HOUSING PRICE PREDICTION

BY-Neetu Bam

Batch No-DS0622

CONTENT

*Over View
*Problem statement
*Problem Understanding
*Exploratory Data Analysis
*Visualization
*Steps and Assumptions
*Model Building
*Variation Inflation Factor
*Cross Validation Score
*Hyper Parameter Tuning
*Conclusion

Import necessary Liabraries

In [1]:
```
# 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')
```

Import Train And Test Data

```
# Importing the train data set

df_train=pd.read_csv('house_train.csv')
df_train.head(5)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condi
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	

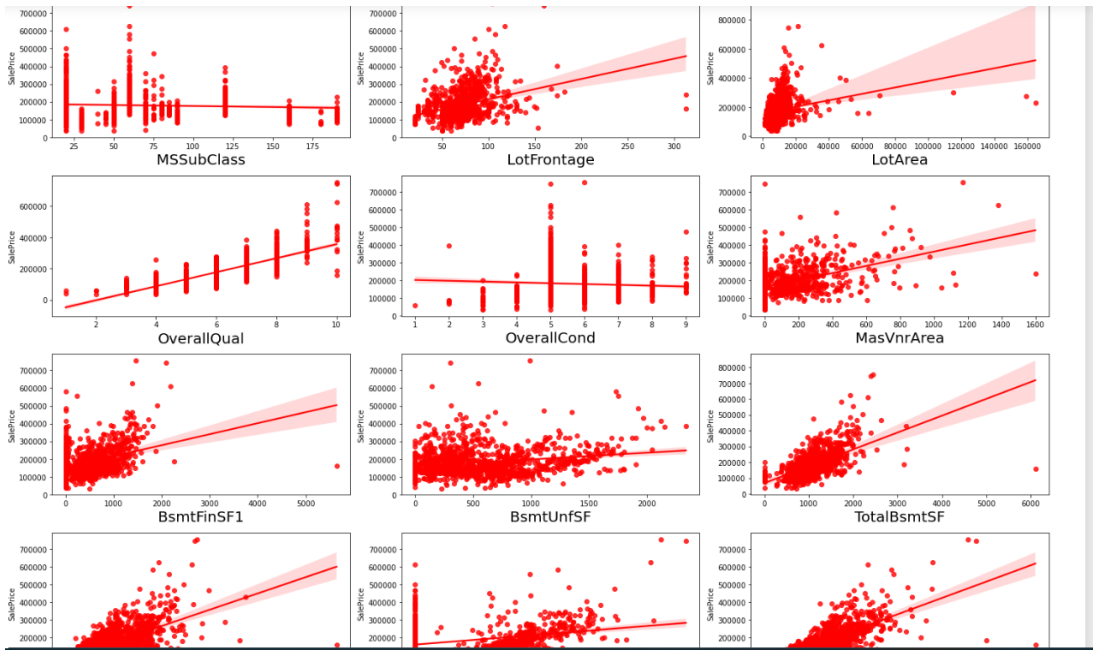
```
# Importing the test data set
df_test=pd.read_csv('house_test.csv')
df_test.head(5)
```

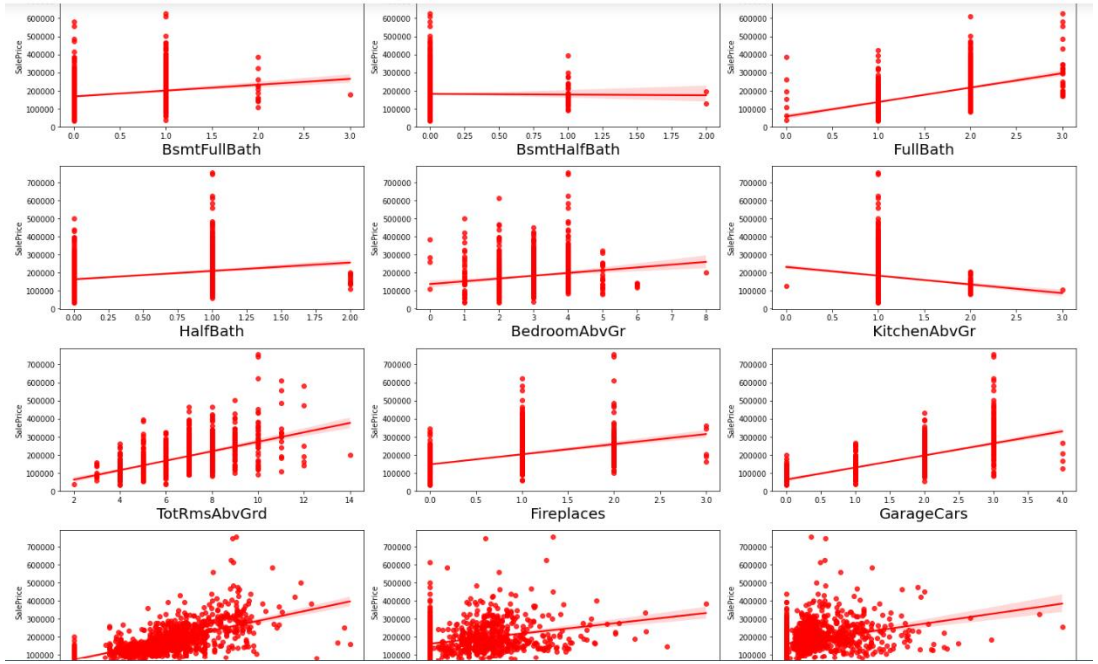
	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Cond
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	Corner	Gtl	StoneBr	Norm	
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	StoneBr	Norm	
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	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	

EDA

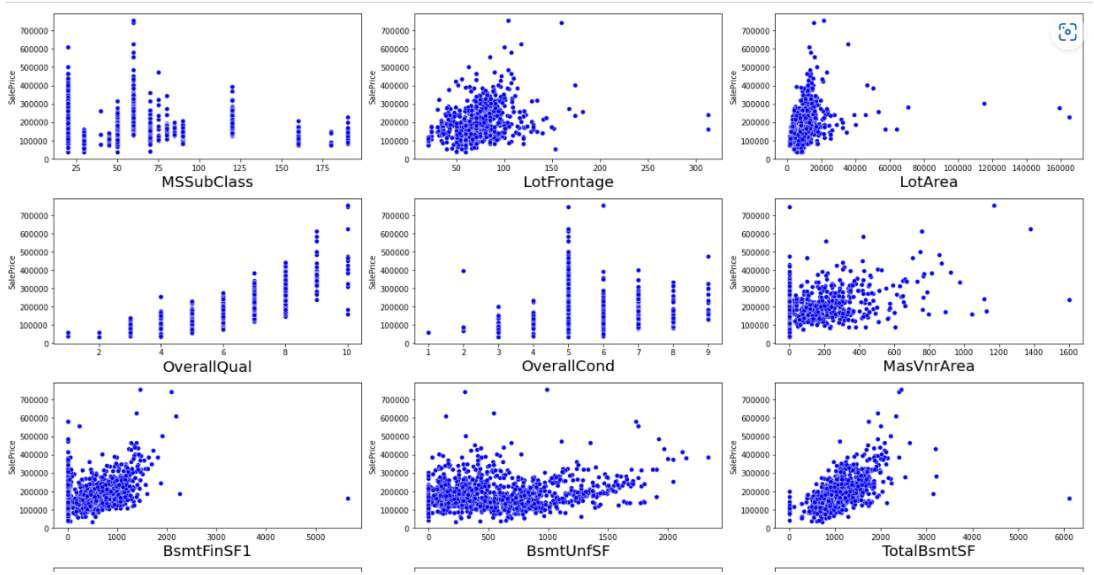
First I check Data shape,column,describe,Null Data,Unique Data,Heat map.After that I do Data Visualization

