FLIP ROBO

HOUSING: PRICE PREDICTION

Submitted by:

ASHUTOSH MISHRA

ACKNOWLEDGMENT

I’d like to extend my gratitude to my mentor Mr. Md. Kashif for giving me this opportunity to work upon this project. Below are all the details of the project which I consumed while preparing and drafting the project –

The references, research papers, data sources, professionals, etc are majorly referred from “Flip Robo Technologies” study collaterals and data repository. Also, some of the other resources like “Research Gate” were explored to gain a deep understanding on the assigned subject.

Above stated details stands correct to the best of my knowledge and I hereby acknowledge the same.

INTRODUCTION

- ## Business Problem Framing

See, now a day's almost most of the people have a desire towards buying his/her own personal houses for their standard of living; and in-order to fulfil the same, it creates a big opportunity amongst the housing real estate companies.

The housing companies who provides better amenities than their counterparts, have higher chances to grab these customers and make more sales and in turn, generate more revenue. This depicts that housing and real estate market is expanding rapidly world-wide to full-fill the need of customers; but there are lots of challenges as competition is really tough in the market b/w the real estate companies.

The problem statement is that one of the housing and Real Estate Company named as "Surprise Housing" has decided to enter in the Australian market and they seek to buy out properties below the mark-up price.

Also, the company wants to know that which variables are vital to predict the price of houses and how the variables impact the prices of the house!
Understanding and interpreting the same in a mathematical and graphical form can help an individual or a firm to invest in a planned manner based on mathematical statistics.

- ## Conceptual Background of the Domain Problem

Data science comes as a vital tool to solve business problems to help companies optimize the standards resulting in overall revenue and profits growth. Moreover, it also improves their marketing strategies and demands focus on changing trends in real estate sale and purchase aspects.

Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques globally used for achieving the business goals for housing companies.

Currently the assigned project utilises Predictive Modelling algorithm to solve the business statement.

Review of Literature

The research project was for a company “Surprise Housing” seeking to do business in Australia w.r.t. Housing Price Prediction. Being a Data Scientist, I have used the Regression Model using multiple algorithms to design and optimise the results.

Some of the classes which I explored from Scikit-learn libraries were – Statistics, Analytical Modelling, Hyper Tuning Method, CV Ratings, Predictive Modelling and Regression Model, etc.

In totality, my project comprises of eight different test cases or regression models where the objective was to train for accuracy and test for accuracy using distribution plot to best understand the linearity of clusters.

Data provided were – X_train (variables having direct impact on price) and Y_train (sales price). In the beginning, I was provided with 80 X_train variable which I condensed to 10 columns keeping in mind the Principle Component Analysis. This resulted in condensed data being trained in a much shorter span of time with utmost accuracy.

Also, during prediction of the sales price; X_Test data was provided where Y_Test required to be found out using the same prediction method on all the algorithm. Utilizing some of the concepts, I also managed to condense the X_Test variable into ten Principle Components so that the columns are similar in nature both X_Train and X_Test data.

As a result of the above analytical modelling, I have managed to achieve below top three scenarios, which are –

Regression Models	Testing Accuracy (in%)
Ada Boost Regressor	81.6
Random Forest Regressor	87.19
Gradient Boosting Regressor	88.74

- ## Motivation for the Problem Undertaken

In today’s era, every individual seeks to buy his or her own personal house but fails to develop a holistic understanding like – locality, use of statistics, past prices, analysis based on past data for better and accurate prediction, etc.

This project guided me to baseline each aspect carefully and be concrete on the decision making process regardless it’s an individual or an entity like a company.

In order to cater to the above project, my current knowledge and skill set has aided me a lot which I’d explore on this exponentially further.

Analytical Problem Framing

• Mathematical/ Analytical Modeling of the Problem

Mathematical Modeling- I have used the below statistical models to find-out the corresponding mean, median, mode and relationship among the variables.

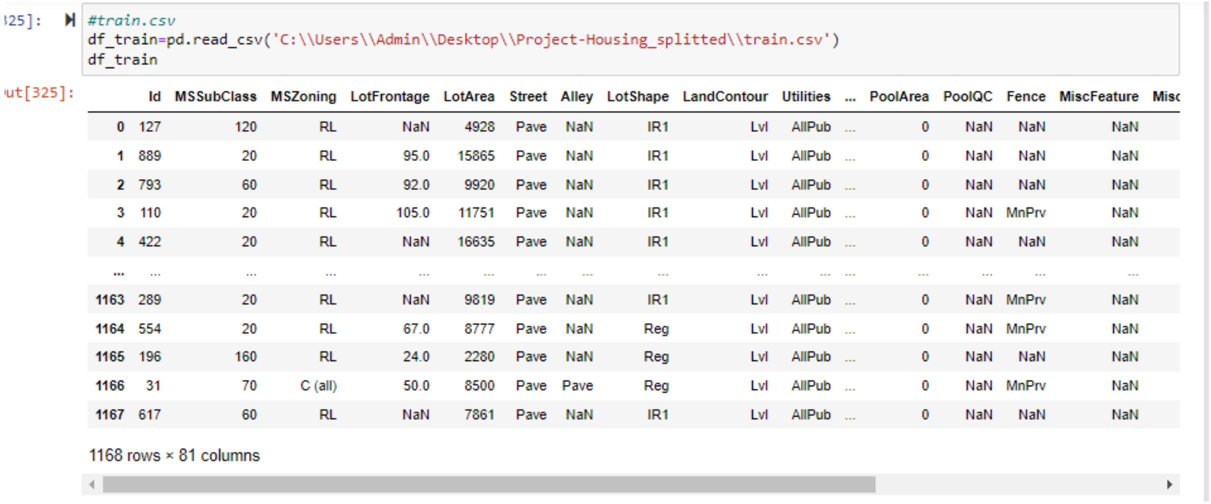
Descriptive Statistics (Describe function)- To find out the mean, median, mode, percentiles and IQR (interquartile_range)
Statistical Modeling, Correlation- To find out the relationship among the variables i.e. finding out the positive, negative and zero correlated attributes.
Multi-collinearity- To find out all those variables who are giving similar information to the target variable ('Sale Price')
Skewness- To check whether the attributes are the skewing left hand side or right hand side (Threshold value is=0.5).

Outliers- To find out variables those are having the value greater than 3.

Analytical Modeling-
Label Encoder- To convert all the categorical columns into numeric category
Simple Imputer-To replace all the null values present in the columns with mean or mode.

• Data Sources and their formats

There are two datasets is given- one is train dataset and other is test dataset. Train dataset (df_train) is given to train the model and test dataset (df_test) is given to test the model and later on deploy the best fit model and predict the Sale Price of the house.

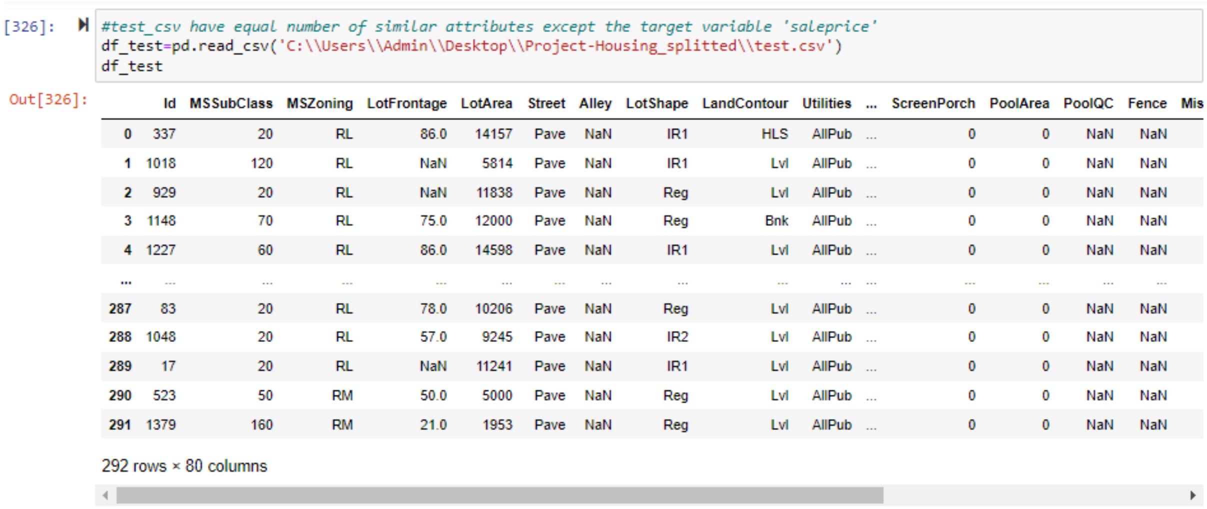


```
325]:  #train.csv
       df_train=pd.read_csv('C:\\Users\\Admin\\Desktop\\Project-Housing_splitted\\train.csv')
       df_train
```

ut[325]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	Misc
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1163	289	20	RL	NaN	9819	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
1164	554	20	RL	67.0	8777	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
1165	196	160	RL	24.0	2280	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
1166	31	70	C (all)	50.0	8500	Pave	Pave	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
1167	617	60	RL	NaN	7861	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	

