HOUSING: PRICE PREDICTION

Submitted by:

VANISREE P.G

ACKNOWLEDGMENT

First I would like to thank the Almighty for his wonderful presence with me throughout this project and helped me to make it as a successful one.

For my internship I had the pleasure of working at FILP ROBO Was a great chance for acquired knowledge, personal and Professional development. I extend whole hearted thanks to FILP ROBO under whom I worked and learned a lot and for enlightening me with their knowledge and experience to grow with the corporate working

This is a great pleasure to express my deep sense of gratitude and thanks to SME for his valuable ideas, instantaneous help, effective support and continued encouragement which enabled for the successful completion of the project. I also like to thank the data trained mentors and Technical team members for helping me with technical queries.

These are the following website which I referred for the references.

1. https://scikit-learn.org/
2. https://kaggle.com
3. www.google.com
4. www.geeksforgeeks.org

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 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.

A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses Logistic Regression is a part of the Supervised Learning method of Machine Learning. It is a statistical method for the analysis of a dataset. It has one or more independent variables that determine an outcome. There is one basic difference between Linear Regression and Logistic Regression which is

that Linear Regression's outcome is continuous whereas Logistic Regression's outcome is only limited. Here, the outcome represents a dependent variable.

- Conceptual Background of the Domain Problem

Real estate plays an integral role in the U.S. economy. Residential real estate provides housing for families. It's the greatest source of wealth and savings for many Americans. Commercial real estate, which includes apartment buildings, creates jobs and spaces for retail, offices, and manufacturing.

The value of property also depends on the proximity of the property, its size its neighbourhood and audience for which the property is subjected to be sold. For example if audience is mainly concerned of commercial purpose. Then the property which is located in densely populated area will be sold very fast and at high prices compared to the one located at remote place. Similarly if audience is concerned only on living place then property with less dense area having large area with all services will be sold at higher prices.

The company is looking at prospective properties to buy houses to enter the market. We 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.

- ## Motivation for the Problem Undertaken

To understand real world problems where Machine Learning and Data Analysis can be applied to help organizations in various domains to make better decisions with the help of which they can gain profit or can be escaped from any loss which otherwise could be possible without the study of data .One of such domain is Real Estate.

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.

Analytical Problem Framing

- **Mathematical/ Analytical Modeling of the Problem**

Regardless of where you stand on the matter of Data Science sexiness, it's simply impossible to ignore the continuing importance of data, and our ability to analyze, organize, and contextualize it. Drawing on their vast stores of employment data and employee feedback, Glassdoor ranked Data Scientist #1 in their 25 Best Jobs in America list. So the role is here to stay, but unquestionably, the specifics of what a Data Scientist does will evolve. With technologies like Machine Learning becoming ever-more common place, and emerging fields like Deep Learning gaining significant - Data Scientists continue to ride the crest of an incredible wave of innovation and technological progress.

While having a strong coding ability is important, data science isn't all about

software engineering (in fact, have a good familiarity with Python and you're good to go). Data scientists live at the intersection of coding, statistics, and critical thinking. As Josh Wills put it, "data scientist is a person who is better at statistics than any programmer and better at programming than any statistician." I personally know too many software engineers looking to transition into data scientist and blindly utilizing machine learning frameworks such as TensorFlow or Apache Spark to their data without a thorough understanding of statistical theories behind them. So comes the study of statistical learning, a theoretical framework for machine learning drawing from the fields of statistics and functional analysis.

It is important to understand the ideas behind the various techniques, in order to know how and when to use them. One has to understand the simpler methods first, in order to grasp the more sophisticated ones. It is important to accurately assess the performance of a method, to know how well or how badly it is working. Additionally, this is an exciting research area, having important applications in science, industry, and finance. Ultimately, statistical learning is a fundamental ingredient in the training of a modern data scientist

- **Data Sources and their formats**

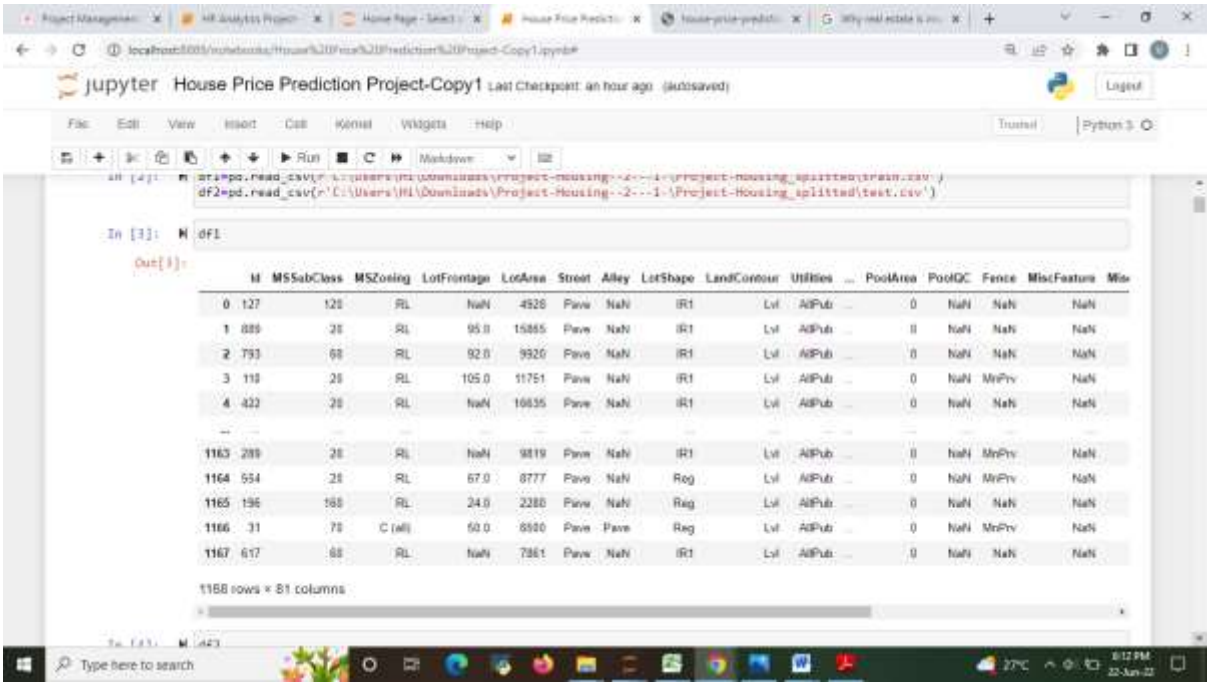
Data files are get from Flip Robo . The data is provided in the CSV file they are test.csv, train.csv file. A detailed description of data also give to know about each attribute in dataset in a ‘txt ‘ file format.

