FLIP ROBO

PROJECT REPORT

ON

HOUSING: PRICE PREDICTION

SUBMITTED BY

V Samyukta

ACKNOWLEDGMENT

It gives me immense pleasure to deliver this report. Working on this project was a great learning experience that helped me attain in-depth knowledge on data analysis process.

Flip Robo Technologies (Bangalore) provided all of the necessary information and datasets, required for the completion of the project.

I express my gratitude to my SME, Ms. Khushboo Garg, for providing the dataset and directions for carrying out the project report procedure.

My heartfelt gratitude to DataTrained institute and FlipRobo company for providing me this internship opportunity. Last but not least to my sincere thanks to my family and all those who helped me directly or indirectly in completion this project.

CONTENTS

1.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.

With the supplied independent variables, we must simulate the price of dwellings. The management will then utilise this model to figure out how the prices change depending on the factors. They may then adjust the firm's strategy and focus on regions that will provide large profits. Furthermore, the model would assist management in comprehending the price dynamics of a new market.

Conceptual Background of the Domain Problem:

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

Buying a house is a stressful thing.

- One has to pay huge sums of money and invest many hours and even then there is a persisting concern whether it’s a good deal or not. Buyers are generally not aware of factors that influence the house prices. Almost all the houses are described by the total area in square foot, the neighborhood and number of bedrooms. Sometimes houses are even priced at X dollars per square foot.
  This creates an illusion that house prices are dependent almost solely on the above stated factors. Most of the houses are bought through real estate agents.

People rarely buy directly from the seller, since there are a
lot of legal terminology involved and people are unaware of them. Hence real estate agents are trusted with the communication between buyers and sellers as well as laying down a legal contract for the transfer. This just creates a middle man and increases the cost of houses. Therefore, the houses are overpriced and a buyer should have a better idea of the actual value of these houses.

- There are various tools, like Zillow and Trulia, available
online to assist a person with buying houses. These tools provide a price estimation of various houses and are generally free for use. These tools incorporate many factors to estimate the house prices by providing weights to each factor. For example, Zillow creates Zestimate of houses which is "calculated three times a week based on millions of public and user-submitted data points"
- The median error rate for these estimates is quite low. The main problem with these tools is that they are heavy on advertisements and they promote real estate agents.
Zillow provides paid premium services for real estate agents and this is their main source of income.
- Estimates of actual house prices will help buyers to have better negotiations with the real estate agents, as the list price of the house and much higher than the actual price
- 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 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.

Motivation for the Problem Undertaken:

The main goal of this research is to create a model that can estimate property prices using other supporting features. Machine Learning methods will be used to forecast.

We obtained the sample data from our customer database. In order to increase customer selection, the client requests certain forecasts that will assist them in making future investments and improving customer selection.

The House Price Index is a popular tool for estimating house price fluctuations. Because housing prices are significantly connected with other characteristics such as location, region, and population, predicting individual house prices requires information other than HPI. There have been a lot of studies that use typical machine learning algorithms to successfully estimate house prices, but they seldom look at the performance of different models and ignore the less popular yet sophisticated models.

As a result, this study will use both classic and advanced machine learning methodologies to analyse the differences between numerous advanced models in order to evaluate the diverse influences of features on prediction methods. This research will also present an optimistic outcome for housing price prediction by thoroughly validating numerous strategies in model implementation on regression.

2.ANALYTICAL PROBLEM FRAMING

Mathematical / Analytical Modelling of the Problem:

This particular problem has two datasets one is train dataset and the other is test dataset. I have built model using train dataset and predicted SalePrice for test dataset. By looking into the target column, I came to know that the entries of SalePrice column were continuous and this was a Regression problem so I have to use all regression algorithms while building the model. Also, I observed some unnecessary entries in some of the columns like in some columns I found more than 80% null values and more than 85% zero values so I decided to drop those columns. If I keep those columns as it is, it will create high skewness in the model. While checking the null values in the datasets I found many columns with nan values and I replaced those nan values with suitable entries like mean for numerical columns and mode for categorical columns. To get better insight on the features I have used ploting like distribution plot, bar plot, reg plot and strip plot. With these ploting I was able to understand the relation between the features in better manner. Also, I found outliers and skewness in the dataset so I removed outliers using percentile method and I removed skewness using yeo-johnson method. I have used all the regression models while building model then tunned the best model and saved the best model. At last I have predicted the sale price fot test dataset using the saved model of train dataset.

Data Sources and their formats:

The data given by Flip Robo was in CSV format (Comma Separated Values). There are 1168 rows and 81 columns in the data. There are two data sets available. There are two types of data: training data and testing data.

- The train file will be used to train the model, which means that the model will learn from it. All of the independent variables are included, as well as the target variable. The training set has 1168 records.
- The independent variables are all present in the test file, but not the target variable. For the test data, we'll use the model to forecast the target variable. The test set has 292 records.

Data Pre-processing Done:

In Machine Learning, data pre-processing refers to the process of cleaning and organising raw data in order to make it appropriate for creating and training Machine Learning models. In other words, anytime data is received from various sources, it is collected in raw format, which makes analysis impossible. Data pre-processing is a crucial stage in Machine Learning since the quality of data and the relevant information that can be gleaned from it has a direct impact on our model's capacity to learn; consequently, we must pre-process our data before feeding it into our model. As a result, it is the first and most important stage in developing a machine learning model.

I have used some following pre- processing steps:

- Loading the training dataset as a dataframe.
- Used pandas to set display I ensuring we do not see any truncated information.
- Checked the number of rows and columns present in our training dataset.

- Checked for missing data and the number of rows with null values.
- Verified the percentage of missing data in each column and decided to discard the one’s that have more than 50% of null values.
- Dropped all the unwanted columns and duplicate data present in our dataframe.
- Separated categorical column names and numeric column names in separate list variables for ease in visualization.

- Checked the unique values information in each column to get a gist for categorical data.
- Performed imputation to fill missing data using mean on numeric data and mode for categorical data columns.

- Used Pandas Profiling during the visualization phase along with pie plot, count plot, scatter plot and the others.
- With the help of ordinal encoding technique converted all object datatype columns to numeric datatype.
- Thoroughly checked for outliers and skewness information.
- With the help of heatmap, correlation bar graph was able to understand the Feature vs Label relativity and insights on multicollinearity amongst the feature columns.
- Separated feature and label data to ensure feature scaling is performed avoiding any kind of biasness.
- Checked for the best random state to be used on our Regression Machine Learning model pertaining to the feature importance details.
- Finally created a regression model function along with evaluation metrics to pass through various model formats.

• Data Inputs-Logic-Output Relationships:

- I have used box plot for each pair of categorical features that shows the relation with the median sale price for all the sub categories in each categorical feature.
- And also for continuous numerical variables I have used reg plot to show the relationship between continuous numerical variable and target variable.
- I found that there is a linear relationship between continuous numerical variable and SalePrice.

• Hardware and Software Requirements and Tools Used:

Hardware technology being Used:-

- CPU: HP Pavilion
- Chip: intel core13 8th Gen
- RAM: 8 GB

Software Technology being Used:-

- Programming language: Python
- Distribution: Anaconda Navigator
- Browser based language shell: Jupyter Notebook

Libraries/Packages Used:-

Pandas, NumPy, matplotlib, seaborn, scikit-learn and pandas_profiling.

3.DATA ANALYSIS AND VISUALIZATION

- **Identification of possible problem-solving approaches (methods):**

  - To tackle the problem, I employed both statistical and analytical methodologies, which mostly included data pre-processing and EDA to examine the connection of independent and dependent characteristics. In addition, before feeding the input data into the machine learning models, I made sure that it was cleaned and scaled.
  - We need to anticipate the sale price of houses for this project, which implies our goal column is continuous, making this a regression challenge. I evaluated the prediction using a variety of regression methods. After a series of assessments, I determined that Extra Trees Regressor is the best method for our final model since it has the best r2-score and the smallest difference in r2-score and CV-score of all the algorithms tested. Other regression methods are similarly accurate.
  - I used K-Fold cross validation to gain high performance and accuracy. Then hyper parameter tweaked the final model.
  - Once I had my desired final model, I made sure to save it before loading the testing data and beginning to do data pre-processing as the training dataset and retrieving the anticipated selling price values from the Regression Machine Learning Model.