1168 rows × 81 columns

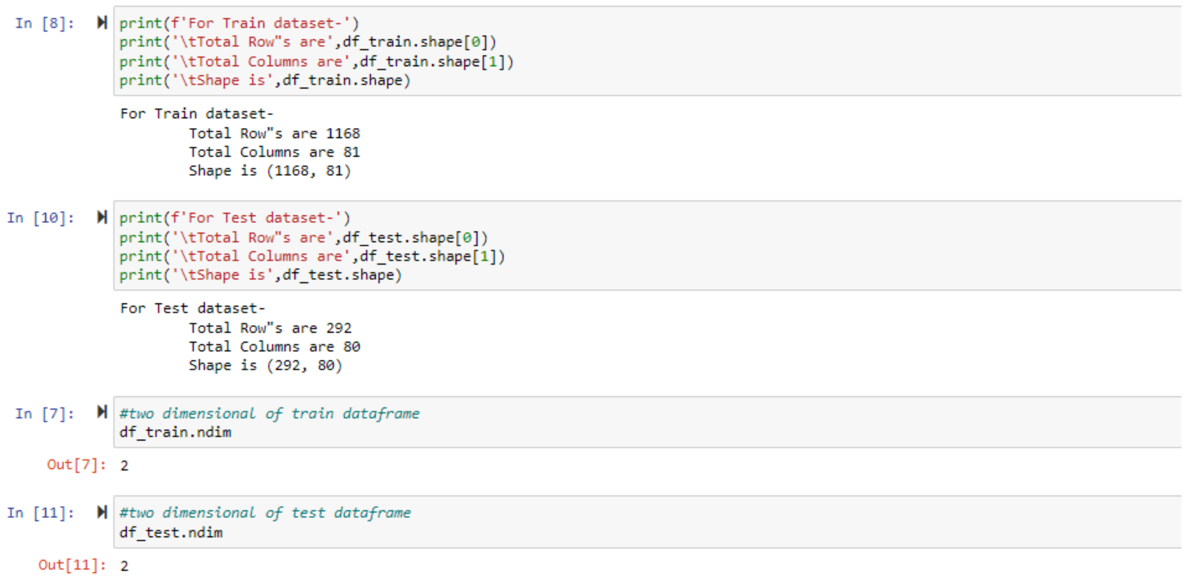


```
[326]:  #test_csv have equal number of similar attributes except the target variable 'saleprice'
        df_test=pd.read_csv('C:\\Users\\Admin\\Desktop\\Project-Housing_splitted\\test.csv')
        df_test
```

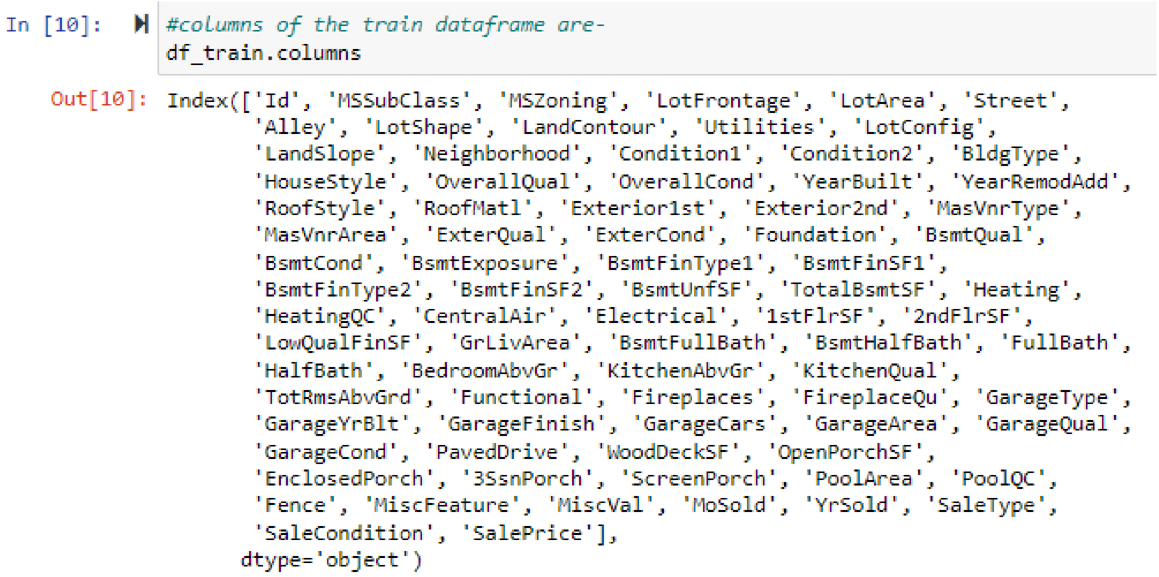
Out[326]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence	Mis
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	...	0	0	NaN	NaN	
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	...	0	0	NaN	NaN	
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
287	83	20	RL	78.0	10206	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
288	1048	20	RL	57.0	9245	Pave	NaN	IR2	Lvl	AllPub	...	0	0	NaN	NaN	
289	17	20	RL	NaN	11241	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
290	523	50	RM	50.0	5000	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
291	1379	160	RM	21.0	1953	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	

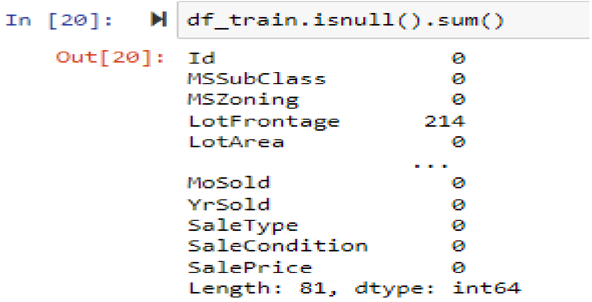
292 rows × 80 columns



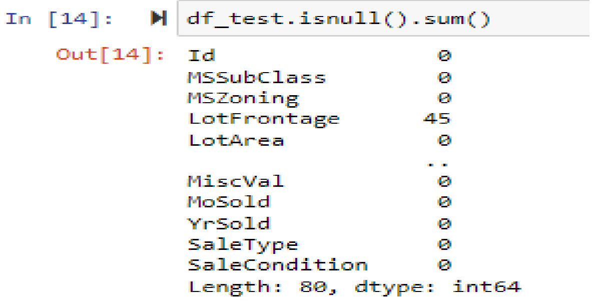
```
In [8]:   print(f'For Train dataset-')
          print('\tTotal Row"s are',df_train.shape[0])
          print('\tTotal Columns are',df_train.shape[1])
          print('\tShape is',df_train.shape)

          For Train dataset-
                  Total Row"s are 1168
                  Total Columns are 81
                  Shape is (1168, 81)

In [10]:  print(f'For Test dataset-')
          print('\tTotal Row"s are',df_test.shape[0])
          print('\tTotal Columns are',df_test.shape[1])
          print('\tShape is',df_test.shape)

          For Test dataset-
                  Total Row"s are 292
                  Total Columns are 80
                  Shape is (292, 80)

In [7]:   #two dimensional of train dataframe
          df_train.ndim

Out[7]:   2

In [11]:  #two dimensional of test dataframe
          df_test.ndim

Out[11]:  2
```



```
In [10]:  #columns of the train dataframe are-
          df_train.columns

Out[10]: Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
                'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
                'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
                'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
                'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
                'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
                'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
                'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
                'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
                'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
                'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
                'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
                'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
                'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
                'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
                'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
                'SaleCondition', 'SalePrice'],
               dtype='object')
```



```
In [20]:  df_train.isnull().sum()

Out[20]:  Id                 0
          MSSubClass         0
          MSZoning           0
          LotFrontage      214
          LotArea            0
                          ...
          MoSold             0
          YrSold             0
          SaleType           0
          SaleCondition      0
          SalePrice          0
          Length: 81, dtype: int64
```



```
In [14]:  df_test.isnull().sum()

Out[14]:  Id               0
          MSSubClass       0
          MSZoning         0
          LotFrontage     45
          LotArea          0
                          ..
          MiscVal          0
          MoSold           0
          YrSold           0
          SaleType         0
          SaleCondition    0
          Length: 80, dtype: int64
```

- Data Preprocessing Done

I’ve dropped the attributes from train and test both the datasets as these columns have less entries (almost 1000 null values) in respective columns and these are 'Alley','PoolQC','Fence','MiscFeature'.

