FLIP ROBO

HOUSING: PRICE PREDICTION

Submitted by:

C.S.Manjunath Reddy

ACKNOWLEDGMENT

I would like to thank FlipRobo for giving me this opportunity. The DataTrained institute classes helped me to solve this problem.

The language used for this project is Python with Pandas, NumPy. I referred to the websites seaborn.pydata, matplotlib.org for visualization purpose, stackoverflow.com to solve the doubts, and scikit-learn for Machine Learning algorithms.

INTRODUCTION

- ## Business Problem Framing

  The company “Surprise Housing” has decided to enter the Australian Housing market and real estate 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. This data trained with various machine learning models to predict the prices for the given test data. The target variable in the training data is continuous. So, we used a regression model to predict the prices.

- ## Conceptual Background of the Domain Problem

  The project is on the housing and real estate market. We used data science techniques and Machine learning techniques to predict the price of the house.

- ## Review of Literature

  I analysed the given data and checked which attributes are performing well to predict the sale price. I dropped some columns which are not useful to train the model and filled null values in the given trained data. Removed skewness and outliers from the data, did an encoding process for categorical data to train the model.

- Motivation for the Problem Undertaken

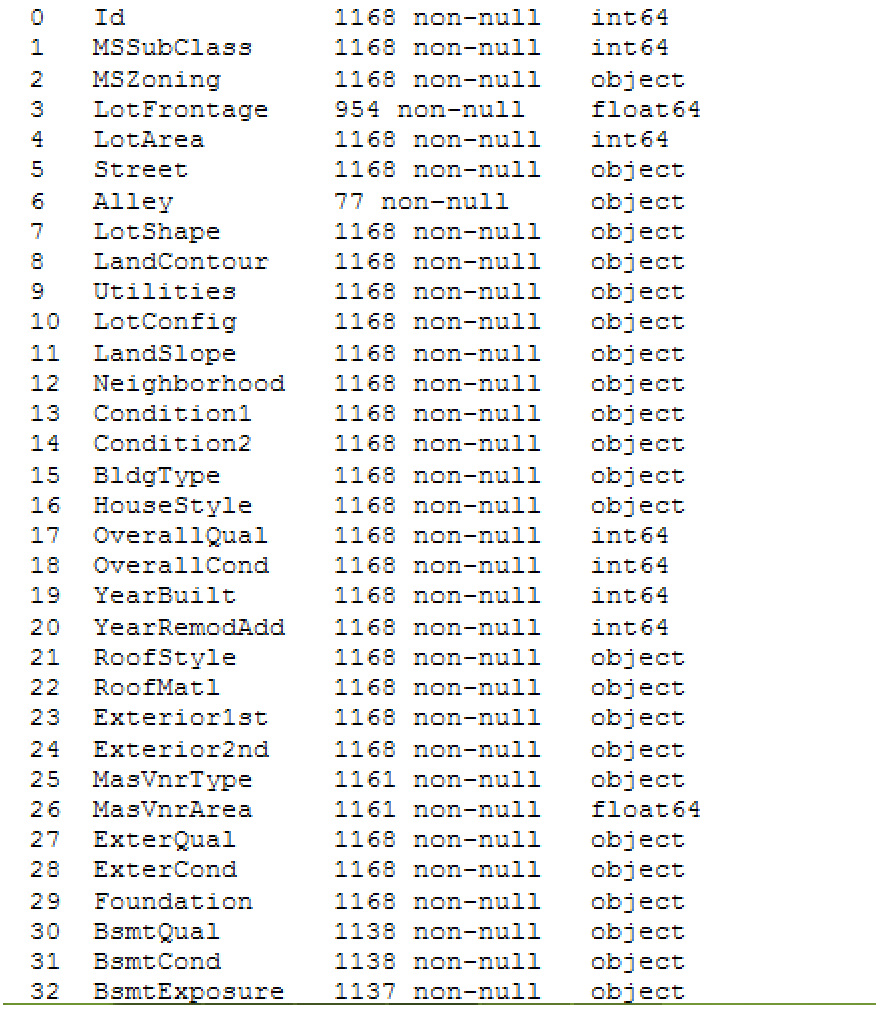
The task is to train the given trained data set including sale price and in future apply the independent variables to predict the sale price. Since the target variable is continuous I used regression model.

