FLIP ROBO

Housing Project Prediction

Submitted by:

Aniruddha Sawant

ACKNOWLEDGMENT

Aniruddha Sawant ("I") acknowledge that I have used the data provided by Flip Robo ("Company") namely Data Description.txt, Train.csv and Test.csv to build the predictive model and have used Housing use case 2 and sample document for guidance and to write report.

INTRODUCTION

- ## Background of the Domain Problem: -

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.

- ## Business Problem: -

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 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. For this company wants to know:

- Which variables are important to predict the price of variable?
- How do these variables describe the price of the house?

• Review of Literature: -

There are total 80 features or variables present in the data that would be used to predict the sales price of the house. But before using those features to predict the outcome we have cleaned the data, transform the data into structured format and run many EDAs to make findings. This process end up in selecting those features which are important to predict the price and build the model.

Have used total 6 machine learning models to train the data however have chose LinearRegression Model since its accuracy of train data and test data is high and close to each other i.e., 88.87% and 88.76% respectively. Using this model, we have predicted the test data where we have no Sales_Price and have add that column in the test data.

• Motivation for the Problem Undertaken: -

We are required 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.

Analytical Problem Framing

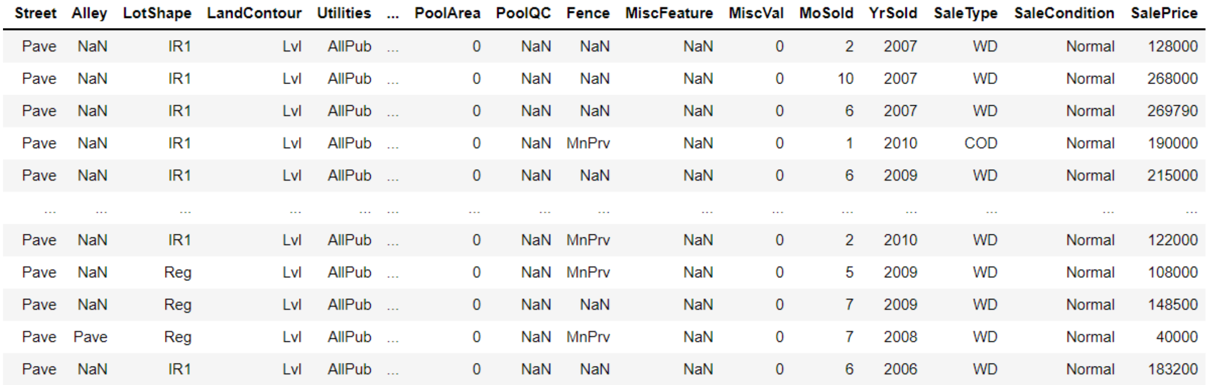
• Mathematical/ Analytical Modelling of the Problem

1. Identifying sources of the data
2. Analysing the data
3. Cleaning and processing the data
4. Selecting most important features
5. Writing down findings and observations
6. Using different models to train the data
7. Selecting best fitted model for predictions
8. Predicting outcome for test data

• Data Sources and their formats

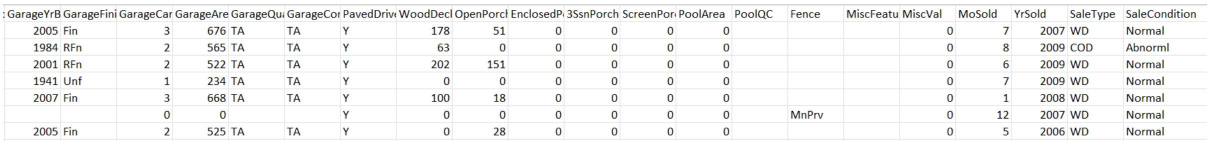
There are two excel sheets one with label and one without label. Below are the two data sources.

1. Train.csv = This excel has train data in it, that will be used to train the machine learning models. This sheet contains total 81 columns including label i.e sales price.
   Snapshot: -



Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2007	WD	Normal	128000
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	10	2007	WD	Normal	268000
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2007	WD	Normal	269790
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	1	2010	COD	Normal	190000
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2009	WD	Normal	215000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	Normal	122000
Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	5	2009	WD	Normal	108000
Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	7	2009	WD	Normal	148500
Pave	Pave	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	7	2008	WD	Normal	40000
Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2006	WD	Normal	183200

2. Test.csv = This excel sheet has test data that needs to be predicted. This data prediction will decide the decision of management.
   Snapshot: -



GarageYrB	GarageFini	GarageCar	GarageAre	GarageQua	GarageCor	PavedDrive	WoodDecl	OpenPorcl	EnclosedP	3SsnPorch	ScreenPor	PoolArea	PoolQC	Fence	MiscFeatu	MiscVal	MoSold	YrSold	SaleType	SaleCondition
2005	Fin	3	676	TA	TA	Y	178	51	0	0	0	0				0	7	2007	WD	Normal
1984	RFn	2	565	TA	TA	Y	63	0	0	0	0	0				0	8	2009	COD	Abnorml
2001	RFn	2	522	TA	TA	Y	202	151	0	0	0	0				0	6	2009	WD	Normal
1941	Unf	1	234	TA	TA	Y	0	0	0	0	0	0				0	7	2009	WD	Normal
2007	Fin	3	668	TA	TA	Y	100	18	0	0	0	0				0	1	2008	WD	Normal
		0	0			Y	0	0	0	0	0	0		MnPrv		0	12	2007	WD	Normal
2005	Fin	2	525	TA	TA	Y	0	28	0	0	0	0				0	5	2006	WD	Normal

Data Pre-processing

1. Checked the data type of each column.
2. Changed the Object data type to Integer.
3. Checked whether the data has any Null Values and fill those Null values using Mean and Mode method.
4. Checked whether the data is categorical data or continuous data.
5. There are many categorical columns which has ratings and level of agreement and disagreement. Updated Those string ratings to Integer.
   E.g: -
   'Ex' – Excellent = 5
   'Gd' – Good = 4
   'TA' - Average/Typical = 3
   'Fa' - Fair = 2
   'Po' - Poor = 1
   'NA' – Not Applicable = 0
6. Encoded remaining Object data type columns to integer using encoder technique.
7. Checked the co-relation of features with label i.e SalePrice.
8. Checked the Multicollinearity between features.
9. Checked the VIF score of features.
10. Checked the Distribution of data.
11. Identified and removed outliers those are not allowed above and below the specific limit.
12. Used Power transformation to remove the skewness from data.

