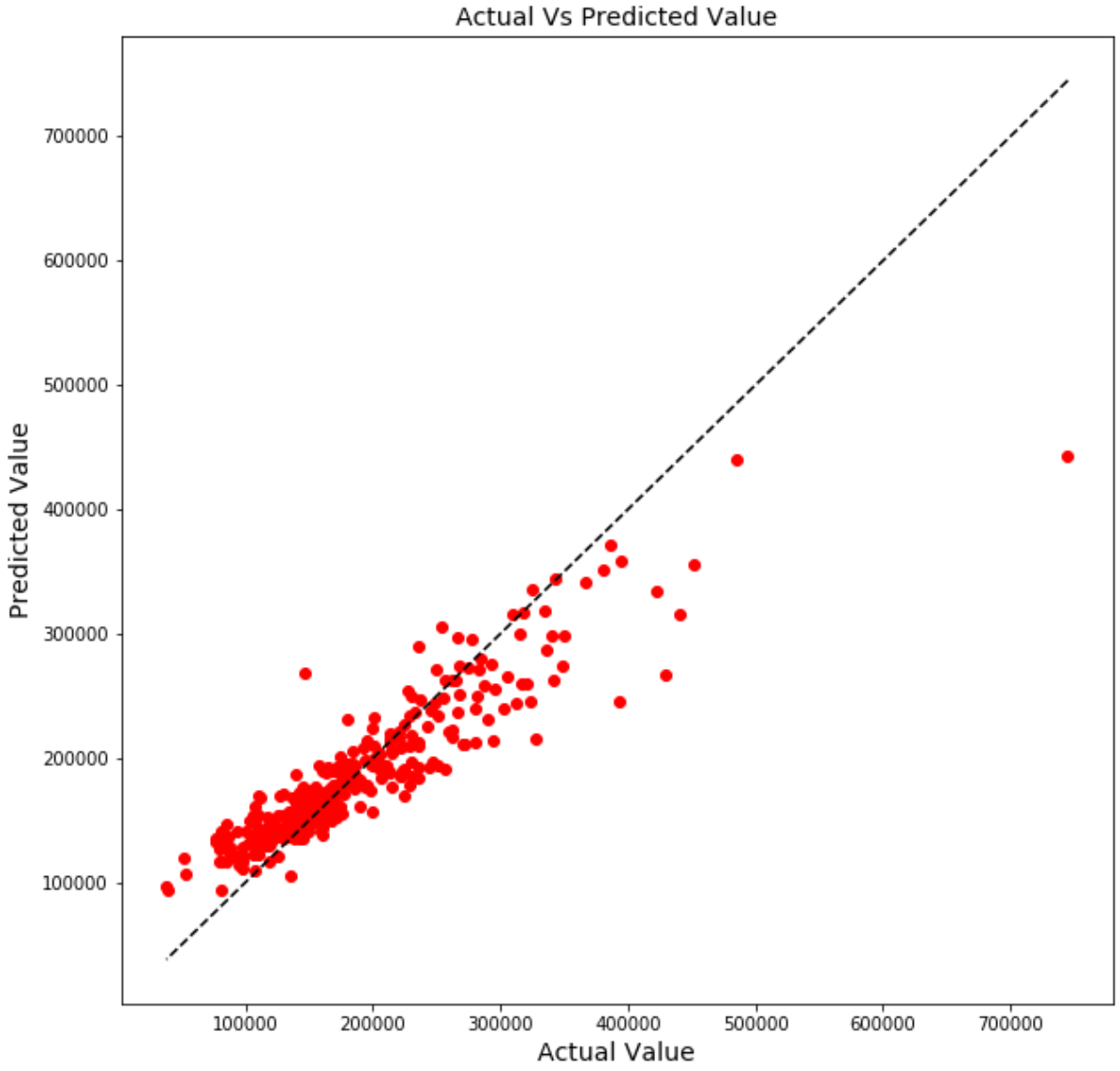
Difference between Predicted and Actual value

```
df=pd.DataFrame([loaded_model.predict(x_test)[:],y_test[:]],index=['Predicted','Acutal'])
df.T
```

