FLIP ROBO

Project Name:-

"HOUSING: PRICE PREDICTION"

Submitted By:-

DIPTIRANJAN PRADHAN

ACKNOWLEDGEMENT:-

I would like to express my special gratitude to the “Flip Robo” team, who has given me thisopportunity to deal with a beautiful dataset and it has helped me to improve my analysing skills. Also, I want to express my huge gratitude to Mrs . Sapna Verma (SME FlipRobo), who has helped me overcome difficulties within this project and others. and also encouraged me a lot with his valuable words and with his unconditional support I have ended up with a beautiful Project.

A huge thanks to my academic team “Data trained” who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life.

And also thank you for many other persons who has helped me directly or indirectly to complete the project.

CONTENTS:-

1.INTRODUCTION:-

➢ Business Problem Framing:-

Houses are one of the necessary need 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 largemarket 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.
House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house. 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 improve Real Estate efficiency. The aim is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analysing previous market trends and price ranges, and also upcoming developments future prices will be predicted.
cost of property depending on number of attributes considered.

Now as a data scientist our work is to analyse the dataset and apply our skills towards predicting house price.

1.2 Conceptual Background of the Domain Problem:-

The real estate market is one of the most competitive in terms of pricing andsame tends to vary significantly based on numerous factors; forecasting property price is an important module in decision making for both the buyersand investors in supporting budget allocation, finding property finding stratagems and determining suitable policies.

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?

✓ 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 improve Real Estate efficiency**.**

There are three factors that influence the price of a house which include physical conditions, concept and location. Hence it becomes one of the prime fields to apply the concepts of machine learning to optimize and predict the prices with high accuracy. Therefore, in this project report we present various
important features to use while predicting housing prices with good accuracy.While using features in a regression model some feature engineering is required for better prediction.

➢ *1.3 Review of Literature:-*

The factors that affect the land price have to be studied and their impact on price has also to be modelled. An analysis of the past data is to be considered. 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 analyse and forecast future trends. As the real estate is fast developing sector,the analysis and forecast of land prices using mathematical modelling and other scientific techniques is an immediate urgent need for decision making by all those concerned.

The increase in population as well as the industrial activity is attributed to various factors, the most prominent being the recent spurt in the knowledge sector viz. Information Technology (IT) and Information technology enabled services. Demand for land started of showing an upward trend and housing and the real estate activity started booming. The need for predicting the
trend in land prices was felt by all in the industry viz. the Government, the regulating bodies, lending institutions, the developers and the investors. 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 andcurate 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.

➢ 1.4 Motivation for the Problem Undertaken:-

I have to 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.

2.ANALYTICAL PROBLEM FRAMING:-

2.1 Mathematical/ Analytical Modeling of the Problem :-

This perticular 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.

2.2 Data Sources and their formats:-

The data was given by my internship company – Flip Robo technologies in csv (comma separated values) format.

Here I was having two datasets one is train and other is test. I have built model using train dataset and predicted Sale Price for test dataset. 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. In this particular datasets I have object, float and integer types of data. I can merge these two datasets and perform my analysis, but I have not done that because of data leakage issue. This is how my datasets looks for me when I import those datasets to my python.

2.3 Data Preprocessing Done:-

- As a first step I have imported required libraries and I have importedboth the datasets which were in csv format.
- Then I did all the statistical analysis like checking shape, nunique, valuecounts, info etc.....
- While checking the info of the datasets I found some columns with more than 80% null values, so these columns will create skewness in datasets.so I decided to drop those columns.
- Then while looking into the value counts I found some columns withmore than 85% zero values this also creates skewness in the model andthere are chances of getting model bias so I have dropped thosecolumns with more than 85% zero values.
- While checking for null values I found null values in most of the columnsand I have used imputation method to replace those null values (modefor categorical column and mean for numerical columns).
- In Id and Utilities column the unique counts were 1168 and 1 respectively, which means all the entries in Id column are unique and ID is the identity number given for perticular asset and all the entries in Utilities column were same so these two column will not help us in model building. So I decided to drop those columns.

- Next as a part of feature extraction I converted all the year columns to there respective age. Thinking that age will help us more than year.
- And all these steps were performed to both train and test datasets separately and simultaneously.

2.4 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.

2.5 Hardware and Software Requirements and Tools Used:-

While taking up the project we should be familiar with the Hardware and software required for the successful completion of the project. Here we need the following hardware and software…

Hardware required: -

1. Processor — core i5 and above
2. RAM — 4 GB or above
3. SSD — 250GB or above

Software required:-

Anaconda

Libraries required :-

Import all the required library.

```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        import warnings
        warnings.filterwarnings('ignore')
```

- import pandas as pd:-

  pandas is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.

- import numpy as np:-

  NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

- Import seaborn as sns:-

  Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

- Import matplotlib.pyplot as plt:-

  matplotlib.pyplot is a collection offunctions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

some other libraries which are used:-

- ✓ from sklearn.preprocessing import OrdinalEncode
- ✓ from sklearn.preprocessing import StandardScaler
- ✓ from statsmodels.stats.outliers_influence import variance_inflation_factor
- ✓ from sklearn.ensemble import RandomForestRegressor
- ✓ from sklearn.tree import DecisionTreeRegressor
- ✓ from xgboost import XGBRegressor

- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.ensemble import ExtraTreesRegressor
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.model_selection import cross_val_score

3.DATA ANALYSIS AND VISUALIZATION:-

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

- ✓ Here I have used imputation method to replace null values.

```
In [29]: #Dropping unnecessary columns in test dataset
         dff=dff.drop(["Alley"],axis=1)
         dff=dff.drop(["PoolQC"],axis=1)
         dff=dff.drop(["Fence"],axis=1)
         dff=dff.drop(["MiscFeature"],axis=1)
```

- ✓ For remove outliers I have used percentile method. And to remove skewness I have used yeo-johnson method.

iii) Percentile Method:-

```
In [135]: for col1 in features1:
              if dff[col1].dtypes != 'object':
                  percentile = dff[col1].quantile([0.01,0.98]).values
                  dff[col][dff[col]<=percentile[0]]=percentile[0]
                  dff[col][dff[col]>=percentile[1]]=percentile[1]
```

- ✓ To encode the categorical columns I have use Ordinal Encoding.
- ✓ And also I have used standardization. Then followed by model building with all regressionalgorithms.

3.2 Testing of Identified Approaches (Algorithms):-

As we know Saleprice was my target and it was a continuous column so thisproblem was regression problem. Here 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. 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:-

- ✓ **RandomForest Regressor**
- ✓ **ExtraTreesRegressor**

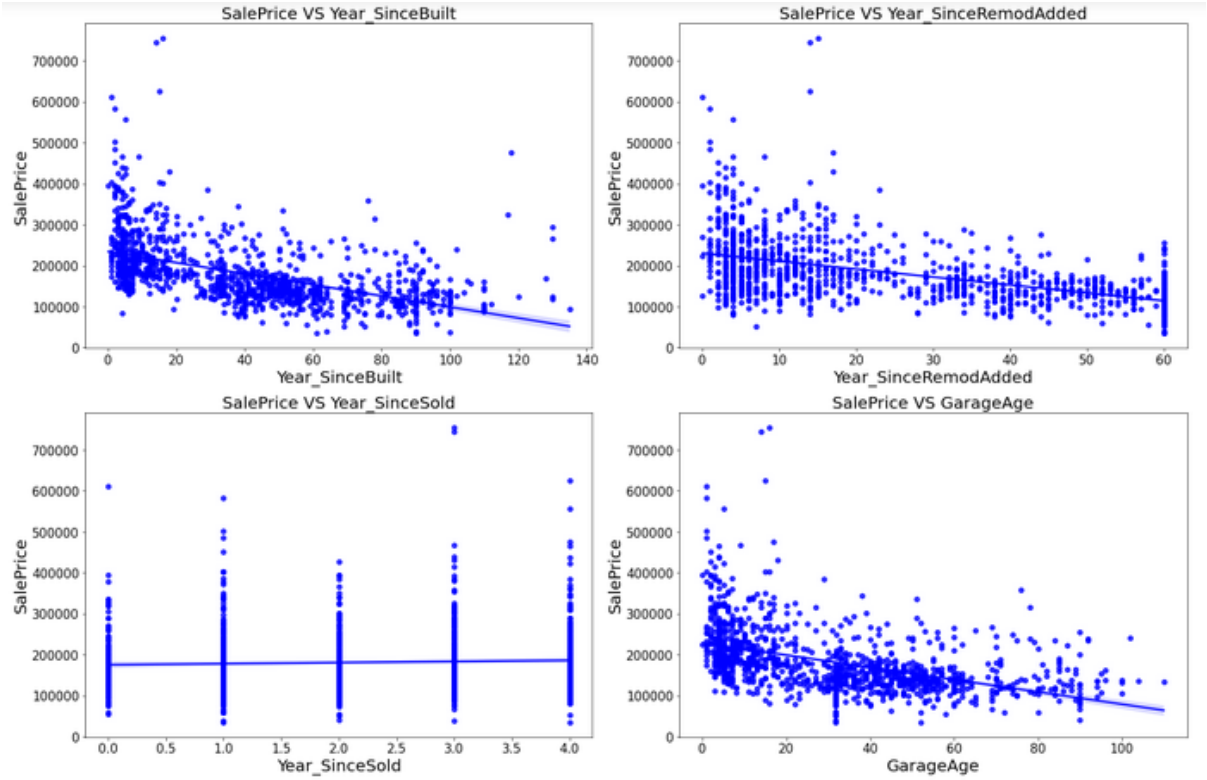
- ✓ **GradientBoostingRegressor**
- ✓ **DecisionTreeRegressor**

