HOUSING: PRICE PREDICTION

Submitted by:
SHUBHAM J. CHOUGUELE

Machine Learning

HOUSING: PRICE PREDICTION

ACKNOWLEDGMENT

I would like to thank FlipRobo Technologies for giving me the opportunity to work on this project. I am very grateful to the DataTrained team for providing me with the knowledge which helped me a lot to work on this project. And I want to express my huge gratitude to Ms. Khushboo Garg (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.

CONTENT

Introduction

1.1 Business Problem Framing:

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors to 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 policie. 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 particular problem has two datasets one is train dataset and the other is test dataset. I have built model using train dataset and predicted SalePrice for test dataset. By looking into the target column, I came to know that the entries of SalePrice column were continuous and this was a Regression problem so I have to use all regression algorithms while building the model. Also, I observed some unnecessary entries in some of the columns like in some columns I found more than 80% null values and more than 85% zero values so I decided to drop those columns. If I keep those columns as it is, it will create high skewness in the model. While checking the null values in the datasets I found many columns with nan values and I replaced those nan values with suitable entries like mean for numerical columns and mode for categorical columns. To get better insight on the features I have used plotting like distribution plot, bar plot, reg plot and strip plot. With these plotting 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 turned the best model and saved the best model. At last I have predicted the sale price for 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 target. In this particular datasets, I have object, float, and integer types of data.

2.3 Data Preprocessing Done:

1) As a first step I have imported required libraries and I have imported both the datasets which were in csv format.
2) Then I did all the statistical analysis like checking shape, nunique, value counts, info etc....
3) 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.
4) 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.
5) 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).
6) 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.
7) 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.
8) And all these steps were performed to both train and test datasets separately and simultaneously.

2.4 Data Inputs- Logic- Output Relationships:

1) 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.
2) And also, for continuous numerical variables I have used reg plot to show the relationship between a continuous numerical variable and target variable.
3) 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:

1) import pandas as pd: pandas are 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.
2) 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.
3) 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.
4) 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 OrdinalEncode
    - from sklearn.preprocessing import StandardScaler from
    - statsmodels. stats.outliers_influence import variance_inflation_factor
    - from sklearn.ensemble import RandomForestRegressor from sklearn.tree import DecisionTreeRegresson
    - from xgboost import XGBRegressor
    - from sklearn.ensemble import GradientBoostingRegressor from sklearn.ensemble import ExtraTreesRegressor

3). Data Analysis and Visualization:

3.1 Identification of possible problem-solving approaches:

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

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

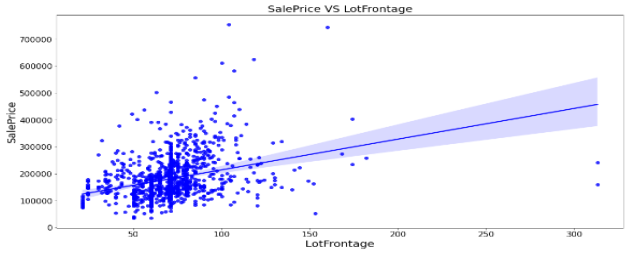
I have used the following metrics for evaluation:

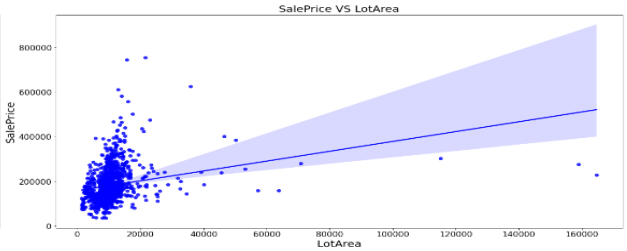
1. I have used mean absolute error which gives magnitude of difference between the prediction of observation and the true value of that observation.
2. I have used root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.
3. I have used r2 score which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