- **Testing of Identified Approaches (Algorithms):**

Since Saleprice was my target and it was a continuous column so this particular problem was regression problem. And I have used all regression algorithms to build my model. By looking into the difference of r2 score and cross validation score I found ExtraTreesRegressor as a best model with least difference. Also to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation. Below are the list of regression algorithms I have used in my project.

- RandomForestRegressor
- XGBRegressor
- ExtraTreesRegressor
- GradientBoostingRegressor

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

r2 score, cross val score, MAE, MSE, and RMSE were the main metrics employed in this study. We used Hyperparameter Tuning to identify the optimal parameters and to improve our results, and we'll be utilising the GridSearchCV technique to do it.

- Cross Validation:

Cross-validation aids in determining the model's overfitting and underfitting. The model is constructed to run on several subsets of the dataset in cross validation, resulting in numerous measurements of the model. If we fold the data five times, it will be separated into five parts, each representing 20% of the whole dataset. During the Cross-validation, the first part (20%) of the 5 parts will be left out as a holdout set for validation, while the rest of the data will be utilised for training. We'll acquire the initial estimate of the dataset's model quality this way.

Further rounds are produced in the same way for the second 20% of the dataset, which is kept as a holdout set while the remaining four portions are utilised for training data during the process. We'll acquire the second estimate of the dataset's model quality this way. During the cross-validation procedure, these stages are repeated to obtain the remaining estimate of model quality.

- R2 Score:

It is a statistical metric that indicates the regression model's quality of fit. The optimal r-square value is 1. The closer the r-square value is to 1, the better the model fits.

- Mean Squared Error:

The average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated is measured by the MSE of an estimator (of a process for estimating an unobserved variable). MSE is a risk function that represents the squared error loss's anticipated value. The Root Mean Squared Error is abbreviated as RMSE.

- Mean Absolute Error:

MAE is a statistic that assesses the average magnitude of mistakes in a set of forecasts without taking into account their direction. It's the average of the absolute differences between forecast and actual observation over the test sample, where all individual deviations are given equal weight.

- Hyperparameter Tunning:

There is a list of several machine learning models available. They're all distinct in some manner, yet the only thing that distinguishes them is the model's input parameters. Hyperparameters are the name given to these input parameters. These hyperparameters will establish the model's architecture, and the greatest thing is that you get to choose the ones you want for your model. Because the list of hyperparameters for each model differs, you must choose from a distinct list for each model.

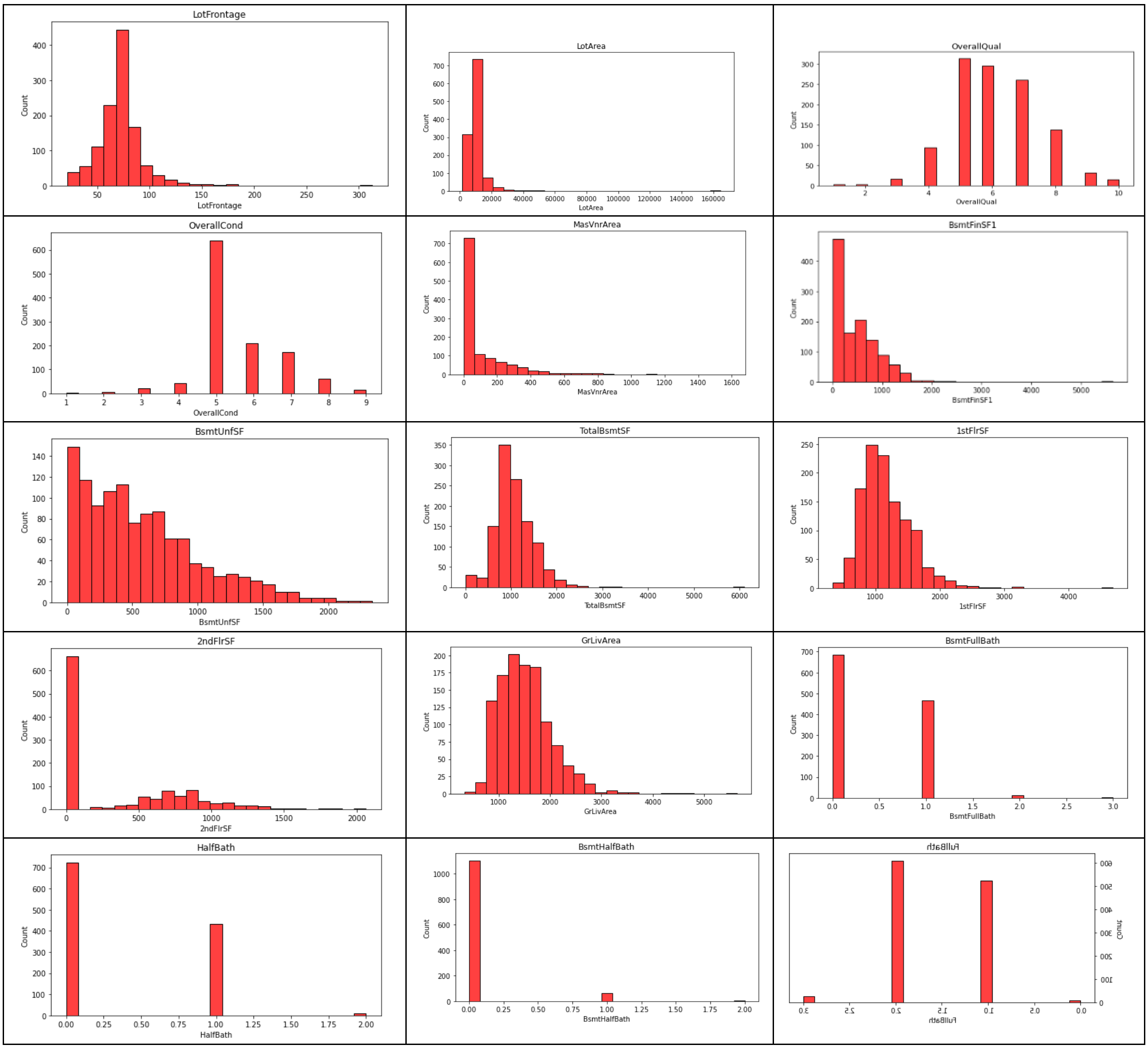
We are unaware of the ideal hyperparameter settings that would produce the best model output. So we instruct the model to automatically explore and choose the best model architecture. Hyperparameter tuning is the term for the method of selecting hyperparameters. GridSearchCV may be used to tune the system.

