House Price prediction

A Step-by-Step guide to Build a Machine Learning Model to predict House Prices

Submitted By:

Razni Nazeem

24.06.2022

FlipROBO Technologies

ACKNOWLEDGEMENT

I would like to express my special gratitude to the "Flip Robo" team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzing skills. Also, I want to express my huge gratitude to Ms. Khushboo Garg (SME FlipRobo), who has helped me overcome difficulties within this project and others.

I would also like to thank various websites like stackoverflow, Kaggle, medium and towardsDatascience for helping me resolve any issues I face during my project.

A huge thanks to my academic team "Data trained" who has helped me grow from a non-Coder to what I am Now. Lastly, I would like to extend my Heartfelt thanks to my Husband and kids because without their support this project would not have been successful. And thank you to many other persons who have helped me directly or indirectly to complete the project.

INTRODUCTION

Business Problem Framing:

In this Article, I will be guiding you to the step-by-step procedure in building a Machine Learning model in Python using popular machine learning libraries NumPy, Pandas & scikit-learn to predict House Prices in the US.

Low inventory, fervid competition and massive price gains have battered buyers since 2020, but now rapidly rising mortgage rates are making it even harder to purchase an affordable home. Housing price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. Therefore, it is important to predict housing prices without bias to help both the buyers and sellers make their decisions. This project uses an open-source dataset, which include 81 explanatory features and 1,168 entries of housing sales in AMES, USA. The aim is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analyzing previous market trends and price ranges, and upcoming developments future prices will be predicted.

Conceptual Background of the Domain Problem:

To help machines understand like humans do and to strengthen AI, machine Learning is required. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, improving their marketing strategies and focusing on changing trends in house sales and purchases.

There are 2 datasets in the link, Train and Test Dataset.

- Size of training set: 10683 records
- Size of test set: 2671 records

Using the Training set, I have built the model and predicted the House Prices in the Test dataset.

Why is house price prediction important?

House Price prediction is important to drive Real Estate efficiency. As earlier, House prices were determined by calculating the acquiring and selling price in a locality. Therefore, the House Price prediction model is very essential in filling the information gap and improving Real Estate efficiency.

Review of Literature

It is inferred that establishing a simple linear mathematical relationship for these time-series data is found not viable for forecasting. Hence it became imperative to establish a non-linear model which can well fit the data characteristic to analyze and forecast future trends. As the real estate sector is a fast-developing sector, the analysis and forecast of land prices using mathematical modeling and other scientific techniques is an immediate urgent need for decision making by all those concerned.

Therefore, in this project report, we present various important features to use while predicting housing prices with good accuracy. We can use regression models, using various features to have lower Residual Sum of Squares error. While using features in a regression model some feature engineering is required for better prediction.

The primary aim of this report is to use these Machine Learning Techniques and curate them into ML models which can then serve the users. The main objective of a Buyer is to acquire a house which fulfills their dreams as well as within their budget. Similarly, a Seller wanted to sell the house and make a good profit out of it.

Motivation for the Problem Undertaken:

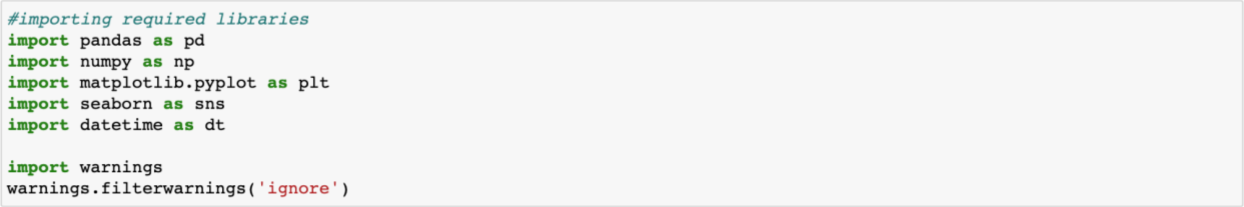
I must model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market. The relationship between house prices and the economy is an important motivating factor for predicting house prices.

LIBRARIES IMPORTED

There are 3 sets of libraries used.

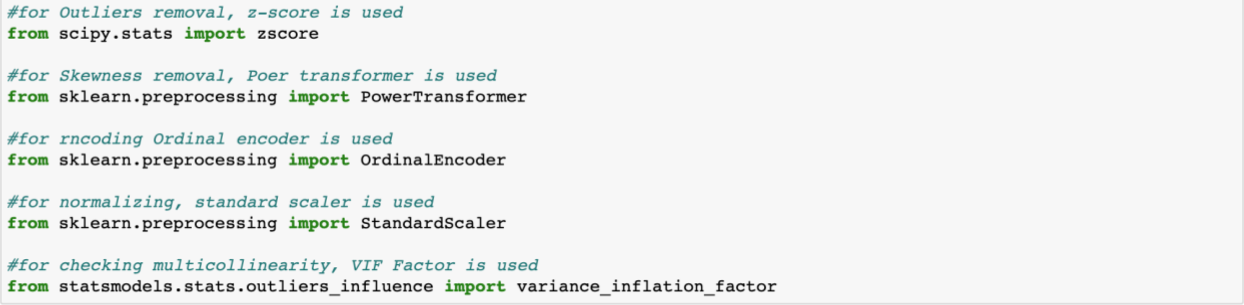
- Basic Libraries for Data Analysis and Visualization
- Libraries for Data Cleaning and Feature Engineering (Data preprocessing)
- Libraries for Building the ML Models

Basic Libraries used are:



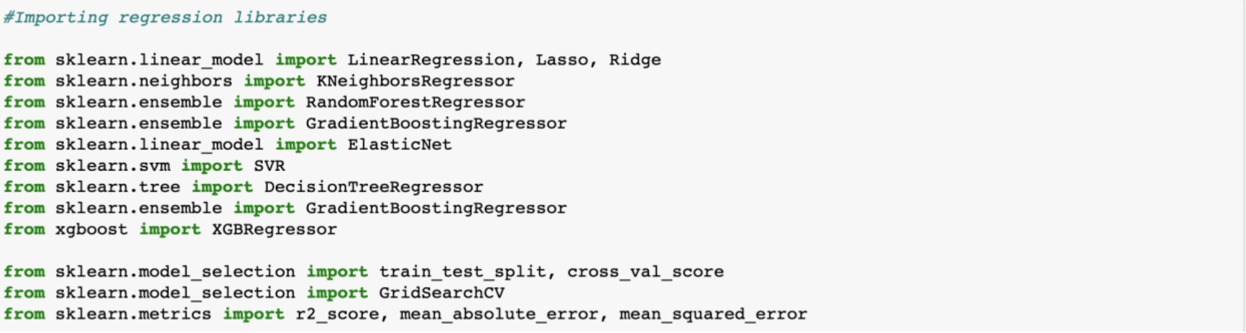
```
#importing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt

import warnings
warnings.filterwarnings('ignore')
```

Libraries for Data Cleaning and Feature Engineering (Data preprocessing):



```
#for Outliers removal, z-score is used
from scipy.stats import zscore

#for Skewness removal, Poer transformer is used
from sklearn.preprocessing import PowerTransformer

#for rncoding Ordinal encoder is used
from sklearn.preprocessing import OrdinalEncoder

#for normalizing, standard scaler is used
from sklearn.preprocessing import StandardScaler

#for checking multicollinearity, VIF Factor is used
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Libraries for Building the ML Models:



```
#Importing regression libraries

from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

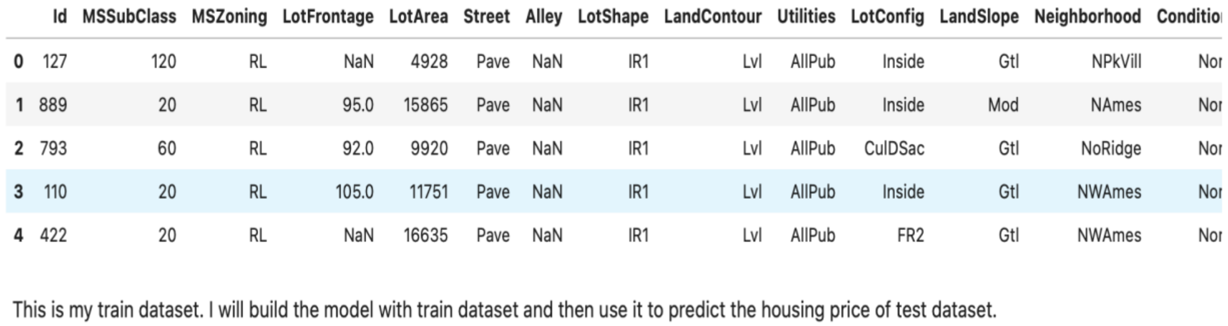
These are the libraries used in my Jupyter notebook for Prediction.