Replaced all the nan values of the numerical columns and the categorical columns with mean and most_frequent(mode) value of the respective columns respectively.
Use Label Encoder to convert all the categorical variables value into numeric form for train and test both the datasets.

Dropped the variable ‘Utilities’ from test and train both the dataset as it has only one category present and that could lead to biasness in the model.
Remove all the zero correlated variables(‘MasVnrType’) w.r.t. target variable from train dataset as well test as well because we need to get the same numbers of columns while training and testing phase.

Dropped all the feature variables those were giving the same amount of information to the target variable.

I haven’t remove outliers as almost maximum attributes are consisting of outliers and if I consider the same the dataset could be decrease and our prediction can’t be good enough then.

I’ve use PowerTransformer method to remove the skewness and standard scaler technique to normalize the feature variable.
I’ve used Principle component analysis and select number of variables as 10.

Note I’ve done the similar process on the test dataset as well to maintain the columns equal in both the datasets.

• Data Inputs- Logic- Output Relationships

In the beginning, I was provided with 80 X_train variable which I condensed to 10 columns keeping in mind the Principle Component Analysis. This resulted in condensed data being trained in a much shorter span of time with utmost accuracy.

Also, below are some of the inputs samples which impacted the output -
GarageCars- Size of garage in car capacity
GarageArea- Size of garage in square feet
TotalBsmtSF- Total square feet of basement area
1stFlrSF- First Floor square feet
FullBath-Full bathrooms above grade

• State the set of assumptions (if any) related to the problem under consideration

No

Hardware and Software Requirements and Tools Used

Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

I have used Python IDE (Integrated Development environment) as a dedicated software throughout solving this project.

Python Libraries that I've used throughout the process are-

- Numpy- It is use for linear algebra
- Pandas- data analysis/manipulation library
- Scipy- Utility function for optimization
- Matplotlib-Data visualization and plotting library
- Seaborn- Data visualization and statistical plotting library
- Sklearn- Machine Learning Tool
- Imblearn- Deal with classification problems of Imbalanced classes
- Statsmodels-Deal with advanced statistics

Classes-

- Label Encoder-Encoding the categorical variables into numbercategory
- Simple Imputer-Replacing the null values with mean,median or mode
- variance_inflation_factor-Calculate multicollinearity
- Power Transformer-remove skewness
- StandardScaler-normalize the feature variables
- Principle Component Analysis- Reduce the dimension of the dataframe
- Cross_val_score- CV score
- GridSearchCV- Find out the best parameters for the model

Model/s Development and Evaluation

• Identification of possible problem-solving approaches (methods)

Statistical method-
When I use describe function then I find out that most of the variables mean are less than that of median and the interquartile range b/w the variables are varying too much and it shows that datasets are skewing left hand side and it indicates that the variables are not normally distributed.

Also, I have used correlation method to check what are the variables that are giving strong correlation w.r.t Target variable ‘Sale Price’.
Analytical Method-

I’ve uses boxplot and distribution plots to check the outliers and skewness of the variables respectively through the plotting’s.

• Testing of Identified Approaches (Algorithms)

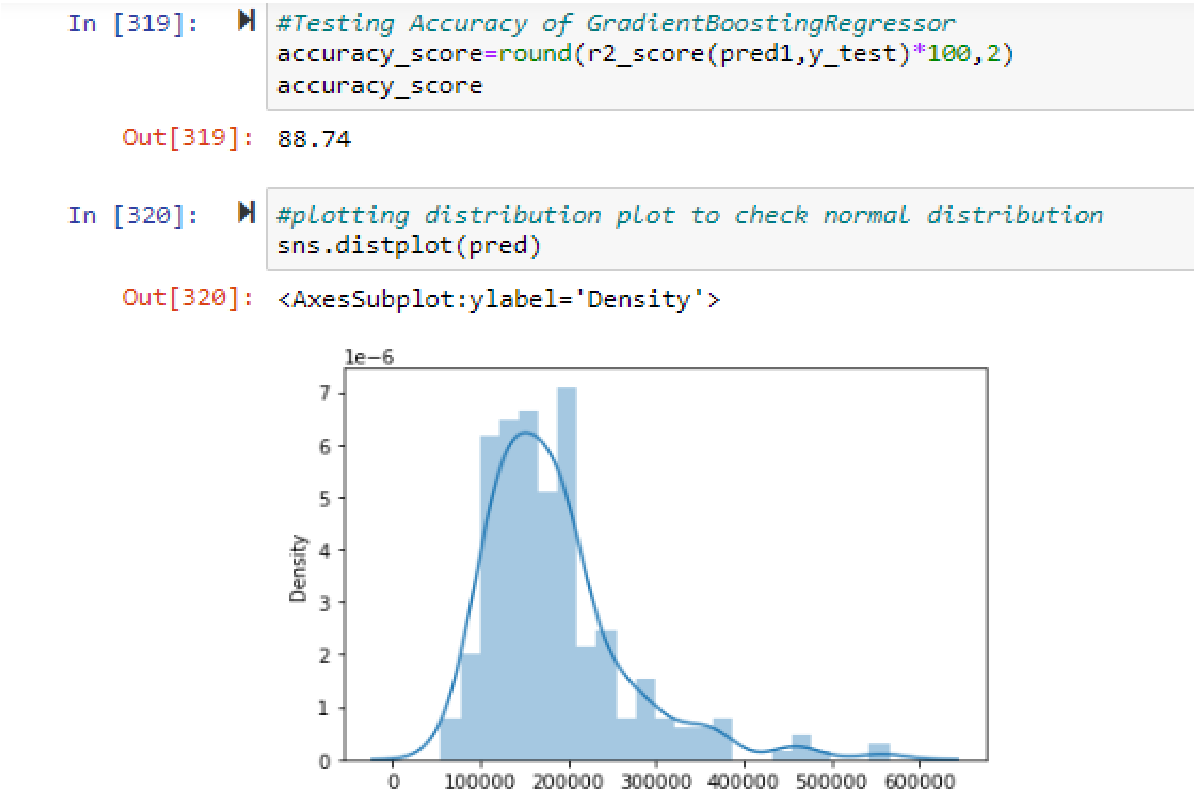
Listing down all the algorithms used for the training and testing.

```
Lr=LinearRegression()
ls=Lasso()
rd=Ridge()
en=ElasticNet()
sgd=SGDRegressor()
rf=RandomForestRegressor()
ad=AdaBoostRegressor()
grd=GradientBoostingRegressor()
```

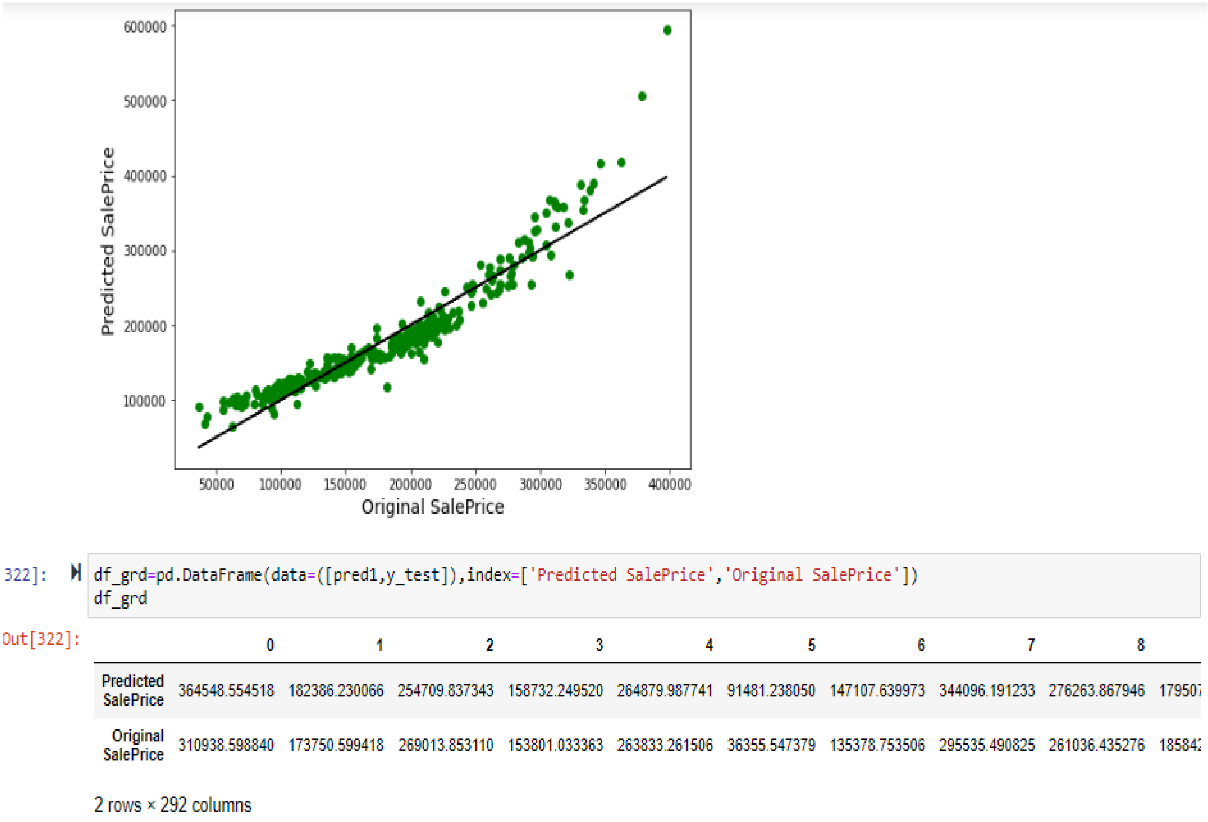
- Run and Evaluate selected models

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

1st Best Model (Gradient Boosting Regressor)



```
In [319]:  #Testing Accuracy of GradientBoostingRegressor
            accuracy_score=round(r2_score(pred1,y_test)*100,2)
            accuracy_score

Out[319]:   88.74

In [320]:  #plotting distribution plot to check normal distribution
            sns.distplot(pred)

Out[320]:   <AxesSubplot:ylabel='Density'>
```