- ## Data Inputs

1. Id column has unified information about houses which is not usable in building model and has no co-relation with label hence have dropped the column.

2. MSSubClass is categorical column that shows the type of dwelling involved in the sale and has no Null values.

3. MSZoning column has categorical data that identifies the general zoning classification of the sale and it has no Null values.

4. LotFrontage column is a Linear foot of street connected to property and has continuous data with NaN values in it.

5. LotArea column is a Lot size in square feet which has continuous data.

6. Street column is a categorical data that shows the type of road access to property and has no Null values.

7. Alley column is a categorical data that shows the type of alley access to property and has 1091 Null values hence we can drop that column since filling NaN values can make a whole model biased.

8. LotShape column is a categorical data that shows the general shape of property and has no Null values.

9. LandContour has categorical data that shows Flatness of the property and has no Null Values.

10. Utilities column has categorical data that shows Type of utilities available but have only one value hence we can drop the column since it will not help model to train in anyway.

11. LotConfig column has categorical data that shows Lot configuration and has no Null values.

12. LandSlope column has categorical data that shows Slope of property and has no Null values.

13. Neighborhood column has categorical data that shows Physical locations within Ame's city limits and has no Null values.

14. Condition1 and Condition2 column has categorical data that shows Proximity to various conditions and has no Null values.

15. BldgType column has categorical data that shows Type of dwelling and has no Null Values.

16. HouseStyle column has categorical data that shows Style of dwelling and has no Null values.

17. OverallQual column has categorical data that shows Rating of overall material and finish of the house and has no Null values.

18. OverallCond column has categorical data that shows Rating of overall condition of the house and has no Null values.

19. YearBuilt column has date information that shows original construction date and has no null values.

20. YearRemodAdd column has date information that shows remodel date if not then it is same as original construction date and has no null values.

21. RoofStyle column has categorical data that shows Type of roof and has no null values.

22. RoofMatl column has categorical data that shows Roof material and has no null values.

23. Exterior1st and Exterior2nd column has categorical data that shows Exterior covering on house and has no null values.

24. MasVnrType column has categorical data that shows Masonry veneer type and has Null values in it.

25. MasVnrArea column has continuous data that shows Masonry veneer area in square feet however there are 692 data points which has 0 values since those home does not have Masonry veneer this question becomes invalid and since around 60% data is invalid in this dataset, we can drop this column.

26. ExterQual column has categorical data that shows the quality of the material on the exterior and has no Null values.

27. ExterCond column has categorical data that shows present condition of the material on the exterior and has no Null Values.

28. Foundation column has categorical data that shows the Type of foundation and has no Null values.

29. BsmtQual column has categorical data that evaluates the height of the basement and has the Null values.

30. BsmtCond column has categorical data that evaluates the general condition of the basement and has the Null values.

31. BsmtExposure column has categorical data that refers to walkout or garden level walls and has Null values.

32. BsmtFinType1 and BsmtFinType2 column has categorical data that shows the Rating of basement finished area and has Null values.

33. BsmtFinSF1 and BsmtFinSF2 column has categorical data that shows Type 1 and 2 finished square feet and has no Null values.

34. BsmtUnfSF column has continuous data that shows Unfinished square feet of basement area and has no Null values.

35. TotalBsmtSF column has continuous data that shows Total square feet of basement area and has no Null values.

36. Heating column has categorical data that shows Type of heating and has no Null values.

37. HeatingQC column has categorical data that shows Heating quality and condition and has no Null values.

38. CentralAir column has categorical data that shows Central air conditioning exist or not and has no Null values.

39. Electrical column has categorical data that shows Electrical system and has no Null values.

40. 1stFlrSF column has continuous data that shows First Floor square feet and has no Null values.

41. 2ndFlrSF column has continuous data that shows Second floor square feet and has no Null values.

42. LowQualFinSF column has continuous data that shows Low quality finished square feet (all floors) and has no Null values.

43. GrLivArea column has continuous data that shows Above grade (ground) living area square feet and has no Null values.

44. BsmtFullBath column has categorical data that shows Number of Basement full bathrooms and has no Null values.

45. BsmtHalfBath column has categorical data that shows Number of Basement half bathrooms and has no Null values.

46. FullBath column has categorical data that shows Number of Full bathrooms above grade and has no Null values.

47. HalfBath column has categorical data that shows number of Half bathrooms above grade and has no Null values.

48. BedroomAbvGr column has categorical data that shows number of Bedrooms above grade (does NOT include basement bedrooms) and has no Null values.

49. KitchenAbvGr column has categorical data that shows Kitchens above grade and has no Null values.

50. KitchenQual column has categorical data that shows Kitchen quality and has no Null values.

51. TotRmsAbvGrd column has categorical data that shows Total rooms above grade (does not include bathrooms) and has no Null values.

52. Functional column has categorical data that shows Home functionality (Assume typical unless deductions are warranted) and has no Null values.

53. Fireplaces column has categorical data that shows Number of fireplaces and has no Null values.

54. FireplaceQu column has categorical data that shows Fireplace quality and half of the data is filled Null values hence we can drop the column.

55. GarageType column has categorical data that shows Garage Type and has Null values.

56. GarageYrBlt column has continuous data that shows Year garage was built and has Null values.

57. GarageFinish column has categorical data that shows Interior finish condition of the garage and has Null values.

58. GarageCars column has categorical data that shows Size of garage in car capacity and has no Null values.

59. GarageArea column has continuous data that shows Size of garage in square feet and has no Null values.

60. GarageQual column has categorical data that shows Garage quality and has Null values.

61. GarageCond column has categorical data that shows Garage condition and has Null values.

62. PavedDrive column has categorical data that shows Paved driveway and has no Null values.

63. WoodDeckSF column has continuous data that shows Wood deck area in square feet and has no Null values.

64. OpenPorchSF column has continuous data that shows Open porch area in square feet and has no Null values.

65. EnclosedPorch column has continuous data that shows Enclosed porch area in square feet and has no Null values.

66. 3SsnPorch column has continuous data that shows Three season porch area in square feet and has no Null values.