Analytical Problem Framing

1. Mathematical/Analytical Modeling of the Problem

After loading the train and test datasets, I checked the first 5 elements in both sets.

- Train Dataset:



	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Conditio
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NPkVill	Nor
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	Inside	Mod	NAmes	Nor
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	NoRidge	Nor
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NWAmes	Nor
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NWAmes	Nor

This is my train dataset. I will build the model with train dataset and then use it to predict the housing price of test dataset.

We have 1,168 rows and 81 columns including our Target "SalePrice" for the train dataset and 292 rows and 80 columns for the Test dataset.

By looking into the Target column "SalePrice" which is continuous, I came to know that it is a Regression problem. I analyzed each column's information and unique values. I saw a huge number of missing values in certain columns and more than 80% of the data in certain columns as 0. So, I dropped those columns to avoid high bias and variance. I used an Imputation technique to replace NaN values in the train and test dataset. Further I have extracted some meaningful columns from the columns available to us. Afterwards, I have analyzed the dataset using plots and other visualization techniques like barplot, distplot etc. I also plotted features with the Target. Then I have removed Outliers, skewness and found correlation between variables and removed multicollinearity, and normalized the dataset using Standard scaler. I have created 5 models and used hyperparameter tuning to find the best score. Regplot has been plotted for each model for their actual and predicted values. Cross Validation score has also been computed and then found the best model and saved for future predictions. Using the saved model, I have predicted house prices of the Test dataset and stored it in a csv file. Using my model, anyone can find the house prices provided the features are similar as in my model.

2. Data Sources and their formats:

Data was collected from my Internship Company FlipRobo Technologies. Train and Test datasets have been in csv format. Another text file which has the necessary information regarding the variables and their values have been included. This really helped me analyze and understand the dataset furthermore. My train dataset was having 1168 rows and 81 columns including target, and my test dataset was having 292 rows and 80 columns excluding target. I have object, float and integer types of data.

3. Data Processing done

* Imported necessary Libraries and loaded the dataset.

* Statistical Analysis done like Shape, info, nunique, value_counts.

* Dropped columns with more than 80% of NaN values and values **"0"**

***** Imputation Technique to replace other NaN values.

* Dropped columns- ID & Utilities which had all unique values and only 1 value respectively.

* Extracted Age from Year mentioned in the dataset.

* All the steps done for both Train and Test dataset.

4. Data Inputs- Logic- Output Relationships

* The relation between Categorical columns with Target has been found using Boxenplot.

* The relation between Numerical columns with Target has been found by using Scatterplot, Swarmplot, Stripplot.

* Various columns have a linear relationship with the Target. While Some columns do not have any specific Pattern.

5. Hardware and Software Requirements and Tools Used

* Hardware: **Processor**- Core i5 and above, **RAM**- 8GB or above, **SSD**- 250 or above

* Software: Anaconda

* Libraries Used mentioned before.

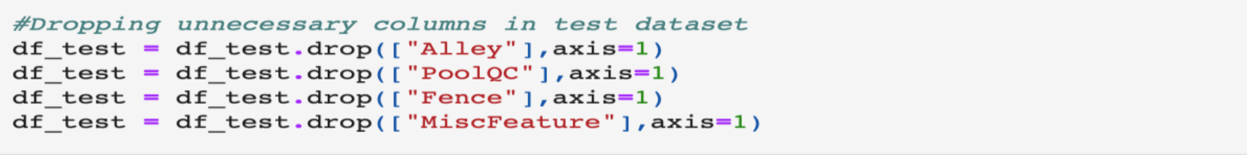
MODEL DEVELOPMENT AND EVALUATION

1. Identification of possible problem-solving approaches

* I have checked the information of the dataset regarding null values and datatypes.

I have 3 float types, 35 INT type and 43 Object type data. I then checked the missing data.

There is a lot of missing data. I have dropped those columns with more than 80% missing values.



```
#Dropping unnecessary columns in test dataset
df_test = df_test.drop(["Alley"],axis=1)
df_test = df_test.drop(["PoolQC"],axis=1)
df_test = df_test.drop(["Fence"],axis=1)
df_test = df_test.drop(["MiscFeature"],axis=1)
```

I used an imputation technique to replace NaN values.

I used the Percentile method to remove Outliers.

I used a Power transformer to remove Skewness.

I used Ordinal Encoder to convert categorical columns to numerical.

I used Pearson Correlation to find the correlation between variables.

I used Standard Scaler to scale and normalize the data.

I used various Machine Learning algorithms to create models to predict House Price.

2. Testing of Identified Approaches (Algorithms)

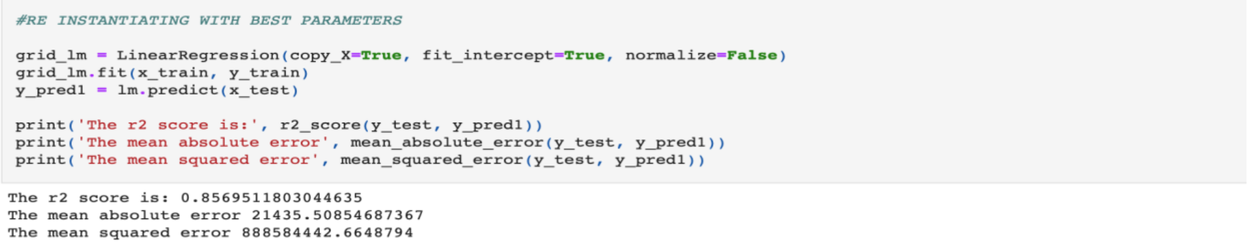
Since our Target is SalePrice which is continuous, I have a Regression Problem. I have used 5 different algorithms to build the models and found the R2 score and CV Score of each one of them. I have finally decided to select the model which has the least difference between r2 and CV Score and that model is Random Forest Regressor model.

I have Used: Linear Regression, KNN Regressor, Random Forest Regressor, XGB Regressor and Gradient Boosting Regressor.

3. Run and evaluate selected models

A. Linear Regression:

At first, I found the best Random state for which I got the best score and performed a train-test-split to fit the model. My score for Linear regression model is 85.69% and CV Score of 79.97%. I have tuned with the best parameters, but score remained the same.

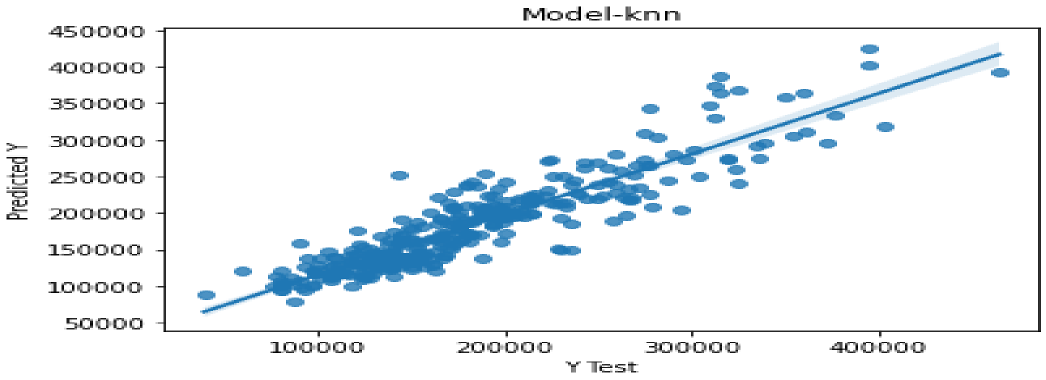


```
#RE INSTANTIATING WITH BEST PARAMETERS

grid_lm = LinearRegression(copy_X=True, fit_intercept=True, normalize=False)
grid_lm.fit(x_train, y_train)
y_pred1 = lm.predict(x_test)

print('The r2 score is:', r2_score(y_test, y_pred1))
print('The mean absolute error', mean_absolute_error(y_test, y_pred1))
print('The mean squared error', mean_squared_error(y_test, y_pred1))
```

```
The r2 score is: 0.8569511803044635
The mean absolute error 21435.50854687367
The mean squared error 888584442.6648794
```