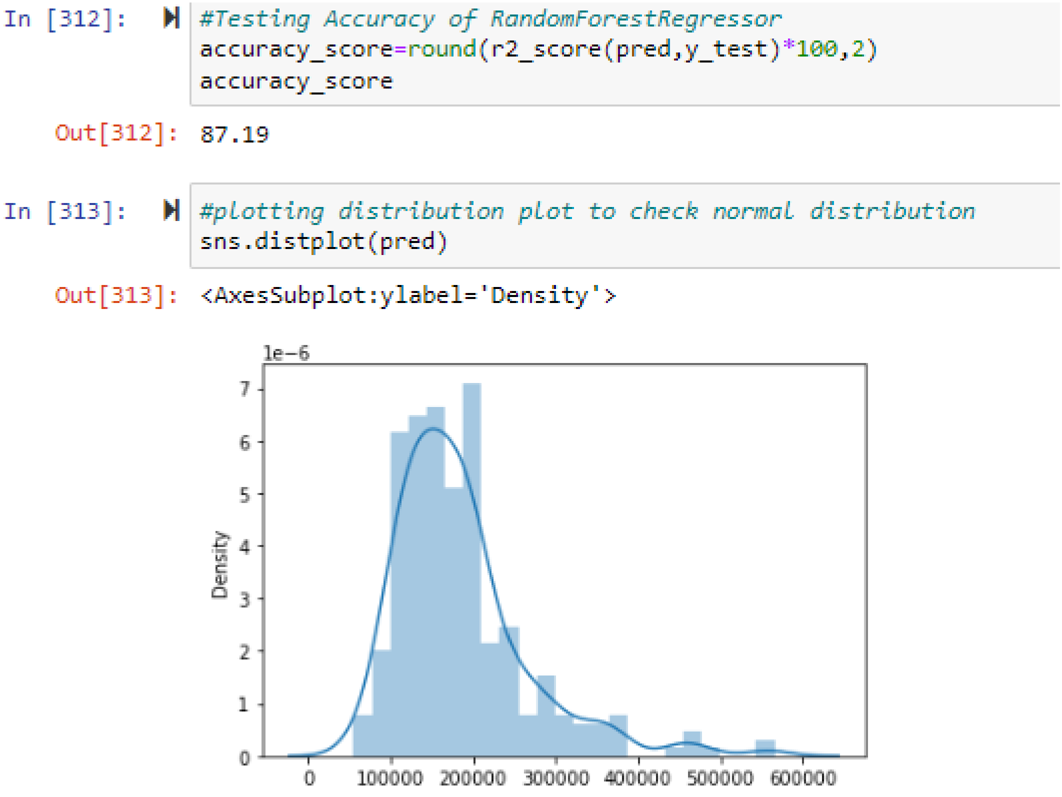
```
322]:  df_grd=pd.DataFrame(data=([pred1,y_test]),index=['Predicted SalePrice','Original SalePrice'])
       df_grd
```

Out[322]:

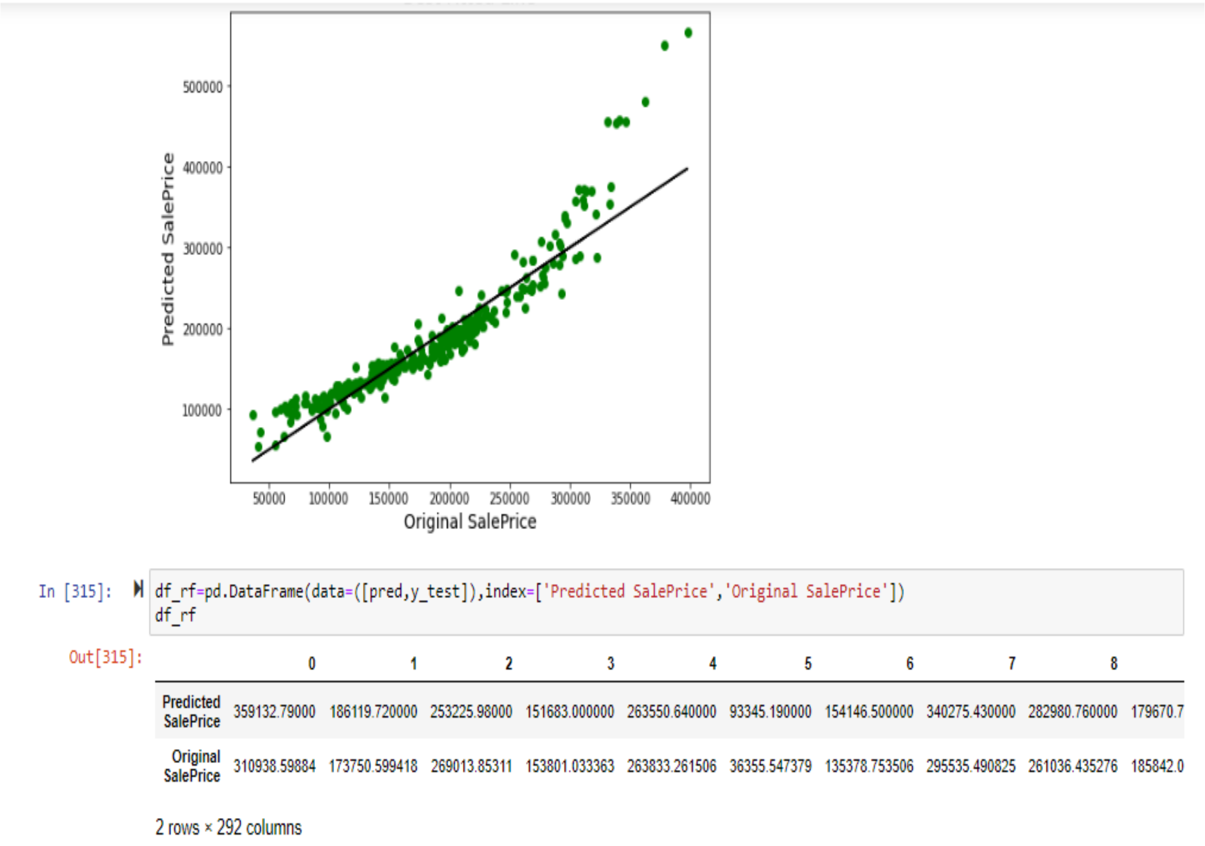
	0	1	2	3	4	5	6	7	8
Predicted SalePrice	364548.554518	182386.230066	254709.837343	158732.249520	264879.987741	91481.238050	147107.639973	344096.191233	276263.867946	179507
Original SalePrice	310938.598840	173750.599418	269013.853110	153801.033363	263833.261506	36355.547379	135378.753506	295535.490825	261036.435276	18584

2 rows × 292 columns

2nd Best Model (Random Forest Regressor)



```
In [312]:  #Testing Accuracy of RandomForestRegressor
           accuracy_score=round(r2_score(pred,y_test)*100,2)
           accuracy_score

Out[312]:  87.19

In [313]:  #plotting distribution plot to check normal distribution
           sns.distplot(pred)

Out[313]:  <AxesSubplot:ylabel='Density'>
```