67. ScreenPorch column has continuous data that shows Screen porch area in square feet and has no Null values.

68. PoolArea column has continuous data that shows Pool area in square feet however out of the entire dataset only 7 homes have pool.

69. PoolQC column has continuous data that shows Pool quality however in this case 99% of data are NaN hence we can drop this column.

70. Fence column has categorical data that shows Fence quality however in this case 80% of data are NaN hence we can drop this column.

71. MiscFeature column has categorical data that shows Miscellaneous feature not covered in other categories and has 1124 NaN in the data hence have dropped the column.

72. MiscVal column has continuous data that shows Value of miscellaneous feature however in 1126 cases there is no value given as there is no MiscFeature available hence we can drop this column.

73. MoSold column has categorical data that shows Month Sold (MM) and has no Null values.

74. YrSold column has categorical data that shows Year Sold (YYYY) and has no Null values.

75. SaleType column has categorical data that shows Type of sale and has no Null values.

76. SaleCondition column has categorical data that shows Condition of sale and has no Null values.

77. SalePrice column is the Label column and has continuous data that shows selling price and also it has no Null values.

- ## Hardware and Software Requirements and Tools Used

1. Libraries and packages used
    - import numpy as np – For Numpy work
    - import pandas as pd – To work on DataFrame
    - import seaborn as sns – Plotting Graphs
    - import matplotlib.pyplot as plt - Plotting Graphs
    - import pickle – To save the Model
    - from sklearn.preprocessing import StandardScaler (To scale the train data), OrdinalEncoder(To encode object data to Integer), PowerTransformer (To remove skewness from dataset)
    - enc = OrdinalEncoder() = Assigned OrdinalEncoder to variable
    - from statsmodels.stats.outliers_influence import variance_inflation_factor – To calculate VIF score
    - from sklearn.model_selection import train_test_split – To split the data into train and test.
    - from sklearn import metrics, from sklearn.linear_model import Ridge, Lasso, RidgeCV, LassoCV – To regularize the model.
    - from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, accuracy_score – To calculate and analyse model metrics.
    - import warnings, warnings.filterwarnings('ignore') – To ignore unwanted Warnings
2. Machine Learning models used
    - from sklearn.linear_model import LinearRegression
    - from sklearn.tree import DecisionTreeRegressor
    - from sklearn.ensemble import AdaBoostRegressor
    - from sklearn.ensemble import GradientBoostingRegressor
    - from sklearn.ensemble import RandomForestRegressor
    - from sklearn.neighbors import KNeighborsRegressor
3. Hardware used – 11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00GHz 3.00 GHz with 8.00 GB RAM and Windows 11
4. Software used – Anaconda and Jupyter Notebook to build the model.

Model/s Development and Evaluation

- **Identification of possible problem**
  1. The data was not structured and organized hence cleaned the data using various data cleaning and pre-processing techniques.
  2. There are many outliers present in the data hence removed outliers
  3. There was a skewness in the data hence have removed the skewness from the data.
  4. Scaled the data using Standard Scalar to make the data standardized to build a model.

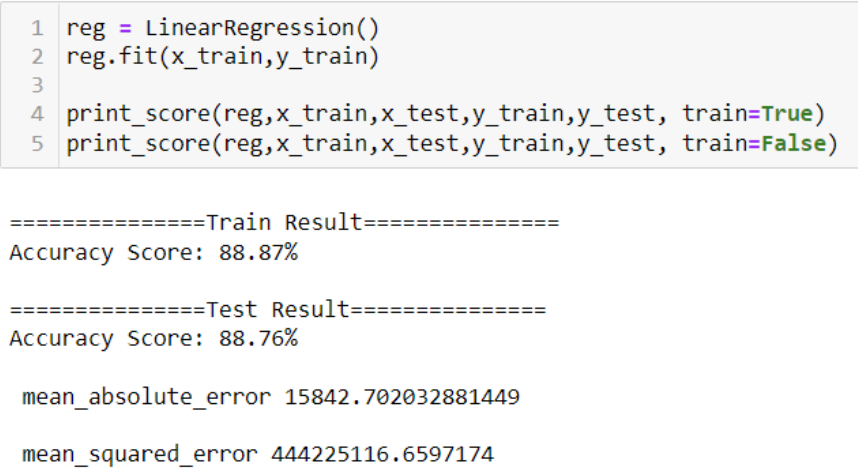
- **Testing of Identified Approaches**

  These are the algorithms which have been used to train and test data.

  1. LinearRegression
  2. DecisionTreeRegressor
  3. AdaBoostRegressor
  4. GradientBoostingRegressor
  5. RandomForestRegressor
  6. KNeighborsRegressor

- **Run and evaluate selected models**

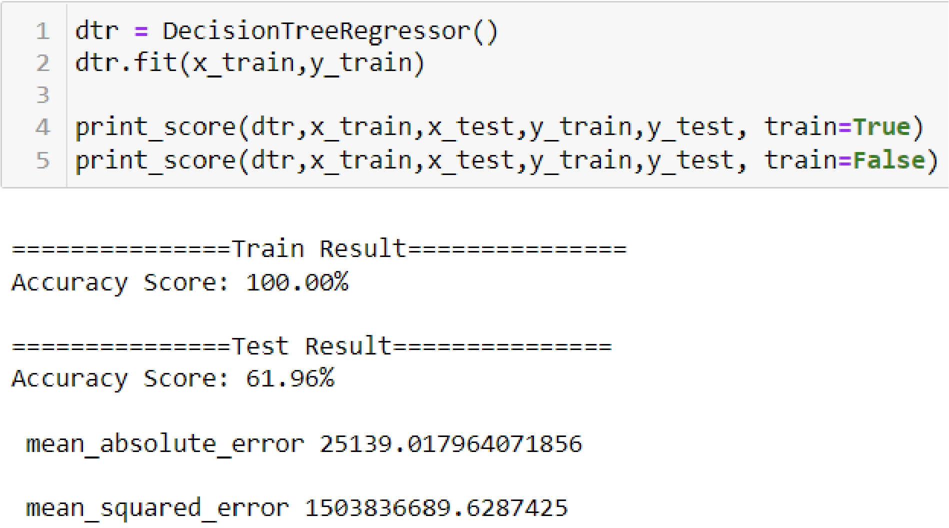
1. **LinearRegression**: -



```
reg = LinearRegression()
reg.fit(x_train,y_train)

print_score(reg,x_train,x_test,y_train,y_test, train=True)
print_score(reg,x_train,x_test,y_train,y_test, train=False)
```

```
==============Train Result==============
Accuracy Score: 88.87%

==============Test Result==============
Accuracy Score: 88.76%

 mean_absolute_error 15842.702032881449

 mean_squared_error 444225116.6597174
```