	Predicted	Acutal
0	178074.709599	169990.0
1	206002.386154	184000.0
2	168672.387830	158000.0
3	315196.641126	440000.0
4	140715.247845	132000.0
5	129596.076999	118500.0
6	263226.967241	341000.0
7	146916.217401	139500.0
8	314980.319461	310000.0
9	135185.254511	130000.0
10	155815.471730	175000.0
11	194001.604137	176000.0
12	148378.109841	153900.0
13	280271.822725	285000.0
14	192181.367205	244000.0
15	157726.805792	142000.0
16	121464.802684	93000.0
17	117259.837056	80000.0
18	271063.720927	250000.0
19	439706.809472	485000.0
20	126634.079042	85000.0
21	137186.157127	119200.0
22	245995.558281	324000.0
23	273805.236386	268000.0
24	186112.710321	230000.0
25	196925.629340	230000.0
26	191917.392029	235000.0
27	141659.452949	133500.0
28	177036.316429	145000.0
29	187821.626474	176500.0
30	137184.959462	87000.0
31	192961.072781	176485.0
32	191408.165900	210000.0
33	260475.103698	317000.0
34	213278.316422	280000.0
35	149994.975338	103600.0
36	273811.658775	348000.0

Plotting the scatter plot for predicted and actual value

```
plt.figure(figsize=(10,10))
plt.scatter(y_test,loaded_model_pred,color='red')
p1=max(max(loaded_model_pred),max(y_test))
p2=min(min(loaded_model_pred),min(y_test))
plt.plot([p1,p2],[p1,p2],color='black',linestyle='--')
plt.xlabel("Actual Value",fontsize=14)
plt.ylabel("Predicted Value",fontsize=14)
plt.title("Actual Vs Predicted Value",fontsize=14)
plt.show()
```



I have predicted the sale price for test data using the best model Gradient Descent Boosting and saved the model using pickle and in the csv format.

- The Housing project have the two data set train and test data set. We have done the analysis for both the data set separately.
- We see that there is lots of null value present in data, we have carefully replaced them with the statistical method.
- We saw in most of the columns have the outliers and skewness. We have removed them using the appropriate method and make sure we don’t loss huge data while cleaning the data.
- We have analysed the data its uniqueness, statistical summary, info of data and its shape.
- In dataset it contains both the categorical and numerical data we have converted the categorical data to numerical data using the Label encoder.
- We have converted the year column to age for better prediction.
- We have scaled the data and removed the high variation inflation factor to avoid multicollinearity.
- Then we have built plot various visualization plot like regression plot, scatterplot, strip plot, distribution plot. Box plot.
- We checked the correlation of data between the feature and the target variables.
- We have used various models in our project and checked the r2 score. We analysis each model and choose the best model which is giving the good score.
- We have done the cross-validation score of various models to avoid over fitness issue.
- We have done the hyper parameter tuning to increase the accuracy and we have used the Metrix like mean squared error and mean absolute error.
- Then finally we have chosen our final model after analysis and predicted the value of SalePrice and saved the model for future interpretation.
- We have plotted the scatterplot for predicted and actual value.

CONCLUSION

- ## Key Findings and Conclusions of the Study

  In this project we have used the various model and analyse the process steps by steps. We have done the cleaning of data and feature engineering to make better prediction. We have the two data set for training and testing. We have analysed both the data set separately. We have built various models and choose the best model which gives the high score as compare the other model. We have done the cross validation of each model. Then we have finally predicted the Sale Price for test data and saved the file in csv format.

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

  It is an interesting data set as it contains two different data set for training and testing. Data cleaning and Feature engineering plays an important role in Data science project. As it helps us to remove the null value and replace with mean, mode, median. It helps to remove the outliers, skewness, variation inflation factor before building the model. We use various plot for graphical representation of data. It helps the other to understand what data is trying to say. If the person is non-technical back ground. Through visualization he can able to understand the insight of data.

  Through this analysis and prediction of SalePrice it will help the real estate agent to make prediction for buying and selling house. Understand which features is impacting the sale price of house. So, that we can improve on that particular area to get it better. Data Science plays important role in every sector it helps to predict what will happen in the future by analysis of past data. It helps many IT Sector to grow much faster than before and predicted the future of tomorrow to make it better.

- **Limitations of this work and Scope for Future Work**
  - ✓ The limitation of this project is if we merge both the data set of training and testing. There is a chance of data leakage.
  - ✓ During the process of removing the outliers and skewness, there is an impact in model accuracy.