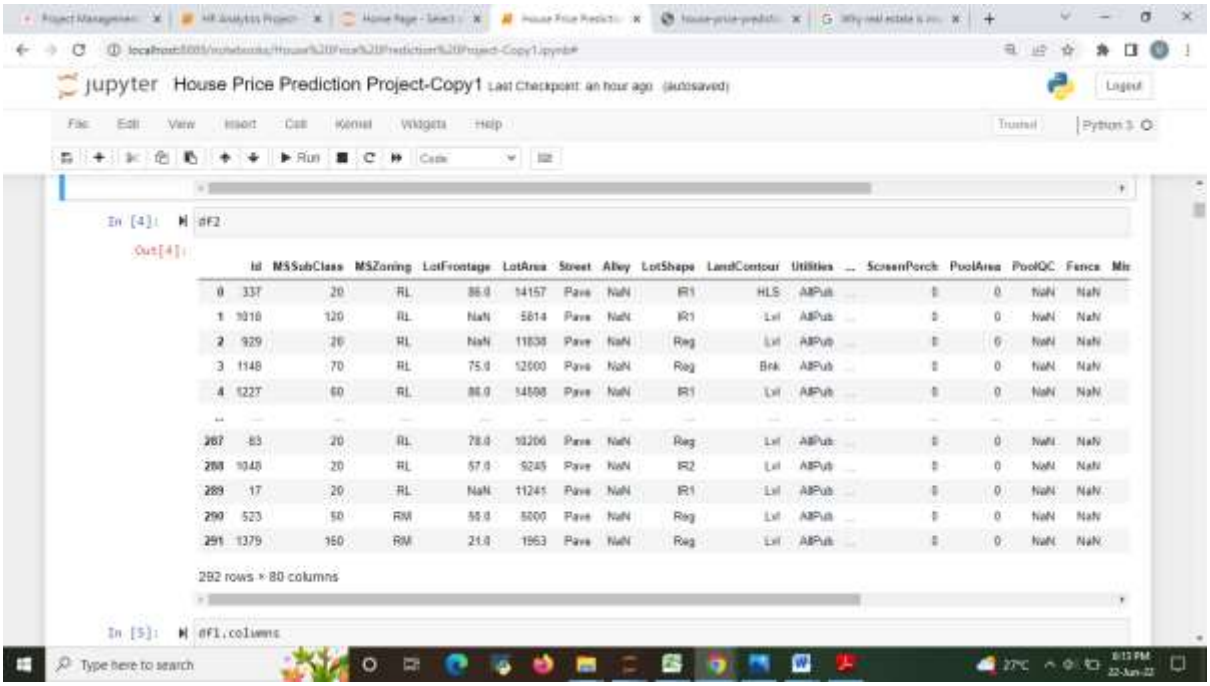
They are totally 1168 rows and 81 columns in a train.csv file. 292 rows and 80 columns in test.csv file . Our target is to find the insights of the data and to do thorough data analysis.

We have to import libraries necessary for data analysis. After we have to uploading the data using the excel file provided in two different Data Frames

- df1-train.csv file
- df2- test.csv file

snapshot of the data set

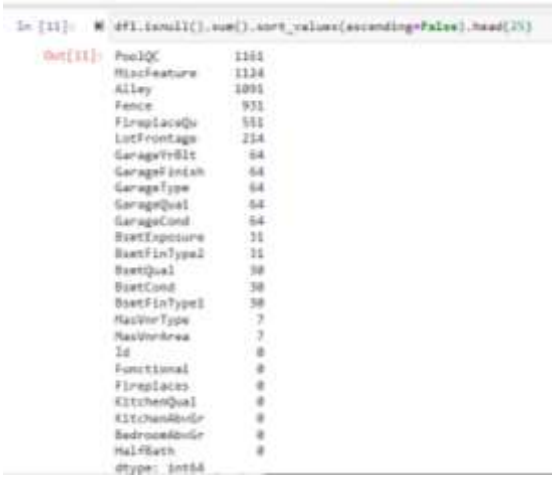




- **Data Pre-processing Done**

The raw data is taken and performed various steps to reduce skewness, outlier, class imbalance and scaling.