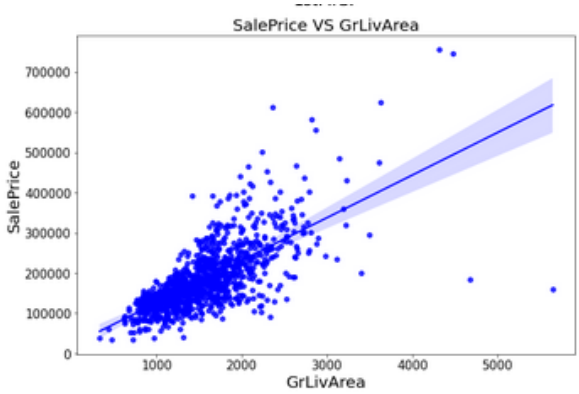
3.3 Key Metrics for success in solving problem under consideration:-

- Here I have used mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.
- I have used r2 score which tells us how accurate our model is.
- I have used root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.

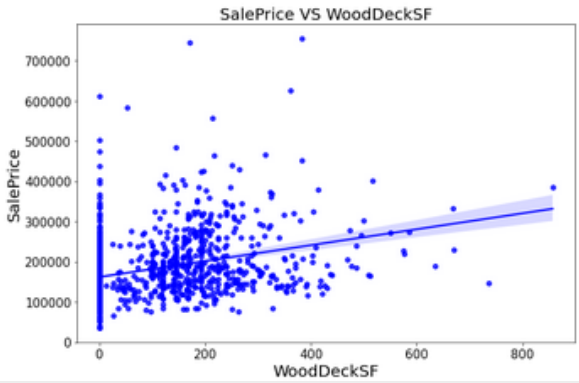
3.4 Data Visualizations:-

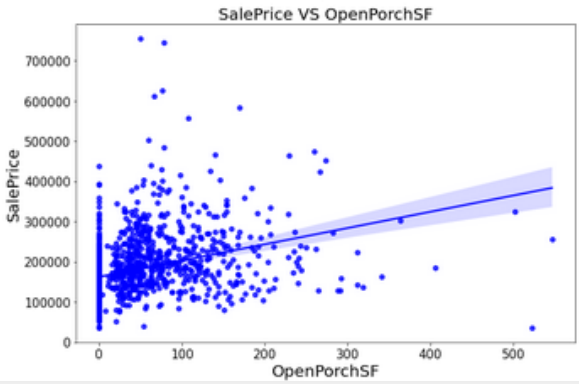
Vizualization of numerical features with target:-