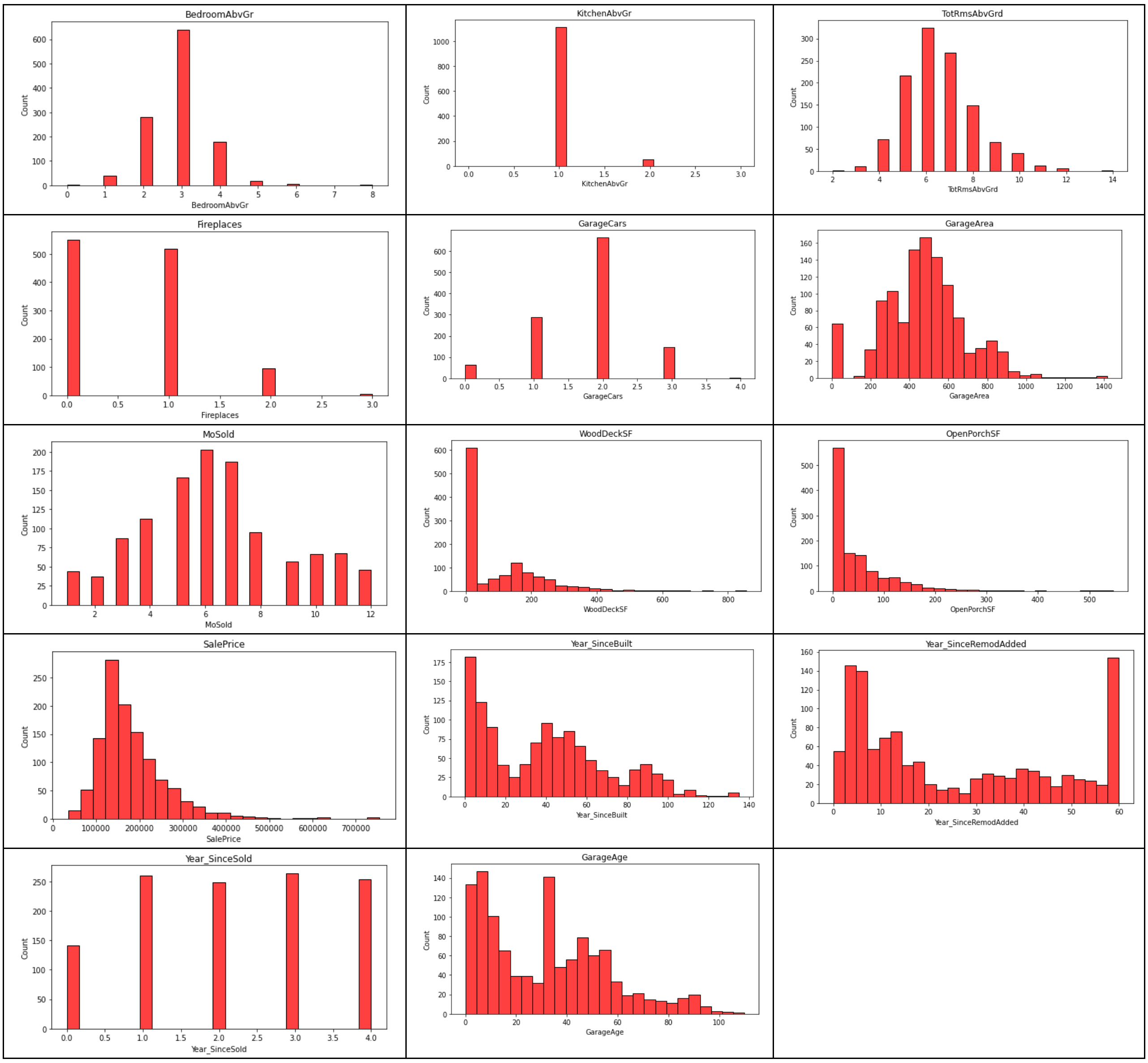
GridSearchCV is a model selection function in the Scikit-learn (or SK-learn) package. It is vital to remember that the Scikit-learn library must be installed on the PC. This function aids in fitting your estimator (model) to your training set by looping over specified hyperparameters. Finally, we may choose the optimal settings from the hyperparameters presented.

• Visualization:

I have used bar plots to see the relation of categorical feature and I have used 2 types of plots for numerical columns one is strip plot for ordinal features and other is reg plot for continuous features.

➢ <u>Visualization of numerical features with target:</u>

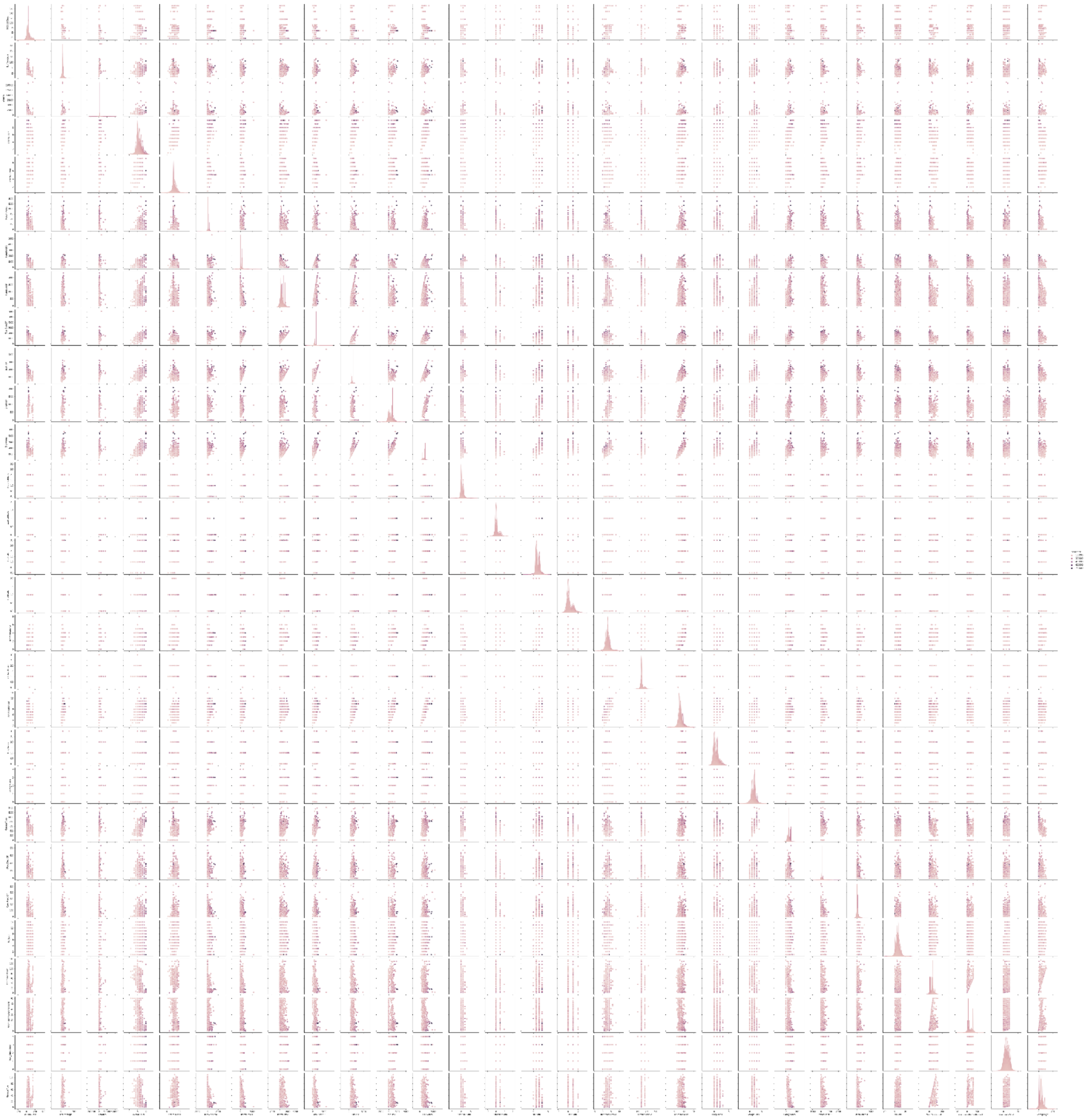




Observations:

1. For 1-STORY 1946 & NEWER ALL STYLES (20) and 2-STORY 1946 & NEWER (60) types of dwelling (MSSuubClass) the sales is good and SalePrice is also high.
2. As Rates the overall material and finish of the house (OverallQual) is increasing linearly sales is also increasing And SalePrice is also increasing linearly.
3. For 5(Average) overall condition of the house (OverallCond) the sales is high and SalePrice is also high.
4. For 0 and 1 Basement full bathrooms (BsmtFullBath) the sales as well as SalePrice is high.
5. For 0 Basement half bathrooms (BsmtHalfBath) the sales as well as SalePrice is high.
6. For 1 and 2 Full bathrooms above grade (FullBath) the sales as well as SalePrice is high.
7. For 0 and 1 Half baths above grade (HalfBath) the sales as well as SalePrice is high.
8. For 2, 3 and 4 Bedrooms above grade (does NOT include basement bedrooms) (BedroomAbvGr) the sales as well as SalePrice is high.
9. For 1 Kitchens above grade (KitchenAbvGr) the sales as well as SalePrice is high.
10. For 4-9 Total rooms above grade (does not include bathrooms) (TotRmsAbvGrd) the sales as well as SalePrice is high.