a) **Checking missing value from the data set.**



```
In [11]: df1.isnull().sum().sort_values(ascending=False).head(25)

Out[11]: PoolQC          1161
         MiscFeature     1124
         Alley           1091
         Fence            931
         FireplaceQu      551
         LotFrontage      214
         GarageYrBlt       64
         GarageFinish      64
         GarageType        64
         GarageQual        64
         GarageCond        64
         BsmtExposure      31
         BsmtFinType2      31
         BsmtQual          30
         BsmtCond          30
         BsmtFinType1      30
         MasVnrType         7
         MasVnrArea         7
         Id                 0
         Functional         0
         Fireplaces         0
         KitchenQual        0
         KitchenAbvGr       0
         BedroomAbvGr       0
         HalfBath           0
         dtype: int64
```

There were null value was present in the dataset but there are some outliers which also get too removed.

```
In [12]: # clouumns of 'Alley', 'MiscFeature', 'PoolQC', 'Fence' got a large number of missing values,
         # so its better to drop these columns
         df1 = df1.drop(columns=['Alley', 'MiscFeature', 'PoolQC', 'Fence'])
         df2 = df2.drop(columns=['Alley', 'MiscFeature', 'PoolQC', 'Fence'])

In [13]: df1['LotFrontage'].fillna(df1['LotFrontage'].median(),inplace=True)
         df1['MasVnrArea'].fillna(df1['MasVnrArea'].median(),inplace=True)
         df1['GarageYrBlt'].fillna(df1['GarageYrBlt'].median(),inplace=True)
         df2['LotFrontage'].fillna(df2['LotFrontage'].median(),inplace=True)
         df2['MasVnrArea'].fillna(df2['MasVnrArea'].median(),inplace=True)
         df2['GarageYrBlt'].fillna(df2['GarageYrBlt'].median(),inplace=True)

In [14]: list1 = ['BsmtFinType1','BsmtQual','BsmtCond', 'FireplaceQu', 'GarageType', 'GarageCond', 'GarageFinish', 'GarageQual', 'Mas
                 'BsmtExposure','BsmtFinType2']

         for item in list1:
             df1[item] = df1[item].fillna(df1[item].mode()[0])
         for item in list1:
             df2[item] = df2[item].fillna(df2[item].mode()[0])
```

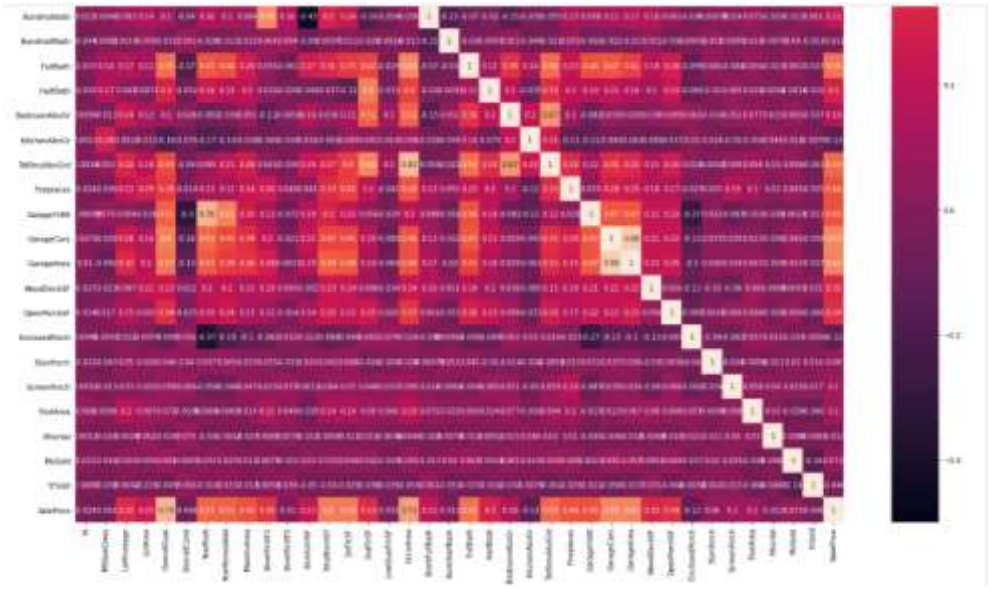
We drop large number of missing value columns. some other columns have null values these are Filled by mean ,median and mode.

b) **CORRELATION**

Correlation between all the columns in the datasets.

In the correlation Heat map, we have the following observations:

- Most of the high correlation factors are FireplaceQu, GarageType, GarageYrBlt, GarageFinish , GarageCars , GarageArea.
- GarageYrBlt and EnclosedPorch has the lowest correlation.
- YearBuilt , OverallQual , GrLivArea and MasVnrArea have high correlation.



Many outlier removal and skewness removal methods are tested and best method Is chosen in order to prevent data loss.

- **Hardware and Software Requirements and Tools Used**

Hardware used for doing the project is a ‘Laptop’ with high end specification and stable internet connection .while coming to the software part I had used ‘python jupyter notebook’ for do my python program and data analysis.

Excel file and Microsoft excel are required for the data handling. In jupyter notebook I had imported lot of python libraries are carried to this project.

1.Pandas-a library which is used to read the data ,visualisation and analysis of data.

2.Numpy-used for working with array and various mathematical operations in python.

3.Seaborn- visualization for plotting different type of plot.