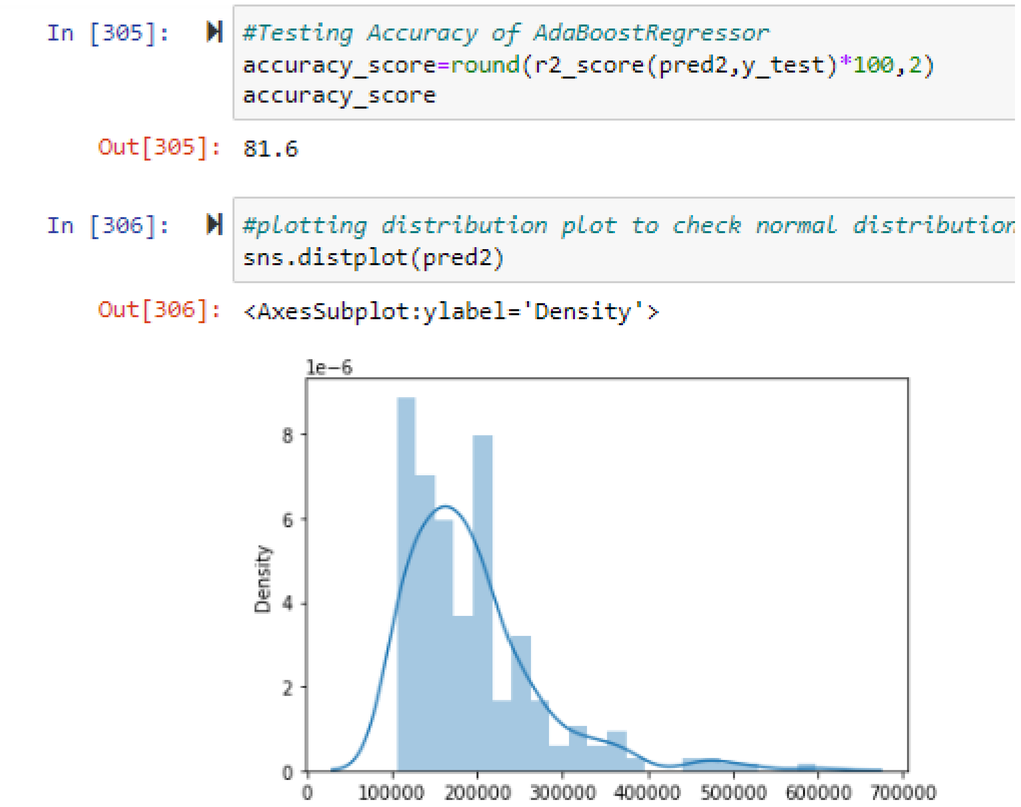
```
In [315]: df_rf=pd.DataFrame(data=([pred,y_test]),index=['Predicted SalePrice','Original SalePrice'])
          df_rf
```

Out[315]:

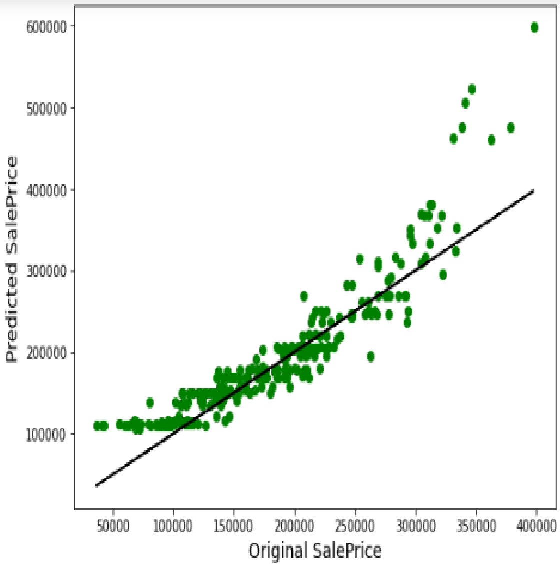
	0	1	2	3	4	5	6	7	8	
Predicted SalePrice	359132.79000	186119.720000	253225.98000	151683.000000	263550.640000	93345.190000	154146.500000	340275.430000	282980.760000	179670.7
Original SalePrice	310938.59884	173750.599418	269013.85311	153801.033363	263833.261506	36355.547379	135378.753506	295535.490825	261036.435276	185842.0

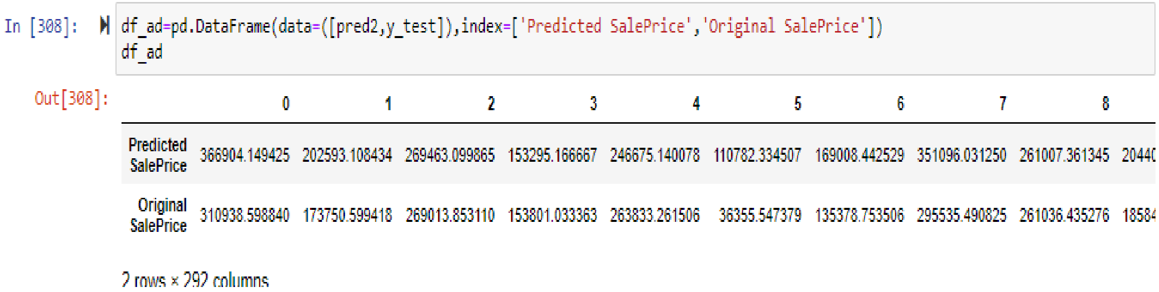
2 rows × 292 columns

3rd Best Model (Ada Boost Regressor)



```
In [305]: #Testing Accuracy of AdaBoostRegressor
          accuracy_score=round(r2_score(pred2,y_test)*100,2)
          accuracy_score

Out[305]: 81.6

In [306]: #plotting distribution plot to check normal distributior
          sns.distplot(pred2)

Out[306]: <AxesSubplot:ylabel='Density'>
```





In [308]:

```
df_ad=pd.DataFrame(data=([pred2,y_test]),index=['Predicted SalePrice','Original SalePrice'])
df_ad
```

Out[308]:

	0	1	2	3	4	5	6	7	8	
Predicted SalePrice	366904.149425	202593.108434	269463.099865	153295.166667	246675.140078	110782.334507	169008.442529	351096.031250	261007.361345	2044
Original SalePrice	310938.598840	173750.599418	269013.853110	153801.033363	263833.261506	36355.547379	135378.753506	295535.490825	261036.435276	1858

2 rows × 292 columns

- ## Key Metrics for success in solving problem under consideration

Below is the best regression models where I used below metrics -

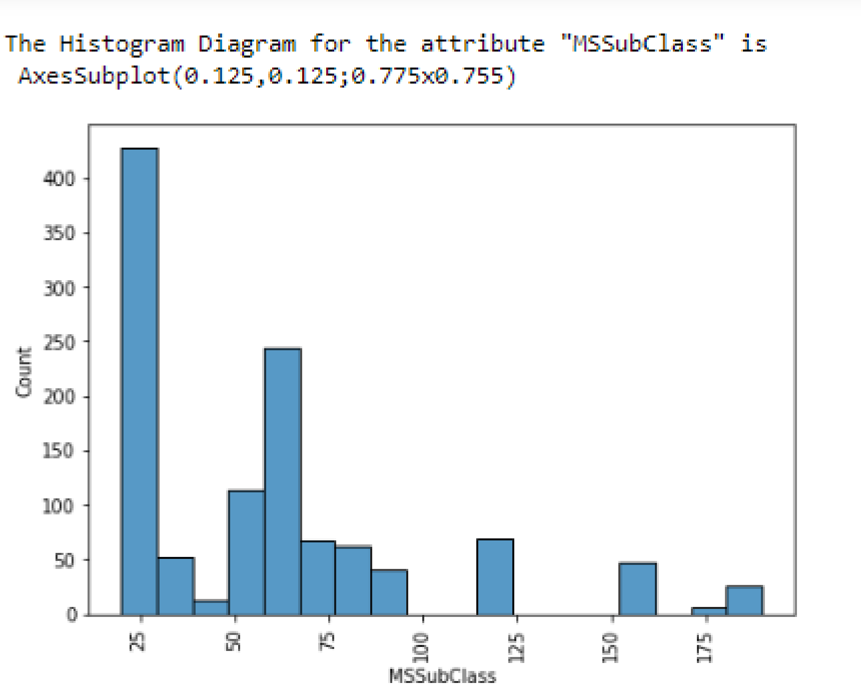
Regression Models	Testing Accuracy (in%)
Ada Boost Regressor	81.6
Random Forest Regressor	87.19
Gradient Boosting Regressor	88.74

1. CV Score – Model testing Accuracy
2. Hyper Tuning Method – Best parameter for the respective models
3. Principle Component Analysis – Course of reduction of dimensions, etc