11. For 0 and 1 Number of fireplaces (Fireplaces) the sales as well as SalePrice is high.
12. For 1 and 2 Size of garage in car capacity (GarageCars) the sales is high and for 3 Size of garage in car capacity (GarageCars) the SalePrice is high.
13. In between april to august for Month Sold (MoSold) the sales is good with SalePrice.
14. For all the Year_SinceSold the SalePrice and sales both are same.

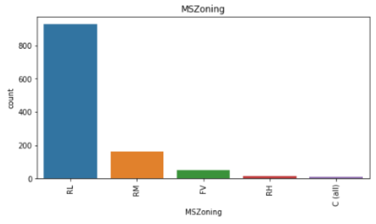


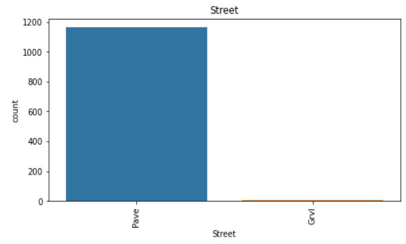
Observations:

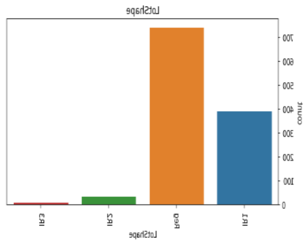
1. As Linear feet of street connected to property(LotFrontage) is increasing sales is decreasing and the SalePrice is ranging between 0-3 lakhs.
2. As Lot size in square feet(LotArea) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
3. As Masonry veneer area in square feet (MasVnrArea) is increasing sales is decreasing and SalePrice is ranging between 0-4 lakhs.

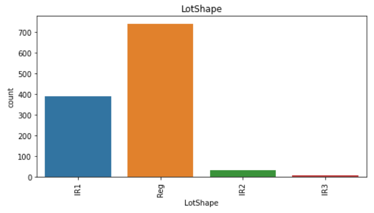
4. As Type 1 finished square feet(BsmtFinSF1) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs.
5. As Unfinished square feet of basement area (BsmtUnfSF) is increasing sales is decreasing and the SalePrice is in between 0-4 lakhs. There are some outliers also.
6. As Total square 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.

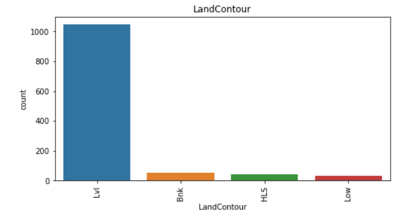
2. <u>Visualization of categorical features with target:</u>