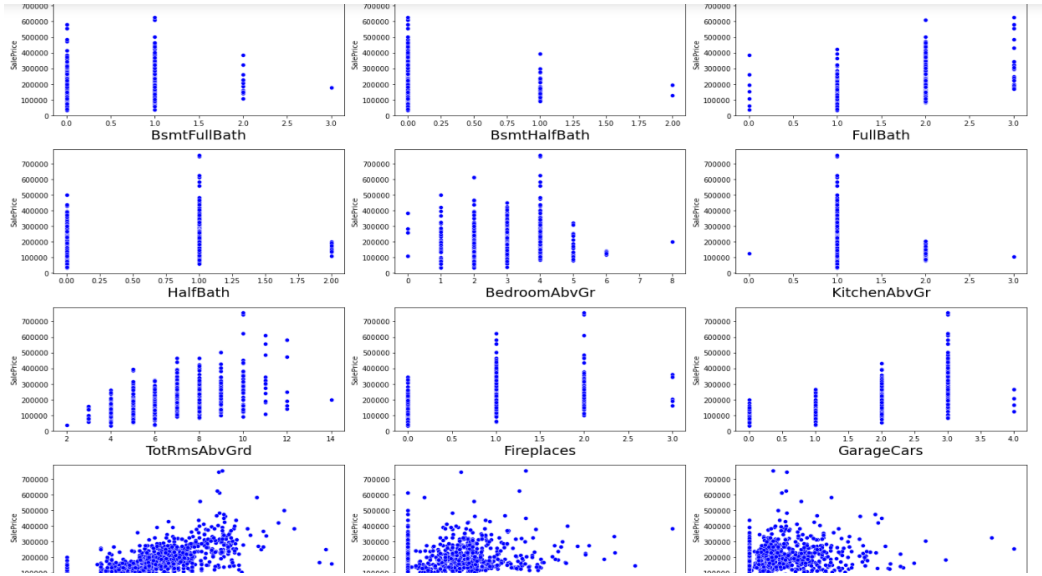
Visualization of Regression Plot



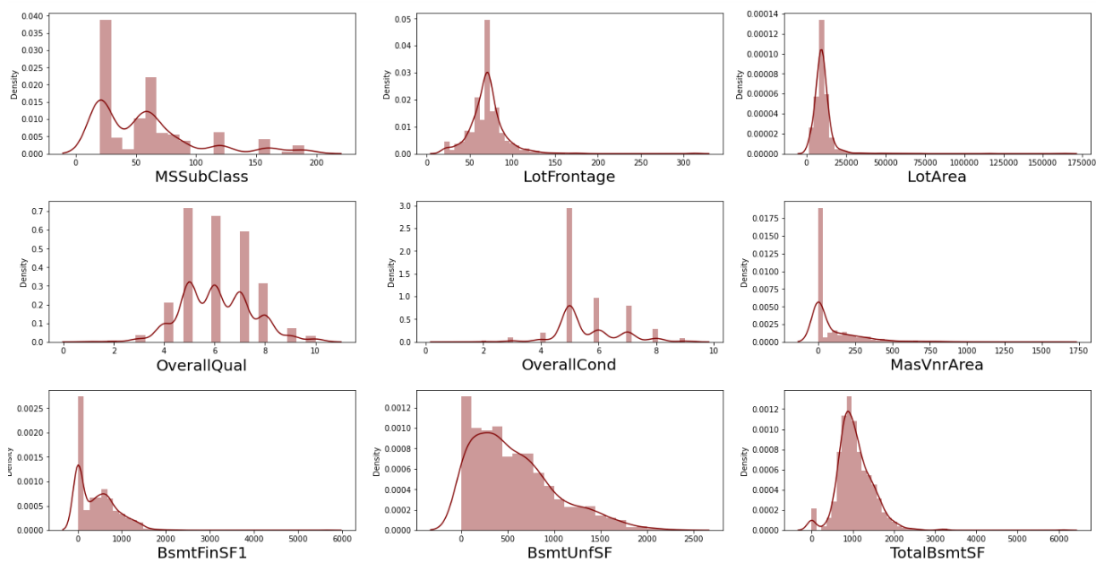


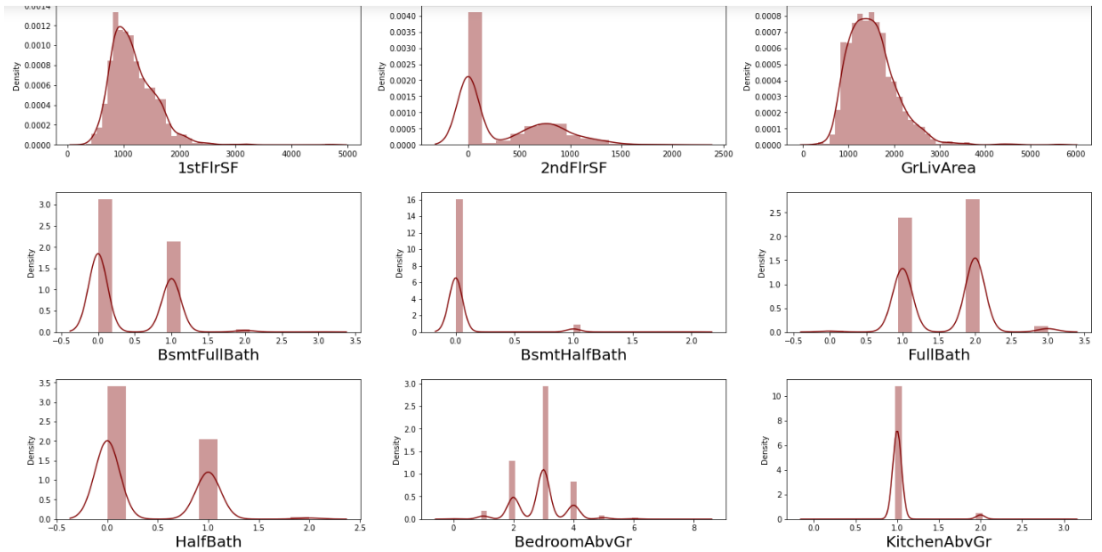
Visualization of Scatterplot for Numerical Data



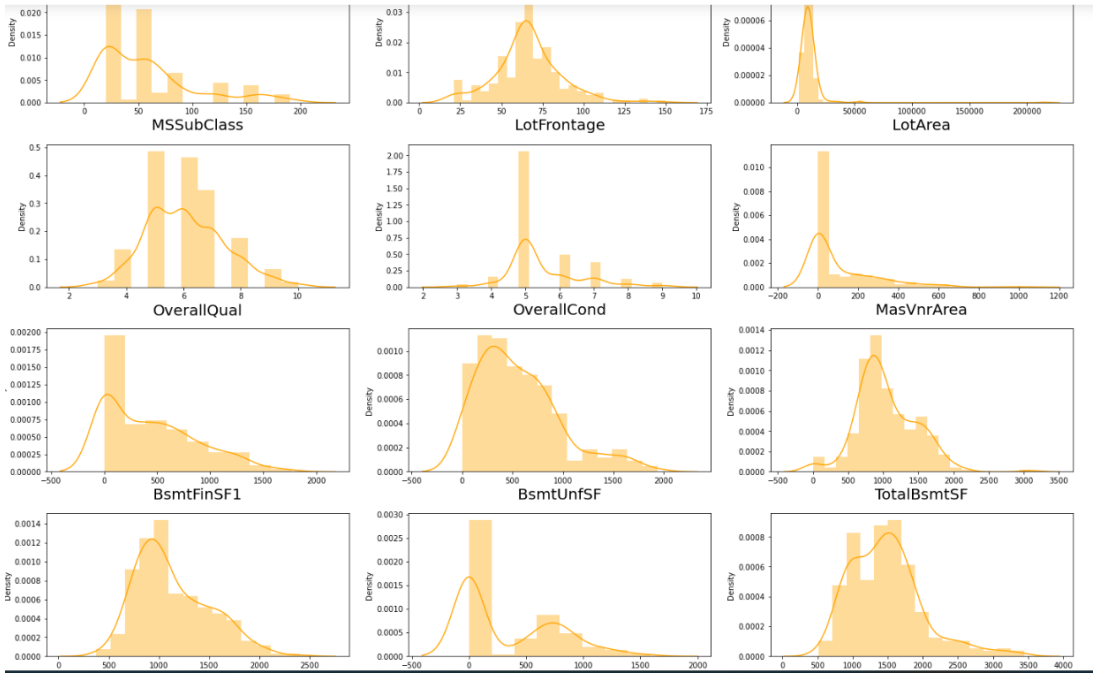


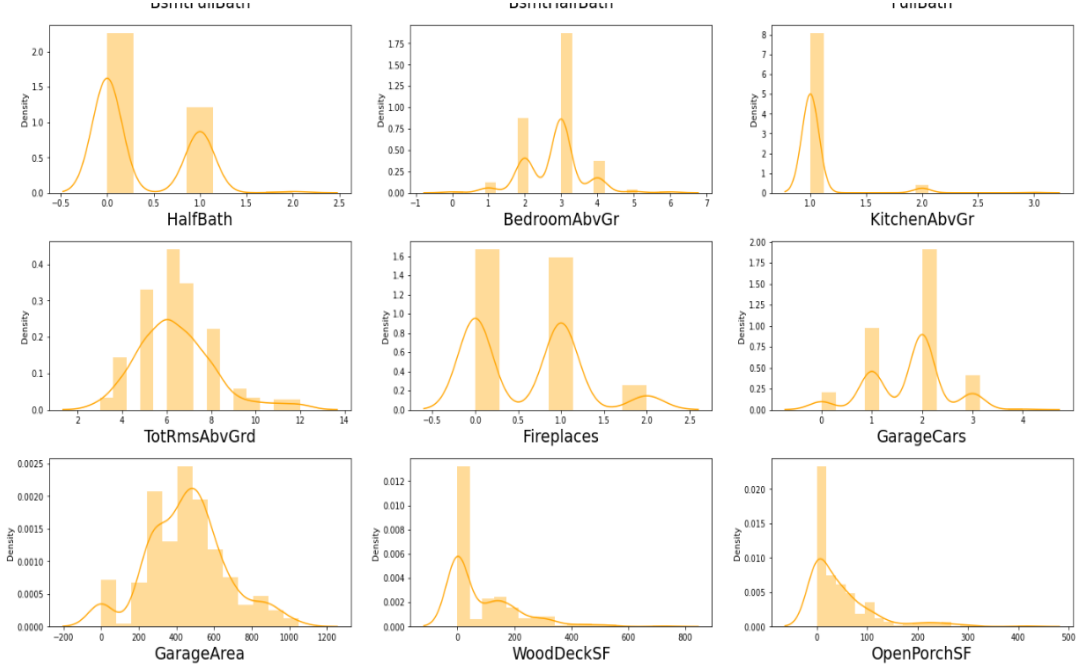
Distribution Plot of Numerical data for Train Data



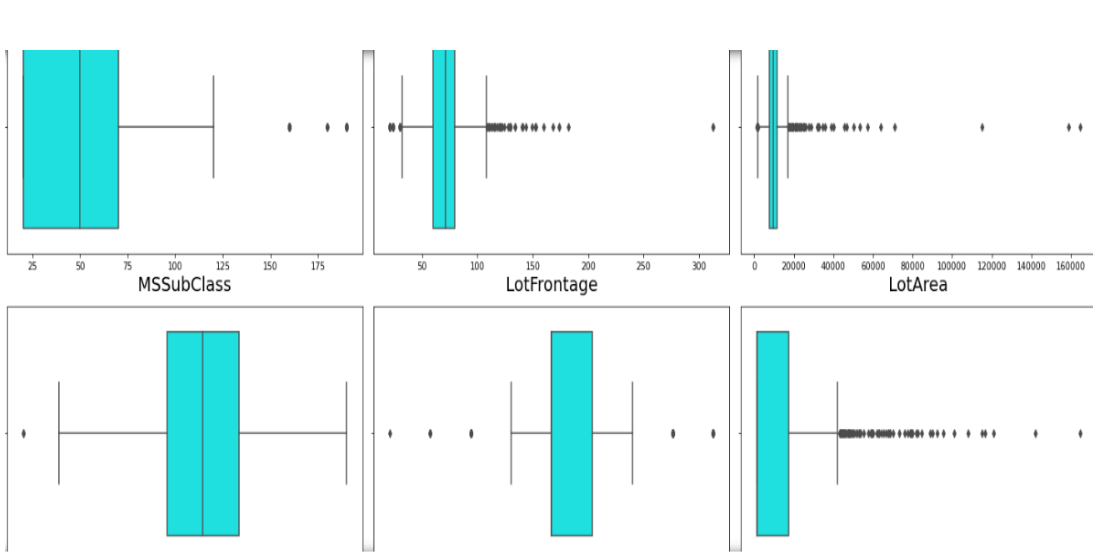


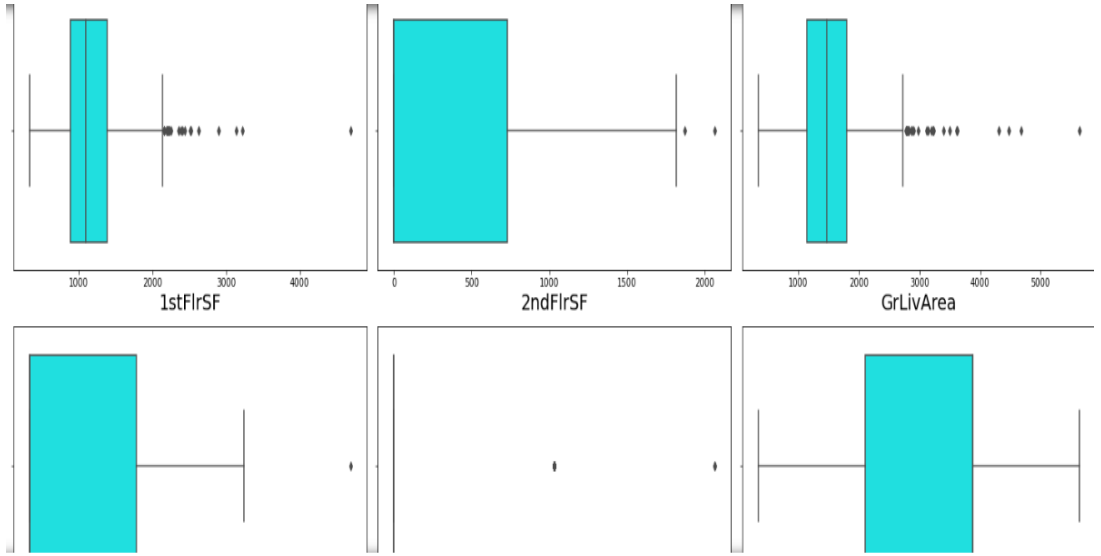
Distribution Plot of Numerical data for Test Data



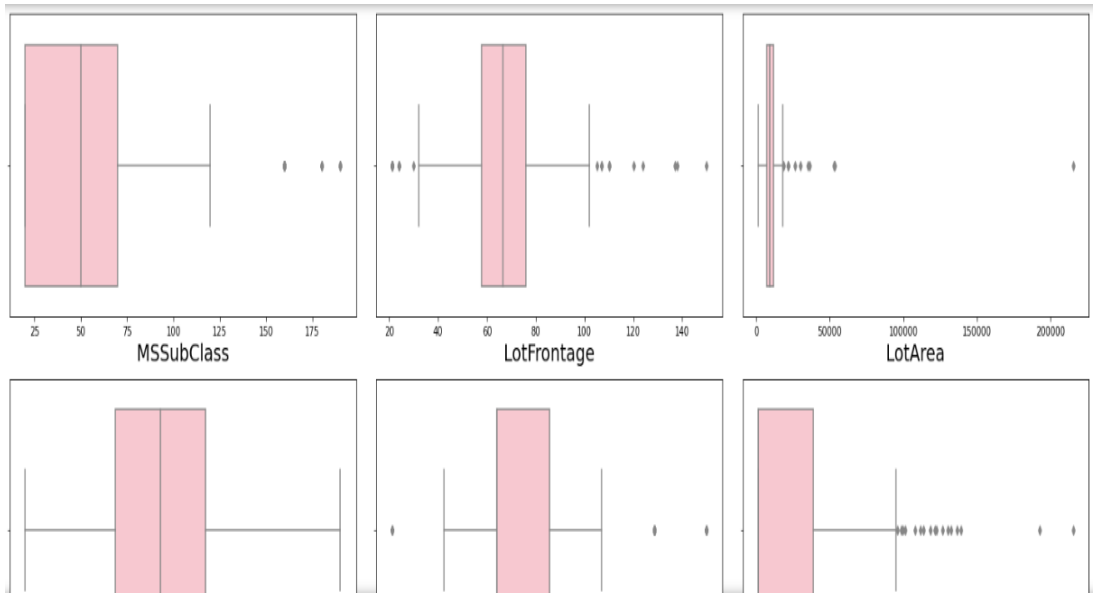


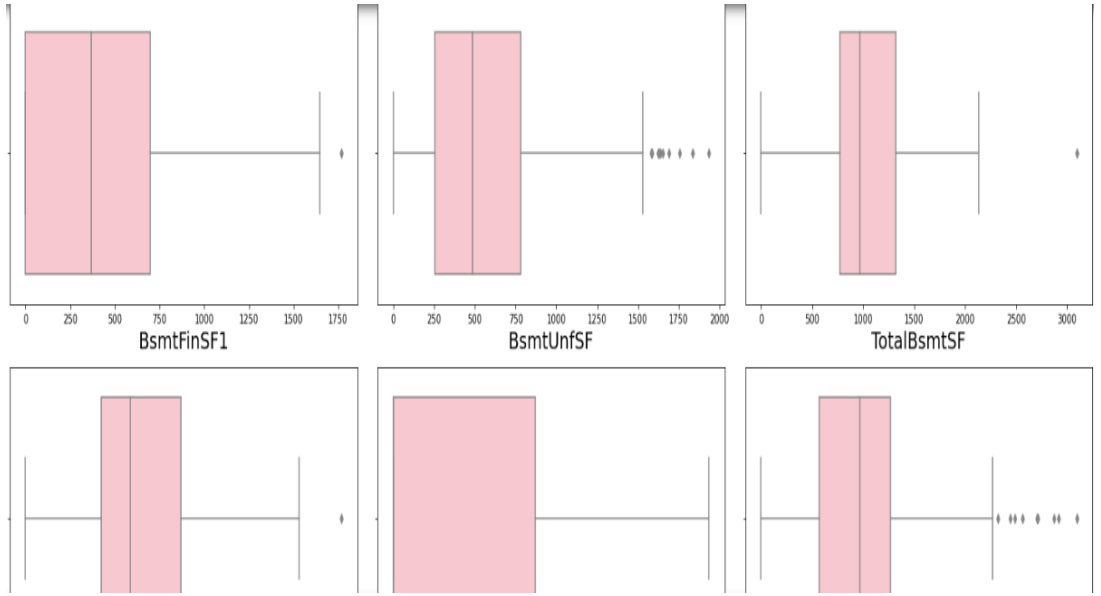
Outliers in Train Data



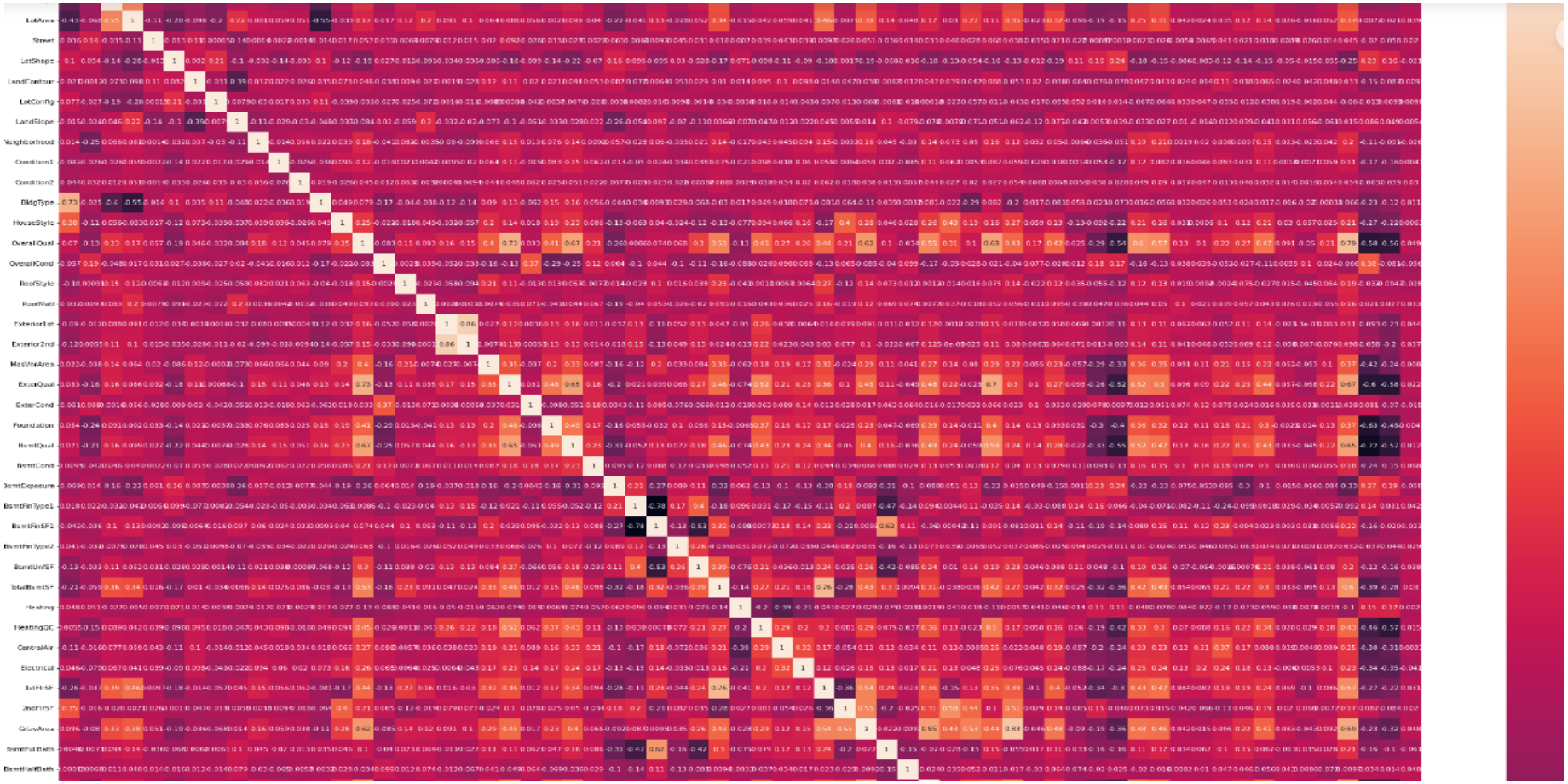