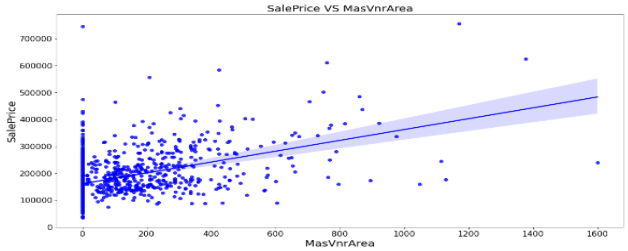
3.4 Visualizations:

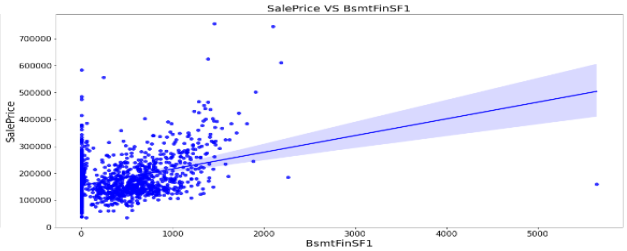
I have used bar plots to see the relation of categorical feature and I have used 2 types of plots for numerical columns one is strip plot for ordinal features and other is reg plot for continuous features.

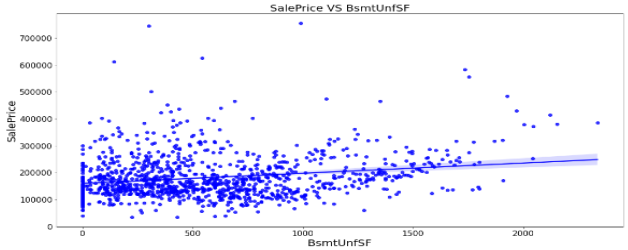
1. **Vizualization of numerical features with target :**