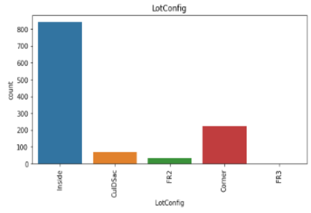


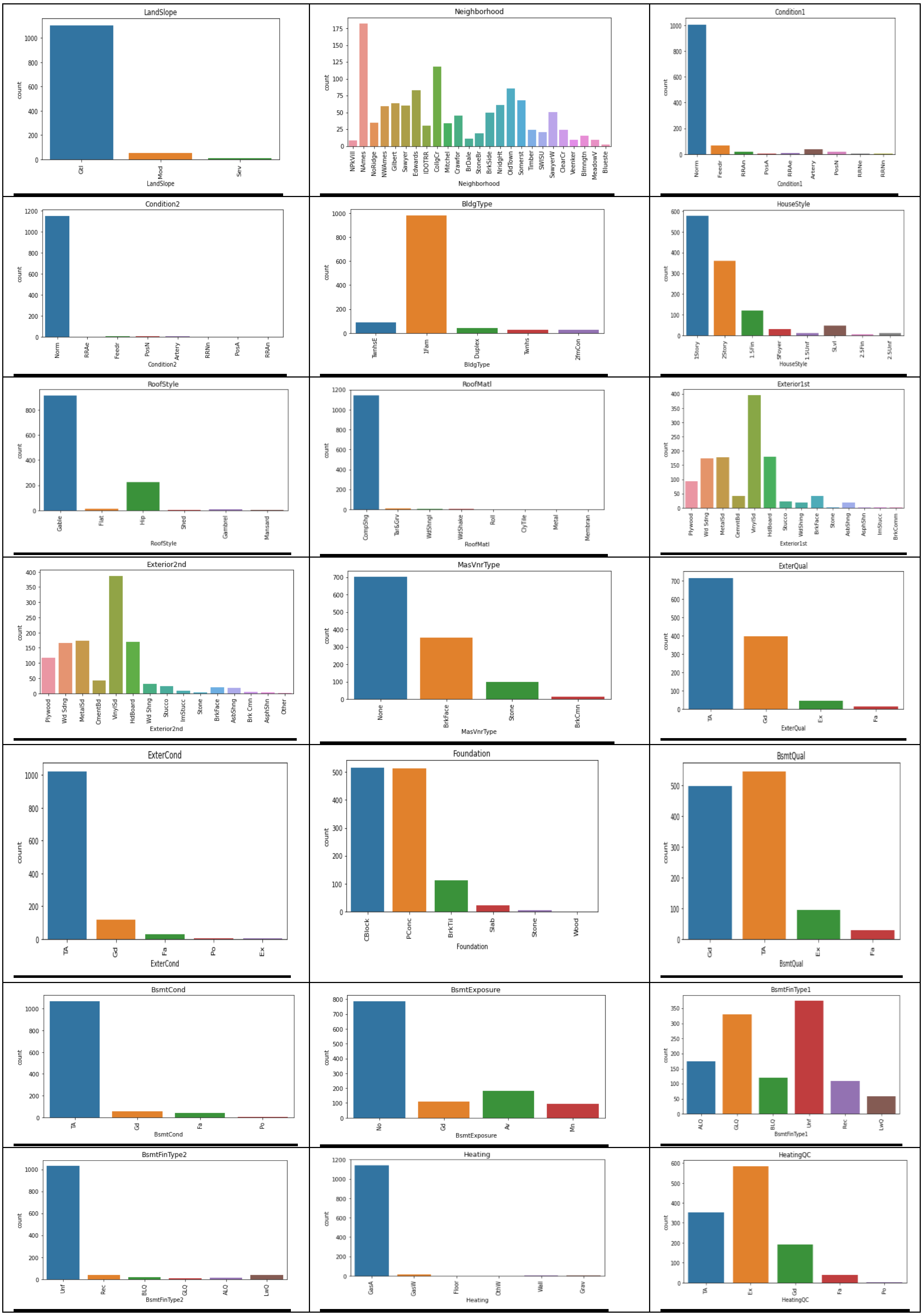


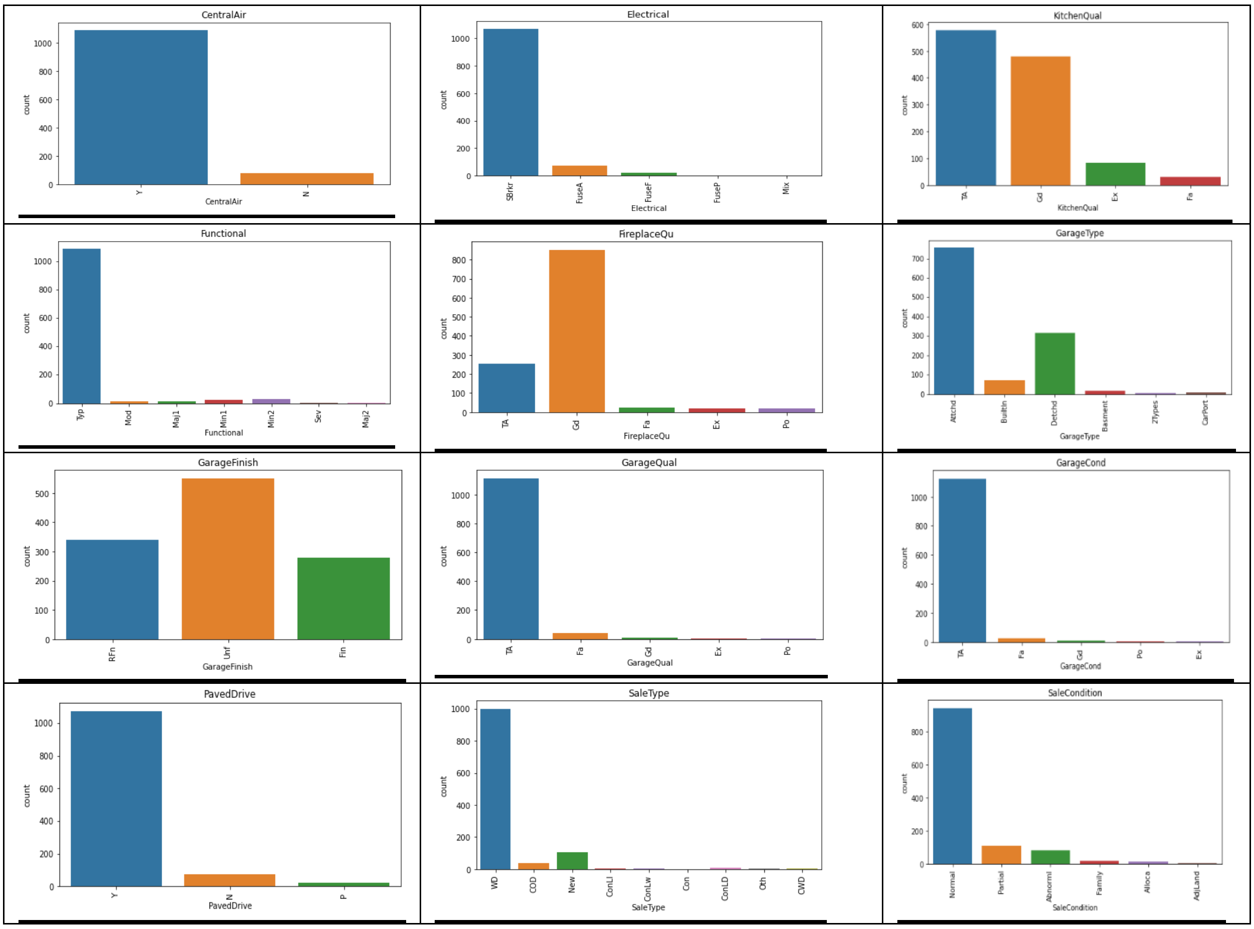









LandSlope
count
0 200 400 600 800 1000
Gtl Mod Sev
LandSlope
Neighborhood
count
0 25 50 75 100 125 150 175
NPkVill NAmes NoRidge NWAmes Gilbert Sawyer Edwards IDOTRR CollgCr Mitchel Crawfor BrDale StoneBr BrkSide NridgHt OldTown Somerst Timber SWISU SawyerW ClearCr Veenker Blmngtn MeadowV Blueste
Neighborhood
Condition1
count
0 200 400 600 800 1000
Norm Feedr RRAn PosA RRAe Artery PosN RRNe RRNn
Condition1
Condition2
count
0 200 400 600 800 1000 1200
Norm RRAe Feedr PosN Artery RRNn PosA RRAn
Condition2
BldgType
count
0 200 400 600 800 1000
TwnhsE 1Fam Duplex Twnhs 2fmCon
BldgType
HouseStyle
count
0 100 200 300 400 500 600
1Story 2Story 1.5Fin SFoyer 1.5Unf SLvl 2.5Fin 2.5Unf
HouseStyle
RoofStyle
count
0 200 400 600 800
Gable Flat Hip Shed Gambrel Mansard
RoofStyle
RoofMatl
count
0 200 400 600 800 1000 1200
CompShg Tar&Grv WdShngl WdShake Roll ClyTile Metal Membran
RoofMatl
Exterior1st
count
0 50 100 150 200 250 300 350 400
Plywood Wd Sdng MetalSd CemntBd VinylSd HdBoard Stucco WdShing BrkFace Stone AsbShng AsphShn ImStucc BrkComm
Exterior1st
Exterior2nd
count
0 50 100 150 200 250 300 350 400
Plywood Wd Sdng MetalSd CmentBd VinylSd HdBoard Wd Shng Stucco ImStucc Stone BrkFace AsbShng Brk Cmn AsphShn Other
Exterior2nd
MasVnrType
count
0 100 200 300 400 500 600 700
None BrkFace Stone BrkCmn
MasVnrType
ExterQual
count
0 100 200 300 400 500 600 700
TA Gd Ex Fa
ExterQual
ExterCond
count
0 200 400 600 800 1000
TA Gd Fa Po Ex
ExterCond
Foundation
count
0 100 200 300 400 500
CBlock PConc BrkTil Slab Stone Wood
Foundation
BsmtQual
count
0 100 200 300 400 500
Gd TA Ex Fa
BsmtQual
BsmtCond
count
0 200 400 600 800 1000
TA Gd Fa Po
BsmtCond
BsmtExposure
count
0 100 200 300 400 500 600 700 800
No Gd Av Mn
BsmtExposure
BsmtFinType1
count
0 50 100 150 200 250 300 350
ALQ GLQ BLQ Unf Rec LwQ
BsmtFinType1
BsmtFinType2
count
0 200 400 600 800 1000
Unf Rec BLQ GLQ ALQ LwQ
BsmtFinType2
Heating
count
0 200 400 600 800 1000 1200
GasA GasW Floor OthW Wall Grav
Heating
HeatingQC
count
0 100 200 300 400 500 600
TA Ex Gd Fa Po
HeatingQC



Observations:

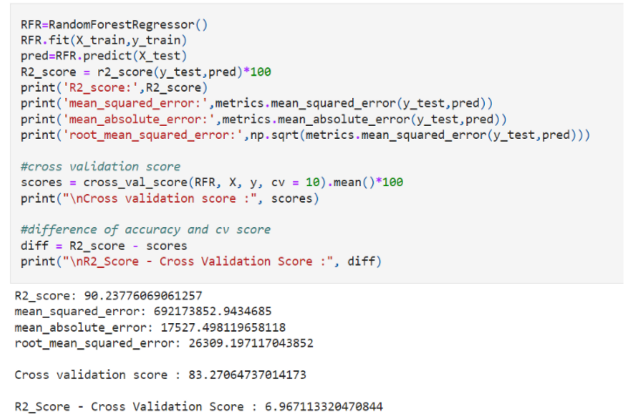
1. For Floating Village Residential (FV) and Residential Low Density(RL) zoning classification of the sale(MSZoning) the SalePrice is high.
2. For paved type of road access to property (Street) the SalePrice is high.
3. For Slightly irregular (IR1), Moderately Irregular (IR2) and Irregular (IR3) shape of property (LotShape) the SalePrice is high.
4. For Hillside - Significant slope from side to side (HLS) Flatness of the property (LandContour) the SalePrice is High.
5. For Cul-de-sac (CulDSac) Lot configuration (LotConfig) the SalePrice is High.
6. For all types of Slope of property (LandSlope) i.e., Gentle slope (Gtl), Moderate Slope (Mod) and Severe Slope (Sev) the SalePrice is High.
7. For Northridge (NoRidge) locations within Ames city limits (Neighborhood) the SalePrice is High.
8. For Within 200' of North-South Railroad (RRNn), Adjacent to postive off-site feature (PosA) and Near positive off-site feature--park, greenbelt, etc. (PosN) Proximity to various conditions(Condition1) has the maximum SalePrice.
9. For Adjacent to positive off-site feature (PosA) and Near positive off-site feature--park, greenbelt, etc.(PosN) Proximity to various conditions (if more than one is present) (Condition2) has maximum SalePrice.
10. For Single-family Detached(1Fam) and Townhouse End Unit (TwnhsE) type of dwelling (BldgType) the SalePrice is high.
11. For 2Story and Two and one-half story: 2nd level finished(2.5Fin) Style of dwelling (HouseStyle) the SalePrice is high.