Outliers in Test Data

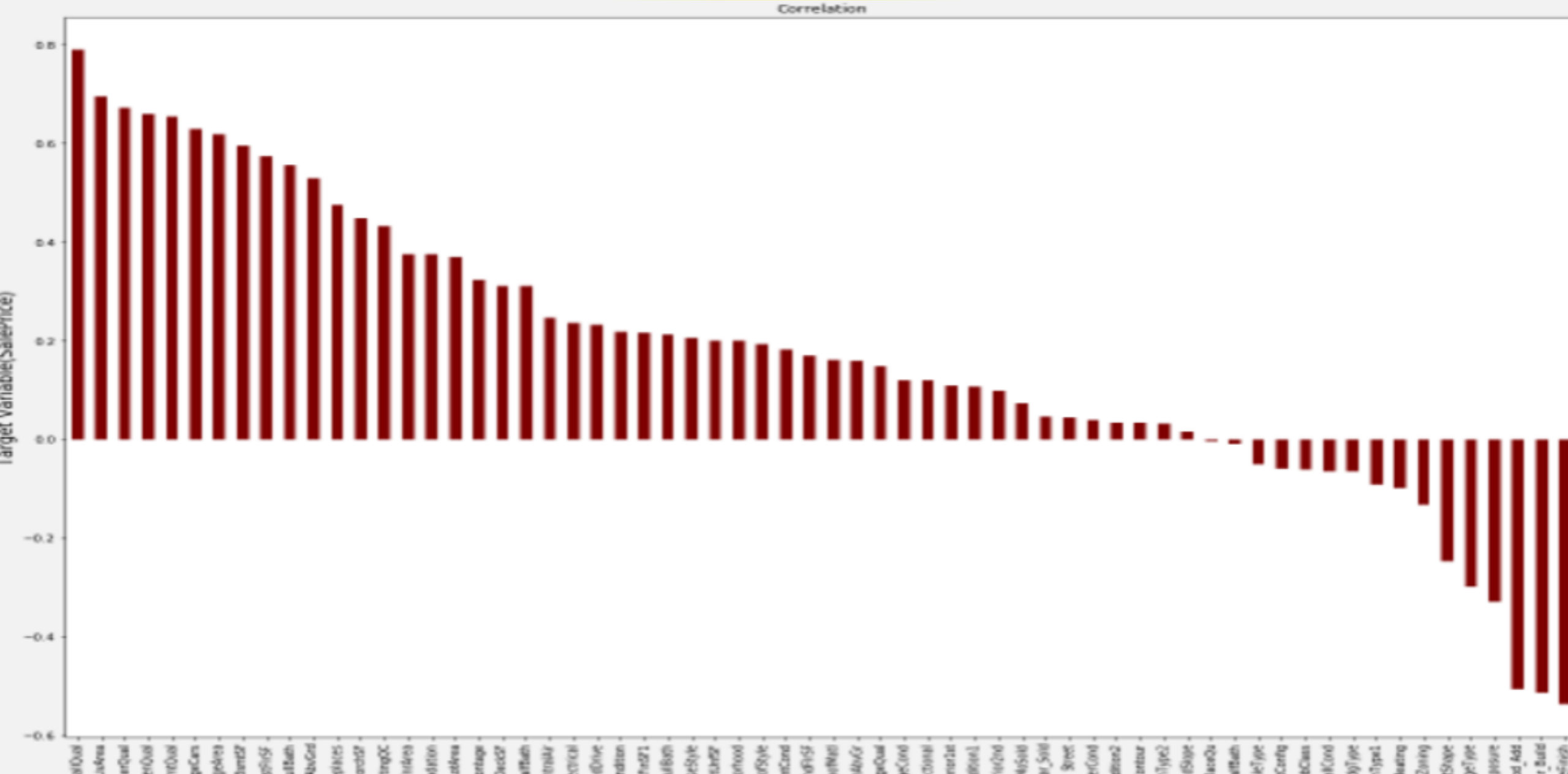




Correlation of Train Data



Correlation of Features and Target variables



Data cleaning

- We use imputation technique to replace the null value.
- To remove outliers we use the Zscore method.
- To remove skewness we use preprocessing technique using Power Transformer ‘yeo-johnson’ method.
- We converted the categorical data to numerical data using Label Encoder.
- We use the correlation technique to check the correlation of each features.
- We use Standard Scaler to scale the data.
- Used various model and check the cross validation score to make the better prediction.