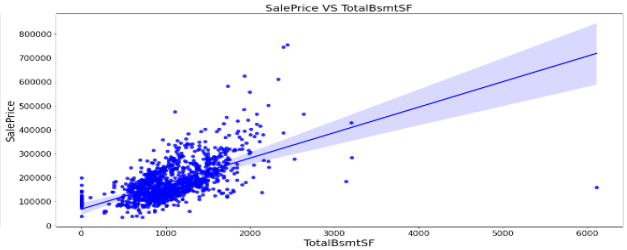

SalePrice VS LotFrontage
SalePrice
LotFrontage

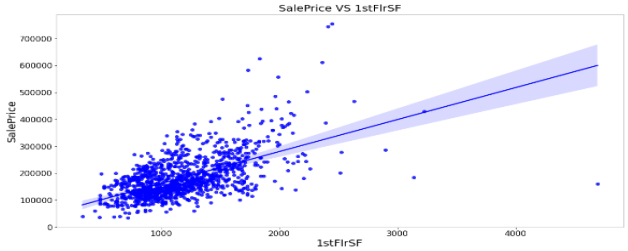

SalePrice VS LotArea
SalePrice
LotArea

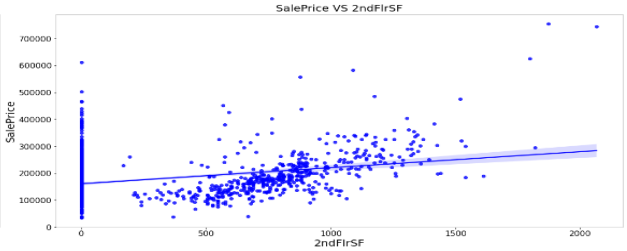

SalePrice VS MasVnrArea
SalePrice
MasVnrArea

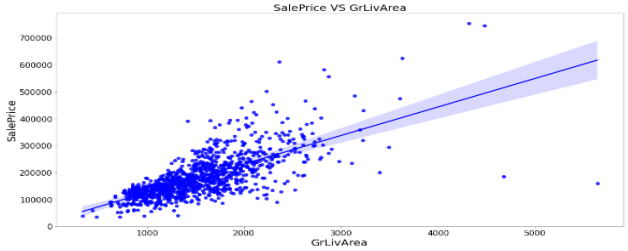

SalePrice VS BsmtFinSF1
SalePrice
BsmtFinSF1

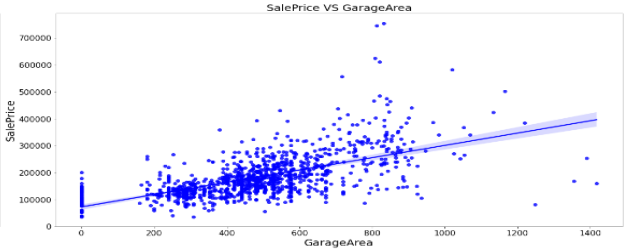

SalePrice VS BsmtUnfSF
SalePrice
BsmtUnfSF

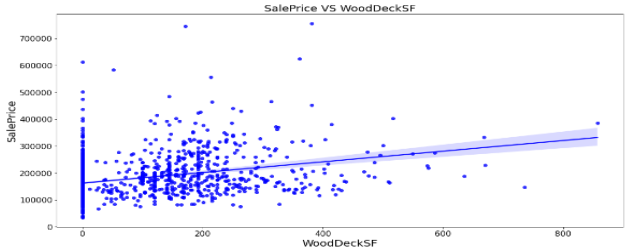

SalePrice VS TotalBsmtSF
SalePrice
TotalBsmtSF

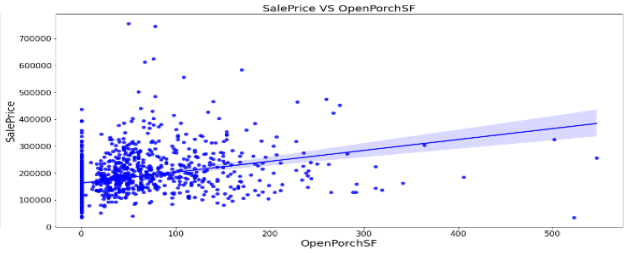

SalePrice VS 1stFlrSF
SalePrice
1stFlrSF

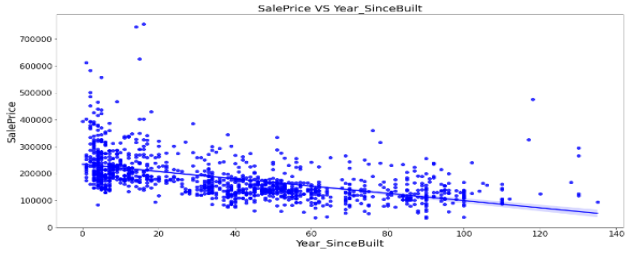

SalePrice VS 2ndFlrSF
SalePrice
2ndFlrSF

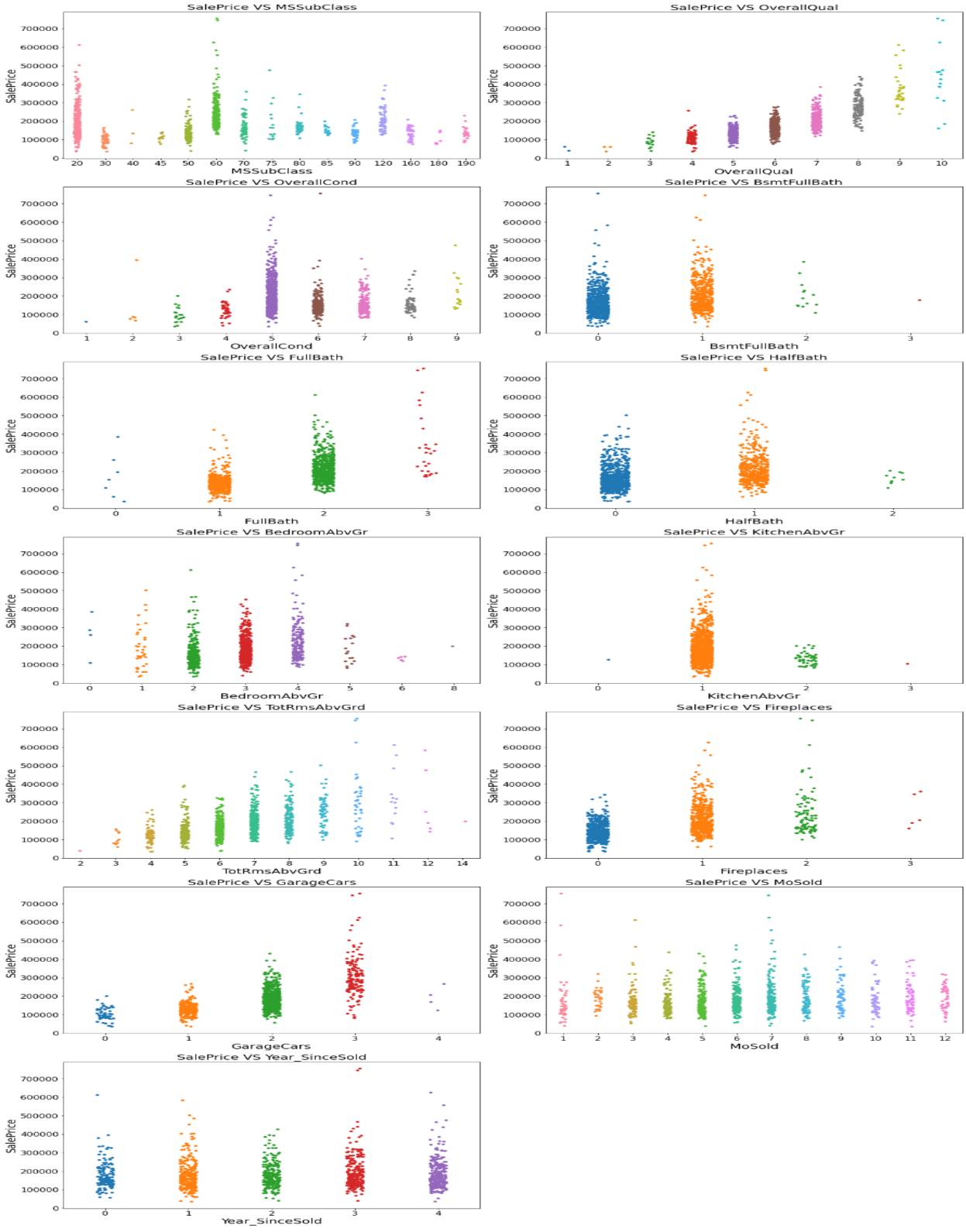

SalePrice VS GrLivArea
SalePrice
GrLivArea


SalePrice VS GarageArea
SalePrice
GarageArea


SalePrice VS WoodDeckSF
SalePrice
WoodDeckSF


SalePrice VS OpenPorchSF
SalePrice
OpenPorchSF


SalePrice VS Year_SinceBuilt
SalePrice
Year_SinceBuilt


SalePrice VS MSSubClass
SalePrice
MSSubClass
SalePrice VS OverallQual
OverallQual
SalePrice VS OverallCond
OverallCond
SalePrice VS BsmtFullBath
BsmtFullBath
SalePrice VS FullBath
FullBath
SalePrice VS HalfBath
HalfBath
SalePrice VS BedroomAbvGr
BedroomAbvGr
SalePrice VS KitchenAbvGr
KitchenAbvGr
SalePrice VS TotRmsAbvGrd
TotRmsAbvGrd
SalePrice VS Fireplaces
Fireplaces
SalePrice VS GarageCars
GarageCars
SalePrice VS MoSold
MoSold
SalePrice VS Year_SinceSold
Year_SinceSold

2. Visualization of categorical features with target:

3.5 Run and Evaluate selected models:

1) Model Building:

1) RandomForestRegressor :

In [171]:
```
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: 90.32583221726799
mean_squared_error: 582292594.7573317
mean_absolute_error: 16531.84735042735
root_mean_squared_error: 24130.739623089295

Cross validation score : 83.41653825107949

R2_Score - Cross Validation Score : 6.9092939661885
```