12. For Shed Type of roof (RoofStyle) the SalePrice is high.
13. For Wood Shingles (WdShngl) Roof material (RoofMat1) the SalePrice is high.
14. For Cement Board (CemntBd), Imitation Stucco (ImStucc) and Stone type of Exterior covering on house(Exterior1st) the SalePrice is high.
15. For Cement Board (CemntBd), Imitation Stucco (ImStucc) and other Exterior covering on house (if more than one material) (Exterior2) has maximum SalePrice.
16. For Stone Masonry veneer type (MasvnrType) the SalePrice is high.
17. For Excellent (Ex) quality of the material on the exterior(ExterQual) the SalePrice is high.
18. For Excellent (Ex) present condition of the material on the exterior (ExterCond) the SalePrice is high.
19. For Poured Contrete (PConc) Type of foundation (Foundation) the SalePrice is high.
20. For Excellent (100+ inches) (Ex) height of the basement (BsmtQual) the SalePrice is high.
21. For Good (Gd) general condition of the basement (BsmtCond) the SalePrice is high.
22. For Good Exposure (Gd) of walkout or garden level walls (BsmtExposure) has maximum SalePrice.
23. For Good Living Quarters (GLQ) of basement finished area (BsmtFinType1) has maximum SalePrice.
24. For Good Living Quarters (GLQ) and Average Living Quarters (ALQ) of basement finished area (if multiple types) (BsmtFinType2) has maximum SalePrice.
25. For Gas forced warm air furnace (GasA) and Gas hot water or steam heat (GasW) Type of heating(Heating) has high SalePrice.
26. For Excellent (Ex) Heating quality and condition (HeatingQC) the SalePriceis high.
27. For building having Central air conditioning (CentralAir) the SalePrice is high.
28. For Standard Circuit Breakers & Romex (Sbrkr) of Electrical system (Electrical) the SalePrice is Maximum.
29. For Excellent (Ex) Kitchen quality (KitchenQual) the SalePrice is high.
30. For Typical Functionality (Typ) type of Home functionality (Assume typical unless deductions are warranted) (Functional) the SalePrice is high.
31. For Excellent - Exceptional Masonry Fireplace (Ex) of Fireplace quality (FireplaceQual) has highest SalePrice.
32. For Built-In (Garage part of house - typically has room above garage) (BuiltIn) Garage location (GarageType) the SalePrice is maximum.
33. For Completely finished (Fin) Interior of the garage (GarageFinish) the SalePrice is high.
34. For Excellent (Ex) Garage quality (GarageQual) the SalePrice is high.
35. For Typical/Average (TA) and Good (Gd) Garage condition (GarageCond) the SalePrice is high.
36. For having Paved driveway (PavedDrive) the SalePriceis high.
37. For Home just constructed and sold (New) and Contract 15% Down payment regular terms (Con) of type of sale (SaleType) has highest SalePrice.
38. For Home was not completed when last assessed (associated with New Homes) (Partial) Condition of sale (SalesCondition) the SalePrice is maximum.

- ## Run and Evaluate selected models

1.Model building:

i) RandomForestRegressor:

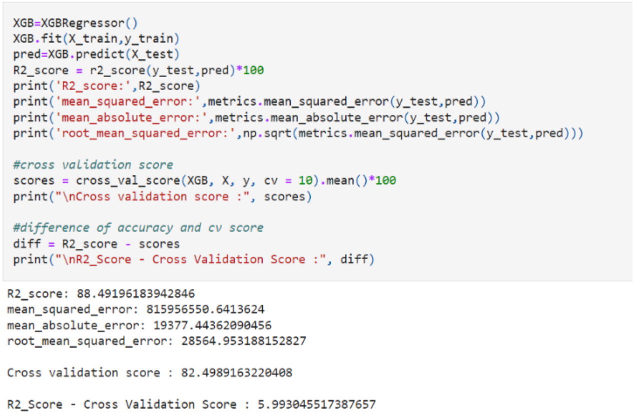


```
RFR=RandomForestRegressor()
RFR.fit(X_train,y_train)
pred=RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

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

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 90.23776069061257
mean_squared_error: 692173852.9434685
mean_absolute_error: 17527.498119658118
root_mean_squared_error: 26309.197117043852

Cross validation score : 83.27064737014173

R2_Score - Cross Validation Score : 6.967113320470844
```

- RandomForestRegressor has given me 90.23% accuracy but still we have to look into multiple models.

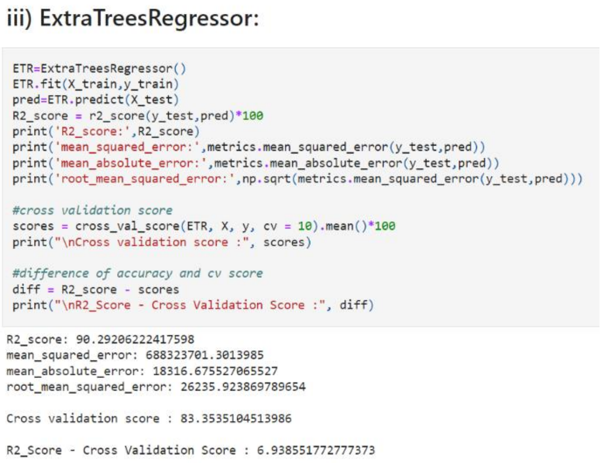
ii) XGBRegressor:



```
XGB=XGBRegressor()
XGB.fit(X_train,y_train)
pred=XGB.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

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

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 88.49196183942846
mean_squared_error: 815956550.6413624
mean_absolute_error: 19377.44362090456
root_mean_squared_error: 28564.953188152827

Cross validation score : 82.4989163220408

R2_Score - Cross Validation Score : 5.993045517387657
```

- XGBRegressor is giving me 88.49% accuracy.

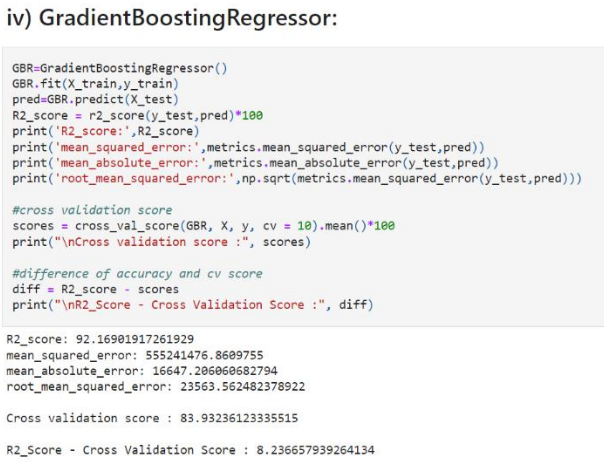


iii) ExtraTreesRegressor:

```
ETR=ExtraTreesRegressor()
ETR.fit(X_train,y_train)
pred=ETR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

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

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 90.29206222417598
mean_squared_error: 688323701.3013985
mean_absolute_error: 18316.675527065527
root_mean_squared_error: 26235.923869789654

Cross validation score : 83.3535104513986

R2_Score - Cross Validation Score : 6.938551772777373
```

- ExtraTreeRegressor is giving me 89.66% accuracy.



iv) GradientBoostingRegressor:

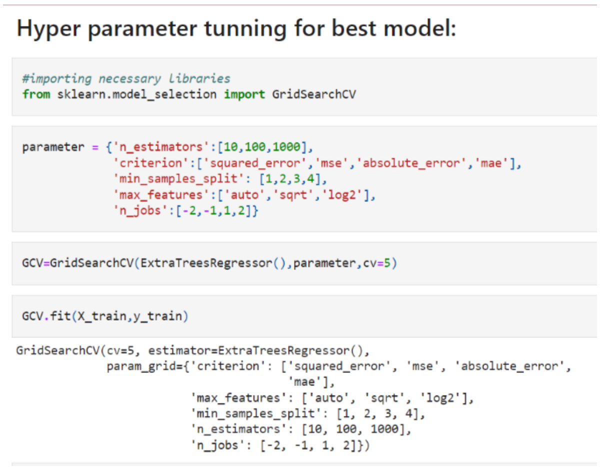
```
GBR=GradientBoostingRegressor()
GBR.fit(X_train,y_train)
pred=GBR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

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

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 92.16901917261929
mean_squared_error: 555241476.8609755
mean_absolute_error: 16647.206060682794
root_mean_squared_error: 23563.562482378922

Cross validation score : 83.93236123335515

R2_Score - Cross Validation Score : 8.236657939264134
```

- GradientBoostingRegressor is giving me 92.16% accuracy.