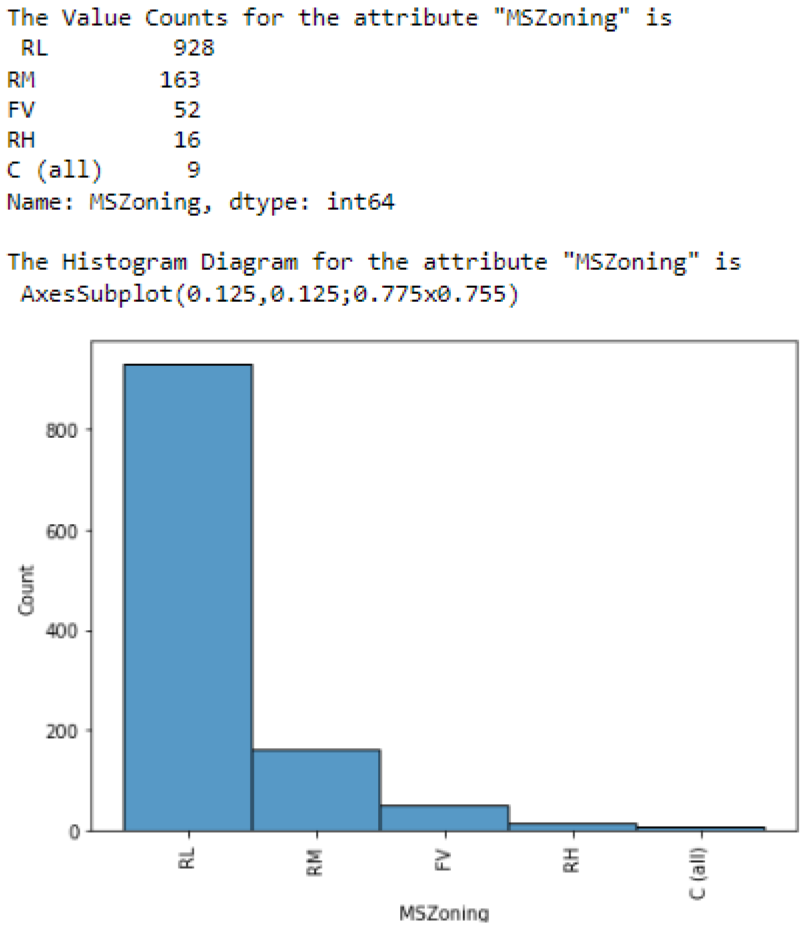
- ## Visualizations

The plots are –

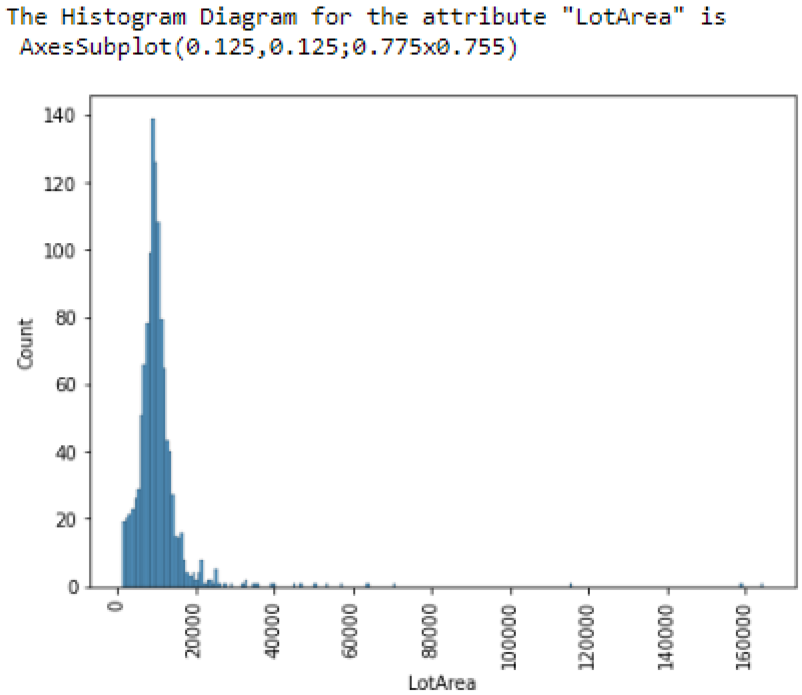
1. Histograms
2. Scatter Plot
3. Distribution plot, box plot, etc.



```
The Histogram Diagram for the attribute "MSSubClass" is
 AxesSubplot(0.125,0.125;0.775x0.755)
```

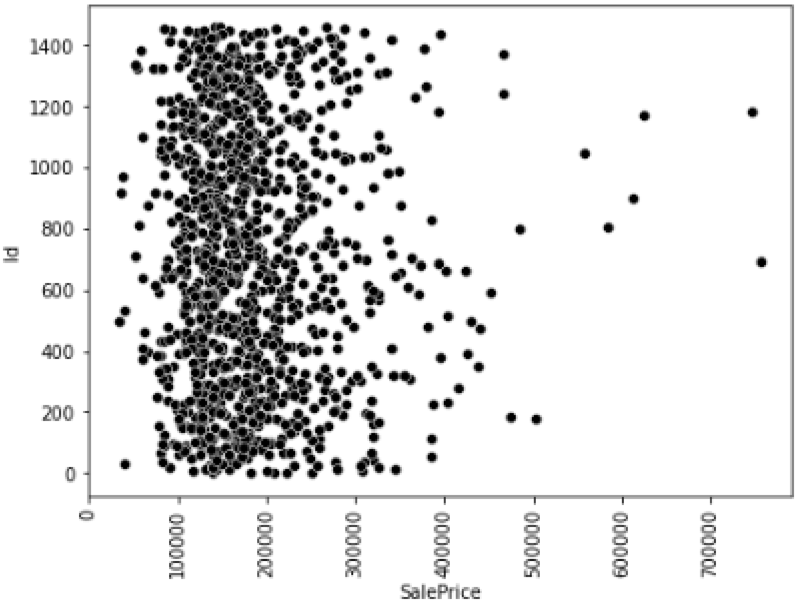


```
The Value Counts for the attribute "MSZoning" is
 RL          928
RM         163
FV          52
RH          16
C (all)      9
Name: MSZoning, dtype: int64

The Histogram Diagram for the attribute "MSZoning" is
 AxesSubplot(0.125,0.125;0.775x0.755)
```

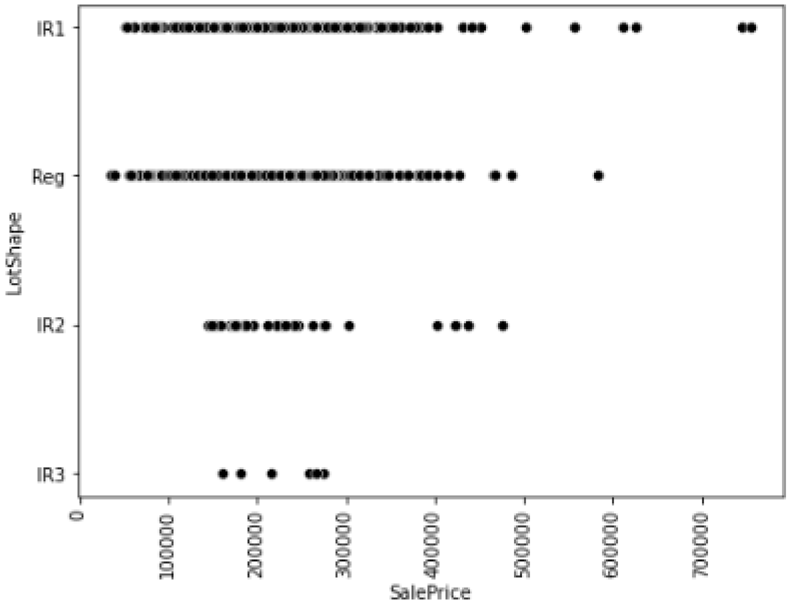


```
The Histogram Diagram for the attribute "LotArea" is
 AxesSubplot(0.125,0.125;0.775x0.755)
```