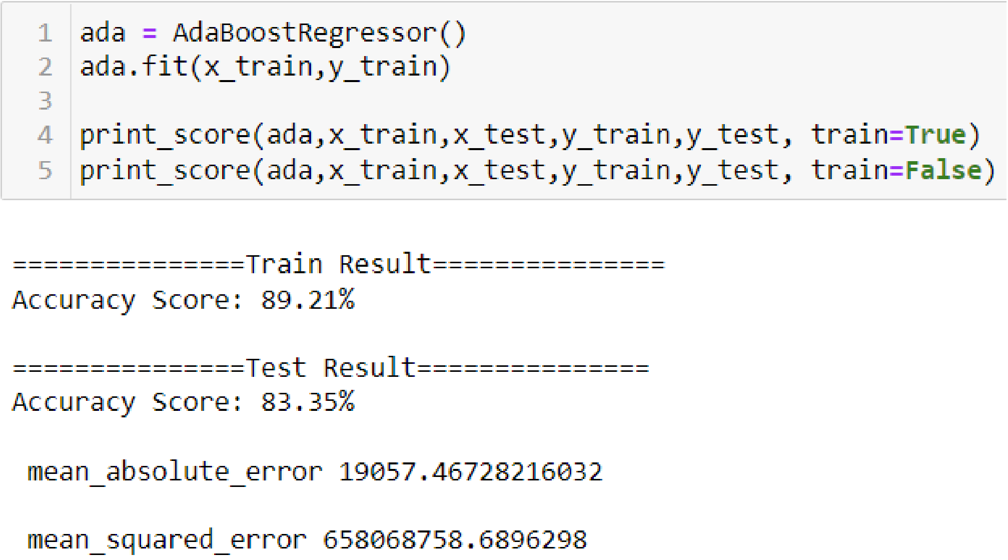
2. **DecisionTreeRegressor**: -



```
dtr = DecisionTreeRegressor()
dtr.fit(x_train,y_train)

print_score(dtr,x_train,x_test,y_train,y_test, train=True)
print_score(dtr,x_train,x_test,y_train,y_test, train=False)
```

```
==============Train Result==============
Accuracy Score: 100.00%

==============Test Result==============
Accuracy Score: 61.96%

 mean_absolute_error 25139.017964071856

 mean_squared_error 1503836689.6287425
```

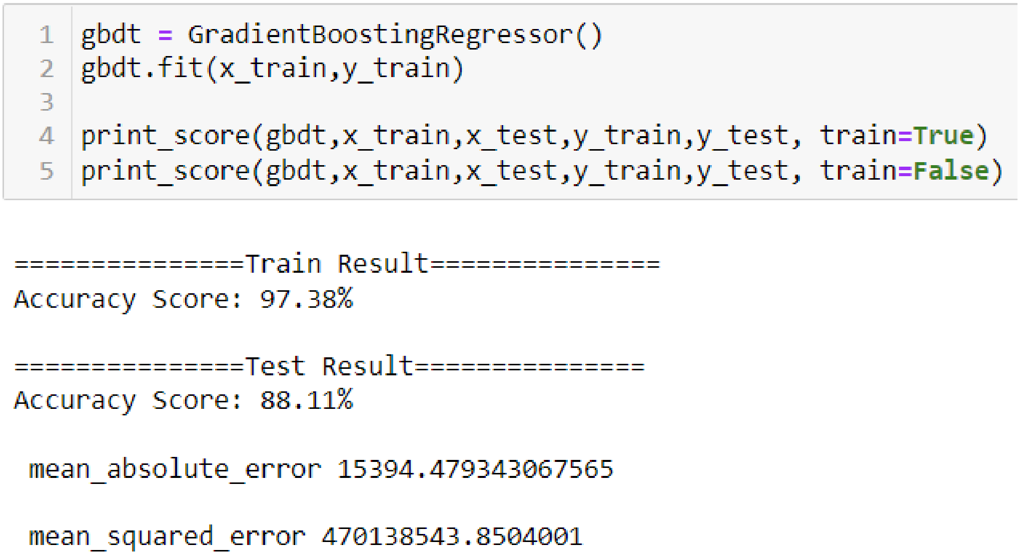
3. **AdaBoostRegressor**: -



```
ada = AdaBoostRegressor()
ada.fit(x_train,y_train)

print_score(ada,x_train,x_test,y_train,y_test, train=True)
print_score(ada,x_train,x_test,y_train,y_test, train=False)
```

```
==============Train Result==============
Accuracy Score: 89.21%

==============Test Result==============
Accuracy Score: 83.35%

 mean_absolute_error 19057.46728216032

 mean_squared_error 658068758.6896298
```

