FLIP ROBO

PROJECT REPORT ON:

"HOUSING: PRICE PREDICTION"

SUBMITTED BY

VISHAL LAKHERA

ACKNOWLEDGMENT

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analytical and statistical skills. And I want to express my huge gratitude to Mrs. Sapna Verma (SME Flip Robo), She is the person who has helped me to get out of all the difficulties I faced during the project and also inspired me in so many aspects . I have ended up with a project worth your while. A huge thanks to my academic team “Data trained” who helped me learn and nurtured me through these months.

Contents

1.INTRODUCTION

1.1 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 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. 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 and same tends to vary significantly based on numerous factors; forecasting property price is an important module in decision making for both the buyers and 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:

1. Which variables are important to predict the price of variable?
2. 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 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.

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 collected for my internship company – Flip Robo technologies in csv (comma separated values) format. Also, I was having two datasets one is train and other is test. I have bui lt model using train dataset and predicted SalePrice for test dataset. M y train dataset was having 1168 rows and 81 columns including target, and my test dataset was having 292 rows and 80 columns excluding ta rget. In this perticular 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 datas ets looks for me when I import those datasets to my python.

2.3 Data Preprocessing Done

I. As a first step I have imported required libraries and I have imported both the datasets which were in csv format

II. Then I did all the statistical analysis like checking shape, nunique, value counts, info etc…..

III. 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.

IV. Then while looking into the value counts I found some columns with more than 85% zero values this also creates skewness in the model and there are chances of getting model bias so I have dropped those columns with more than 85% zero values.

V. While checking for null values I found null values in most of the columns and I have used imputation method to replace those null values (mode for categorical column and mean for numerical columns).

VI. 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.

VII. 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.

VIII. 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 — 8 GB or above

3. SSD — 250GB or above

Software/s required: -

1.Anaconda

Libraries required :-

- To run the program and to build the model we need some basic libraries as follows:
- 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 of functions 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.
- from sklearn.preprocessing import OrdinalEncoder
- 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)

I have used imputation method to replace null values. To remove outliers I have used percentile method. And to remove skewness I have used yeo-johnson method. To encode the categorical columns I have use Ordinal

Encoding. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Also I have used standardization. Then followed by model building with all regression algorithms.

3.2 Testing of Identified Approaches (Algorithms)

Since Saleprice was my target and it was a continuous column so this perticular 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
- DecisionTreeRegressor

3.3 Key Metrics for success in solving problem under consideration

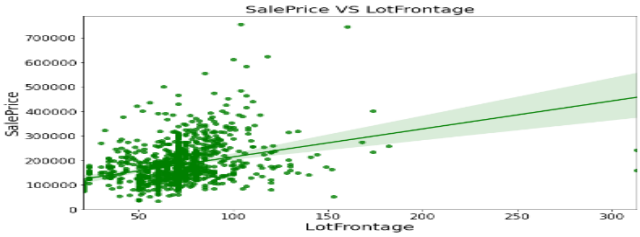
I have used the following metrics for evaluation:

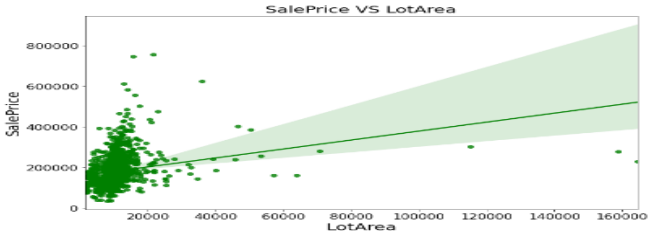
- I have used mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.
- I have used root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.
- I have used r2 score which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

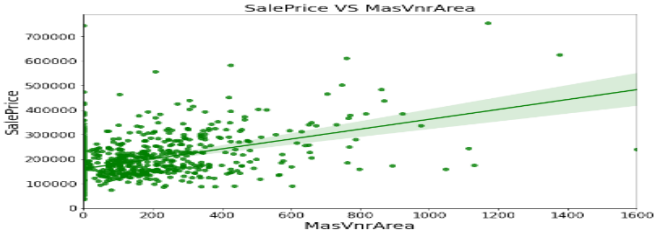
3.4 Visualizations

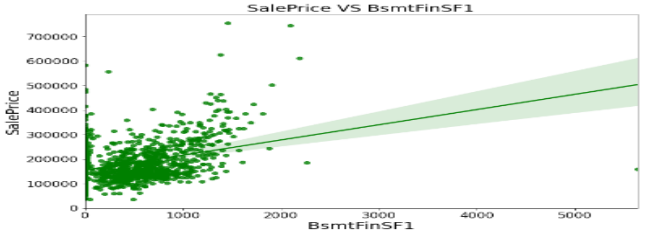
I have used bar plots to see the relation of categorical feature and I have used 2 types of plots for numerical columns one is strip plot for ordinal features and other is reg plot for continuous features.