4.Matplotlib- It provides an object-oriented API for embedding plots into applications .

Model/s Development and Evaluation

problem-solving approaches

Classification Model with following algorithms

- Linear Regression
- Random forest Regression
- Decision Tree Regression
- SVM
- Gradient Boosting Regression

Evaluation metrics

- Mean Absolute error
- Mean square error
- Root mean squared error
- Variance
- R2 score

Testing of Identified Approaches (Algorithms)

All the algorithms used for the training and testing.

- LR= Linear Regression()
- DT=Decision Tree Regressor()
- RF=Random Forest Regressor()
- svr=SVR()
- GBR=GradientBoostingRegressor()

- **Run and Evaluate selected models**

Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

1.LinearRegression

In [59]:
```
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

In [60]:
```
LR=LinearRegression()
LR.fit(x_train,y_train)
print(LR.score(x_train,y_train))
LR_predict=LR.predict(x_test)
```

```
0.883698026309843
```

In [61]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,LR_predict))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,LR_predict))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,LR_predict)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,LR_predict))
print('r2_score:',r2_score(y_test,LR_predict))
```

```
Mean Absolute Error:  0.08755197195731042
Mean Squared Error:  0.012712522979371475
Root Mean Squared Error:  0.11274982474208763
Explained Variance Score:  0.9218198222757136
r2_score: 0.921768284429249
```

2.Random Forest Regressor

In [62]:
```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
predictions1=rf.predict(x_test)
print(rf.score(x_train,y_train))
```

```
0.9811131921098327
```

In [63]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,predictions1))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,predictions1))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,predictions1)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,predictions1))
print('r2_score:',r2_score(y_test,predictions1))
```

```
Mean Absolute Error:  0.10036402432685654
Mean Squared Error:  0.0199687063204292
Root Mean Squared Error:  0.14131067305914724
Explained Variance Score:  0.878581596975157
r2_score: 0.8771143890390111
```

3. Decision Tree

In [64]:
```
from sklearn.tree import DecisionTreeRegressor

DTR=DecisionTreeRegressor()
DTR.fit(x_train,y_train)
print(DTR.score(x_train,y_train))
DTR_PRED=DTR.predict(x_test)
```

```
1.0
```

In [65]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,DTR_PRED))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,DTR_PRED))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,DTR_PRED)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,DTR_PRED))
print('r2_score:',r2_score(y_test,DTR_PRED))
```

```
Mean Absolute Error:  0.14492756756427871
Mean Squared Error:  0.042908498381006153
Root Mean Squared Error:  0.20714366604124335
Explained Variance Score:  0.7359943810572049
r2_score: 0.7359449854007762
```

4.SVR

In [66]:
```
from sklearn.svm import SVR

svr=SVR()
svr.fit(x_train,y_train)
print(svr.score(x_train,y_train))
svr_predict=svr.predict(x_test)
```

```
0.8639244509050528
```

In [67]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,svr_predict))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,svr_predict))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,svr_predict)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,svr_predict))
print('r2_score:',r2_score(y_test,svr_predict))
```

```
Mean Absolute Error:  0.11865487321009272
Mean Squared Error:  0.027071438482631303
Root Mean Squared Error:  0.1645340040314807
Explained Variance Score:  0.8346822125358819
r2_score: 0.8334048183117617
```

5.Gradient Boosting Regressor

In [68]:
```
from sklearn.ensemble import GradientBoostingRegressor

GBR=GradientBoostingRegressor()
GBR.fit(x_train,y_train)
print(GBR.score(x_train,y_train))
GBR_PRED=GBR.predict(x_test)
```

```
0.9669533782621488
```

In [69]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,GBR_PRED))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,GBR_PRED))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,GBR_PRED)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,GBR_PRED))
print('r2_score:',r2_score(y_test,GBR_PRED))
```

```
Mean Absolute Error:  0.08977022535951591
Mean Squared Error:  0.015268781639892629
Root Mean Squared Error:  0.12356691159000709
Explained Variance Score:  0.9068219365900744
r2_score: 0.9060373000463955
```

After evaluating the model based on MAE, MSE, RMSE, EVS, R2 SCORE the best model choose for hyper parameter tuning are Gradient Boosting Regressor, Random Forest Regressor.

Hyper parameter tuning

➢ GradientBoostingRegressor

In [70]:
```
from sklearn.ensemble import GradientBoostingClassifier  #GBM algorithm
from sklearn.model_selection import GridSearchCV
parameter = {"loss":['ls', 'lad', 'huber', 'quantile'],
             "criterion":['friedman_mse', 'mse', 'mae']}
GBR = GridSearchCV(GradientBoostingRegressor(),parameter,cv=5)
```

In [71]:
```
GBR.fit(x_train,y_train)
```

Out[71]:
```
GridSearchCV(cv=5, estimator=GradientBoostingRegressor(),
             param_grid={'criterion': ['friedman_mse', 'mse', 'mae'],
                         'loss': ['ls', 'lad', 'huber', 'quantile']})
```

In [72]:
```
GBR.best_params_
```

Out[72]:
```
{'criterion': 'friedman_mse', 'loss': 'huber'}
```

In [73]:
```
from sklearn.ensemble import GradientBoostingRegressor

GBR=GradientBoostingRegressor(criterion='friedman_mse',loss='huber')
GBR.fit(x_train,y_train)
GBR_final=GBR.predict(x_test)
```