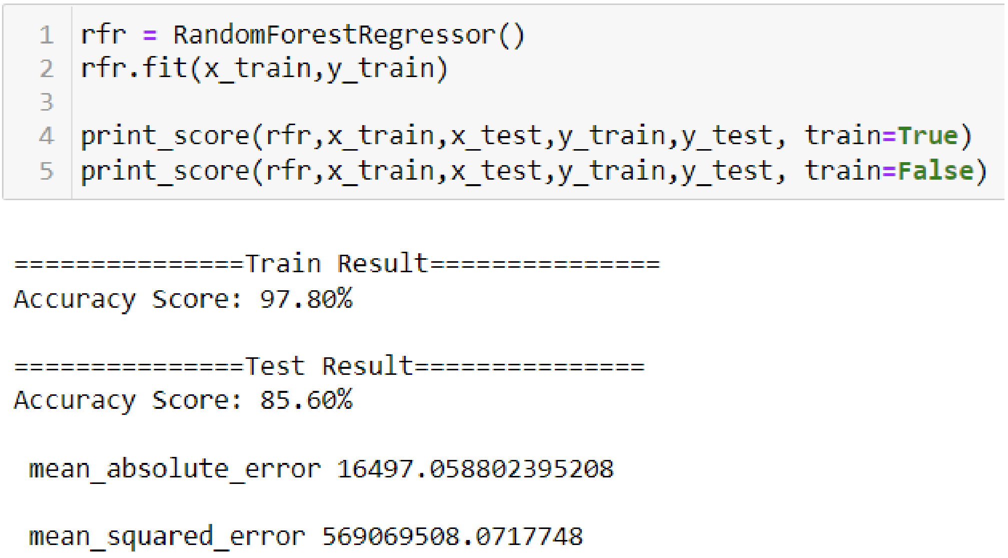
4. **GradientBoostingRegressor**: -



```python
gbdt = GradientBoostingRegressor()
gbdt.fit(x_train,y_train)

print_score(gbdt,x_train,x_test,y_train,y_test, train=True)
print_score(gbdt,x_train,x_test,y_train,y_test, train=False)
```

```
===============Train Result===============
Accuracy Score: 97.38%

===============Test Result===============
Accuracy Score: 88.11%

 mean_absolute_error 15394.479343067565

 mean_squared_error 470138543.8504001
```

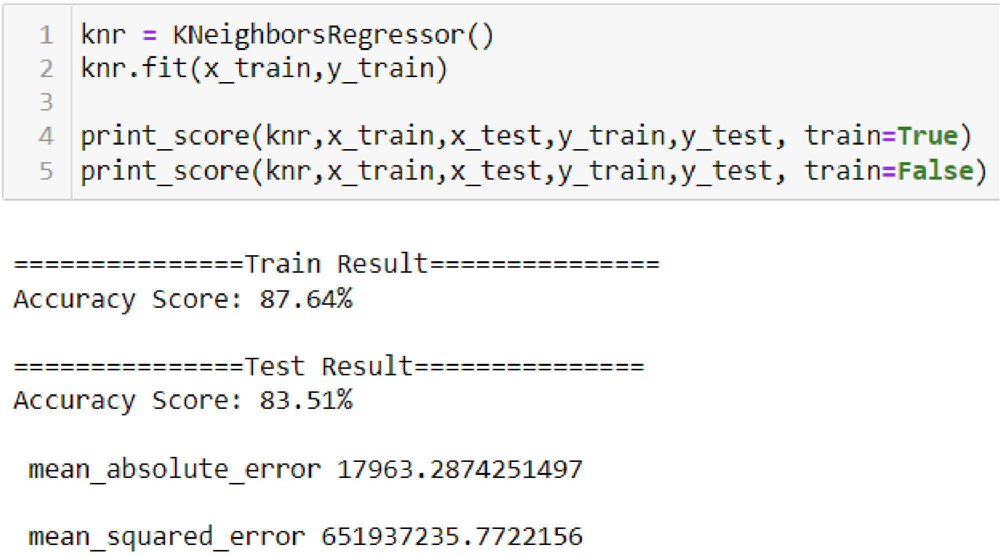
5. **RandomForestRegressor**: -



```python
rfr = RandomForestRegressor()
rfr.fit(x_train,y_train)

print_score(rfr,x_train,x_test,y_train,y_test, train=True)
print_score(rfr,x_train,x_test,y_train,y_test, train=False)
```

```
===============Train Result===============
Accuracy Score: 97.80%

===============Test Result===============
Accuracy Score: 85.60%

 mean_absolute_error 16497.058802395208

 mean_squared_error 569069508.0717748
```

6. **KNeighborsRegressor**: -

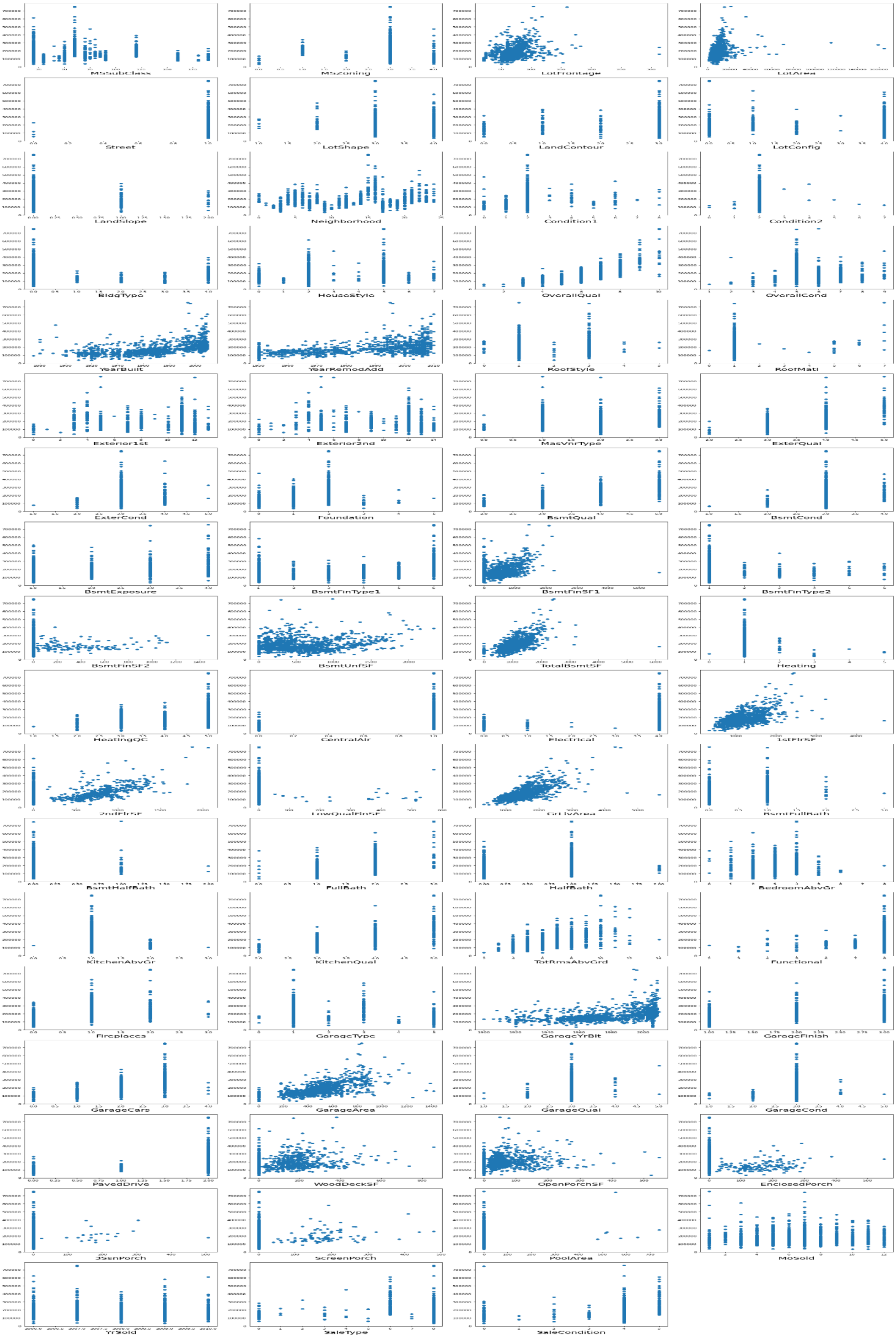


```python
knr = KNeighborsRegressor()
knr.fit(x_train,y_train)

print_score(knr,x_train,x_test,y_train,y_test, train=True)
print_score(knr,x_train,x_test,y_train,y_test, train=False)
```