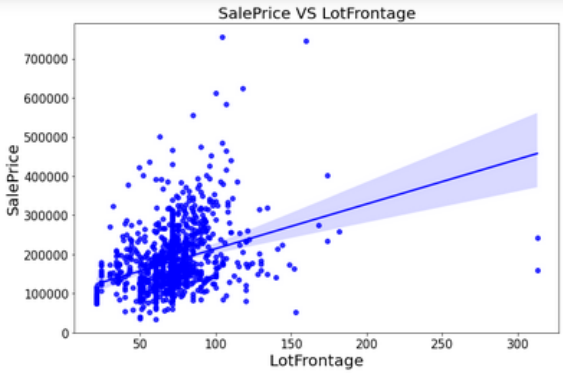


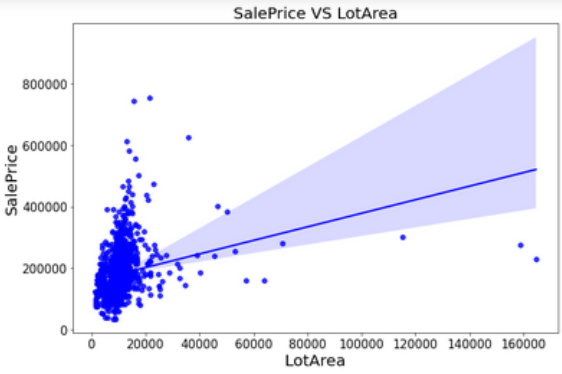

SalePrice VS GrLivArea
SalePrice
GrLivArea

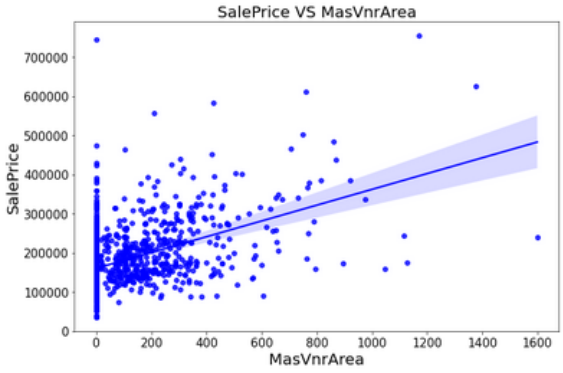

SalePrice VS GarageArea
SalePrice
GarageArea

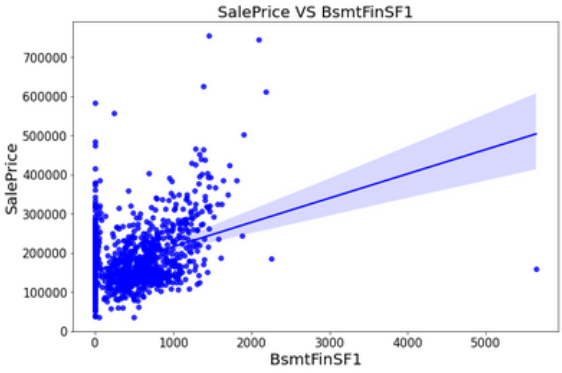

SalePrice VS WoodDeckSF
SalePrice
WoodDeckSF

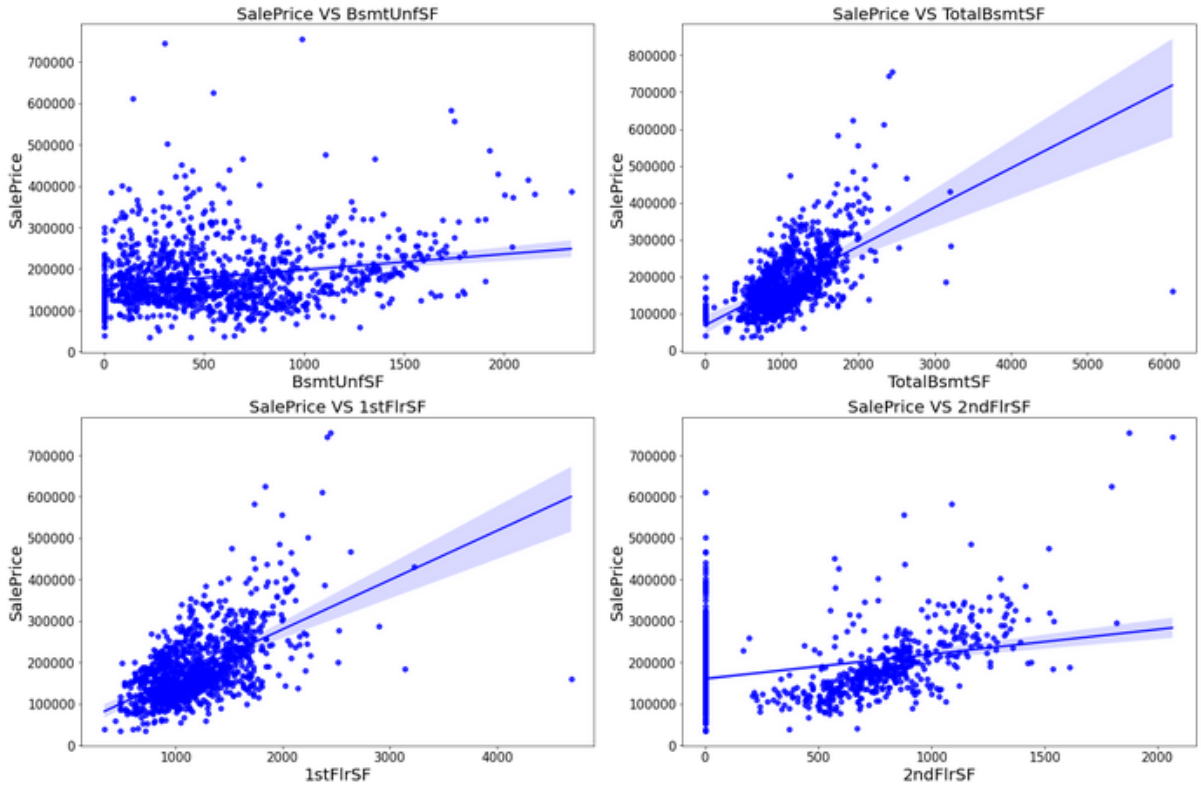

SalePrice VS OpenPorchSF
SalePrice
OpenPorchSF


SalePrice VS LotFrontage
SalePrice
LotFrontage


SalePrice VS LotArea
SalePrice
LotArea


SalePrice VS MasVnrArea
SalePrice
MasVnrArea


SalePrice VS BsmtFinSF1
SalePrice
BsmtFinSF1



Obervation:-

From Above fig we can see that:-

1.As Linear feet of street connected to property(LotFrontage) is increseing sales is decreasing and the SalePrice is rangeing between 0-3 lakhs.

2.As Lot size in square feet(LotArea) is increseing 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 rangeing between 0-4 lakhs.

4.As Type 1 finished square feet(BsmtFinSF1) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

5.As Unfinished square feet of basement area(BsmtUnfSF) is increseing 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 increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

7.As First Floor square feet(1stFlrSF) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

8.As Second floor square feet(2ndFlrSF) is increseing 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 increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

10.As Size of garage in square feet(GarageArea) is increseing sales is increseing and the saleprice is in between 0-4 lakhs.

11.As Wood deck area in square feet(WoodDeckSF) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

12.As Open porch area in square feet(OpenPorchSF) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

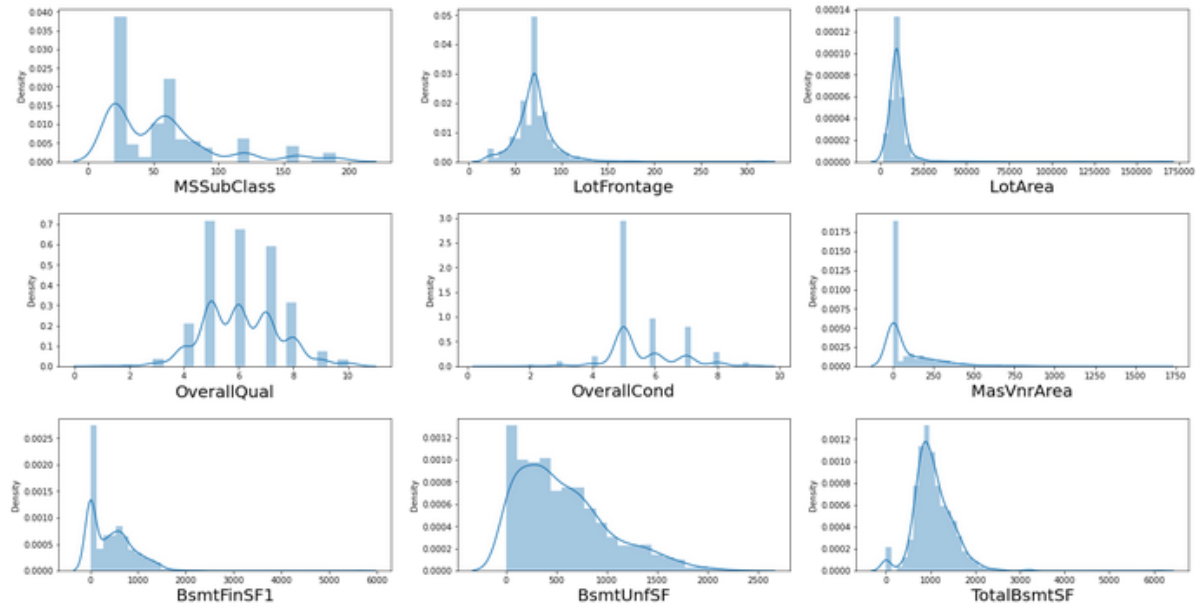
13.As Year_SinceBuilt is increseing sales is decreasing and the saleprice is high for newly built building and the sales price is in between 0-4 lakhs.

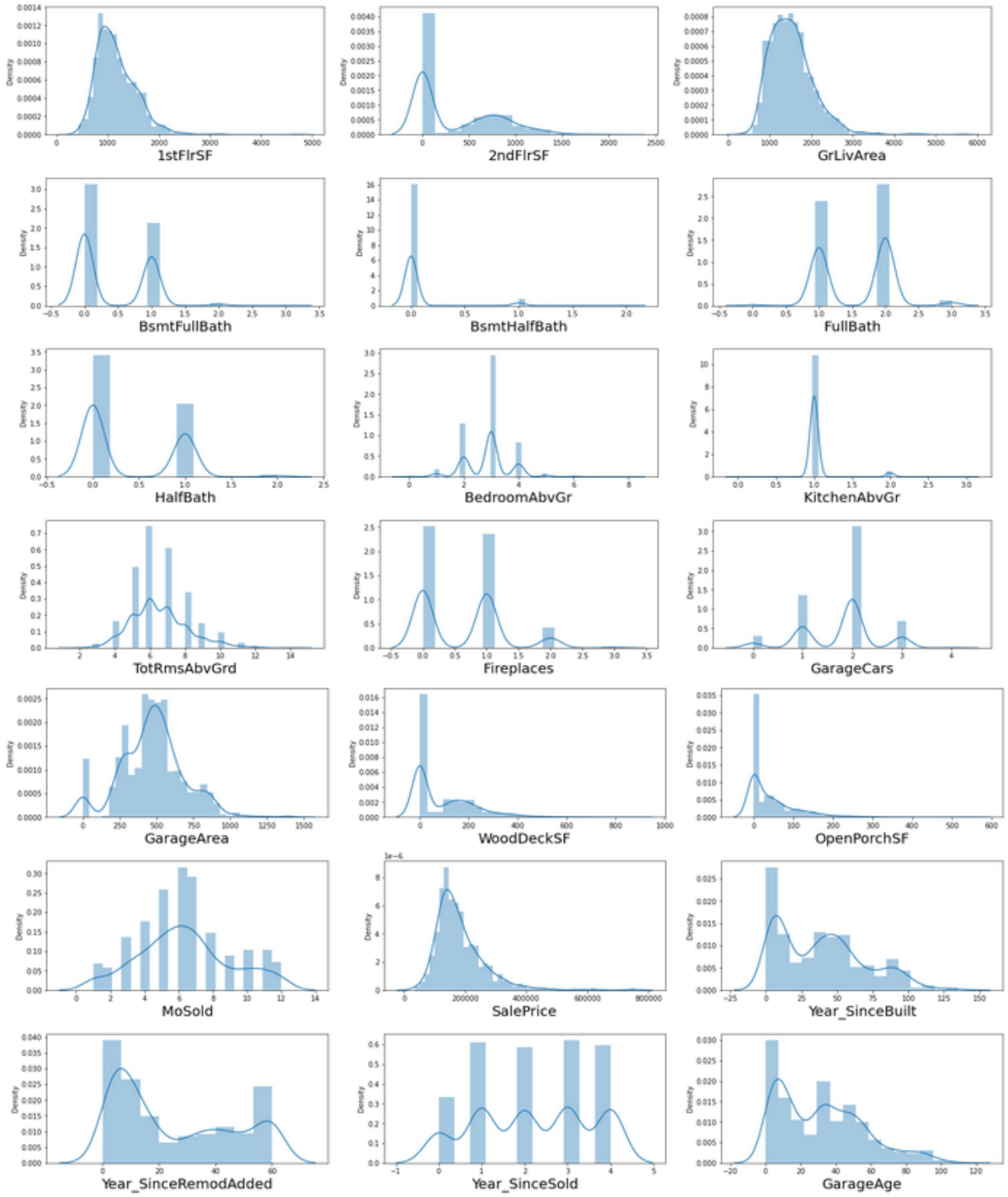
14.As Since Remodel date (same as construction date if no remodeling or additions)(Year_SinceRemodAdded) is increseing sales is decreasing and the saleprice is in between 1-4 lakhs.