In [74]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,GBR_final))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,GBR_final))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,GBR_final)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,GBR_final))
print('r2_score:',r2_score(y_test,GBR_final))
```

```
Mean Absolute Error:  0.09254705809797661
Mean Squared Error:  0.017361233604325524
Root Mean Squared Error:  0.13176203400192912
Explained Variance Score:  0.8940629407615017
r2_score: 0.8931605400835932
```

The best model after hyper parameter tuning is GradientBoostingRegressor has 87% of accuracy.

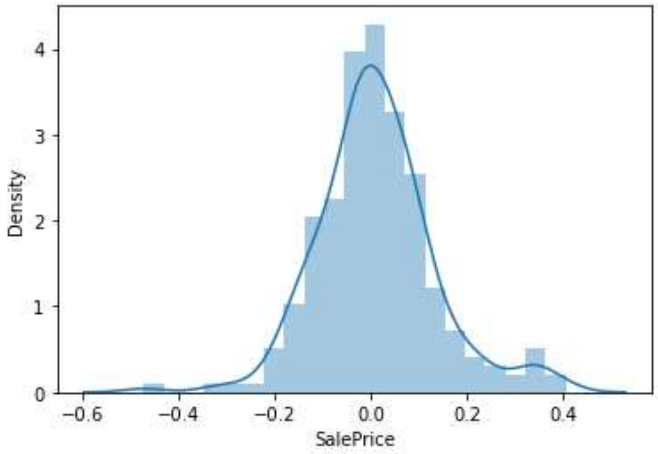
Final model

In [81]:
```
print("FINAL MODEL")
print("---------------------------------------")
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,GBR_final))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,GBR_final))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,GBR_final)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,GBR_final))
print('r2_score:',r2_score(y_test,GBR_final))
```

```
FINAL MODEL
---------------------------------------
Mean Absolute Error:  0.09254705809797661
Mean Squared Error:  0.017361233604325524
Root Mean Squared Error:  0.13176203400192912
Explained Variance Score:  0.8940629407615017
r2_score: 0.8931605400835932
```

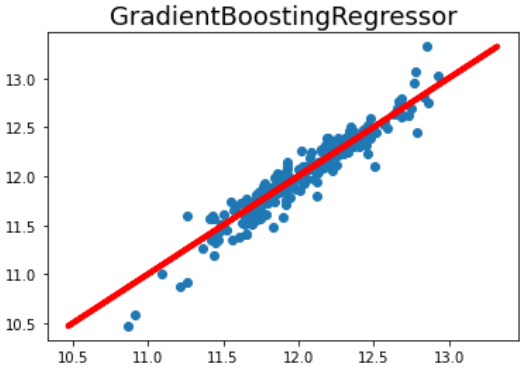
In [82]:
```
sns.distplot(GBR_PRED-y_test)
```

Out[82]:
```
<AxesSubplot:xlabel='SalePrice', ylabel='Density'>
```



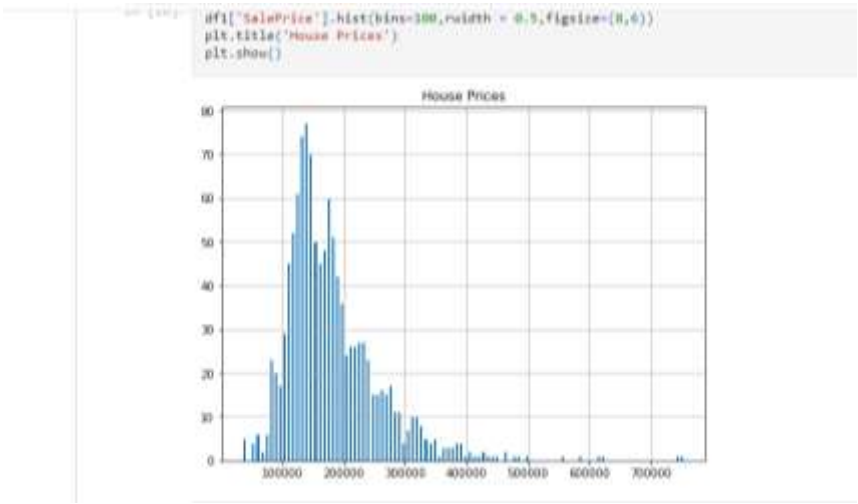
In [89]:
```
plt.scatter(GBR_PRED,y_test)
plt.plot(y_test,y_test,linewidth=4,color='Red')
plt.title('GradientBoostingRegressor',fontsize=18)
```

Out[89]:
```
Text(0.5, 1.0, 'GradientBoostingRegressor')
```

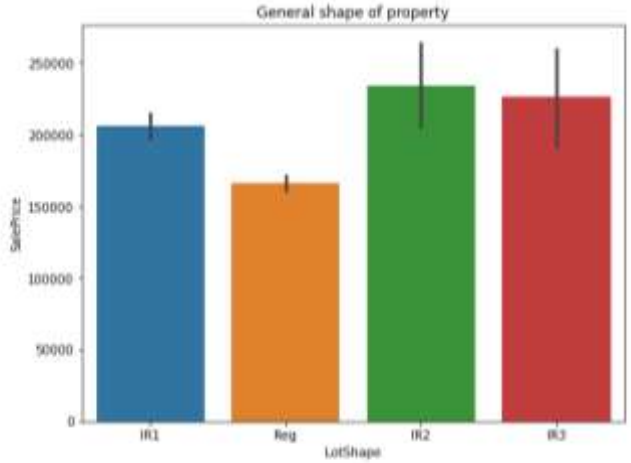


Finally Our model predicted the output.

- Key Metrics for success in solving problem under consideration
- Mean Absolute error
- Mean square error
- Root mean squared error
- Varance
- R2 score
- Data Visualizations



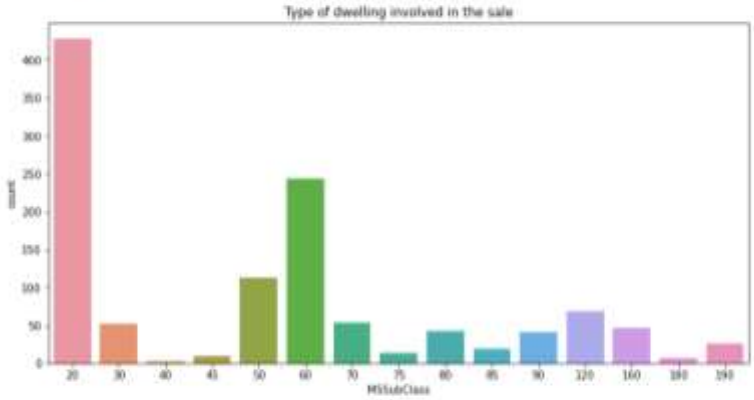
Graph based on sale price of the house.100000 to 400000 are range of sale price of house.



Inference:

Moderately Irregular and Irregular shape plot are high sale price compare with the Regular shape polt.