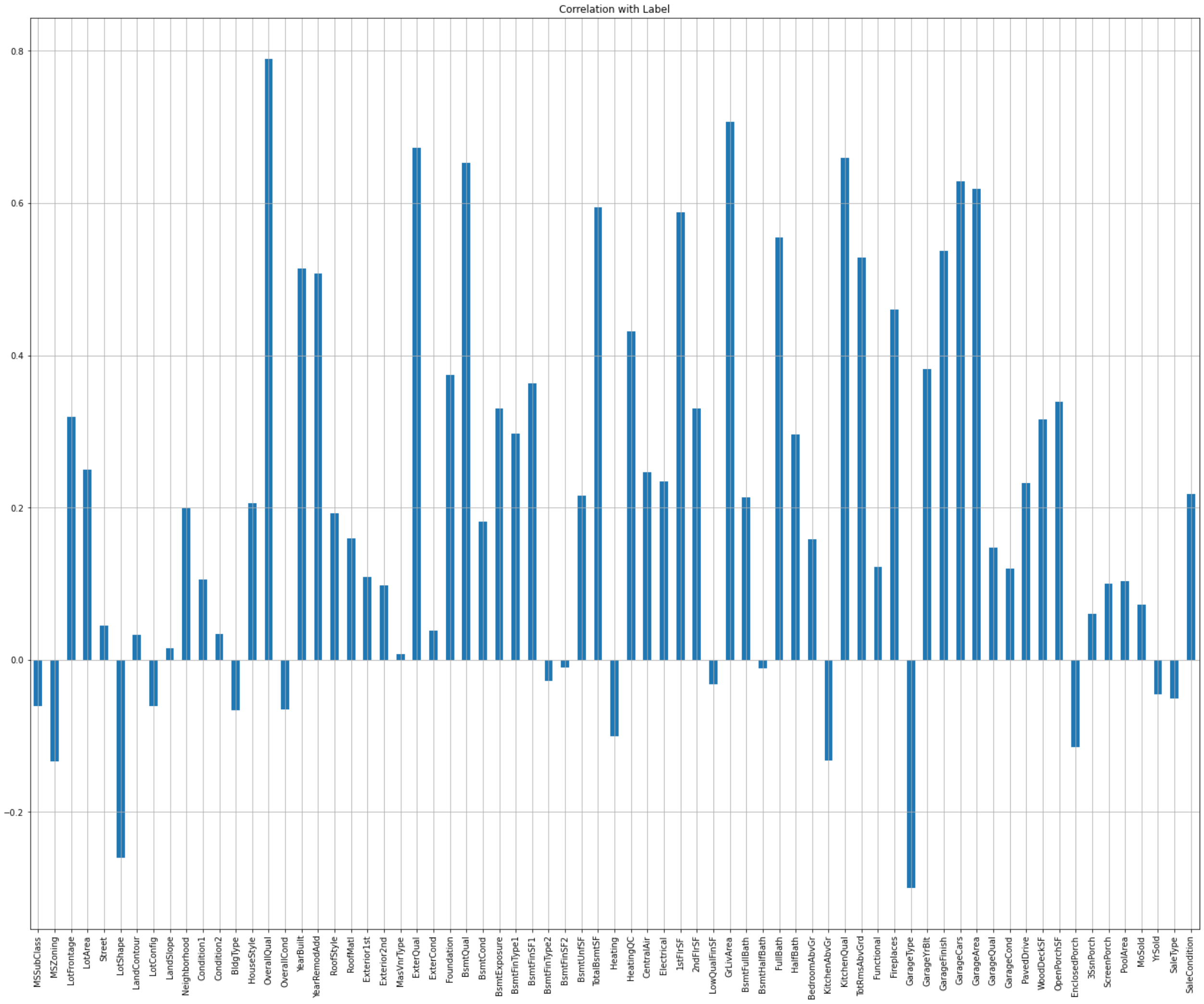
```
===============Train Result===============
Accuracy Score: 87.64%

===============Test Result===============
Accuracy Score: 83.51%

 mean_absolute_error 17963.2874251497

 mean_squared_error 651937235.7722156
```