15.As Since Year garage was built(GarageAge) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

For Numerical columns:-

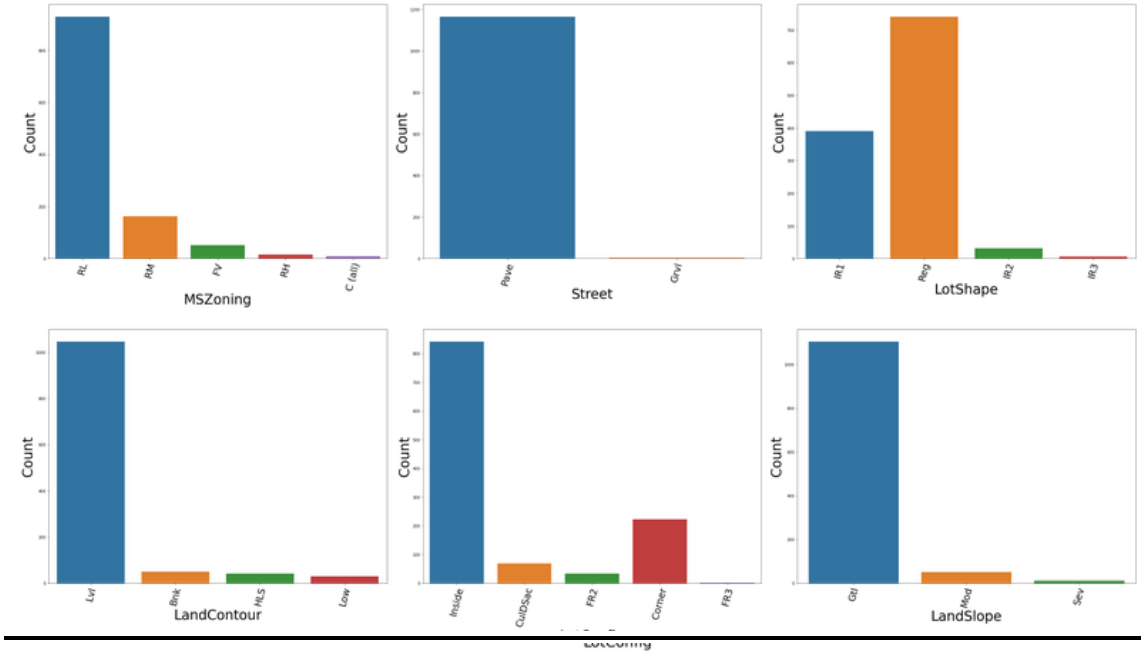
```
plt.figure(figsize = (20,40))
plotnumber = 1
for column in df[numerical_columns]:
    if plotnumber <=35:
        ax = plt.subplot(12,3,plotnumber)
        sns.distplot(df[column])
        plt.xlabel(column,fontsize = 20)
    plotnumber+=1
plt.tight_layout()
```