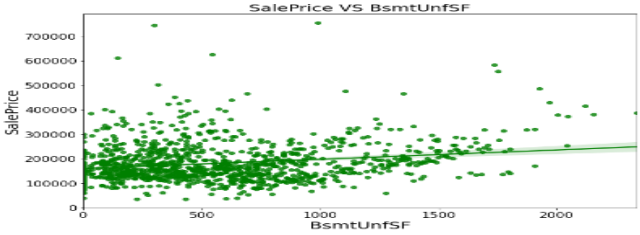
1. Vizualization of numerical features with target:

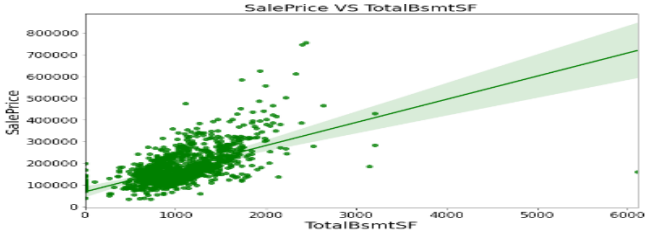

SalePrice VS LotFrontage
SalePrice
LotFrontage

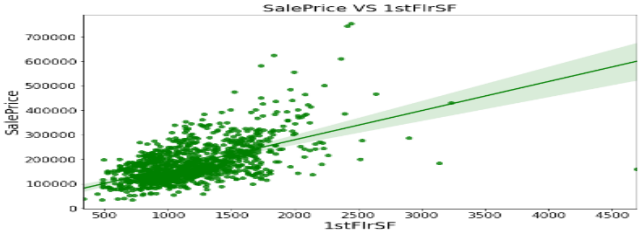

SalePrice VS LotArea
SalePrice
LotArea


SalePrice VS MasVnrArea
SalePrice
MasVnrArea

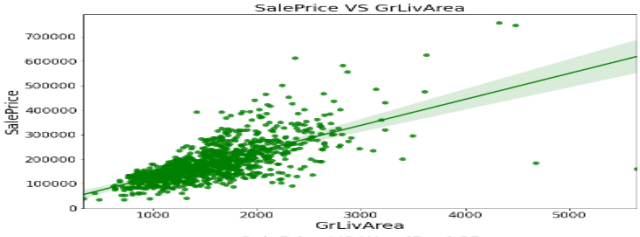

SalePrice VS BsmtFinSF1
SalePrice
BsmtFinSF1

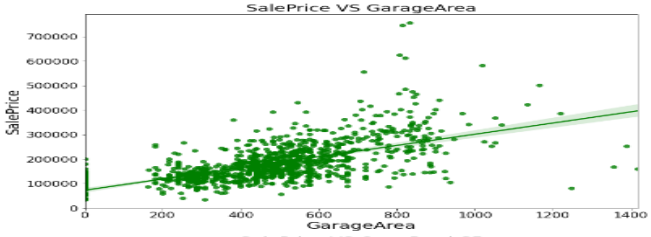

SalePrice VS BsmtUnfSF
SalePrice
BsmtUnfSF

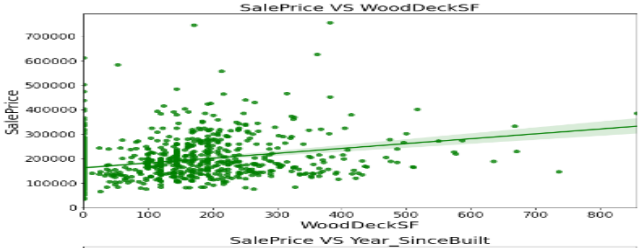

SalePrice VS TotalBsmtSF
SalePrice
TotalBsmtSF

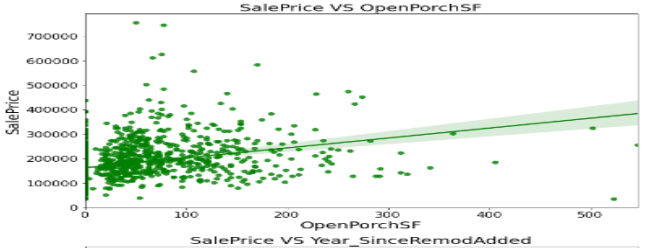

SalePrice VS 1stFlrSF
SalePrice
1stFlrSF

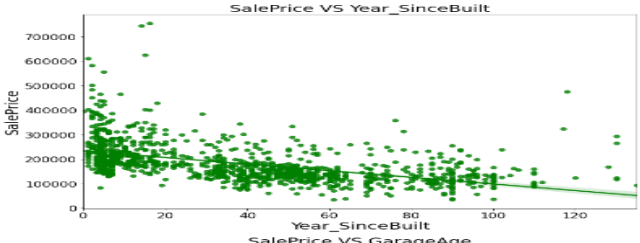

SalePrice VS 2ndFlrSF
SalePrice
2ndFlrSF

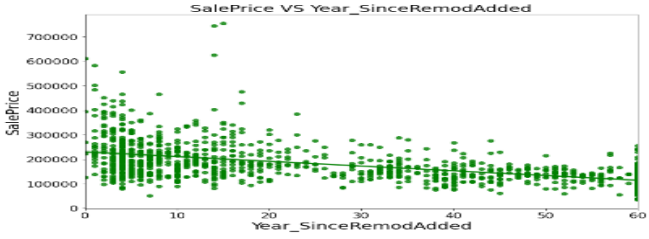

SalePrice VS GrLivArea
SalePrice
GrLivArea

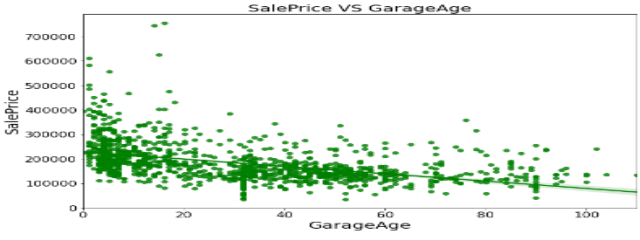