2) XGBRegressor :

In [172]:
```
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: 87.45624031388263
mean_squared_error: 755014647.221571
mean_absolute_error: 18498.67612624644
root_mean_squared_error: 27477.529860261657

Cross validation score : 83.7399614253858

R2_Score - Cross Validation Score : 3.7162788884968307
```

3) ExtraTreeRegressor :

In [173]:
```
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: 88.38121034338118
mean_squared_error: 699340277.0177733
mean_absolute_error: 17378.189002849
root_mean_squared_error: 26445.04257923918

Cross validation score : 83.60528681917934

R2_Score - Cross Validation Score : 4.775923524201843
```

4) GradientBoostingRegressor :

In [174]:
```
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: 90.96273573775039
mean_squared_error: 543957079.8189039
mean_absolute_error: 15707.479286560227
root_mean_squared_error: 23322.88746744073

Cross validation score : 82.73750772325684

R2_Score - Cross Validation Score : 8.225228014493553
```

5) DecisionTreeRegressor :

In [175]:
```
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: 76.93882717996058
mean_squared_error: 1388062566.3219373
mean_absolute_error: 26027.99715099715
root_mean_squared_error: 37256.71169496763

Cross validation score : 66.21604205852161

R2_Score - Cross Validation Score : 10.722785121438974
```

After seeing the difference of model accuracy and cross validation score,and the R_2 score I found ExtraTreesClassifier as the best model.

After seeing the difference in model accuracy and cross-validation score, and the R_2 score I found ExtraTreesClassifier as the best model.

2) Hyper Parameter Tuning:

Hyper Parameter Tunning For Best Model :

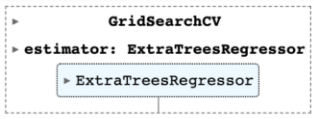
In [176]:
```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
parameter = {'n_estimators':[10,100],
             'criterion':['squared_error','mae'],
             'min_samples_split': [2,4],
             'max_features':['auto','sqrt'],
             'n_jobs':[-2,2]}
```

In [177]:
```
# Giving estimator as ExtraTreesRegressor

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

In [178]:
```
GCV.fit(X_train,y_train)
```

Out[178]:



In [179]:
```
GCV.best_params_
```

Out[179]:
```
{'criterion': 'squared_error',
 'max_features': 'sqrt',
 'min_samples_split': 2,
 'n_estimators': 100,
 'n_jobs': 2}
```

In [180]:
```
Best_mod=ExtraTreesRegressor(criterion='mae',max_features='sqrt',min_samples_split=2,n_estimators=100,n_jobs=-2)
Best_mod.fit(X_train,y_train)
pred=Best_mod.predict(X_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 88.63803456961863
mean_squared_error: 683881909.0784365
mean_absolute_error: 17062.027521367523
RMSE value: 26151.135904171286
```

With the help of hyper parameter tuning i've increased the accuracy(r2_score) from 88.38 to 88.63

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

Model Saving :

In [181]:
```
# Saving the model using .pkl