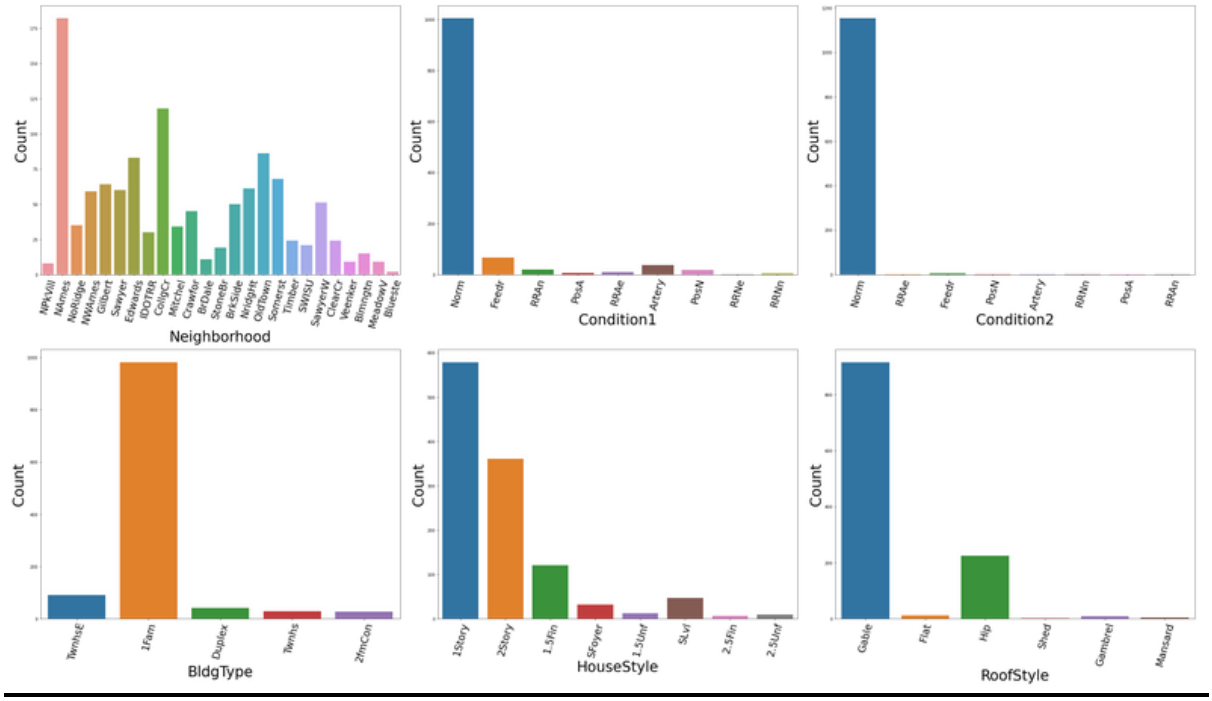


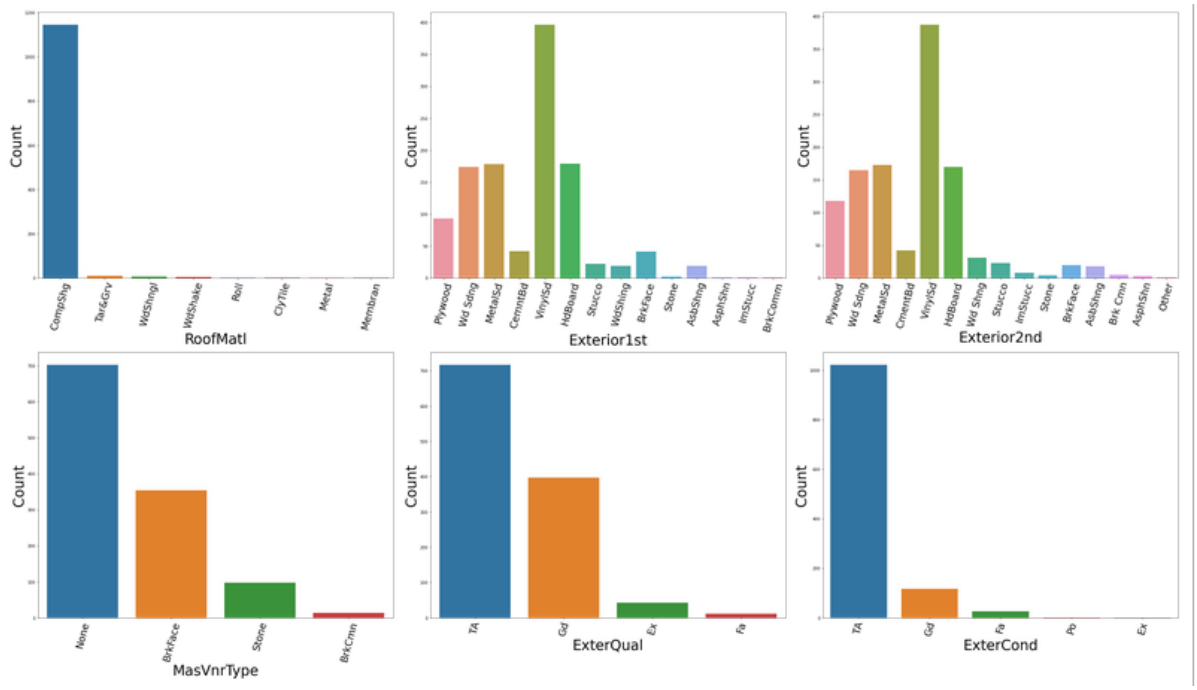


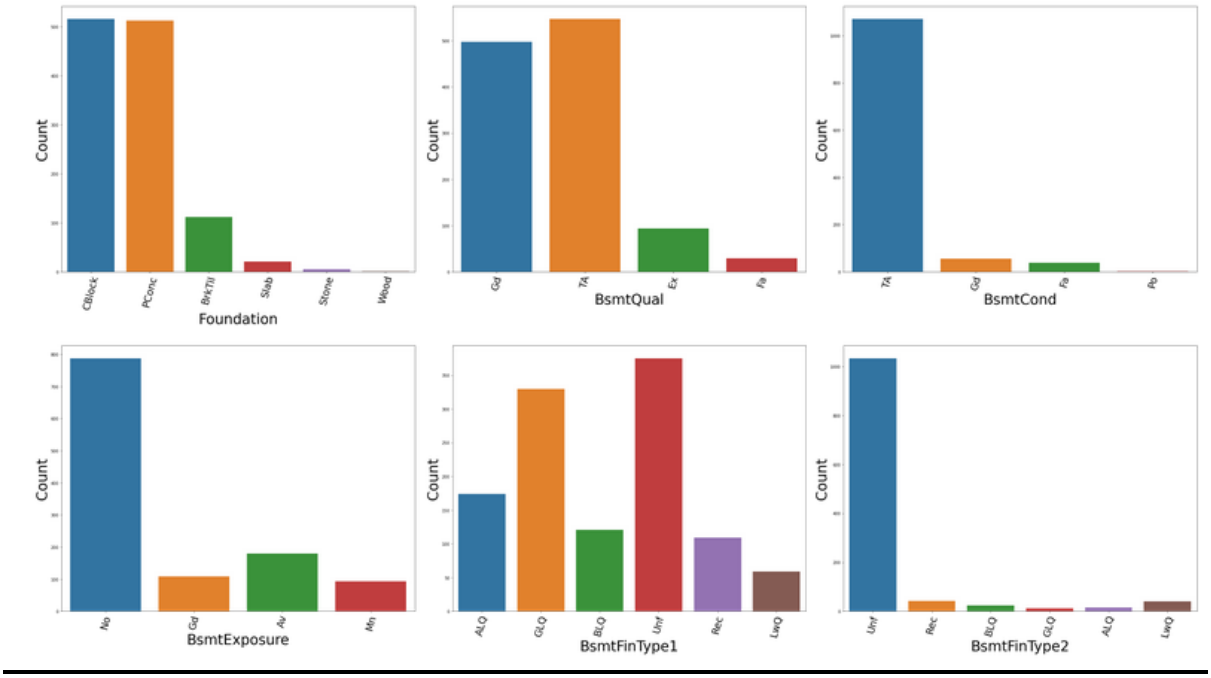
As we can see there is skewness in almost all numerical columns.I have to remove this skewness.

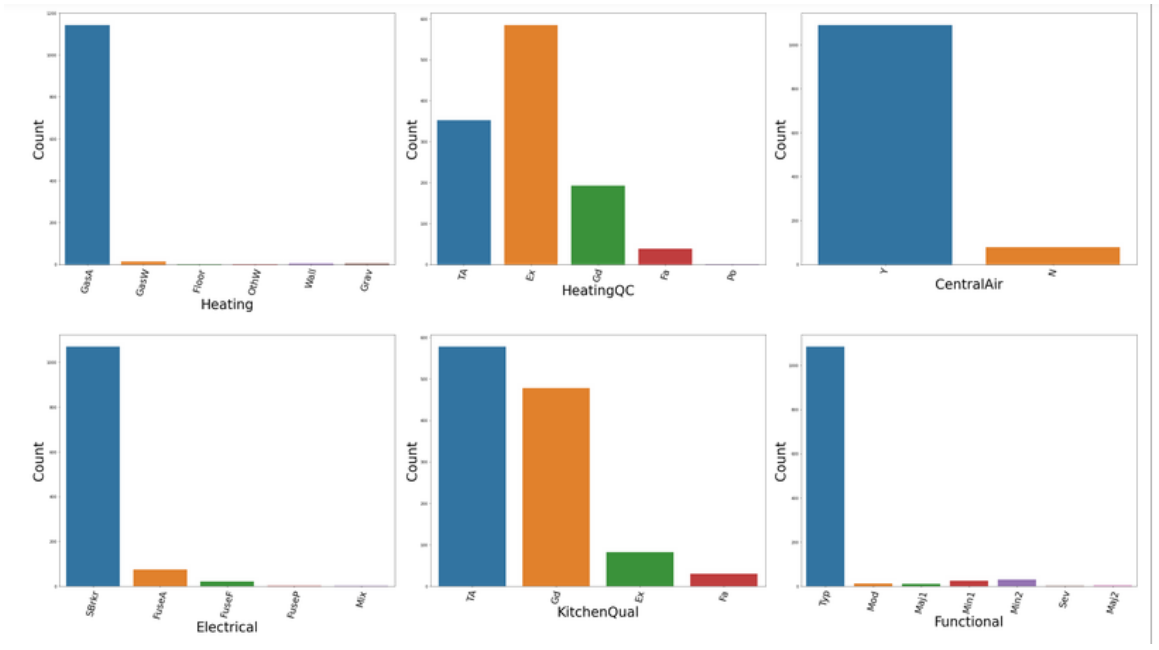
2. Visualization of categorical features with target:-

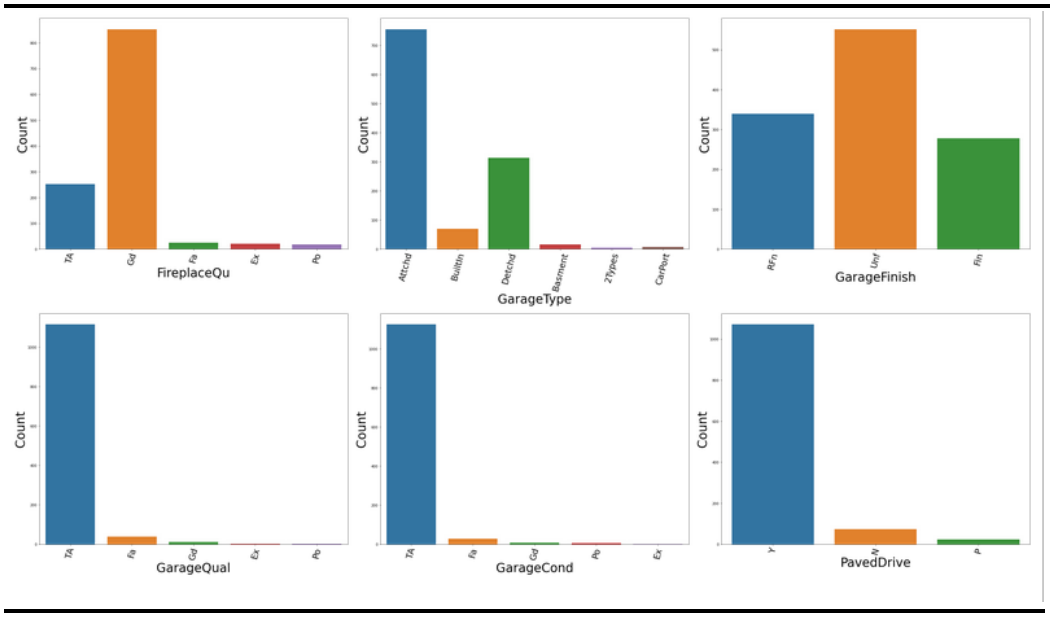


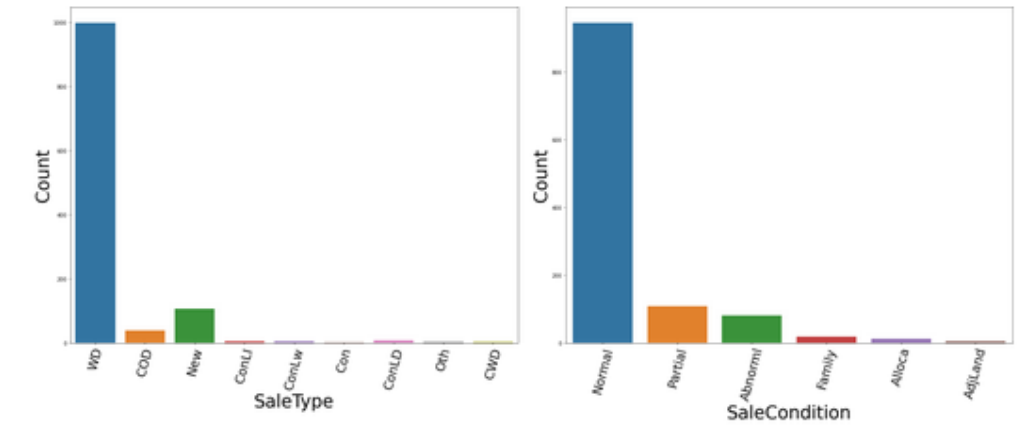



Count
CompShg
Tar&Grv
WdShngl
WdShake
Roll
ClyTile
Metal
Membran
RoofMatl
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
Exterior1st
Plywood
Wd Sdng
MetalSd
CmentBd
VinylSd
HdBoard
Wd Shng
Stucco
ImStucc
Stone
BrkFace
AsbShng
Brk Cmn
AsphShn
Other
Exterior2nd
None
BrkFace
Stone
BrkCmn
MasVnrType
TA
Gd
Ex
Fa
ExterQual
TA
Gd
Fa
Po
Ex
ExterCond