SalePrice VS GarageArea
SalePrice
GarageArea


SalePrice VS WoodDeckSF
SalePrice
WoodDeckSF


SalePrice VS OpenPorchSF
SalePrice
OpenPorchSF


SalePrice VS Year_SinceBuilt
SalePrice
Year_SinceBuilt


SalePrice VS Year_SinceRemodAdded
SalePrice
Year_SinceRemodAdded


SalePrice VS GarageAge
SalePrice
GarageAge

Observation

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.

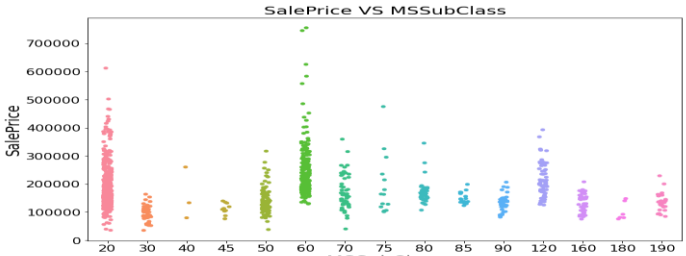
11.As Wood deck area in square feet(WoodDeckSF) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

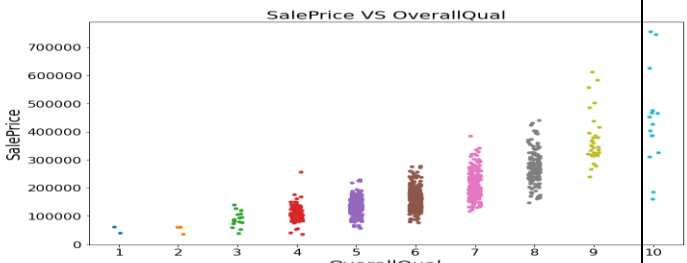
12.As Open porch area in square feet(OpenPorchSF) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

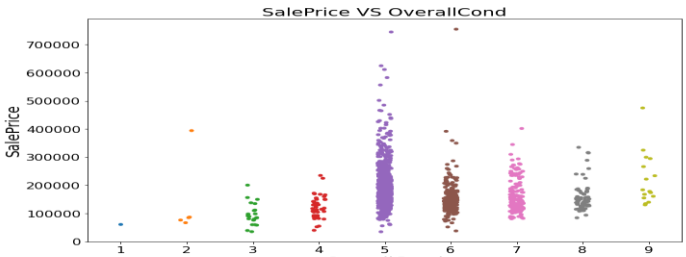
13.As Year_SinceBuilt is increseing sales is decreasing and the saleprice is high for newly built building and the sales price is in between 0-4 lakhs.