B. KNN Regressor:

I found the best random state which yields the best score and then fit the model. R2 score for KNN Regressor was 82.75% and CV Score of 73.43%. I have tuned with different parameters, but the score has not improved. The regplot of actual and predicted values using KNN is:



C. XGB Regressor:

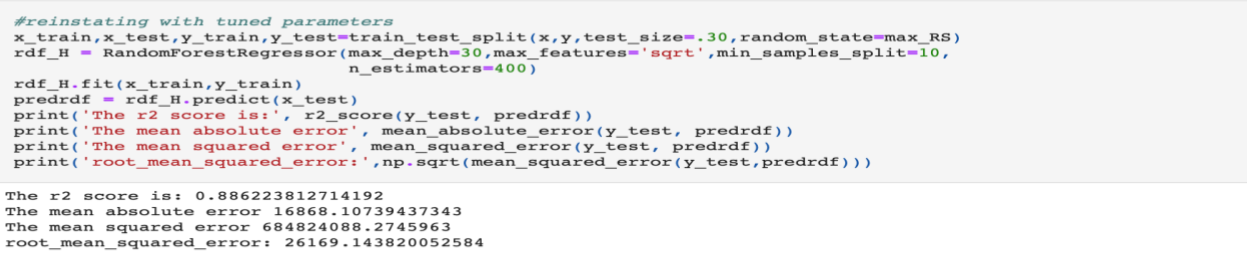
The R2 score of XGB Regressor was 88.92% and CV Score of 83.30%. Score did not improve after Hyper parameter tuning.

D. Gradient Boosting Regressor:

The r2 score of Gradient boosting was the highest among all other models which is 90.78%. But the CV score was comparatively low, and the difference is more than my best model. CV Score of GBR is 83.14%.

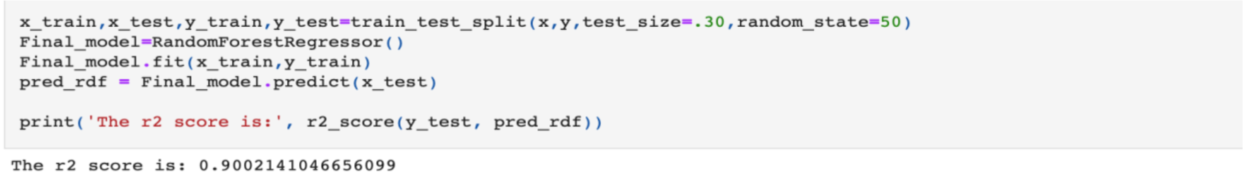
E. **Random Forest Regressor (Best Model):**

I first found the best random state and fit the model. I got a score of 89.89% and a CV Score of 83.92%. Hence, I selected the random forest as the best model and saved the model. While performing the fitting of the final model, my score was improved, and it became **90.02%.** Hyper parameter tuning of the model did not increase my score.



```
#reinstating with tuned parameters
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=max_RS)
rdf_H = RandomForestRegressor(max_depth=30,max_features='sqrt',min_samples_split=10,
                              n_estimators=400)
rdf_H.fit(x_train,y_train)
predrdf = rdf_H.predict(x_test)
print('The r2 score is:', r2_score(y_test, predrdf))
print('The mean absolute error', mean_absolute_error(y_test, predrdf))
print('The mean squared error', mean_squared_error(y_test, predrdf))
print('root_mean_squared_error:',np.sqrt(mean_squared_error(y_test,predrdf)))
```

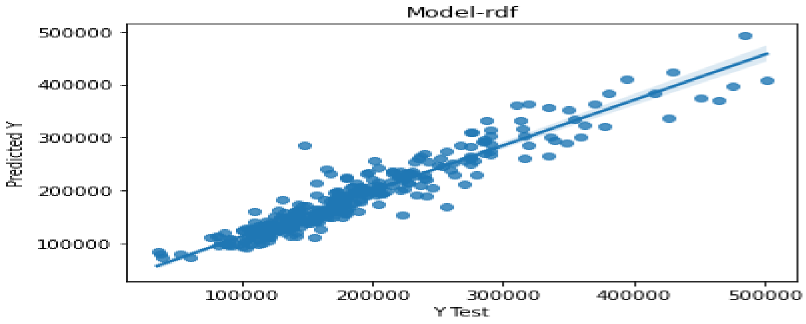
```
The r2 score is: 0.886223812714192
The mean absolute error 16868.10739437343
The mean squared error 684824088.2745963
root_mean_squared_error: 26169.143820052584
```



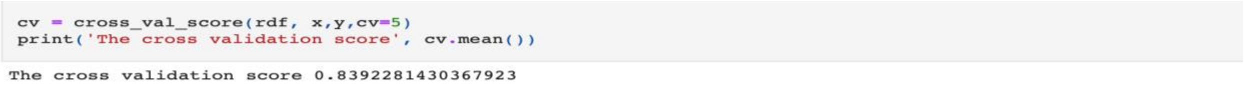
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=50)
Final_model=RandomForestRegressor()
Final_model.fit(x_train,y_train)
pred_rdf = Final_model.predict(x_test)

print('The r2 score is:', r2_score(y_test, pred_rdf))
```

```
The r2 score is: 0.9002141046656099
```

The regplot of actual VS Predicted for Random Forest model shows it's a good model.



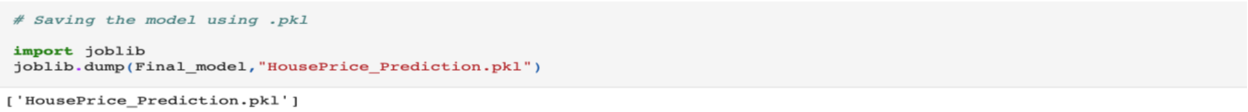
The CV Score:



```
cv = cross_val_score(rdf, x,y,cv=5)
print('The cross validation score', cv.mean())
```

```
The cross validation score 0.8392281430367923
```

I saved the Final Model using joblib:



```
# Saving the model using .pkl

import joblib
joblib.dump(Final_model,"HousePrice_Prediction.pkl")
```

```
['HousePrice_Prediction.pkl']
```

While saving the Final model I got accuracy of 90.02%. Hence, my model is saved for further predictions.

4. Key Metrics for success in solving problem under consideration

* I have used the r2 score as the accuracy score of the model.

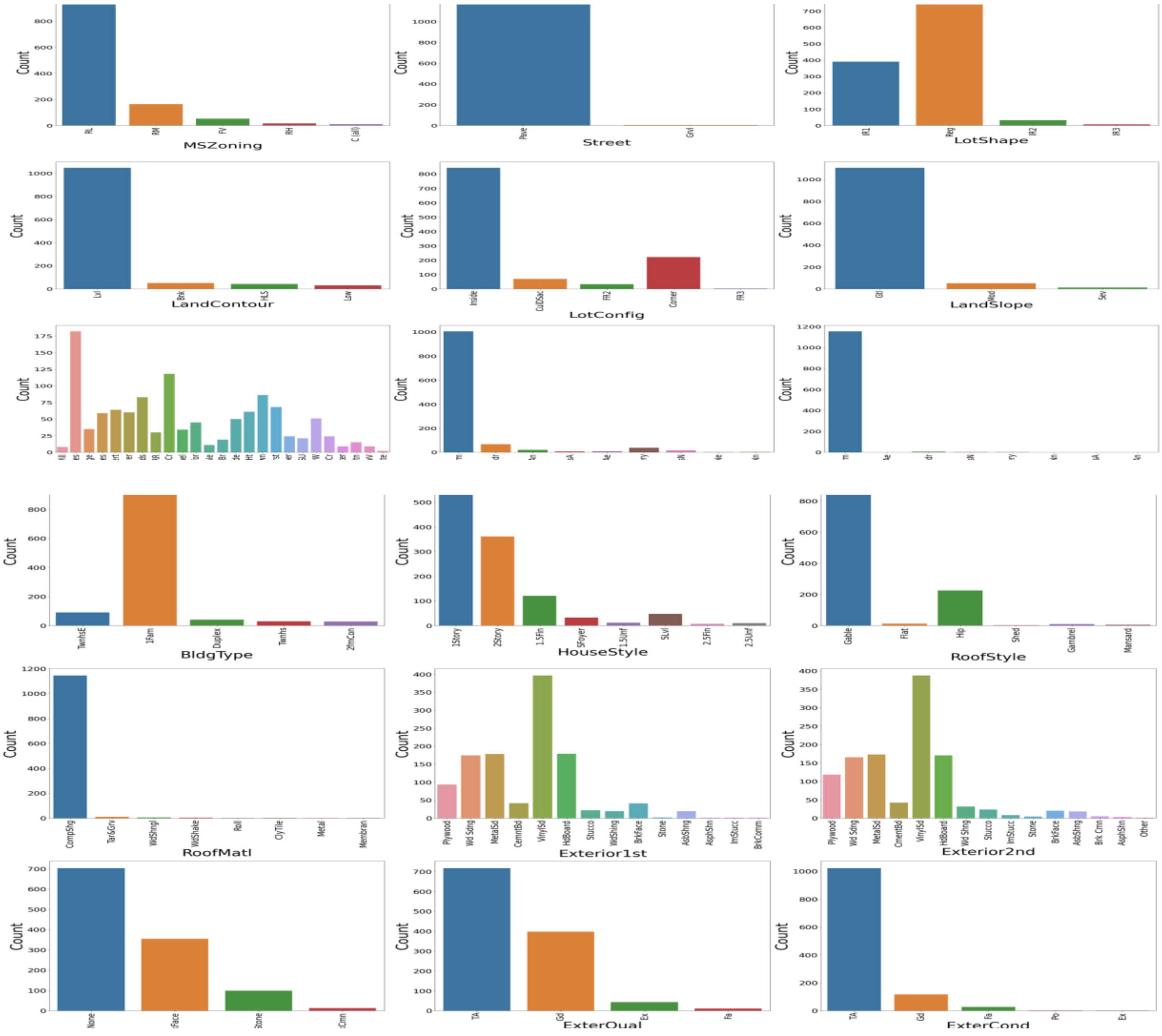
* I have used mean squared error, mean absolute error to find the error rate in the model.

* I have used root mean squared error and took the least amount as the best fit model.

* I also used Cross Validation Score to cross verify with r2 score and find the best model which has the least difference between r2 score and cv Score mean.

5. Visualizations

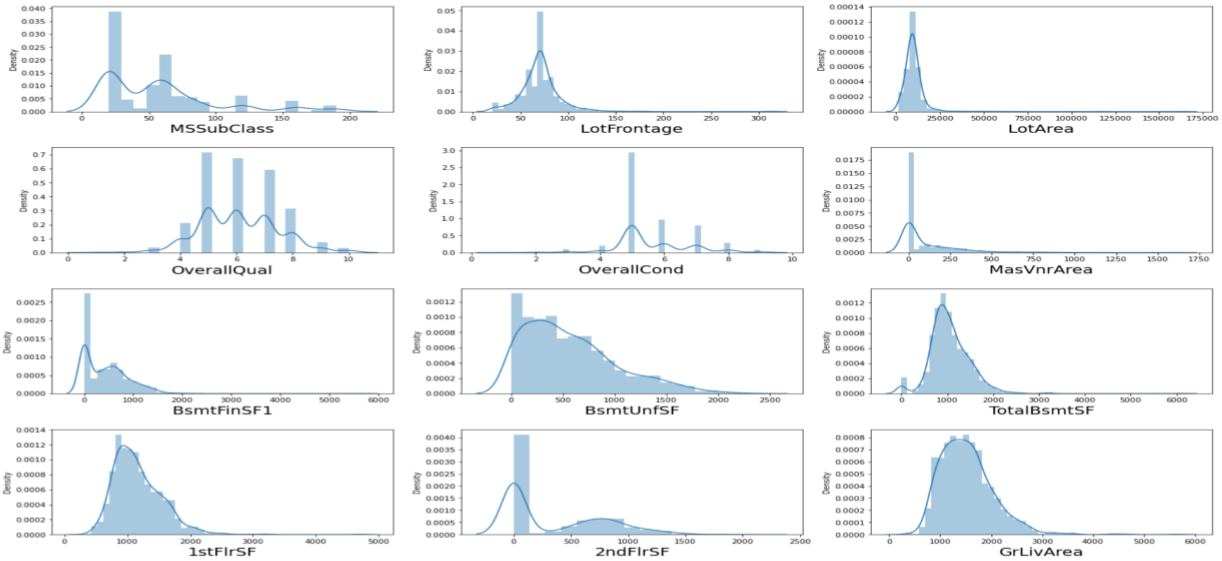
- **I Have used barplots to visualize the count of Categorical data.**

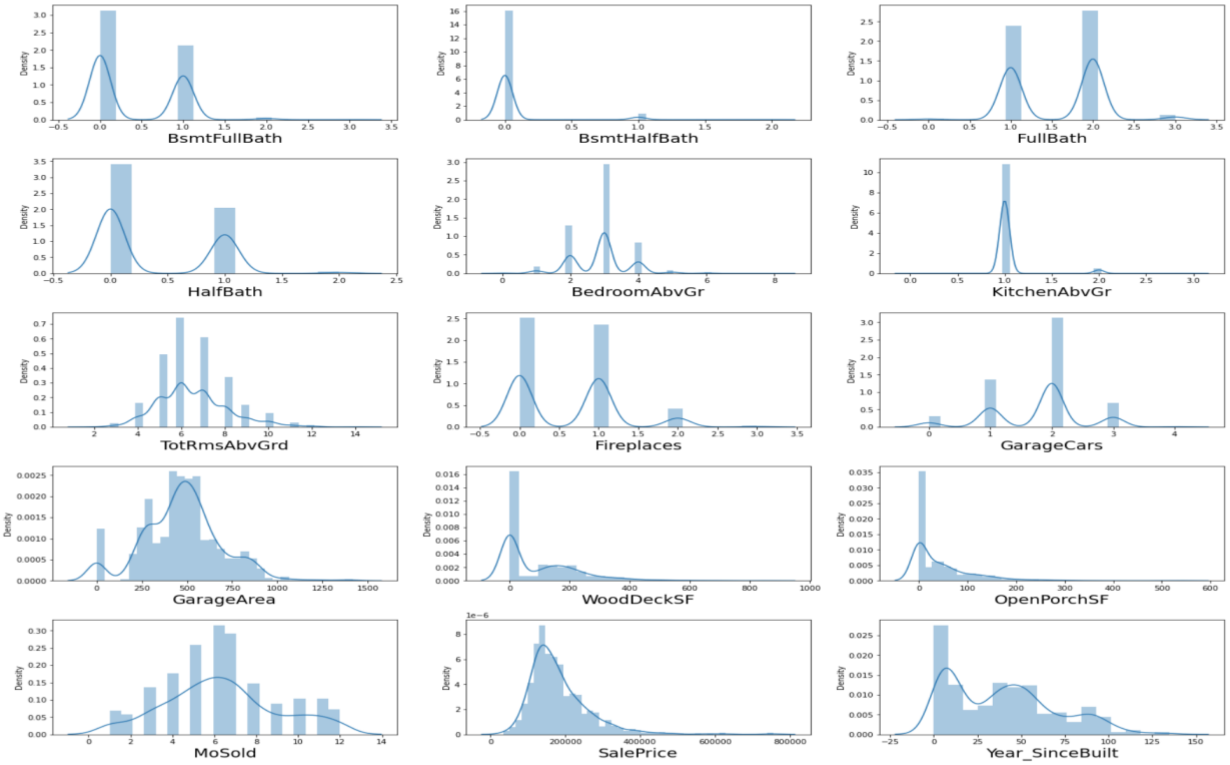


Observations:

- MSZoning-in zoning classification residential low density is having more count.
- Lot Config- inside lot is preferred by more people as a configuration of the lot.
- Neighborhood-northwest ames is having more count and the least is blue stem for the physical location within ames city limits.
- Condition1 & Condition2-Normal condition is having more count for both condition1 and condition2 which is proximity to various conditions.
- Bldg Type -single family detached is preferred by more people.
- House Style-one story dwelling is having more count among all types.
- Roofstyle-gable has more count as the type of roof and a very few counts of shed.
- Exterior 1st & Exterior 2nd- most of the houses in the dataset have vinyl siding as the exterior covering on house and asphalt shingles imitation shicco, brick common has the least count.
- Exterqual & Extercont-the quality of exterior material is average or typical for
- Foundation-Pouredconcrete and cinderblock is having more count as a type of foundation and the least is wood.
- BSMTQuality-a greater number of people prefer 80 to 89 inches height of basement, and the least count is 70 to 79 inches.
- BSMTCondition-slight dampness is having more count and the least count is for poor severe cracking or settling wetness in the basement.