```
plt.figure(figsize=[12,6])
sns.countplot(df1.MSSubClass)
plt.title('Type of dwelling involved in the sale')
plt.show()
```

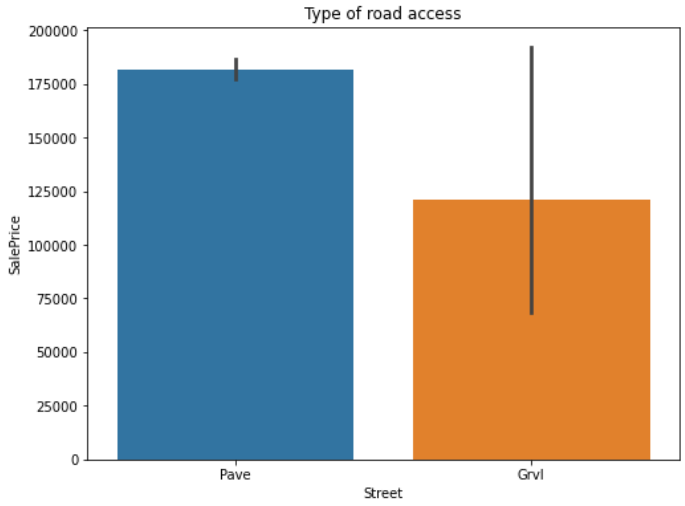


Inference: 1-STORY 1946 & NEWER ALL STYLES(20) are high demand for dwelling involved in the sale.

2-STORY 1946 & NEWER(60) second higher demand.

- 1-STORY 1946 & NEWER ALL STYLES(20) are high demand for dwelling involved in the sale.
- 2-STORY 1946 & NEWER(60) second higher demand.
- the property with the road access of Pave is in more demand and so its price is also high

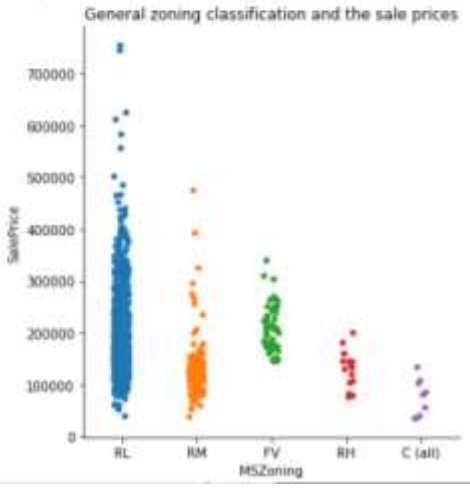
```
n [23]: plt.figure(figsize=[8,6])
sns.barplot(x='Street', y='SalePrice', data = df1.sort_values('SalePrice', ascending=False))
plt.title('Type of road access')
plt.show()
```



- **Identifies the general zoning classification of the sale.**

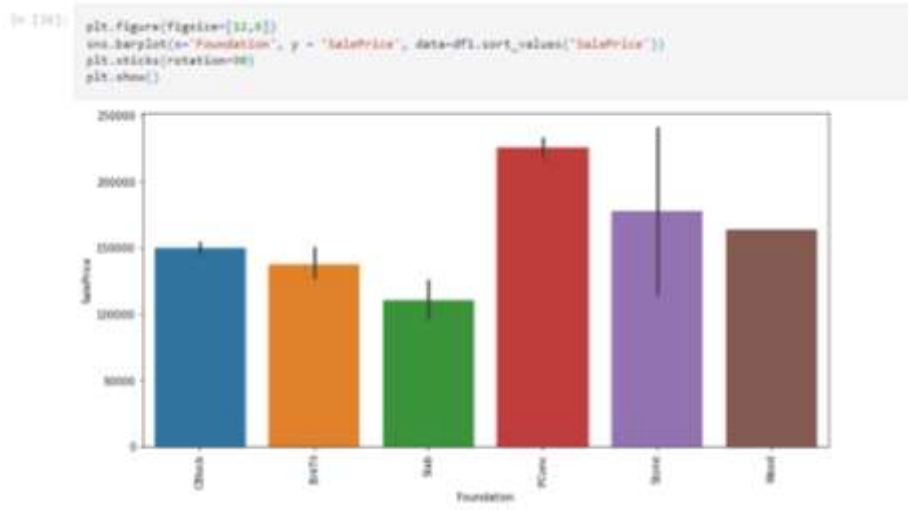
```
In [24]: plt.figure(figsize=[12,8])
         sns.catplot(x='MSZoning', y='SalePrice',data=df1.sort_values('SalePrice',ascending=False))
         plt.title('General zoning classification and the sale prices')
         plt.show()
```