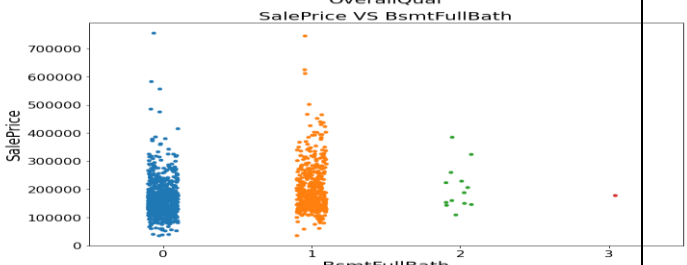
14.As Since Remodel date (same as construction date if no remodeling or additions)(Year_SinceRemodAdded) is increseing sales is decreasing and the saleprice is in between 1-4 lakhs.

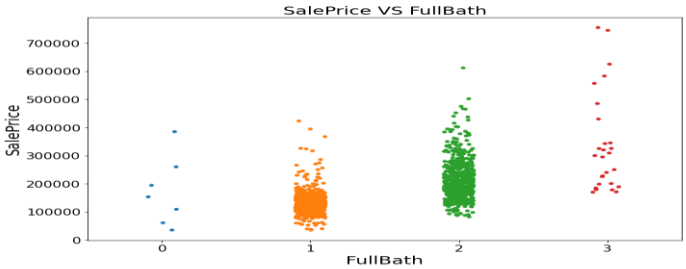
15.As Since Year garage was built(GarageAge) is increseing sales is decreasing and the saleprice is in between 0-4 lakhs.