- BSMTExposure-no exposure is having the most count and minimum exposure is the least for walkout or garden level walls.
- BSMF type-unfinished followed by good living quarters is having more count as the basement finished type and low quality is having the least count.
- Heating-gas A is having more count.
- Central Air-most of the houses in the dataset have central air conditioning.
- Electrical-standard circuit breaker is used by more houses and a very few houses use a mixed type of electrical system.
- Saletype-warranty deed is preferred by more people as the type of sale.
- Sale Condition-normal sale condition is having the most count and the very few counts of adjacent land purchase is a condition of sale.
- Garagetype-attached to home type of garage is having the most count in the dataset than 2 types.

- **I have used distplot to analyze distribution of numerical data.**

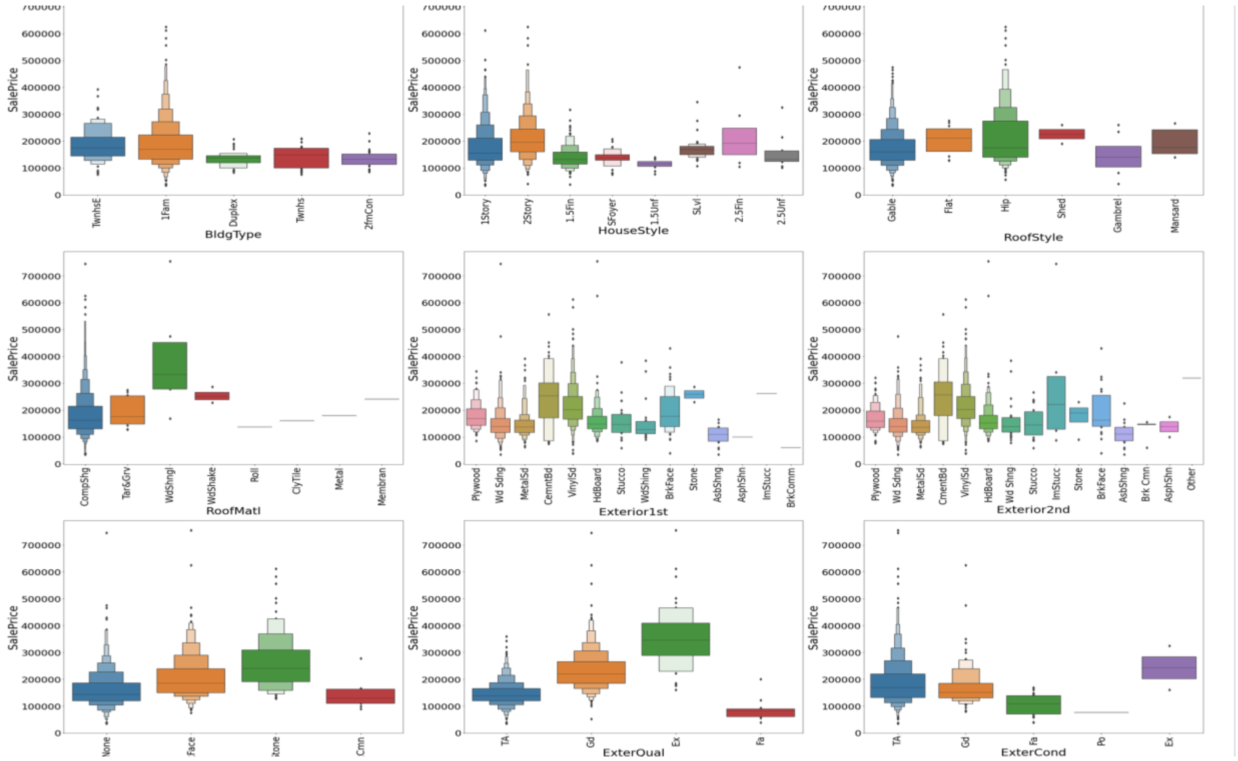


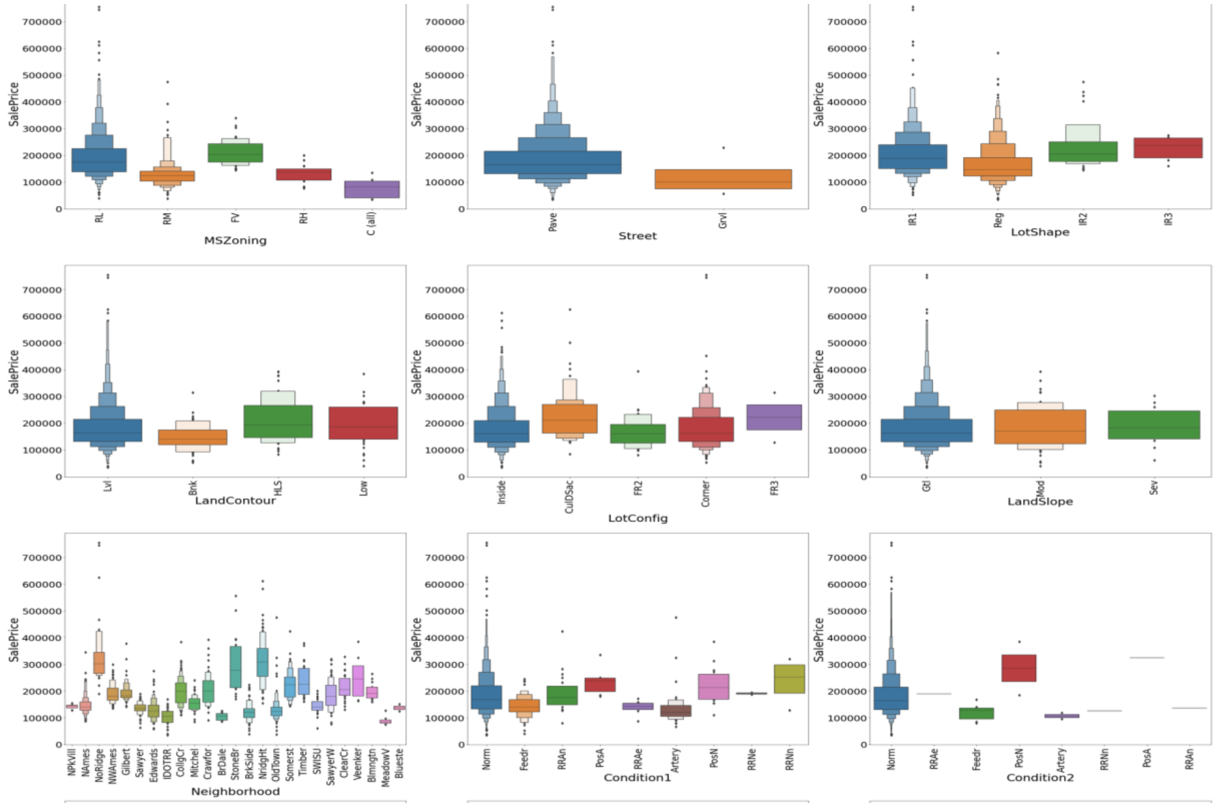


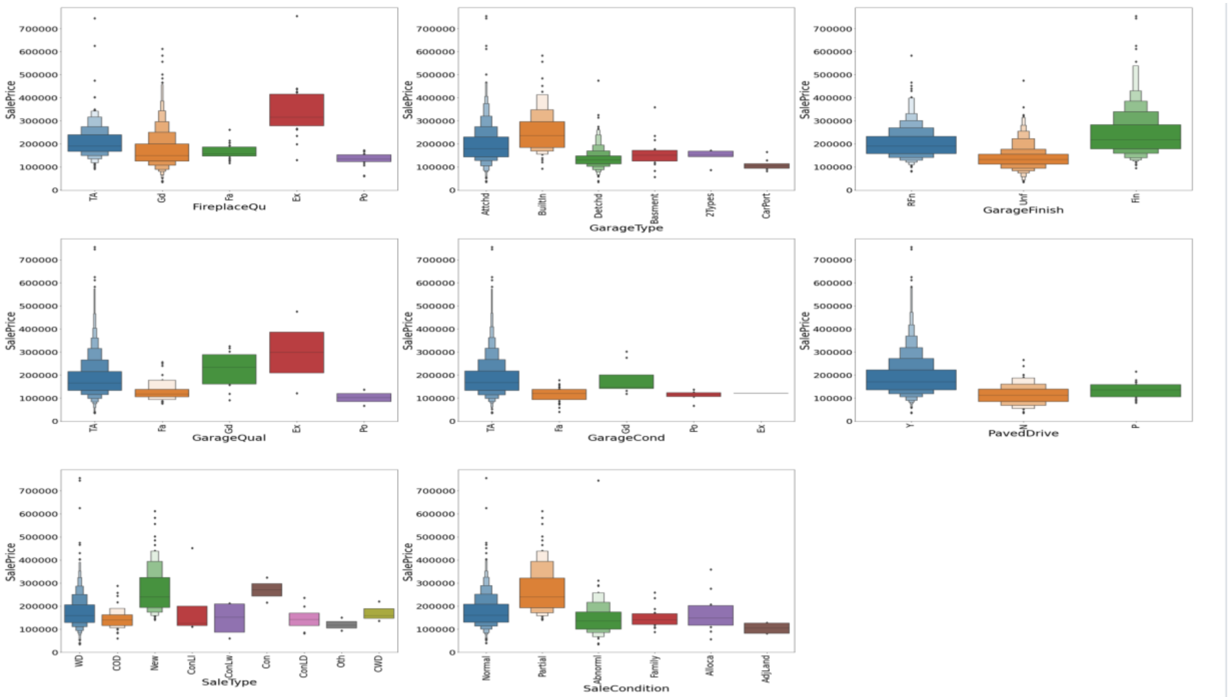
Observations:

- There is skewness present in almost all numerical columns. I will be removing the skewness using the power transformer.

- **I have used a boxen plot to find the relation between categorical and target.**



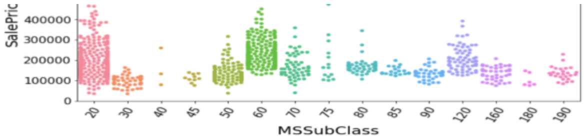


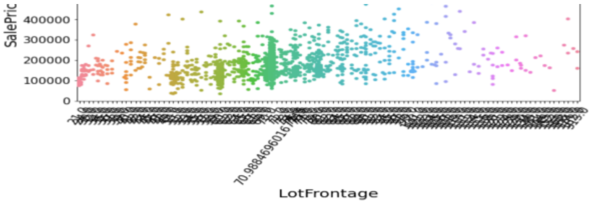


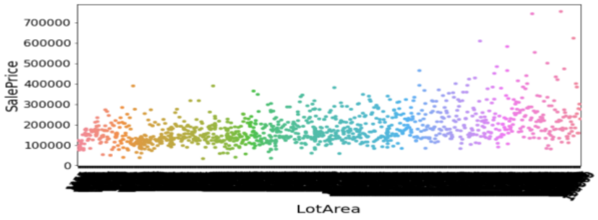
Observations:

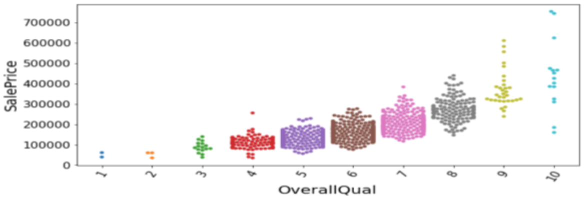
- The Price and demand is highest for Residential Low density Zone property.
- Paved roads are preferred more and of higher price.
- Slightly Irregular Lot shape is having higher Price and range than regular Lot shape which is preferred by more customers.
- Northridge and Northridge Heights neighborhoods are having higher prices.
- Normal proximity to various conditions is costlier than any others.
- 1 Family detached dwelling is in more demand and costlier than any other types.
- 2 Storey houses are costlier.
- Hip roof style is more expensive.
- Poured concrete Foundation is very expensive than slab which is cheapest.
- For Standard Circuit Breakers & Romex (Sbrkr) of Electrical system (Electrical) the SalePrice is Maximum.
- For Completely finished (Fin) Interior of the garage(Garage Finish) the SalePrice is high.
- For Home just constructed and sold (New) and Contract 15% Down payment regular terms(Con) of type of sale(Sale Type) has highest SalePrice.