Count
CBlock
PConc
BrkTil
Slab
Stone
Wood
Foundation
Gd
TA
Ex
Fa
BsmtQual
TA
Gd
Fa
Po
BsmtCond
No
Gd
Av
Mn
BsmtExposure
ALQ
GLQ
BLQ
Unf
Rec
LwQ
BsmtFinType1
Unf
Rec
BLQ
GLQ
ALQ
LwQ
BsmtFinType2


Count
GasA
GasW
Floor
OthW
Wall
Grav
Heating
TA
Ex
Gd
Fa
Po
HeatingQC
Y
N
CentralAir
SBrkr
FuseA
FuseF
FuseP
Mix
Electrical
TA
Gd
Ex
Fa
KitchenQual
Typ
Mod
Maj1
Min1
Min2
Sev
Maj2
Functional


Count
TA
Gd
Fa
Ex
Po
FireplaceQu
Attchd
Builtin
Detchd
Basment
2Types
CarPort
GarageType
RFn
Unf
Fin
GarageFinish
TA
Fa
Gd
Ex
Po
GarageQual
TA
Fa
Gd
Po
Ex
GarageCond
Y
N
P
PavedDrive


Count
WD
COD
New
ConLI
ConLw
Con
ConLD
Oth
CWD
SaleType
Normal
Partial
Abnorml
Family
Alloca
AdjLand
SaleCondition

Observations:-

From above observation we can see that:-

1)In MSZoning we can see that Residential Low Density zoning has maximum count, and the least count is commercial.

2)In street more people have used pave type of road access to the peoperty than gravel.

3)In Lotshape Regular shaped property has maximum count.

4)In Landcolor Near Level property has maximum count,compare to other.

5)In LotConfig Inside lot configured property has maximum count and very low number of people choses frontage of 3 sides of property.

6)In Landslope gentle slope is having more count and least is severe slope.

7)In Neighbourhood wecan see If the property is located in North Ames then count is good compared to other locations.

8)IN condition1&2 Normal condition is having more count for both condition1 and condition2 which is approximity to various condition.

9)IN Bidgtype we can see single family detached is preferred by more people and a very few people prefer townhouse inside.

10)In Housestyle we can see One story dwelling housestyle has maximum count.

11)In roofstyle,Gable roof style's count is high.

12)In RoofMati standard composite shingle is having more count as the roof material and roll and membrane is having the least count.

13)In Exterior1st & Exterior2nd-most of the houses in the dataset is having vinyl siding as the exterior covering on house and asphalt shingles imitation shicco, brick common have the least count.

14)IN MasVnrType None has maximum count & very few have brick common.

15)In Exterqual & Extercontthe quality of exterior material is average or typical for both number of houses and very few with fair quality. Excellent and poor is having the least countfor the present condition of exterior material.

16)In foundation Cinder Block and Poured Contrete foundations the count is maximum.

17)In BsmtQual good and average quality heights of the basement the count is high.

18)IN BsmtCond slight dampness is having more count and the least count is for poor severe cracking or settling wetness basement.

19)In SMTExposure no exposure is having the more count and minimum exposure is the least for walkout or garden level walls.

20)In BsmtFinType1 unfinished Rating of basement finished area-1 the count is maximum.

21)In BsmtFinType2 unfinished Rating of basement finished area-2 the count is maximum.

22)In Heating gas A is having more count.

23)In Heating QC Excellent Heating quality and condition the count is high for the feature Heating quality and condition.

24)In Central air conditioning-yes has maximum count.

25)In Electrical, standard circute breaker is used by more houses and a very few houses uses mixed type of electrical system.

26)In Kitchen Qual, Typical/Average(TA) and good Kitchen quality the count is maximum.

27)In functional, Typical Functionality has highest count for Home functionality.

28)In FireplaceQu, good Fireplace quality the count is high .

29)In Garagetype ,attatched to home type of garage is having the more count in the dataset than others.

30)In Garage finish, Unfinished Interior of the garage the count is maximum.

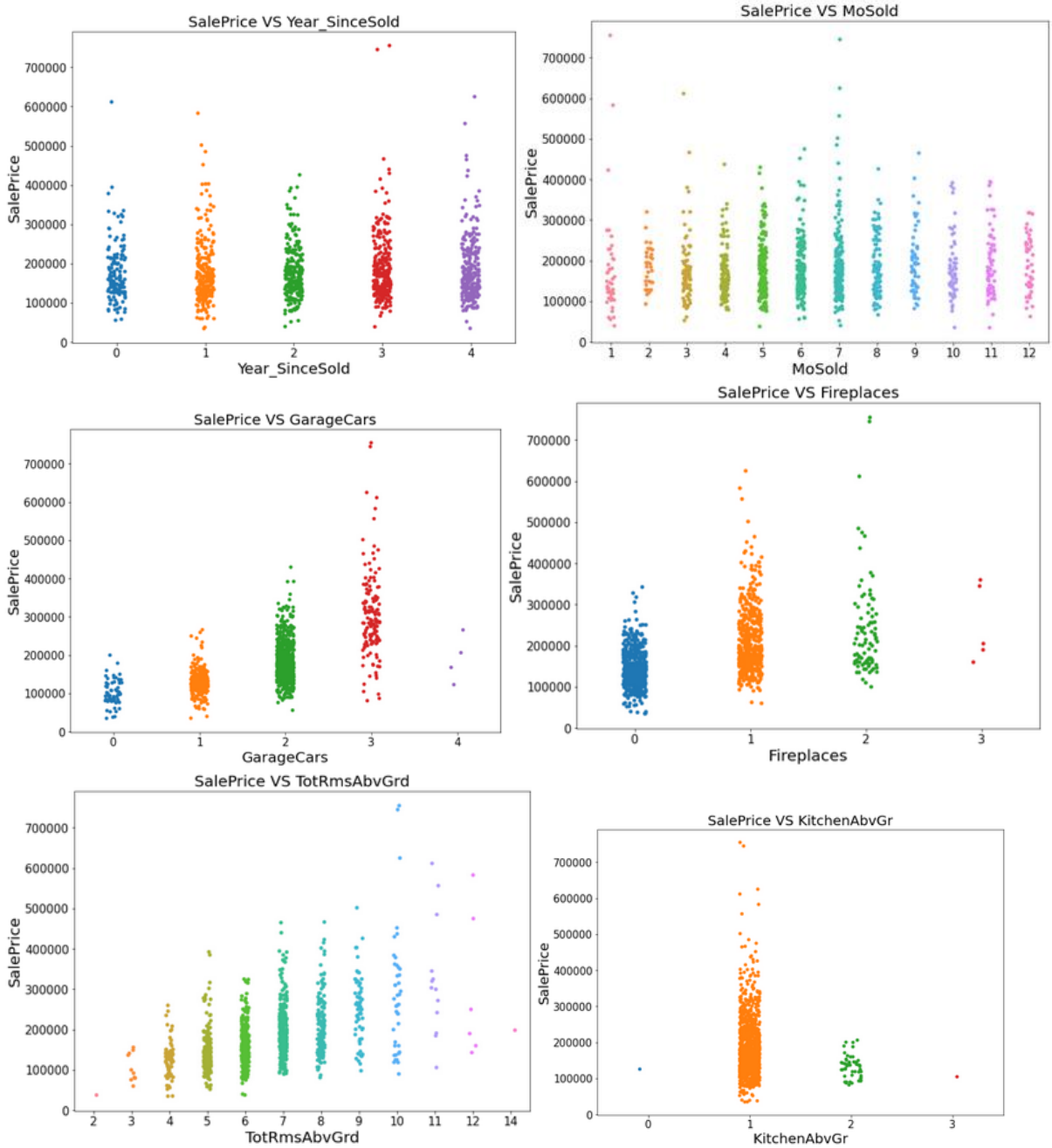
31)In GarageQual, Typical/Average(TA) Garage quality the count is high.