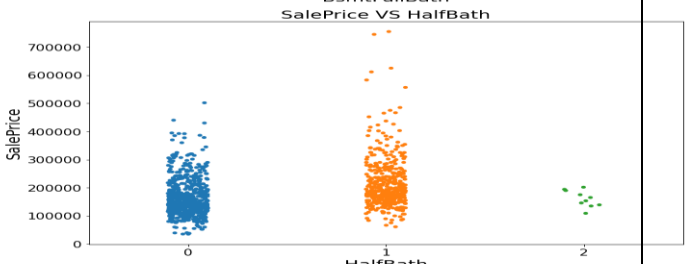

SalePrice VS MSSubClass
SalePrice
MSSubClass

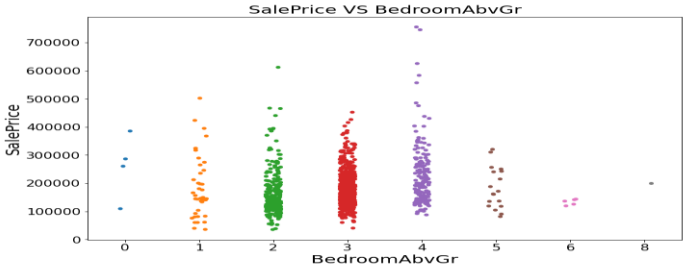

SalePrice VS OverallQual
SalePrice
OverallQual

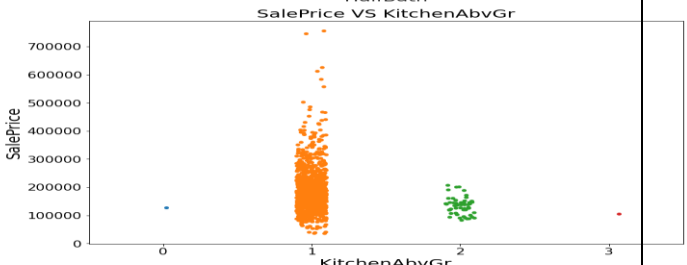

SalePrice VS OverallCond
SalePrice
OverallCond

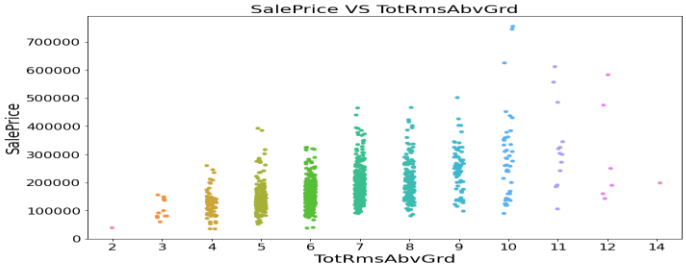

SalePrice VS BsmtFullBath
SalePrice
BsmtFullBath


SalePrice VS FullBath
SalePrice
FullBath

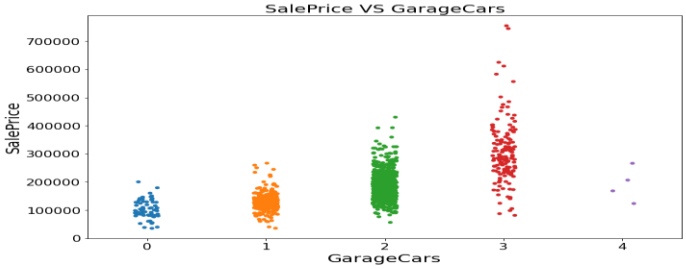

SalePrice VS HalfBath
SalePrice
HalfBath

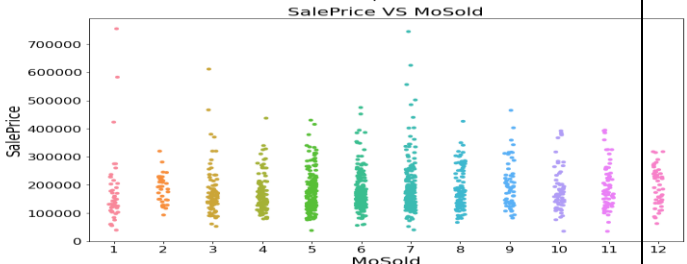

SalePrice VS BedroomAbvGr
SalePrice
BedroomAbvGr

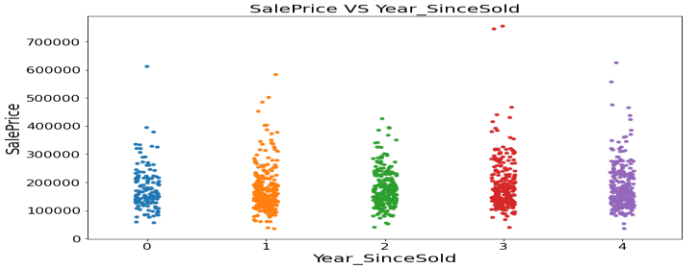

SalePrice VS KitchenAbvGr
SalePrice
KitchenAbvGr


SalePrice VS TotRmsAbvGrd
SalePrice
TotRmsAbvGrd


SalePrice VS Fireplaces
SalePrice
Fireplaces


SalePrice VS GarageCars
SalePrice
GarageCars


SalePrice VS MoSold
SalePrice
MoSold


SalePrice VS Year_SinceSold
SalePrice
Year_SinceSold

Observation

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 the overall material quality 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 1 and 2 Full bathrooms above grade(FullBath) the sales as well as SalePrice is high.

6.For 0 and 1 Half baths above grade(HalfBath) the sales as well as SalePrice is high.

7.For 2, 3 and 4 Bedrooms above grade (does NOT include basement bedrooms)(BedroomAbvGr) the sales as well as SalePrice is high.

8.For 1 Kitchens above grade(KitchenAbvGr) the sales as well as SalePrice is high.

9.For 4-9 Total rooms above grade (does not include bathrooms)(TotRmsAbvGrd) the sales as well as SalePrice is high.