- For Home was not completed when last assessed (associated with New Homes) (Partial) Condition of sale (Sales Condition) the SalePrice is maximum.

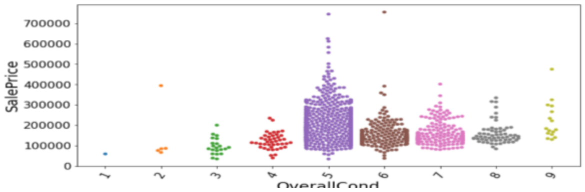
- **I have used swarmplot, strip plot and scatter plot for visualization of numerical columns with target.**

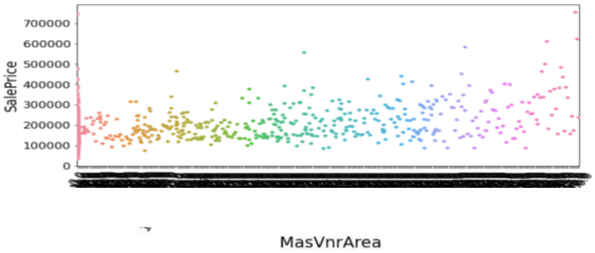


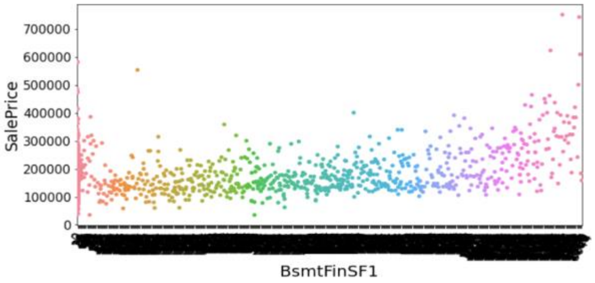


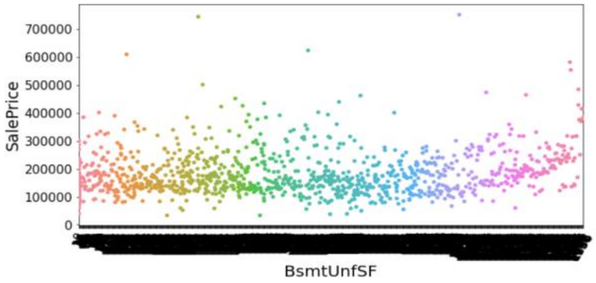


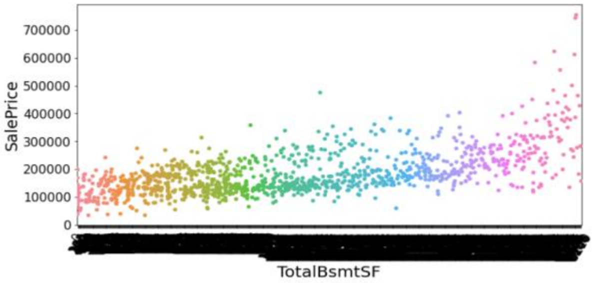


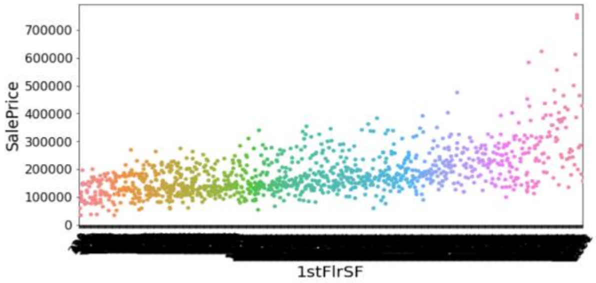


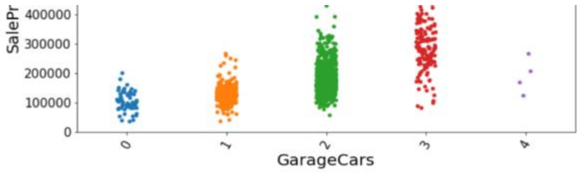


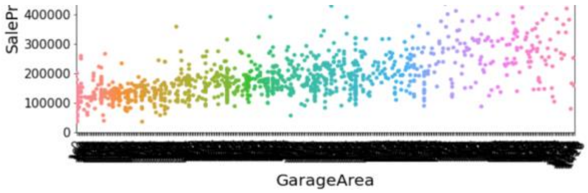


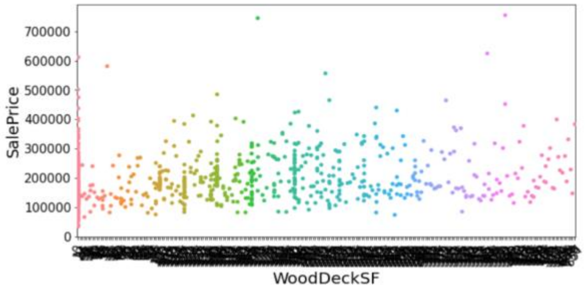


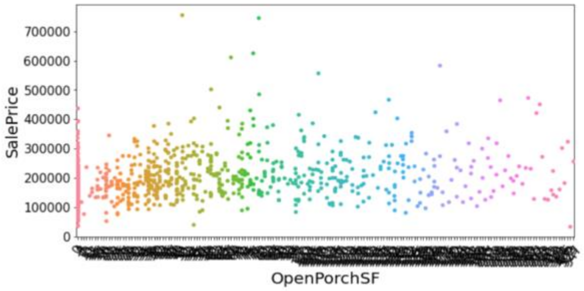


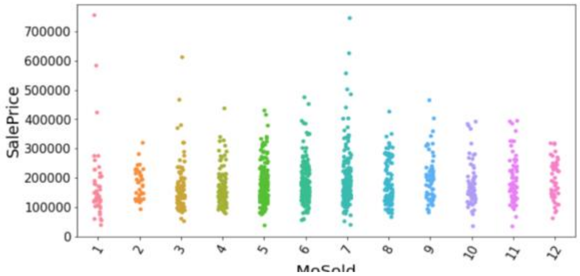


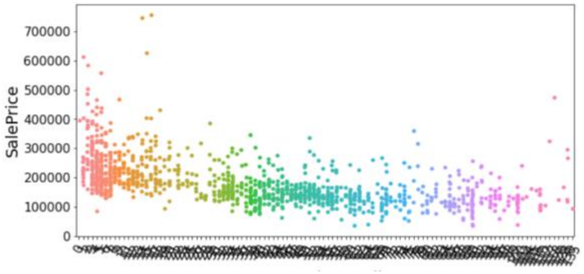

SalePr
400000
300000
200000
100000
0
0
1
2
3
4
GarageCars

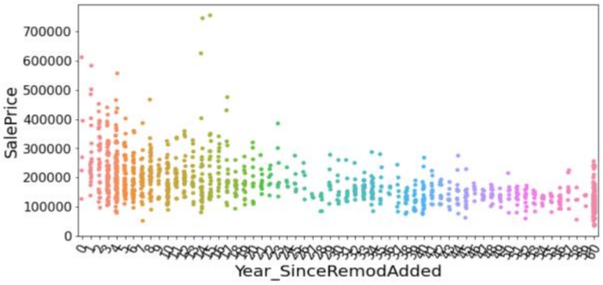

SalePr
400000
300000
200000
100000
0
GarageArea

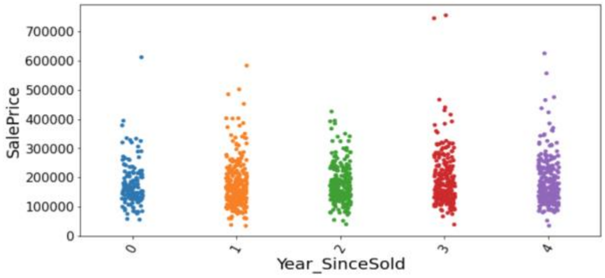

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
WoodDeckSF

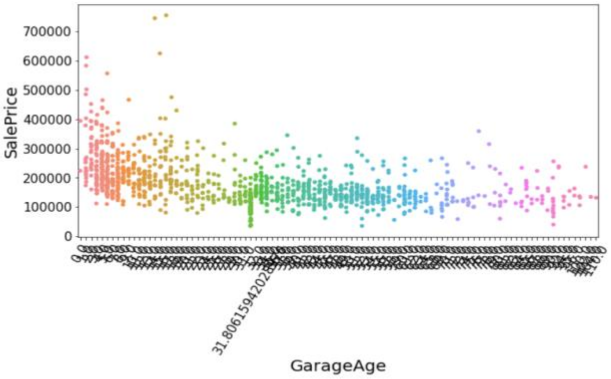

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
OpenPorchSF

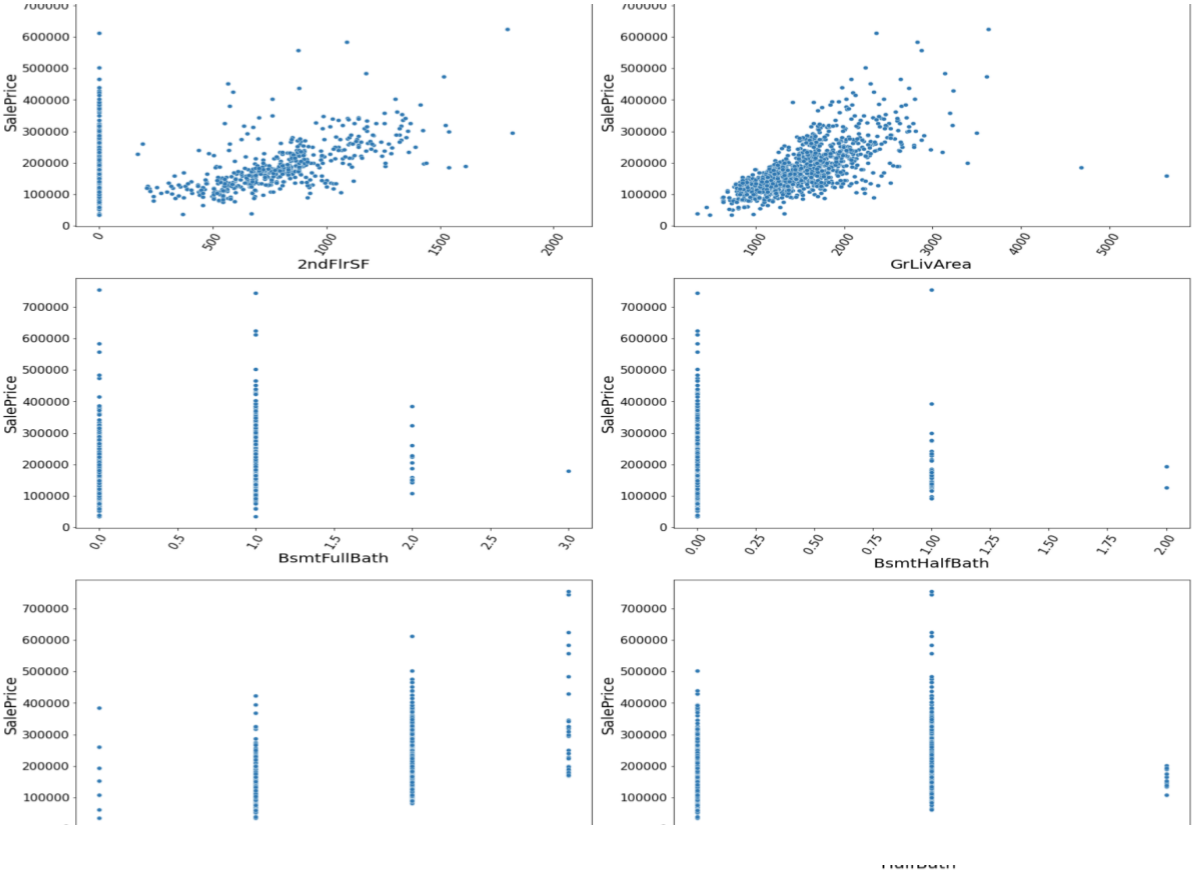

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
11
12
MoSold

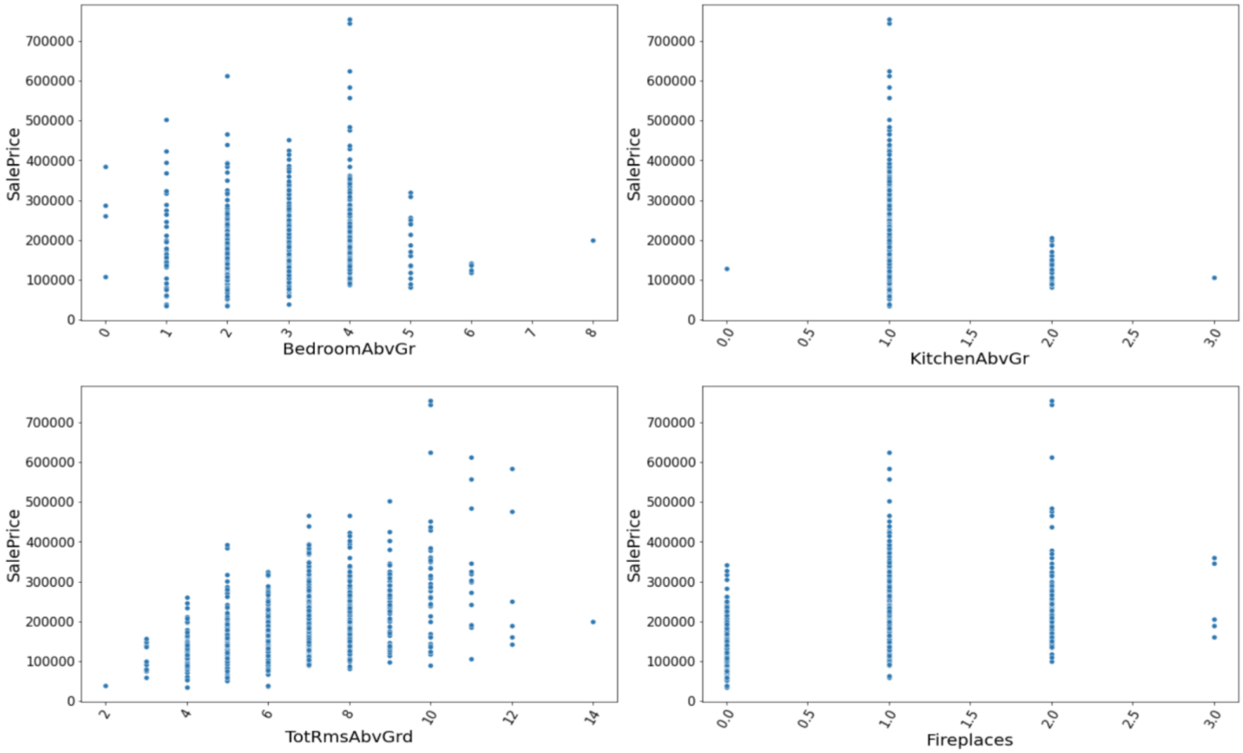

SalePrice
700000
600000
500000
400000
300000
200000
100000
0


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Year_SinceRemodAdded


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
Year_SinceSold


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
31.806159420289
GarageAge


700000
600000
500000
400000
300000
200000
100000
0
SalePrice
0
500
1000
1500
2000
2ndFlrSF
1000
2000
3000
4000
5000
GrLivArea
0.0
0.5
1.0
1.5
2.0
2.5
3.0
BsmtFullBath
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
BsmtHalfBath
HalfBath


700000
600000
500000
400000
300000
200000
100000
0
SalePrice
0
1
2
3
4
5
6
7
8
BedroomAbvGr
0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr
2
4
6
8
10
12
14
TotRmsAbvGrd
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Fireplaces

Observations:

- For size of the garage upto 3 cars, the price increases.
- Generally, as Area of the garage increases, the Price also increases.