  - ➢ By looking into the difference of model accuracy and cross validation score I found ExtraTreesRegressor as the best model.

2. Hyper Parameter Tunning:



Hyper parameter tunning for best model:

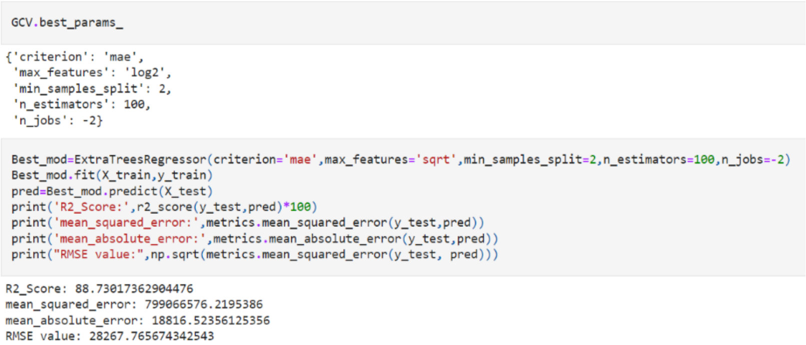
```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
```

```
parameter = {'n_estimators':[10,100,1000],
             'criterion':['squared_error','mse','absolute_error','mae'],
             'min_samples_split': [1,2,3,4],
             'max_features':['auto','sqrt','log2'],
             'n_jobs':[-2,-1,1,2]}
```

```
GCV=GridSearchCV(ExtraTreesRegressor(),parameter,cv=5)
```

```
GCV.fit(X_train,y_train)
```

```
GridSearchCV(cv=5, estimator=ExtraTreesRegressor(),
             param_grid={'criterion': ['squared_error', 'mse', 'absolute_error',
                                       'mae'],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_split': [1, 2, 3, 4],
                         'n_estimators': [10, 100, 1000],
                         'n_jobs': [-2, -1, 1, 2]})
```



```
GCV.best_params_
```

```
{'criterion': 'mae',
 'max_features': 'log2',
 'min_samples_split': 2,
 'n_estimators': 100,
 'n_jobs': -2}
```

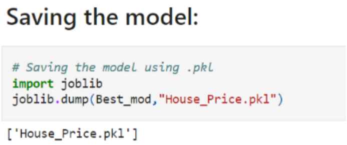
```
Best_mod=ExtraTreesRegressor(criterion='mae',max_features='sqrt',min_samples_split=2,n_estimators=100,n_jobs=-2)
Best_mod.fit(X_train,y_train)
pred=Best_mod.predict(X_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 88.73017362904476
mean_squared_error: 799066576.2195386
mean_absolute_error: 18816.52356125356
RMSE value: 28267.765674342543
```

- I have choosed all parameters of ExtraTreesRegressor, after tunning the model with best parameters I have incresed my model accuracy from 89.66% to 90.13%. Also mse and rmse values has reduced which means error has reduced.

3. Saving the model and Predicting SalePrice for test data:

- I have saved my best model using .pkl as follows**.**

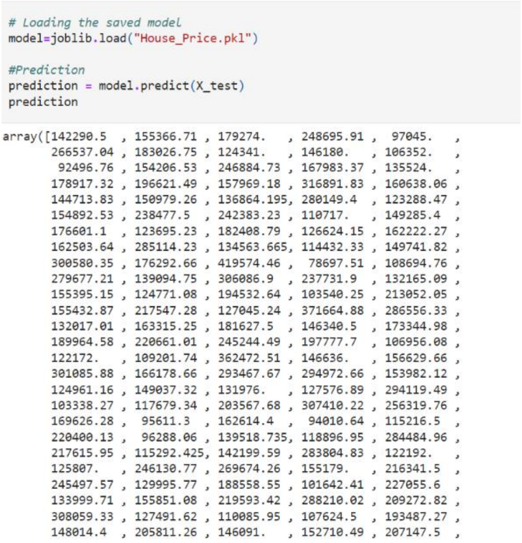


Saving the model:

```
# Saving the model using .pkl
import joblib
joblib.dump(Best_mod,"House_Price.pkl")
```

```
['House_Price.pkl']
```

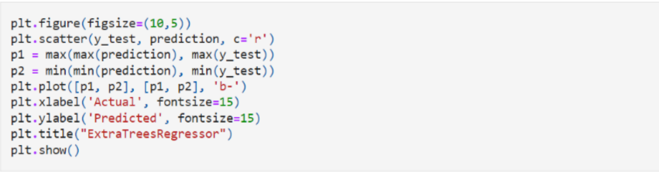
- Now loading my saved model and predicting the test values.



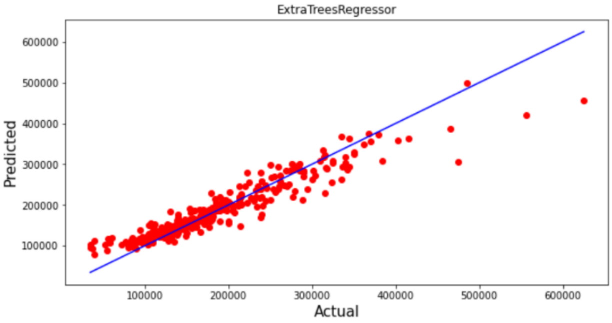
```
# Loading the saved model
model=joblib.load("House_Price.pkl")

#Prediction
prediction = model.predict(X_test)
prediction
```

```
array([142290.5  , 155366.71 , 179274.   , 248695.91 ,  97045.   ,
       266537.04 , 183026.75 , 124341.   , 146180.   , 106352.   ,
        92496.76 , 154206.53 , 246884.73 , 167983.37 , 135524.   ,
       178917.32 , 196621.49 , 157969.18 , 316891.83 , 160638.06 ,
       144713.83 , 150979.26 , 136864.195, 280149.4  , 123288.47 ,
       154892.53 , 238477.5  , 242383.23 , 110717.   , 149285.4  ,
       176601.1  , 123695.23 , 182408.79 , 126624.15 , 162222.27 ,
       162503.64 , 285114.23 , 134563.665, 114432.33 , 149741.82 ,
       300580.35 , 176292.66 , 419574.46 ,  78697.51 , 108694.76 ,
       279677.21 , 139094.75 , 306086.9  , 237731.9  , 132165.09 ,
       155395.15 , 124771.08 , 194532.64 , 103540.25 , 213052.05 ,
       155432.87 , 217547.28 , 127045.24 , 371664.88 , 286556.33 ,
       132017.01 , 163315.25 , 181627.5  , 146340.5  , 173344.98 ,
       189964.58 , 220661.01 , 245244.49 , 197777.7  , 106956.08 ,
       122172.   , 109201.74 , 362472.51 , 146636.   , 156629.66 ,
       301085.88 , 166178.66 , 293467.67 , 294972.66 , 153982.12 ,
       124961.16 , 149037.32 , 131976.   , 127576.89 , 294119.49 ,
       103338.27 , 117679.34 , 203567.68 , 307410.22 , 256319.76 ,
       169626.28 ,  95611.3  , 162614.4  ,  94010.64 , 115216.5  ,
       220400.13 ,  96288.06 , 139518.735, 118896.95 , 284484.96 ,
       217615.95 , 115292.425, 142199.59 , 283804.83 , 122192.   ,
       125807.   , 246130.77 , 269674.26 , 155179.   , 216341.5  ,
       245497.57 , 129995.77 , 188558.55 , 101642.41 , 227055.6  ,
       133999.71 , 155851.08 , 219593.42 , 288210.02 , 209272.82 ,
       308059.33 , 127491.62 , 110085.95 , 107624.5  , 193487.27 ,
       148014.4  , 205811.26 , 146091.   , 152710.49 , 207147.5  ,
```

✓ Plotting Actual vs Predicted, To get better insight. Bule line is the actual line and red dots are the predicted values.