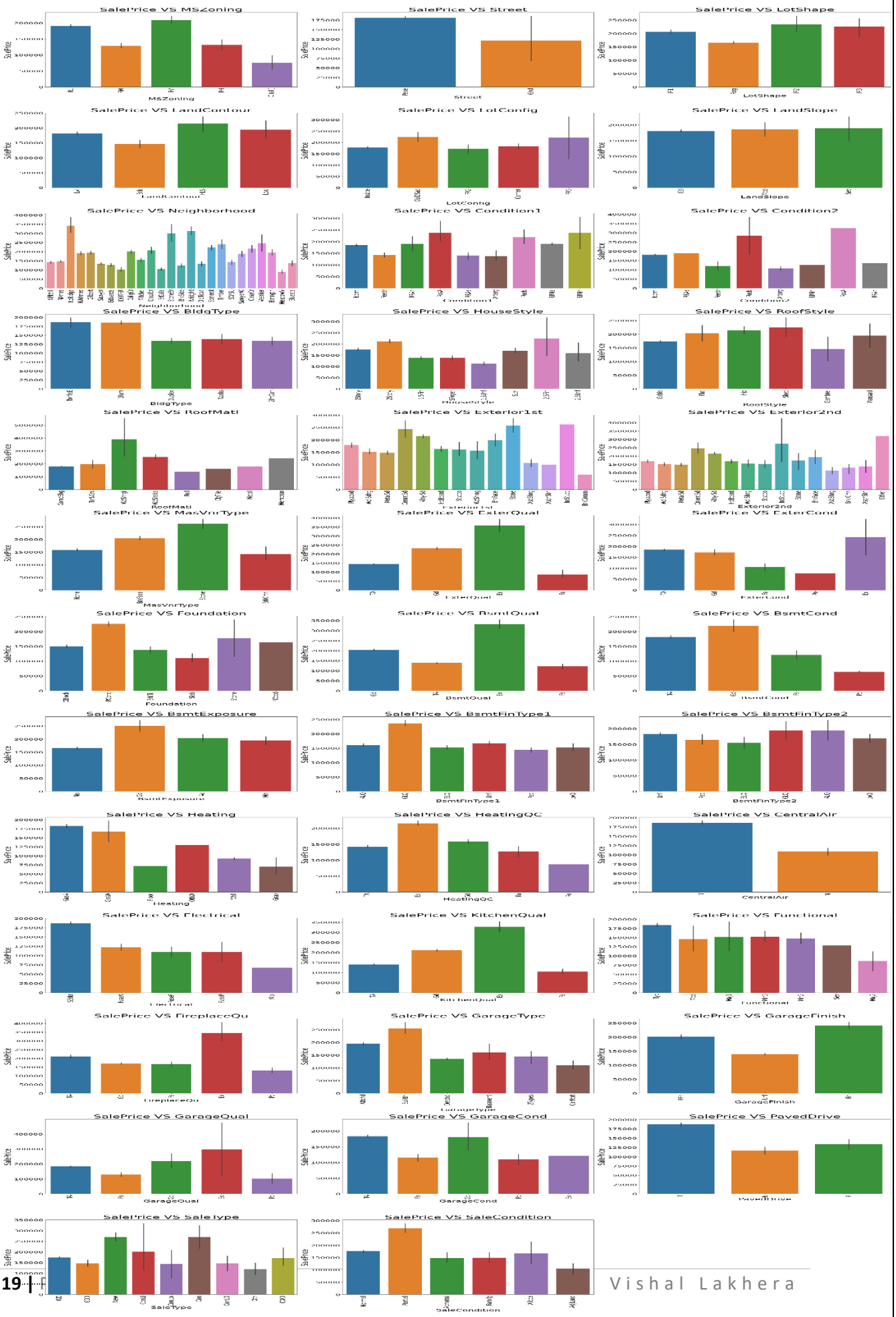
10.For 0 and 1 Number of fireplaces(Fireplaces) the sales as well as SalePrice is high.

11.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.

12.In between april to august for Month Sold(MoSold) the sales is good with SalePrice.

13.For all the Year_SinceSold the salePrice and sales both are same.

2. Visualization of categorical features with target:


SalePrice VS MSZoning
SalePrice VS Street
SalePrice VS LotShape
SalePrice VS LandContour
SalePrice VS LotConfig
SalePrice VS LandSlope
SalePrice VS Neighborhood
SalePrice VS Condition1
SalePrice VS Condition2
SalePrice VS BldgType
SalePrice VS HouseStyle
SalePrice VS RoofStyle
SalePrice VS RoofMatl
SalePrice VS Exterior1st
SalePrice VS Exterior2nd
SalePrice VS MasVnrType
SalePrice VS ExterQual
SalePrice VS ExterCond
SalePrice VS Foundation
SalePrice VS BsmtQual
SalePrice VS BsmtCond
SalePrice VS BsmtExposure
SalePrice VS BsmtFinType1
SalePrice VS BsmtFinType2
SalePrice VS Heating
SalePrice VS HeatingQC
SalePrice VS CentralAir
SalePrice VS Electrical
SalePrice VS KitchenQual
SalePrice VS Functional
SalePrice VS FireplaceQu
SalePrice VS GarageType
SalePrice VS GarageFinish
SalePrice VS GarageQual
SalePrice VS GarageCond
SalePrice VS PavedDrive
SalePrice VS SaleType
SalePrice VS SaleCondition

Observation

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) in 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 Equally 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 postive 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) 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.