Model Building:

- Our target variable is Sale Price and it is continuous columns. This is a regression problem. We have to use the regression model to make the prediction.

- We have used various Regression model and checked the r2 score, mean squared error and mean absolute error.

- We checked the cross validation score of various model.

- These are some of the model used in House price prediction Project

- Linear Regression
- Random Forest Regressor
- KNeighborsRegressor
- Decision Tree Regressor
- Gradient Boosting Regressor
- Ada Boost Regressor
- Support Vector Regressor
- Bagging Regressor
- SGBRegressor

LINEAR REGRESSION

```
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
```

Linear Regression Model gives 78% accuracy

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
```

Random Forest Regressor gives 86% accuracy

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
```

Kneighbors Regressor gives 77% accuracy

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
```

Gradient Boosting Regressor gives 86% accuracy

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
```

Ada Boost Regressor gives 78% accuracy

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
```

Decision Tree Regressor gives 72% accuracy

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
```

Support Vector Regressor gives 80% accuracy

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
```

This is the cross validation score for various model. We see that Random Forest Regressor gives a good cross validation score 84%.

SAVING THE MODEL

```
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
```

We use the Pickle to save our model. Filename as house_price_prediction. We load the data in the read mode.

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

CONCLUSION

- In this project we have used various machine learning model and cross validation score to make the prediction.
- We have checked the r2 score, mean squared error and mean absolute error of data.
- We have done the feature engineering, exploratory data analysis and cleaning of data.
- We analyse the each features and check the correlation of data. Removed the skewness and Outliers using the appropriate technique to give the good accuracy.
- We use the hyper parameter tuning to increase the accuracy.
- We use the variance inflation factor to remove the multi-collinearity.
- Based on our prediction, it will provide a good insight to make the prediction of house price.