- There is no specific pattern for WoodDeckSF and OpenPorhSF.
- As years have passed, the Price decreases.
- As garage age increases, price decreases.
- As the 2ndFlrSF increases price also increases.
- As ground level area increases, Price Increases.
- The price and demand is high for 0 and 1 full baths in the basement.
- If full bath is 3 Price is higher than 0,1 or 2
- for 4 bedrooms houses above ground level then price is higher.
- Price is higher for 1 kitchen above ground level.
- As total rooms above grade increase, the Price also increases.
- Price is higher if there are 2 fireplaces.
- MSSubClass- 2-STORY 1946 & NEWER and 1-STORY 1946 & NEWER ALL STYLES are having higher sale Price.
- LotFrontage & LotArea- There is no specific pattern for Linear feet of street connected to property and Lot size in square feet.
- OverallQual- As the overall quality of the material and finish of the house increases, Sale Price also increases.
- OverallCond- Though the overall condition increases, Sale Price also increases, the Average rating houses are in more demand and Sale Price is higher.
- MasVnrArea, BsmtFinsF1, BsmtUnfSF, TotalBsmtSF & 1stFlrSF have an increasing pattern with Price.

CONCLUSION

1. Key Findings and Conclusions of the Study

For many buyers, higher mortgage rates mean they can no longer afford homes in specific price ranges. The problem is that even modest single-family homes cost as much as lavish pads did a few years ago, so buyers are stuck either waiting for more inventory to come online or moving to a more affordable area.

In this project report, I have used machine learning algorithms to predict the house prices. I have mentioned the step-by-step procedure to analyze the dataset and find the correlation between the features. Hence, we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then we have also saved the data frame of predicted prices of the test dataset.

I have observed that certain features like OverallCond, ExterQual, etc contribute the most to the Price of the house. Also, conditions like years since built negatively affect the price. As years passed the value decreased.

2. Learning Outcomes of the Study in respect of Data Science