```
<Figure size 864x576 with 0 Axes>
```



- For Residential Low Density (RL), the maximum prices are ranging between 50,000 to 4,00,000.
- For Floating Village Residential (FV), the maximum prices are ranging between 150000 to 250000.

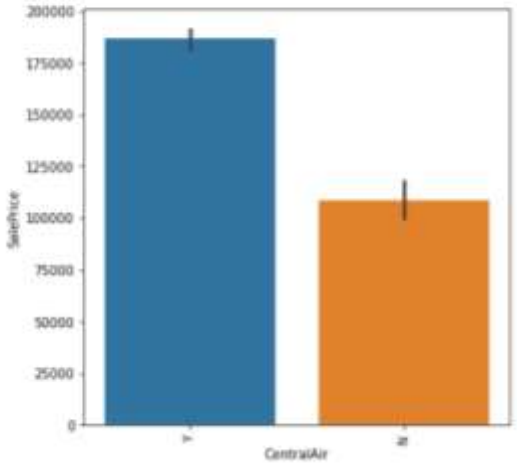
Type of foundation



- slab foundation have least sale price.
- Poured Contrete foundation have high sale price

Central air conditioning

```
In [29]: plt.figure(figsize=[6,6])
         sns.barplot(x='CentralAir', y='SalePrice', data= df1.sort_values('SalePrice', ascending=False))
         plt.xticks(rotation=90)
         plt.show()
```

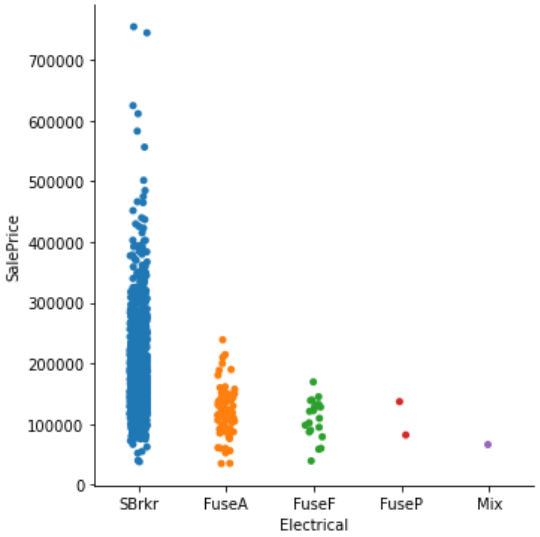


Houses having the option of central air conditioning have more price.

Electrical system

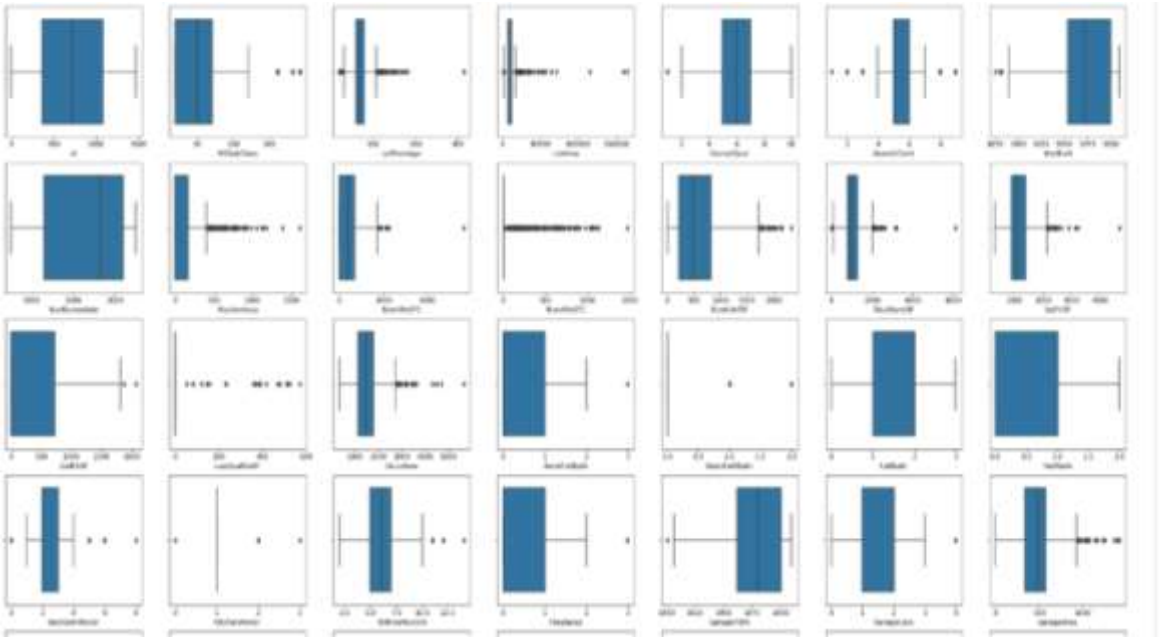
```
In [30]: plt.figure(figsize=[8,4])
         sns.catplot(x='Electrical', y='SalePrice', data=df1.sort_values('SalePrice', ascending=False))

         plt.show()
```

```
<Figure size 576x288 with 0 Axes>
```



- Most of the houses are having the electrical system of standard circuit breakers and romex.
- Least of the houses are having the Fuse Box and mostly knob & tube wiring (poor) and mixed.

- **Checking the outliers**



If we perform outliers removal the data loss will be high so not performing outliers removal Technique.

- **Checking the skewness**