- ## Visualizations

1. Scatter Plot: -



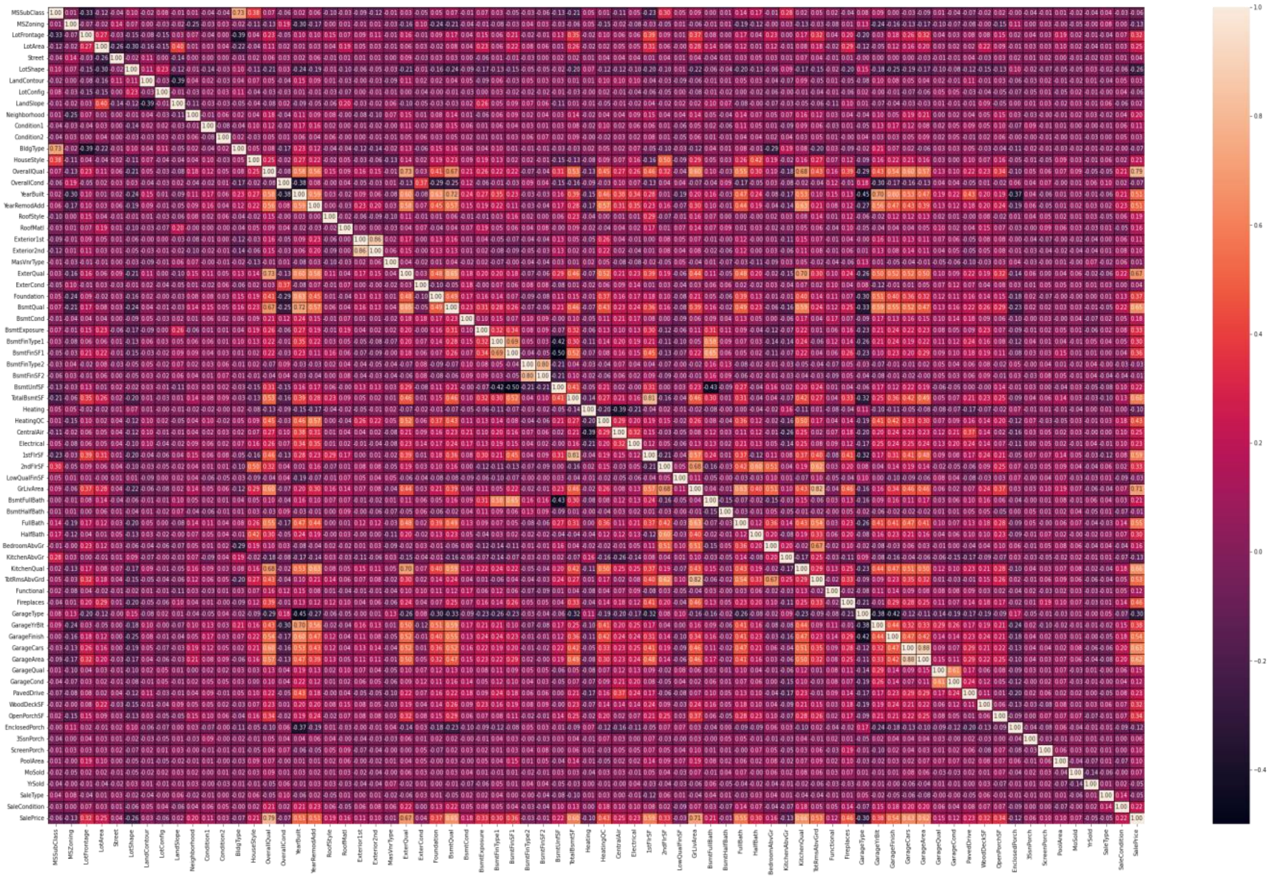
2. BarPlot: -



Observations:

- OverallQual, YearBuilt, YearRemodAdd, ExterQual, BsmtQual, TotalBsmtSF, 1stFlrSF, GrLivArea, FullBath, KitchenQual, TotRmsAbvGrd, GarageFinish, GarageCars and GarageArea columns have high co-relation with Label.
- MSSubClass, LandContour, LotConfig, LandSlope, Condition2, BldgType, OverallCond, MasVnrType, ExterCond, BsmtFinType2, BsmtFinSF2, LowQualFinSF, BsmtHalfBath, 3SsnPorch, YrSold and SaleType columns have Low/No co-relation with Label.