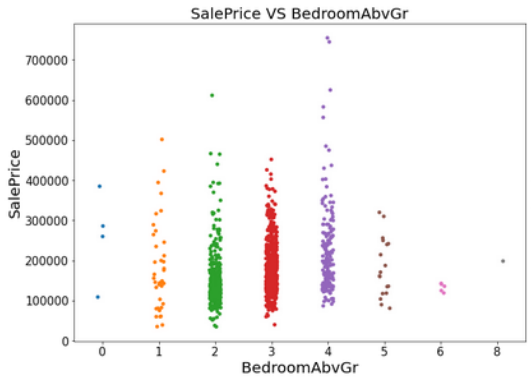
32)In Garage cond ,Typical/Average(TA) Garage condition the count is high.

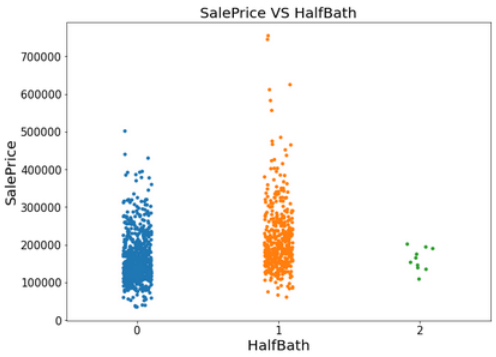
33)In PavadDrive ,Paved driveway the count is maximum.

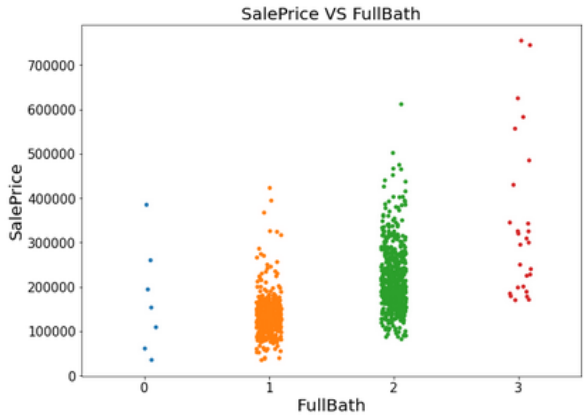
34)In Sale type , Warranty Deed - Conventional type of sales the count is maximum compare to others.

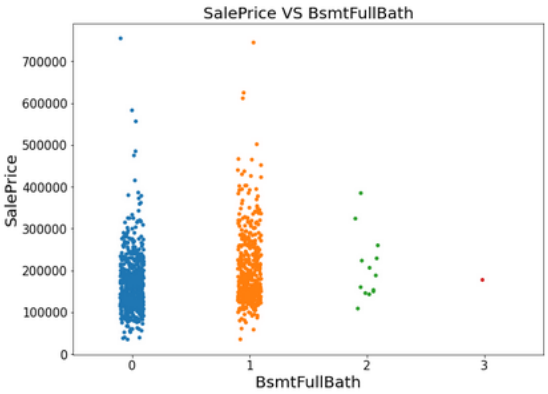
35)In Saleconditio,Normal sales condition the count is high.

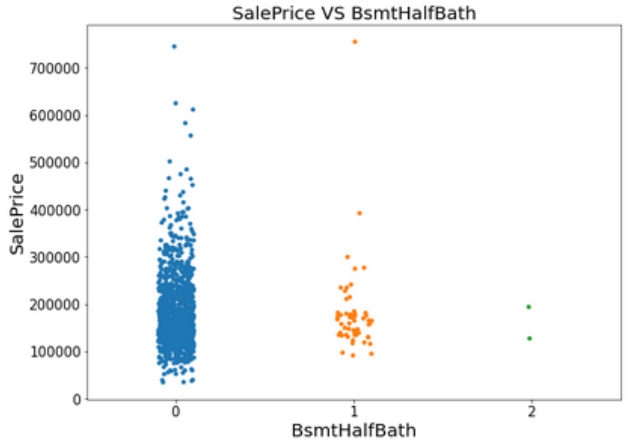


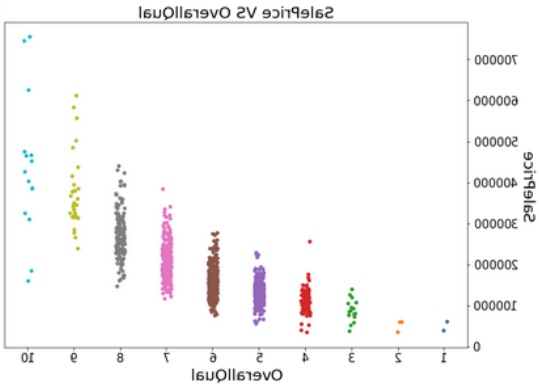

SalePrice VS BedroomAbvGr
SalePrice
BedroomAbvGr

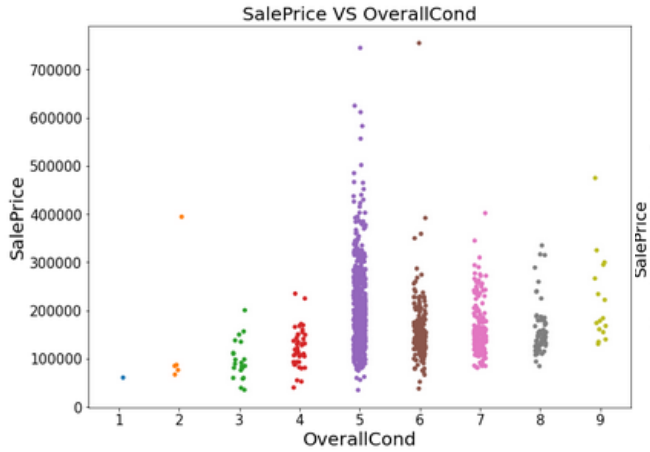

SalePrice VS HalfBath
SalePrice
HalfBath

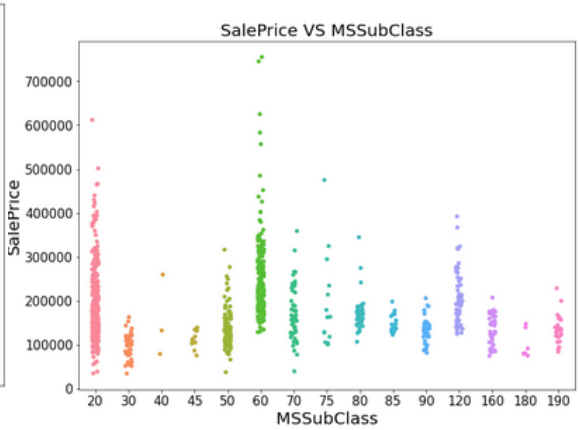

SalePrice VS FullBath
SalePrice
FullBath


SalePrice VS BsmtFullBath
SalePrice
BsmtFullBath


SalePrice VS BsmtHalfBath
SalePrice
BsmtHalfBath


SalePrice VS OverallQual
SalePrice
OverallQual


SalePrice VS OverallCond
SalePrice
OverallCond


SalePrice VS MSSubClass
SalePrice
MSSubClass

Observations:-

1.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.

3.5 Run and Evaluate selected models:-

Finding Best Random state and accuracy:-

```
[181]: from sklearn.metrics import accuracy_score
       from sklearn.metrics import r2_score
       from sklearn.model_selection import train_test_split
       from sklearn.ensemble import RandomForestRegressor
```

```
[182]: maxAccu=0
       maxRS=0
       for i in range(1,100):
           X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.30, random_state =i)
           mod=RandomForestRegressor()
           mod.fit(X_train, y_train)
           pred = mod.predict(X_test)
           acc=r2_score(y_test, pred)
           print('accuracy : ',acc,'random state :',i)
           if acc>maxAccu:
               maxAccu=acc
               maxRS=i
       print("Best accuracy is ",maxAccu," on Random_state ",maxRS)
```

```
[183]: print("Best accuracy is ",maxAccu," on Random_state ",maxRS)
```

```
Best accuracy is  0.9009191560715581  on Random_state  50
```

Here i got the best accuracy and random state.

Here I got best accuracy score by using RandomForestRegressor.