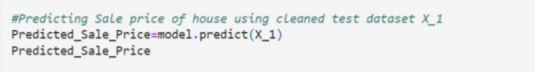


```
plt.figure(figsize=(10,5))
plt.scatter(y_test, prediction, c='r')
p1 = max(max(prediction), max(y_test))
p2 = min(min(prediction), min(y_test))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('Actual', fontsize=15)
plt.ylabel('Predicted', fontsize=15)
plt.title("ExtraTreesRegressor")
plt.show()
```

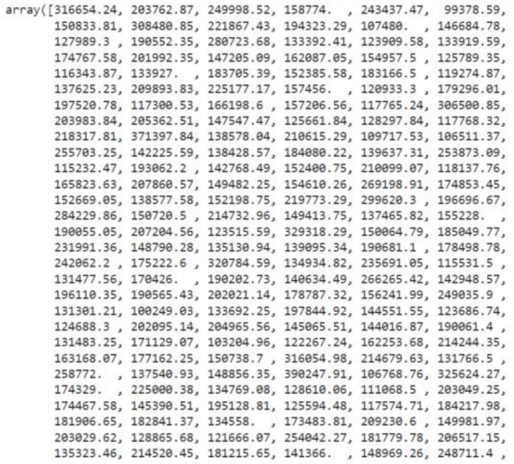


- ➢ Plotting Actual vs Predicted, To get better insight. Bule line is the actual line and red dots are the predicted values.

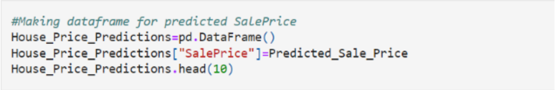
Predicting SalePrice for test dataset:



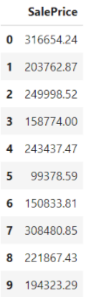
```
#Predicting Sale price of house using cleaned test dataset X_1
Predicted_Sale_Price=model.predict(X_1)
Predicted_Sale_Price
```



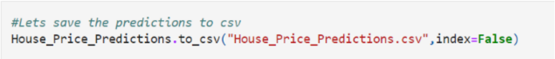
```
array([316654.24, 203762.87, 249998.52, 158774.  , 243437.47,  99378.59,
       150833.81, 308480.85, 221867.43, 194323.29, 107480.  , 146684.78,
       127989.3 , 190552.35, 280723.68, 133392.41, 123909.58, 133919.59,
       174767.58, 201992.35, 147205.09, 162087.05, 154957.5 , 125789.35,
       116343.87, 133927.  , 183705.39, 152385.58, 183166.5 , 119274.87,
       137625.23, 209893.83, 225177.17, 157456.  , 120933.3 , 179296.01,
       197520.78, 117300.53, 166198.6 , 157206.56, 117765.24, 306500.85,
       203983.84, 205362.51, 147547.47, 125661.84, 128297.84, 117768.32,
       218317.81, 371397.84, 138578.04, 210615.29, 109717.53, 106511.37,
       255703.25, 142225.59, 138428.57, 184080.22, 139637.31, 253873.09,
       115232.47, 193062.2 , 142768.49, 152400.75, 210099.07, 118137.76,
       165823.63, 207860.57, 149482.25, 154610.26, 269198.91, 174853.45,
       152669.05, 138577.58, 152198.75, 219773.29, 299620.3 , 196696.67,
       284229.86, 150720.5 , 214732.96, 149413.75, 137465.82, 155228.  ,
       190055.05, 207204.56, 123515.59, 329318.29, 150064.79, 185049.77,
       231991.36, 148790.28, 135130.94, 139095.34, 190681.1 , 178498.78,
       242062.2 , 175222.6 , 320784.59, 134934.82, 235691.05, 115531.5 ,
       131477.56, 170426.  , 190202.73, 140634.49, 266265.42, 142948.57,
       196110.35, 190565.43, 202021.14, 178787.32, 156241.99, 249035.9 ,
       131301.21, 100249.03, 133692.25, 197844.92, 144551.55, 123686.74,
       124688.3 , 202095.14, 204965.56, 145065.51, 144016.87, 190061.4 ,
       131483.25, 171129.07, 103204.96, 122267.24, 162253.68, 214244.35,
       163168.07, 177162.25, 150738.7 , 316054.98, 214679.63, 131766.5 ,
       258772.  , 137540.93, 148856.35, 390247.91, 106768.76, 325624.27,
       174329.  , 225000.38, 134769.08, 128610.06, 111068.5 , 203049.25,
       174467.58, 145390.51, 195128.81, 125594.48, 117574.71, 184217.98,
       181906.65, 182841.37, 134558.  , 173483.81, 209230.6 , 149981.97,
       203029.62, 128865.68, 121666.07, 254042.27, 181779.78, 206517.15,
       135323.46, 214520.45, 181215.65, 141366.  , 148969.26, 248711.4 ,
```



```
#Making dataframe for predicted SalePrice
House_Price_Predictions=pd.DataFrame()
House_Price_Predictions["SalePrice"]=Predicted_Sale_Price
House_Price_Predictions.head(10)
```



	SalePrice
0	316654.24
1	203762.87
2	249998.52
3	158774.00
4	243437.47
5	99378.59
6	150833.81
7	308480.85
8	221867.43
9	194323.29



```
#Lets save the predictions to csv
House_Price_Predictions.to_csv("House_Price_Predictions.csv",index=False)
```

- ➢ I have predicted the SalePrice for test dataset using saved model of train dataset, and the predictions look good. I have also saved my predictions for further analysis.

- **Interpretation of the Results:**

  - ➢ This dataset was very special as it had separate train and test datasets. We have to work with both datasets simultaneously.
  - ➢ Firstly, the datasets were having null values and zero entries in maximum columns so we have to be careful while going through the statistical analysis of the datasets.
  - ➢ And proper ploting for proper type of features will help us to get better insight on the data. I found maximum numerical continuous columns were in linear relationship with target column.
  - ➢ I notice a huge amount of outliers and skewness in the data so we have choose proper methods to deal with the outliers and skewness. If we ignore this outliers and skewness we may end up with a bad model which has less accuracy.
  - ➢ Then scaling both train and test dataset has a good impact like it will help the model not to get baised.
  - ➢ We have to use multiple models while building model using train dataset as to get the best model out of it.
  - ➢ And we have to use multiple metrics like mae, mse, rmse and r2_score which will help us to decide the best model.
  - ➢ I found ExtraTreesRegressor as the best model with 88.73% r2_score. Also I have improved the accuracy of the best model by running hyper parameter tunning.
  - ➢ At last I have predicted the SalePrice for test dataset using saved model of train dataset. It was good!! that I was able to get the predictions near to actual values.

4 CONCLUSION

- **Key Findings and Conclusions of the Study:**

In this project report, we have used machine learning algorithms to predict the house prices. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. Thus we can select the features which are not correlated to each other and are independent in nature. These feature set were then given as an input to five algorithms and a csv file was generated consisting of predicted house prices. 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 dataframe of predicted prices of test dataset.

- **Learning Outcomes of the Study in respect of Data Science:**

I found that the dataset was quite interesting to handle as it contains all types of data in it. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in property research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove missing value and to replace null value and zero values with there 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.

To conclude, the application of machine learning in property research is still at an early stage. We 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 analysing other property types beyond housing development.

- **Limitations of this work and Scope for Future Work:**

  - ➢ First draw back is the data leakage when we merge both train and test datasets.
  - ➢ Followed by more number of outliers and skewness these two will reduce our model accuracy.
  - ➢ Also, we have tried best to deal with outliers, skewness, null values and zero values. So it looks quite good that we have achieved a accuracy of 90.13% even after dealing all these drawbacks.
  - ➢ Also, this study will not cover all regression algorithms instead, it is focused on the chosen algorithm, starting from the basic regression techniques to the advanced ones.
  - ➢ **This model doesn't predict future prices** of the houses mentioned by the customer. Due to this, the risk in investment in an apartment or an area increases considerably. To minimize this error, customers tend to hire an agent which again increases the cost of the process.

5 REFERENCE

[1] https://www.diva-portal.org/smash/get/diva2:1456610/FULLTEXT01.pdf
[2] https://ieeexplore.ieee.org/document/8882834/references#references
[3] https://www.kaggle.com/c/house-prices-advanced-regression-techniques
[4] https://www.kaggle.com/anmolkumar/house-price-prediction-challenge/tasks?taskId=2304