```
In [38]: df1.skew()

Out[38]: MSSubClass         1.422019
         LotFrontage        2.733440
         LotArea           10.659285
         OverallQual        0.175082
         OverallCond        0.580714
         YearBuilt         -0.579204
         YearRemodAdd      -0.495864
         MasVnrArea         2.835718
         BsmtFinSF1         1.871606
         BsmtFinSF2         4.365829
         BsmtUnfSF          0.909857
         TotalBsmtSF        1.744591
         1stFlrSF           1.513707
         2ndFlrSF           0.823479
         LowQualFinSF       8.666142
         GrLivArea          1.449952
         BsmtFullBath       0.627106
         BsmtHalfBath       4.264403
         FullBath           0.057809
         HalfBath           0.656492
         BedroomAbvGr       0.243855
         KitchenAbvGr       4.369259
         TotRmsAbvGrd       0.644657
         Fireplaces         0.671966
         GarageYrBlt       -0.674913
         GarageCars        -0.358556
         GarageArea         0.189665
         WoodDeckSF         1.504929
         OpenPorchSF        2.410840
         EnclosedPorch      3.043610
         3SsnPorch          9.770611
         ScreenPorch        4.105741
         PoolArea          13.243711
         MiscVal           23.065943
         MoSold             0.220979
         YrSold             0.115765
         SalePrice          1.953878
         dtype: float64
```

```
In [41]: df2.skew()

Out[41]: MSSubClass         1.358597
         LotFrontage        0.499491
         LotArea           12.781605
         OverallQual        0.397312
         OverallCond        1.289714
         YearBuilt         -0.755233
         YearRemodAdd      -0.535600
         MasVnrArea         1.978463
         BsmtFinSF1         0.739790
         BsmtFinSF2         3.698543
         BsmtUnfSF          0.960708
         TotalBsmtSF        0.519257
         1stFlrSF           0.692047
         2ndFlrSF           0.765511
         LowQualFinSF      10.929928
         GrLivArea          1.010586
         BsmtFullBath       0.463685
         BsmtHalfBath       3.544934
         FullBath          -0.049800
         HalfBath           0.758892
         BedroomAbvGr       0.075315
         KitchenAbvGr       4.849432
         TotRmsAbvGrd       0.805535
         Fireplaces         0.540164
         GarageYrBlt       -0.677213
         GarageCars        -0.280324
         GarageArea         0.133547
         WoodDeckSF         1.708321
         OpenPorchSF        2.185030
         EnclosedPorch      3.177046
         3SsnPorch         12.277476
         ScreenPorch        4.182351
         PoolArea           0.000000
         MiscVal           13.264758
         MoSold             0.186504
         YrSold             0.018412
         dtype: float64
```

Data is highly skewed from the mean so skewness removal methods need to be followed.

- **Interpretation of the Results**

  Data is highly skewed from the mean so skewness removal methods need to be followed.

  Box plot tells that there are many outliers are present in the data so need to remove the outliers too.

  Heat map tells that which data is highly correlated with class are more important in constructing the model.

CONCLUSION

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.

- Gradient Boosting algorithm: Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models. Gradient boosting algorithm can be used to train models for both regression and classification problem. Gradient Boosting Regression algorithm is used to fit the model which predicts the continuous value.

- Design a specific question or data and get the source to determine the required data.
- Make sure the data is in an accessible format else convert it to the required format.
- Specify all noticeable anomalies and missing data points that may be required to achieve the required data.

- Create a machine learning model
- Set the baseline model that you want to achieve
- Train the data machine learning model.
- Provide an insight into the model with test data
- Now compare the performance metrics of both the test data and the predicted data from the model.
- If it doesn't satisfy your expectations, you can try improving your model accordingly or dating your data or use another data modeling technique.
- At this stage you interpret the data you have gained and report accordingly.
-

- Key Findings and Conclusions of the Study

From this dataset I get to know that each feature play a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms. Importance of removing the skewness and outlier is important. Finding the best parameters for the algorithm also plays a important role in performance and accuracy of the model.

- Learning Outcomes of the Study in respect of Data Science

  Learnt how to process the large number of data. Tried and learnt more about distribution of the data. The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important step to remove missing value or null value fill it by mean median or by mode or by o.Setting a good parameters is more important for the model accuracy. Finding a best random state played a vital roll in finding a better model.

- Limitations and Scope for Future Work

The used pre-processing methods do help in the prediction accuracy. However, experimenting with different combinations of pre-processing methods to achieve better prediction accuracy.

Make use of the available features and if they could be combined as binning features has shown that the data got improved.

The correlation has shown the association in the local data. Thus, attempting to enhance the local data is required to make rich with features that vary and can provide a strong correlation relationship.