Creating train test split:-

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=.30,random_state=maxRS)
```

Model Building:-

i) RandomForestRegressor:-

In [186]:
```
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: 89.7667121853309
mean_squared_error: 615946285.8540052
mean_absolute_error: 16693.20854700855
root_mean_squared_error: 24818.26516608293

Cross validation score : 83.29921379463056

R2_Score - Cross Validation Score : 6.467498390700342
```

RandomForestRegressor is giving me 89.76% r2_score.

- **By using RandomForestRegressor It has give me 89.72% accuracy.**
- **It's a pretty good score but still I have to look into different models.**

ii)ExtraTreesRegressor:-

[187]:
```
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: 88.64651494577248
mean_squared_error: 683371471.3492131
mean_absolute_error: 16877.51031339031
root_mean_squared_error: 26141.37470274303

Cross validation score : 84.03627907708042

R2_Score - Cross Validation Score : 4.610235868692058
```

ExtraTreesRegressor is giving me 88.64% r2_score.

---

iii)GradientBoostingRegressor:-

[188]:
```
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: 90.66728635306325
mean_squared_error: 561740313.7561958
mean_absolute_error: 15806.488414792702
root_mean_squared_error: 23701.061447880256

Cross validation score : 82.17345062643153

R2_Score - Cross Validation Score : 8.493835726631715
```

---

iv)DecisionTreeRegressor:-

```
[190]: DTR=DecisionTreeRegressor()
DTR.fit(X_train,y_train)
pred=DTR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

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

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

```
R2_score: 69.43798402861839
mean_squared_error: 1839541754.977208
mean_absolute_error: 28754.749287749288
root_mean_squared_error: 42889.879400357466

Cross validation score : 64.99093226173737

R2_Score - Cross Validation Score : 4.447051766881017
```

DecisionTreeRegressor is giving me 69.43% r2_score.

- **Here after seeing the difference of model accuracy and cross validation score i found ExtraTreesClassifier as the best model.**

2. Hyper Parameter Tunning:-

```
]: 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)

]: GCV.best_params_

]: #Its take too much tim to load..already it takes 3-4hrs for run...so i take best parameters from myself
```

Here it takes too much time for run..almost it takes 4-5 hours soo I take best_params by it myself.

```
97]: 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: 89.00803141703516
     mean_squared_error: 661611629.177072
     mean_absolute_error: 16663.759629629625
     RMSE value: 25721.812322950187
```

- Here I have choosed all parameters of ExtraTreesRegressor, after tunning the model with best parameters I have incresed my model accuracy from 88.64% to 89%. Also mse and rmse values has reduced whichmeans error has reduced**.**

3. Saving the model:-

```
198]: import joblib
      joblib.dump(Best_mod,"House_Price_prediction.pkl")

198]: ['House_Price_prediction.pkl']
```

Here I am saving the model by using .pkl

Predicting House Price for test dataset using Saved model:-

```
In [199]: model=joblib.load("House_Price_prediction.pkl")

          #Prediction
          prediction = model.predict(X_test)
          prediction

Out[199]: array([138920.91, 179446.32, 120134.68, 238441.11, 134828.94,  93772.93,
                  97373.91, 358474.29, 264034.71, 210915.92, 261215.49, 139135.85,
                 196309.31, 216870.06, 169953.  , 203982.07, 157727.06, 214463.06,
                 161068.5 , 159243.5 , 170749.86, 344818.14, 199540.07, 227460.81,
                 118692.14, 138138.33, 161987.5 , 225852.37, 123540.5 , 133085.3 ,
                 314200.76, 178923.03, 133625.57, 204110.  ,  94270.12, 198613.05,
                 156845.23,  90998.3 , 163093.74, 204807.72, 218212.01, 217040.9 ,
```

```
n [200]: pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"]).T
```

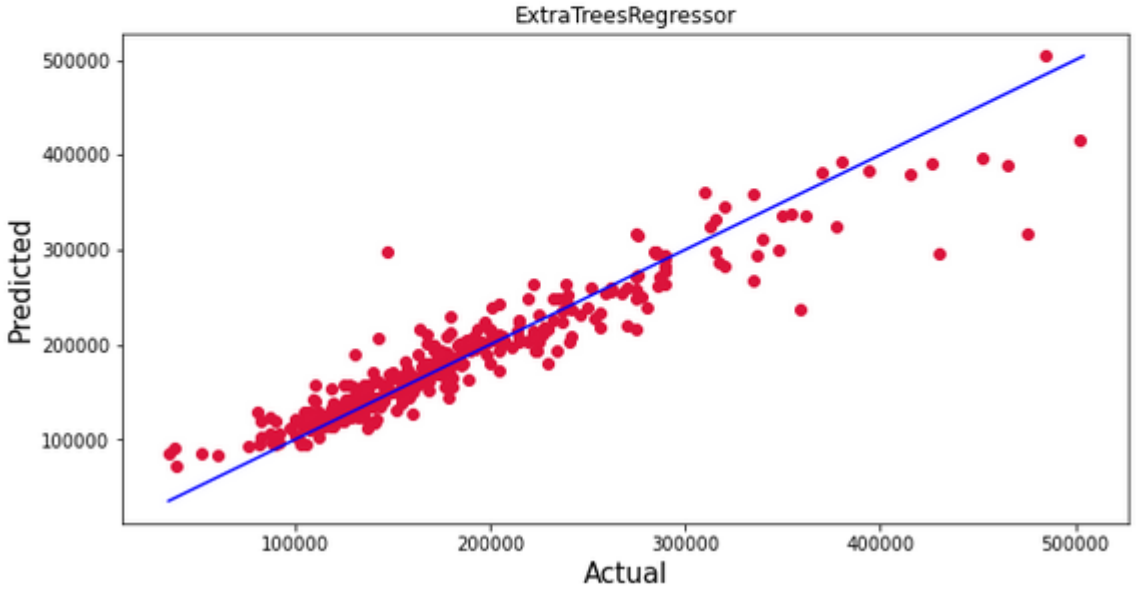
ut[200]:

	Predicted	Actual
0	138920.91	137000.0
1	179446.32	168500.0
2	120134.68	115000.0
3	238441.11	280000.0
4	134828.94	140000.0
...	...	...
346	122330.23	110000.0
347	157611.63	137500.0
348	174925.36	170000.0
349	118592.39	139000.0
350	336725.06	361919.0

351 rows × 2 columns

We can see above are the predicted and actual values.

```
]: plt.figure(figsize=(10,5))
plt.scatter(y_test, prediction, c='crimson')
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()
```



- Here I am ploting actual and predicted values.

```
#for test dataset
```

```
Predicted_Sale_Price=model.predict(X_1)
Predicted_Sale_Price
```

```
array([338161.82, 223501.51, 247919.46, 169062.  , 245190.21,  84030.26,
       147207.61, 332175.23, 239310.53, 170052.03,  90861.76, 140647.5 ,
       123488.09, 205090.4 , 287281.29, 135495.01, 121003.75, 133548.  ,
       177645.13, 200908.21, 149387.  , 158520.37, 158870.  ,  98818.08,
       117435.95, 134901.  , 180037.79, 152512.  , 186934.  , 103960.93,
       148363.62, 199067.32, 224376.46, 166809.5 , 123503.09, 181078.47,
       203776.57, 122181.03, 170374.  , 151761.5 , 115998.4 , 296150.74,
       203564.18, 192709.99, 144303.4 , 126942.  , 130122.  , 104786.37,
       213140.15, 348045.44, 142211.08, 224261.6 , 111422.76, 102121.5 ,
       254744.36, 136280.5 , 141196.56, 189328.7 , 124400.43, 258248.89,
        98692.16, 209423.57, 133796.36, 151185.8 , 202768.52,  95001.  ,
       156794.  , 210997.37, 147362.84, 163266.  , 278385.94, 172865.48,
       166682.21, 149417.68, 146519.74, 230061.93, 319403.3 , 190205.75,
       300934.63, 145948.62, 221100.9 , 135118.75, 152203.25, 160964.7 ,
       199585.83, 230230.81, 117807.52, 357516.41, 155943.08, 180065.09,
       240627.55, 140426.93, 135471.42, 132210.84, 205539.9 , 161363.  ,
```

```
House_Price_Predictions=pd.DataFrame()
House_Price_Predictions["SalePrice"]=Predicted_Sale_Price
House_Price_Predictions.head(10)
```

	SalePrice
0	338161.82
1	223501.51
2	247919.46
3	169062.00
4	245190.21
5	84030.26
6	147207.61
7	332175.23
8	239310.53
9	170052.03

- **Here I have predicted the SalePrice for test dataset using saved model of train dataset, and the predictions look good.**

```
In [207]: House_Price_Predictions.to_csv("House_Price_Predictions.csv",index=False)

In [ ]:
```

Here I have sved the model in CSV format.

3.6 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 89.66% 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 value

4.CONCLUSION:-

4.1 Key Findings and Conclusions of the Study:-

In this project report, we have used machine learning algorithms to predict thehouse 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.

4.2 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.

4.3 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 89% even after dealing all these drawbacks.
- Also, this study will not cover all regression algorithms instead, it is focusedon the chosen algorithm, starting from the basic regression techniques tothe 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.