import joblib
joblib.dump(Best_mod,"House_Price.pkl")
```

Out[181]:
```
['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 [182]:
```
# Loading the saved model
model=joblib.load("House_Price.pkl")

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

Out[182]:
```
array([134325.96, 179163.12, 118808.71, 227383.37, 135987.21,  92434.97,
        94910.34, 365235.8 , 280014.15, 217998.34, 269747.59, 141034.47,
       201886.08, 208215.78, 163457.5 , 203750.89, 163807.48, 219086.42,
       157687.57, 161776.81, 170060.46, 331341.48, 196560.49, 224760.61,
       120372.37, 134258.5 , 159158.  , 231779.33, 124242.31, 144161.79,
       343498.82, 184824.37, 128804.76, 203508.08,  94180.66, 194836.93,
       161575.84,  89922.5 , 157082.37, 210896.12, 225908.2 , 220166.4 ,
       148800.6 , 183344.  , 191013.57, 194273.72, 261891.43, 200370.  ,
       172146.41, 174979.34, 170087.74,  73718.65, 176584.02, 118938.96,
       122072.88, 267052.31, 303482.89, 149993.93,  97854.  , 217232.  ,
        98834.68,  93667.28, 177290.81, 367502.9 , 152933.53, 205908.33,
       316979.38,  83859.79, 154447.87, 166754.43, 200215.09, 208656.01,
        95719.9 , 216558.28, 168613.04, 241652.5 , 134469.44, 198041.24,
       248865.24, 139529.79, 470769.71, 121582.91, 205188.84, 200137.91,
       188408.  , 224564.57, 197000.15, 307355.88, 161796.3 , 105519.07,
       119290.15, 180525.79, 182077.18, 134090.36, 139474.5 , 110048.5 ,
       126057.72, 139489.19, 239644.86, 131395.88, 147960.59, 291670.61,
       229258.8 , 189865.86, 296221.92, 266585.34, 193810.42, 203819.82,
       120747.32, 114304.96, 145351.  , 205400.67, 196287.06, 201302.37,
       191514.5 , 148256.29, 138959.07, 260390.41, 212344.82, 210679.43,
       146643.25, 141563.54, 209785.05, 139155.8 , 197386.05, 247477.41,
       215808.6 , 163367.1 , 147347.  , 324633.48, 185143.97, 366137.04,
       238553.07, 329500.27,  82038.36, 208598.06, 207001.78, 183196.37,
       180898.  , 130531.13, 173354.97, 377858.76, 220011.43, 160100.01,
       206611.5 ,  97574.13, 117676.91, 248099.65, 107750.88, 182260.31,
       157457.71, 142728.15, 126274.11, 224726.54, 133239.65, 130377.07,
```

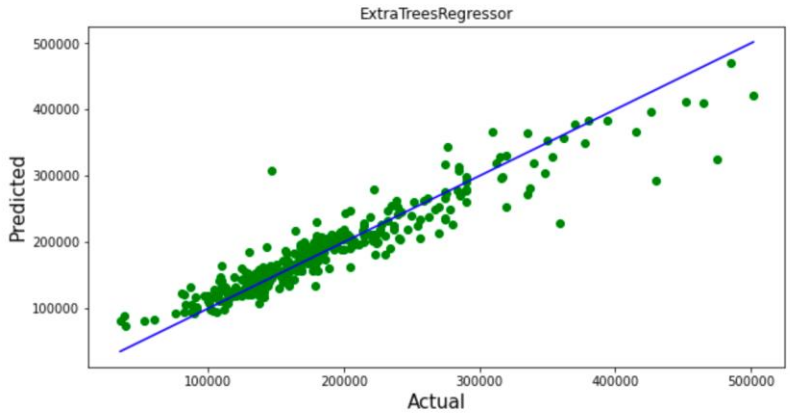
In [183]:
```
pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[183]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
Predicted	134325.96	179163.12	118808.71	227383.37	135987.21	92434.97	94910.34	365235.8	280014.15	217998.34	269747.59	141034.47	201886.08
Actual	137000.00	168500.00	115000.00	280000.00	140000.00	76000.00	88000.00	335000.0	222000.00	227000.00	286000.00	132250.00	189000.00

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

```
In [187]: plt.figure(figsize=(10,5))
          plt.scatter(y_test, prediction, c='g')
          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()
```



3.6 Interpretation of the Results:

- This dataset was very special as it had a 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 plotting 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 number of outliers and skewness in the data so we have chosen proper methods to deal with the outliers and skewness. If we ignore this outlier 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 biased.
- 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 88.38% r2_score. Also, I have improved the accuracy of the best model by running hyperparameter 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 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 data frame 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 analyzed. 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 analyzing 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.

Drawback in Model Building are:

- First drawback is the data leakage when we merge both train and test datasets.
- Followed by a greater 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 88.63% 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
- Kaggle
- Stackoverflow
- DataTrained Notes


THANK YOU