3.5 Run and Evaluate selected models

1. Model Building:

1) RandomForestRegressor

In [156]:
```
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 = 5).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: 89.88458803854769
mean_squared_error: 608851284.1990457
mean_absolute_error: 16432.058404558404
root_mean_squared_error: 24674.912040350733

Cross validation score : 82.9677345317904

R2_Score - Cross Validation Score : 6.916853506757292
```

2)XGBRegressor

In [157]:
```
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 = 5).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: 88.3761155182305
mean_squared_error: 699646936.8797324
mean_absolute_error: 17771.03940749644
root_mean_squared_error: 26450.84000329162

Cross validation score : 83.32256763716677

R2_Score - Cross Validation Score : 5.053547881063736
```

3) ExtraTreesRegressor

In [158]:
```
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 = 5).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: 88.69650321706224
mean_squared_error: 680362654.3790638
mean_absolute_error: 16865.081766381765
root_mean_squared_error: 26083.76227423996

Cross validation score : 83.2414357685765

R2_Score - Cross Validation Score : 5.4550674484857495
```

4) GradientBoostingRegressor

In [159]:
```
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 = 5).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: 91.49055228433869
mean_squared_error: 512187559.85902166
mean_absolute_error: 15412.597795469865
root_mean_squared_error: 22631.561144981177

Cross validation score : 82.97835880252798

R2_Score - Cross Validation Score : 8.512193481810712
```

5) DecisionTreeRegressor

In [160]:
```
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 = 5).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: 72.78311645317979
mean_squared_error: 1638196700.4928775
mean_absolute_error: 27547.02279202279
root_mean_squared_error: 40474.64268517855

Cross validation score : 67.50756780181942

R2_Score - Cross Validation Score : 5.275548651360367
```

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 Tuning For best Model

In [178]:
```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
parameter = {'n_estimators':[10,100],
             'criterion':['squared_error','mae'],
             'min_samples_split': [2,4],
             'max_features':['auto','sqrt'],
             'n_jobs':[-2,2]}
```

In [179]:
```
# Giving estimator as ExtraTreesRegressor
GCV=GridSearchCV(ExtraTreesRegressor(),parameter,cv=5)
```

In [180]:
```
GCV.fit(X_train,y_train)
```

Out[180]:
```
GridSearchCV(cv=5, estimator=ExtraTreesRegressor(),
             param_grid={'criterion': ['squared_error', 'mae'],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_split': [2, 4], 'n_estimators': [10, 100],
                         'n_jobs': [-2, 2]})
```

In [181]:
```
GCV.best_params_
```

Out[181]:
```
{'criterion': 'mae',
 'max_features': 'sqrt',
 'min_samples_split': 2,
 'n_estimators': 100,
 'n_jobs': 2}
```

In [174]:
```
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: 89.21337696585468
mean_squared_error: 649251786.4361374
mean_absolute_error: 16416.3452991453
RMSE value: 25480.419667582744
```

With the help of hyper parameter tuning i've increased the accuracy(r2_score) from 88.69 to 89.21

I have choosed all parameters of ExtraTreesRegressor, after tunning the model with best parameters I have incresed my model accuracy from 88.69% to 89.21%. 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.
- Now loading my saved model and predicting the test values.

Saving the model:

In [182]:
```
# Saving the model using .pkl
import joblib
joblib.dump(Best_mod,"House_Price.pkl")
```

Out[182]:
```
['House_Price.pkl']
```

```
I have saved my model as House_Price.Using .pkl
```

Predicting House Price for test dataset using Saved model of train dataset:

In [183]:
```
# Loading the saved model
model=joblib.load("House_Price.pkl")

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

Out[183]:
```
array([135874.66, 179391.32, 120642.05, 231917.47, 137072.  ,  93112.32,
        94436.34, 372105.83, 270959.05, 215136.39, 272745.05, 141238.51,
       197180.98, 213542.17, 165162.  , 206273.65, 162475.62, 214851.21,
       165493.87, 158879.87, 173850.18, 351754.62, 195789.28, 225365.41,
       120430.92, 135463.5 , 160288.61, 232177.71, 128547.  , 140698.6 ,
       342190.61, 183542.46, 130538.3 , 203460.8 ,  90390.02, 189933.34,
       154734.23,  90989.16, 162328.5 , 205528.86, 223690.49, 221453.02,
       148527.33, 177120.44, 192129.01, 198179.52, 265587.5 , 200072.04,
       172512.84, 172438.62, 174277.  ,  72658.22, 178944.68, 120488.3 ,
       118751.05, 257844.77, 301210.22, 149792.87, 102159.4 , 224794.73,
       102756.58,  95603.49, 175983.05, 384196.15, 153019.95, 206269.42,
       307778.06,  87515.08, 153646.37, 162106.96, 197920.97, 203466.27,
        95299.18, 216682.62, 171956.87, 244716.62, 135546.75, 195599.01,
       261246.88, 143812.63, 477429.34, 124554.55, 207777.81, 197069.49,
       189380.6 , 224565.9 , 195554.94, 293801.81, 165928.54, 102123.5 ,
       120546.76, 186065.41, 173683.85, 133412.9 , 142873.  , 112684.53,
       125665.89, 136238.23, 240747.56, 126213.13, 146904.39, 283985.68,
       234177.51, 191874.05, 300679.33, 259372.34, 198264.8 , 213545.45,
       120578.45, 116681.86, 143550.5 , 207234.39, 195586.86, 202777.  ,
       192055.22, 147119.06, 132799.15, 278966.69, 215383.94, 217981.28,
       146160.  , 140742.71, 216423.51, 141268.  , 194199.02, 239843.89,
       212363.73, 178175.14, 146704.52, 329352.93, 186888.2 , 386857.  ,
       242390.7 , 320026.72,  86904.29, 203943.5 , 203781.42, 192677.18,
       184090.5 , 134717.07, 178479.64, 365281.99, 222644.21, 158180.82,
       226806.46, 101532.63, 117745.92, 245657.13, 103143.82, 185988.5 ,
```

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

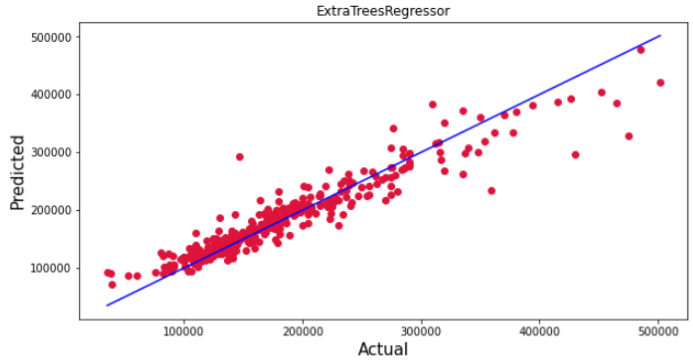
In [184]:
```
pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[184]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	
Predicted	135874.66	179391.32	120642.05	231917.47	137072.0	93112.32	94436.34	372105.83	270959.05	215136.39	272745.05	141238.51	197180.98	213542.
Actual	137000.00	168500.00	115000.00	280000.00	140000.0	76000.00	88000.00	335000.00	222000.00	227000.00	286000.00	132250.00	189000.00	193500.

Above are the predicted values and the actual values.They are almost similar.

In [185]:
```
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()
```



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

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 88.69% 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

4.1 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 stepprocedure 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. 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.

4.3 Limitations of this work and Scope for Future Work

The real estate industry is likely just at the beginning of a significant shift towards greater use of data and data-driven decision making. There are huge opportunities that are now starting to be unlocked by various start-ups and forward-thinking institutions. There is a range of concrete methods — as outlined above — to apply data science to real estate, to help move from millions of rows of data to granular understandings of past, present, and future real estate submarket performance, and make superior investment and business decisions. However, the required skills may often be absent across a good percentage of the industry. There is now the opportunity to learn these techniques and methods — specifically for real estate — and investing the time to upgrade could benefit a range of participants. Real estate researchers could begin to use data and machine learning to produce game-changing insights and unlock the value of large datasets. Those in the Protect industry (or even investing in Protect) could do well to understand these methods better and build (or invest in) disruptive activities. Finally, real estate investors who learn these methods could use data-driven approaches to find exceptional opportunities and beat the market.

Some drawback in Model Building are:

- 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 89.21% 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

- Google
- Kaggles
- Github
- https://scikit-learn.org/stable/

Thank You

Vishal lakhera