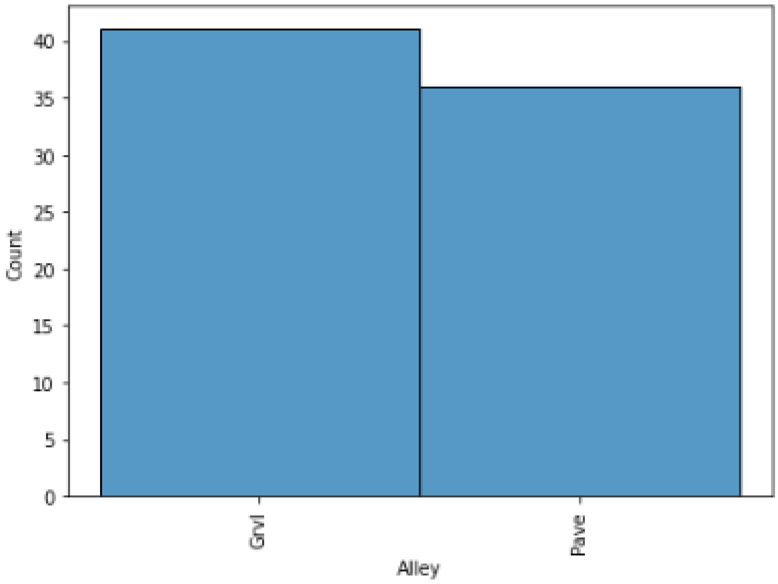
```
The Scatter Plot for the attribute "SalePrice" & "Id" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```



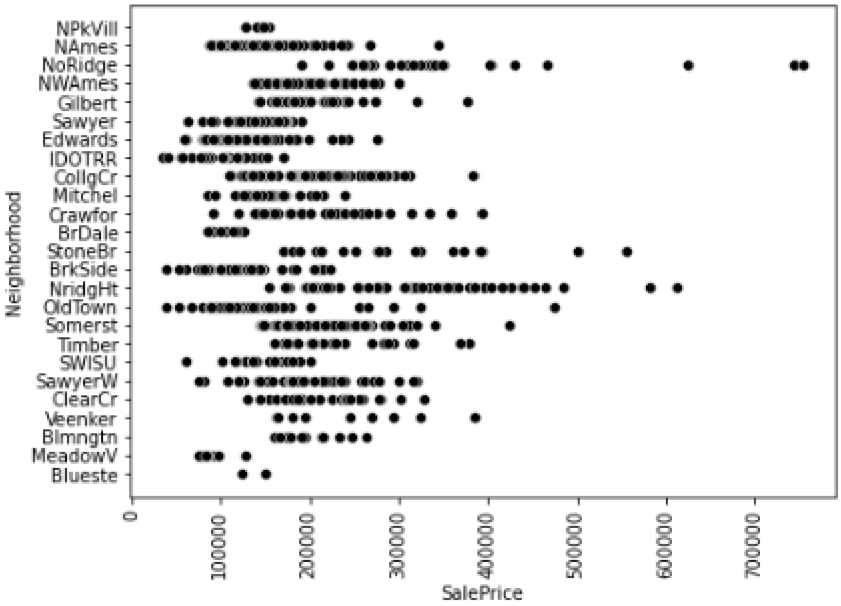
```
The Scatter Plot for the attribute "SalePrice" & "LotShape" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```



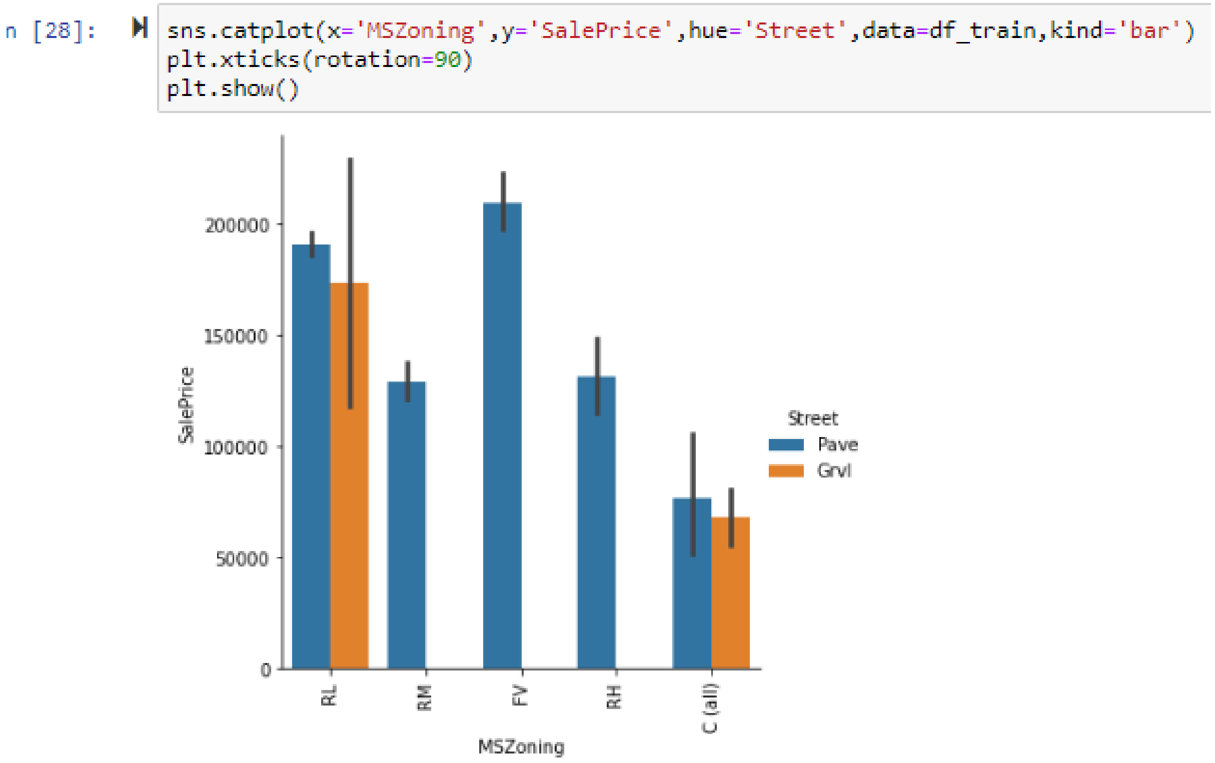
```
The Value Counts for the attribute "Alley" is
 Grvl    41
Pave    36
Name: Alley, dtype: int64

The Histogram Diagram for the attribute "Alley" is
 AxesSubplot(0.125,0.125;0.775x0.755)
```



```
The Scatter Plot for the attribute "SalePrice" & "Neighborhood" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```

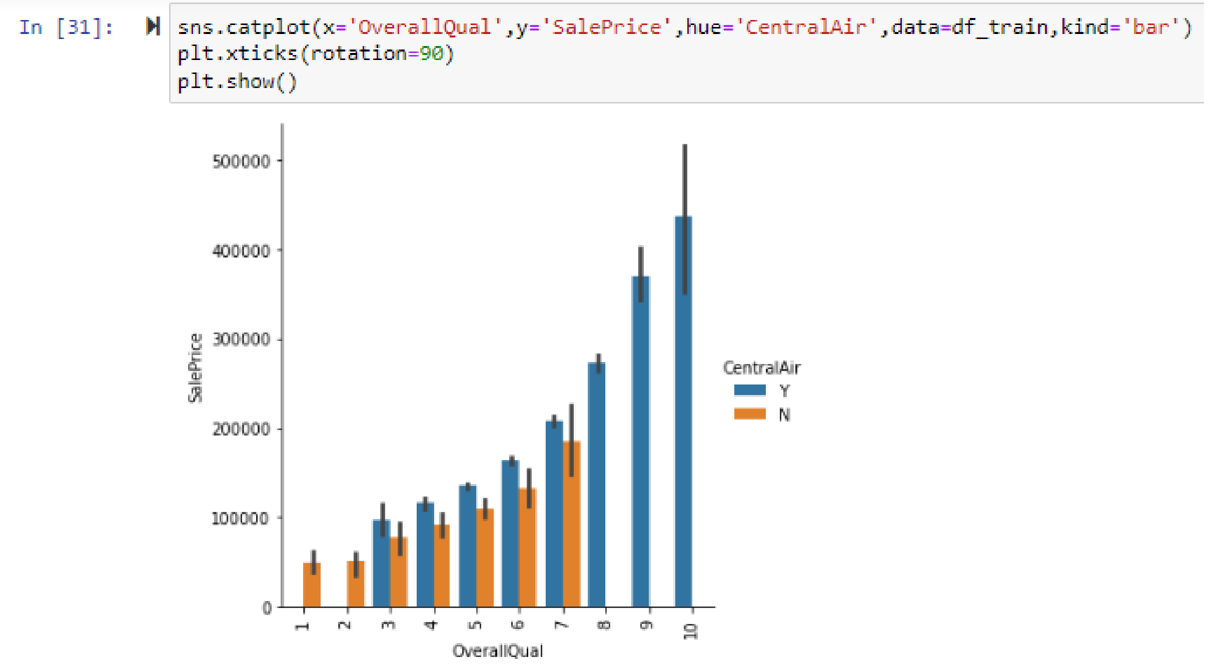


3.Categorical Plot



n [28]:

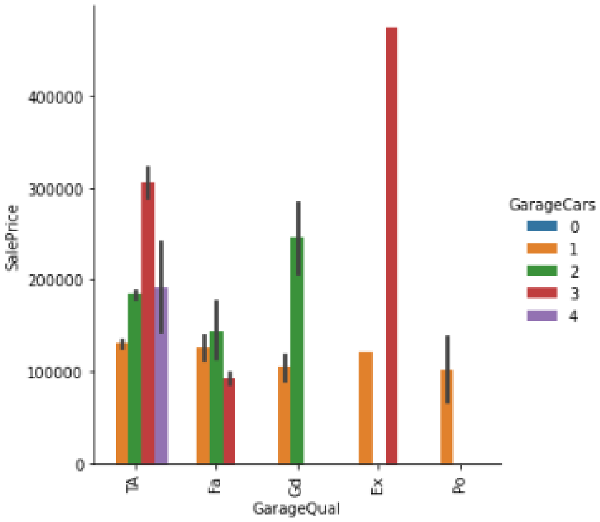
```
sns.catplot(x='MSZoning',y='SalePrice',hue='Street',data=df_train,kind='bar')
plt.xticks(rotation=90)
plt.show()
```



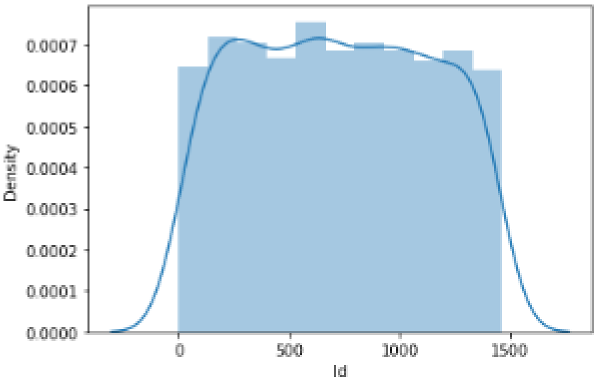
In [31]:

```
sns.catplot(x='OverallQual',y='SalePrice',hue='CentralAir',data=df_train,kind='bar')
plt.xticks(rotation=90)
plt.show()
```

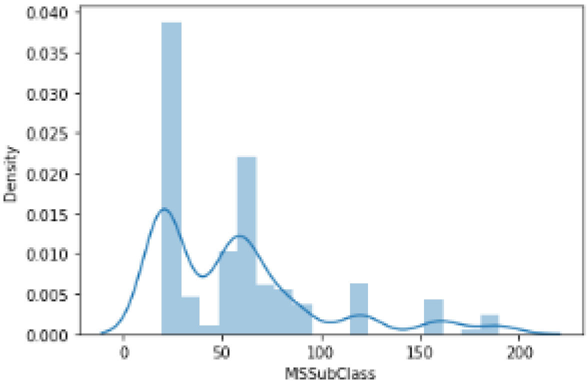
```
In [36]:  sns.catplot(x='GarageQual',y='SalePrice',hue='GarageCars',data=df_train,kind='bar')
          plt.xticks(rotation=90)
          plt.show()
```

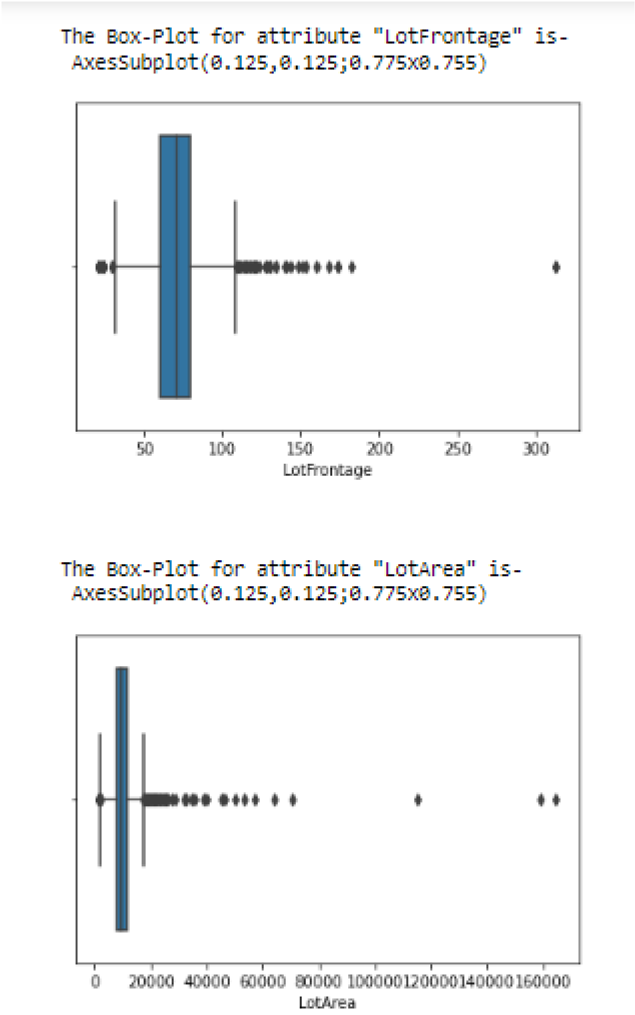


```
The Distribution Plot for attribute "Id" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```



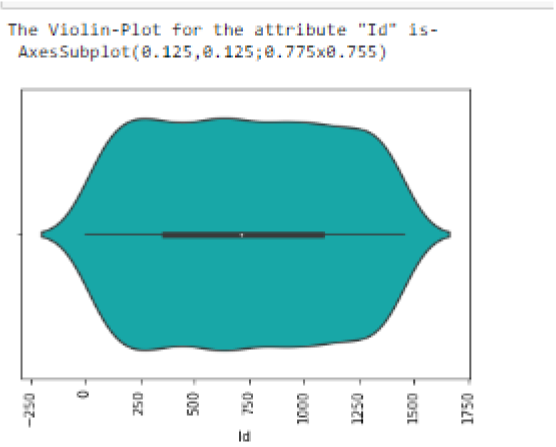
```
The Distribution Plot for attribute "MSSubClass" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```



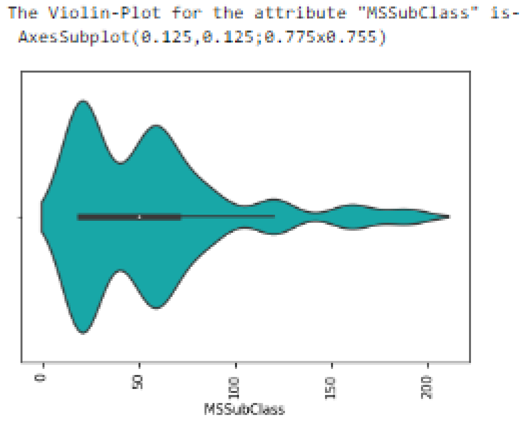


```
The Box-Plot for attribute "LotFrontage" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```

```
The Box-Plot for attribute "LotArea" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```



```
The Violin-Plot for the attribute "Id" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```



```
The Violin-Plot for the attribute "MSSubClass" is-
 AxesSubplot(0.125,0.125;0.775x0.755)
```

- ## Interpretation of the Results

Below are some of the points w.r.t. visualization, pre-processing presented above -

MSSubClass: Identifies the type of dwelling involved in the sale and 428 people are buying out flats of type '1-STORY 1946 & NEWER ALL STYLES'.

MSZoning: Identifies the general zoning classification of the sale and 928 people have bought flats in (RL)'Residential Low Density' category.

LotFrontage: Linear feet of street connected to property 111 flats those have been sold are connected with linear feet of the street

21 flats have lot size of 9600 square feet.

1164 flats are connected to paved road while 4 flats are situated at garvel road side.

CONCLUSION

I have tested out the prediction over the best three models. Out of the three, GradientBoostingRegressor is the top model as it giving the testing accuracy as almost 89% which is higher than that of RandomForestRegressor as well as AdaBoostRegressor which are giving testing accuracy as 87% and 82% respectively.

Also RandomForestRegressor and GradientBoostingRegressor training accuracy is more than 93% as well and it indicates that biasness and variance are optimal and model is regularized as well.
Hence GradientBoostingRegressor is my top accurate model in predicting the SalePrice of the house.

Also as you can see in the original and predicted SalePrice row, almost all the corresponding data's are equal mostly for Gradientboosting and Randomforest model and the best fit line is containing the most data points as well and the distribution is normal distribution

• Learning Outcomes of the Study in respect of Data Science

I have used the Regression Model using multiple algorithms to design and optimise the results. Some of the classes which I explored from Scikit-learn libraries were – Statistics, Analytical Modelling, Hyper Tuning Method, CV Score, Predictive Modelling and Regression Model, etc.

I condensed to 10 columns keeping in mind the Principle Component Analysis. This resulted in condensed data being trained in a much shorter span of time with utmost accuracy.
The visualizations helped in better and quick understanding of the outliers and skewness present in the data sets

With Mathematical Modeling heped me to find-out the corresponding mean, median, mode and relationship among the variables.
Statistical Modeling helped in Correlation for understanding the relationship among the variables

One of the key challenge which I faced was w.r.t. the data set and if the data set were in the same file, then I could have tested for much more in-depth testing accuracy. Root mean square error could have been tested if the data was provided to us in a same file access.

- ## Limitations of this work and Scope for Future Work

Data was limited and more raw data is required to be concrete on the decision making and statistical analysis.

The same analysis which was done for this project can’t be used for another housing related projects as other variables or extra identifiers hasn’t been factored in.