From the results of the tests that have been carried out, the model is declared to have passed the test. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed, and analyzed. The power of visualization has helped us in understanding the data by graphical representation. Data cleaning is one of the most important steps to remove missing values and Zero values and to replace them with respective mean, median or mode. This study is an exploratory attempt to use five machine learning algorithms in estimating housing prices, and then compare their results.

I hope this study has moved a small step ahead in providing some methodological and empirical contributions to property appraisal and presenting an alternative approach to the valuation of housing prices. Future direction of research may consider incorporating additional property transaction data from a larger geographical location with more features or analyzing other property types beyond housing development.

3. Limitations of this work and Scope for Future Work

- There were so many outliers present in the dataset and data loss were high while using the Z-score method.
- We had to use the Percentile method and it is not as effective as the Z-score method in removing outliers.
- There was a lot of skewness present in the dataset which will again affect the model as we must transform it.
- This study did not use all advanced algorithms but only a few simple regression algorithms to a few advanced ones.
- I have not merged the Train and test datasets to prevent data Leakage.
- There were a few columns with more values as 0 and a few with more missing values. I must remove those columns.
- There was multicollinearity and the columns with highest correlation with the Target had to be removed to prevent multicollinearity.

Even after all these Limitations and drawbacks, my model tends to perform well with an accuracy of 90.02% with Random Forest model and a CV Score of 83.92%.

REFERENCES

1. https://www.forbes.com/advisor/mortgages/real-estate/housing-market-predictions/
2. https://scholarworks.sjsu.edu/etd_projects/540/
3. https://scholarworks.sjsu.edu/etd_projects/540/
4. https://rpubs.com/Zetrosoft/lbb-rm
5. https://medium.com/analytics-vidhya/predicting-house-prices-using-classical-machine-learning-and-deep-learning-techniques-ad4e55945e2d
6. https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/

THANK YOU