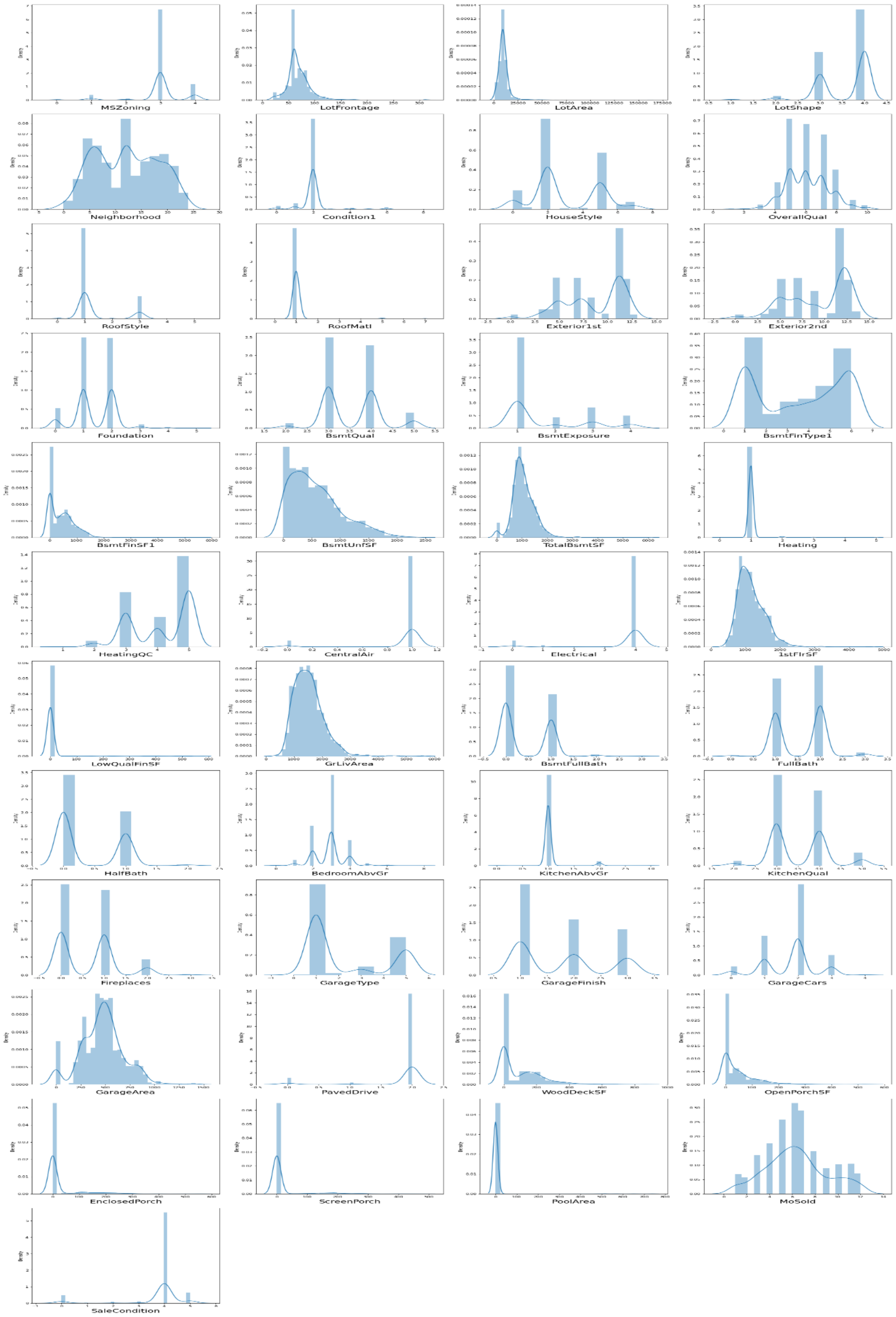
3. HeatMap: -



Observations:

- Multicollinearity problem does not exist in this database

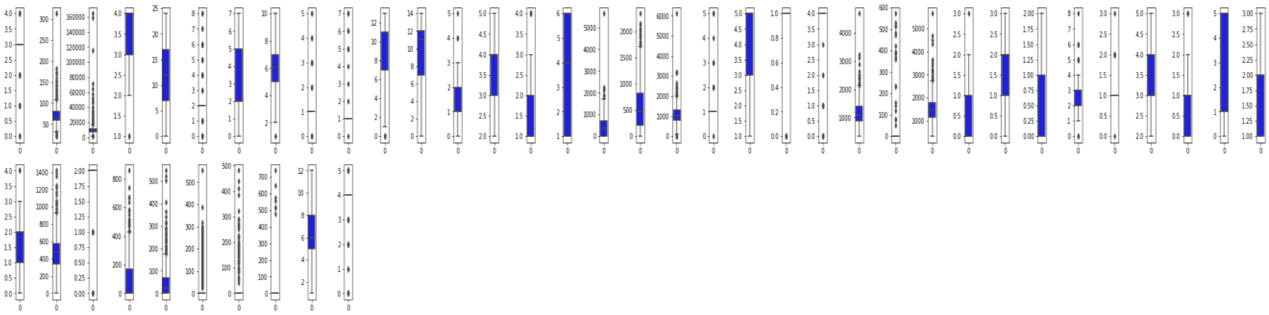
4. Distribution Plot: -



Observations:

- Not considering skewness of categorical data columns.
- LotFrontage, LotArea, BsmtFinSF1, BsmtUnfSF, TotalBsmtSF, 2ndFlrSF, LowQualFinSF, WoodDeckSF, OpenPorchSF, EnclosedPorch, ScreenPorch and PoolArea columns have skewness.

5. Box Plot: -



Observations:

- LotFrontage, LotArea, BsmtFinSF1, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, LowQualFinSF, GrLivArea, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, ScreenPorch and PoolArea columns have outliers

- ## Interpretation of the Results

Below is the list of highly influencing features or variables to predict the sales price of the house.

1. OverallQual = It is very important that the house should be built with good quality materials and this column rates the overall material and finish of the house.

2. YearBuilt = It is important that the construction should be new as it increases to life of the house and this column shows the original year of construction.

3. YearRemodAdd = It is important that if the construction is old then it should be remodel to increase its lifespan and this column shows the remodel date.

4. ExterQual = It is important to use good quality material for exterior of the house and this column evaluates the quality of the material on the exterior.

5. BsmtQual = The price of the house is also depending upon the height of the basement and this column shows the height of the basement.

6. TotalBsmtSF = The price of the house is also depending upon total square feet of basement area and this column shows the basement area.

7. 1stFlrSF = If the house has 1 floor, then the price of the house increases and this column shows the first-Floor square feet.

8. GrLivArea = More the above grade (ground) living area square feet more will be the price of the house and GrLivArea contains the information of living area square feet.

9. FullBath = Number of full bathrooms affects the price of the house and this column shows the number of full bathrooms.

10. KitchenQual = Quality of kitchen is very important. A top-rated kitchen quality can increase the price of the house and this column shows the Quality of kitchen.

11. TotRmsAbvGrd = high number of rooms will lead to high prices and this column gives the count of total rooms.

12. GarageFinish = People want their garages to be fully finished and this result in high price of the house. This column shows the condition of the garage.

13. GarageCars = It is important to know how much cars a house owner can park in his garage and this column gives that count.

14. GarageArea = Large area of garage results in increasing of house price and this column provides information about garage area in feet.

CONCLUSION

- **Key Findings and Conclusions of the Study**
    1. Selecting LinearRegression model since the Accuracy score i.e., 88.87% and test scores i.e., 88.76 % are greater and close to each other.
    2. mean_absolute_error is also low for LinearRegression model.
    3. LinearRegression model is not overfitted since r2_score of LassoCV and RidgeCV is same and close to test score i.e., 88.76%.

- **Learning Outcomes of the Study in respect of Data Science**
    1. Data Cleaning helps to convert unorganized and unstructured data into structured data which will be used to make findings.
    2. Data visualization helps understand and analyse the data.
    3. Model building helps to predict outcomes, in this case LinearRegression model fits perfect for this dataset.
    4. While doing pre-processing the high VIF problem arises as many highly co-related features have high VIF score.

- **Limitations of this work and Scope for Future Work**
    1. There are 80 features present in the dataset however due to pre-processing and visualization we have cutdown some features, hence this could become a disadvantage in the future as we update the dataset and there is a possibility that we may lose some important information.
    2. It is necessary to keep an eye on new and updated data to further train the model and make decision as per new data.