Analytical Problem Framing

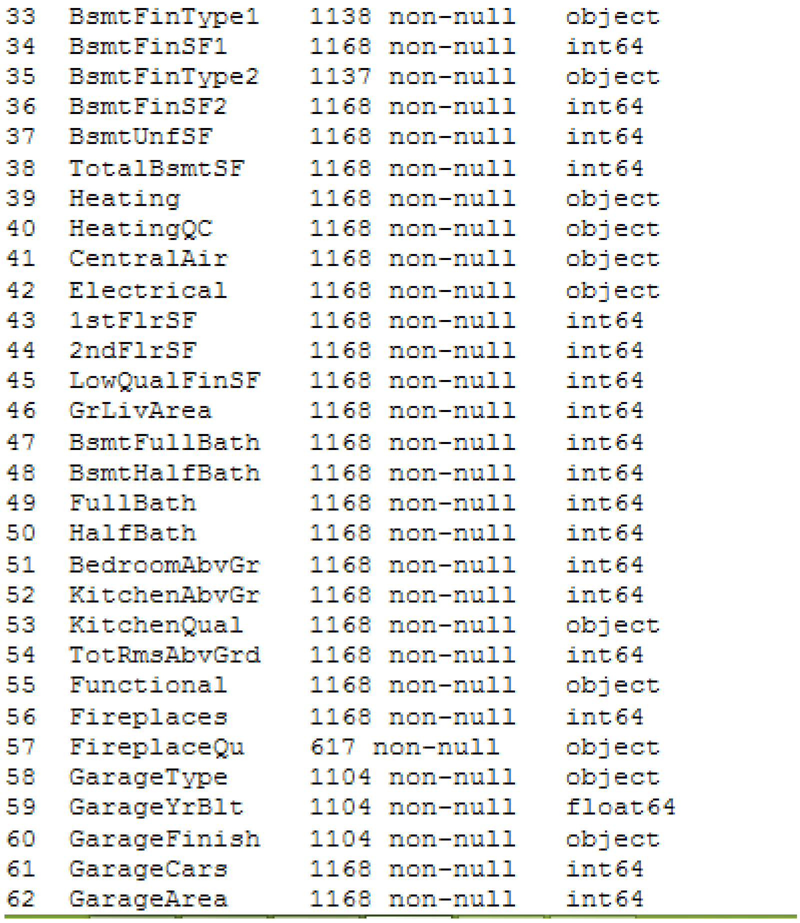
- Mathematical/ Analytical Modeling of the Problem

With the help of “pandas.describe()” function we got information of descriptive statistics (mean, median, min value, max value, percentile). With the help of log and square root, transformation removed skewness and with IQR removed outliers.

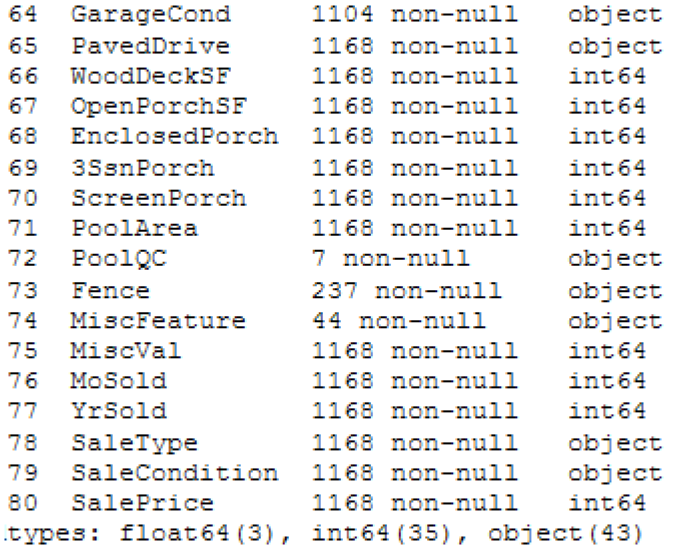
- Data Sources and their formats



```
0   Id             1168 non-null   int64
1   MSSubClass     1168 non-null   int64
2   MSZoning       1168 non-null   object
3   LotFrontage    954 non-null    float64
4   LotArea        1168 non-null   int64
5   Street         1168 non-null   object
6   Alley          77 non-null     object
7   LotShape       1168 non-null   object
8   LandContour    1168 non-null   object
9   Utilities      1168 non-null   object
10  LotConfig      1168 non-null   object
11  LandSlope      1168 non-null   object
12  Neighborhood   1168 non-null   object
13  Condition1     1168 non-null   object
14  Condition2     1168 non-null   object
15  BldgType       1168 non-null   object
16  HouseStyle     1168 non-null   object
17  OverallQual    1168 non-null   int64
18  OverallCond    1168 non-null   int64
19  YearBuilt      1168 non-null   int64
20  YearRemodAdd   1168 non-null   int64
21  RoofStyle      1168 non-null   object
22  RoofMatl       1168 non-null   object
23  Exterior1st    1168 non-null   object
24  Exterior2nd    1168 non-null   object
25  MasVnrType     1161 non-null   object
26  MasVnrArea     1161 non-null   float64
27  ExterQual      1168 non-null   object
28  ExterCond      1168 non-null   object
29  Foundation     1168 non-null   object
30  BsmtQual       1138 non-null   object
31  BsmtCond       1138 non-null   object
32  BsmtExposure   1137 non-null   object
```



```
33   BsmtFinType1    1138 non-null    object
34   BsmtFinSF1      1168 non-null    int64
35   BsmtFinType2    1137 non-null    object
36   BsmtFinSF2      1168 non-null    int64
37   BsmtUnfSF       1168 non-null    int64
38   TotalBsmtSF     1168 non-null    int64
39   Heating         1168 non-null    object
40   HeatingQC       1168 non-null    object
41   CentralAir      1168 non-null    object
42   Electrical      1168 non-null    object
43   1stFlrSF        1168 non-null    int64
44   2ndFlrSF        1168 non-null    int64
45   LowQualFinSF    1168 non-null    int64
46   GrLivArea       1168 non-null    int64
47   BsmtFullBath    1168 non-null    int64
48   BsmtHalfBath    1168 non-null    int64
49   FullBath        1168 non-null    int64
50   HalfBath        1168 non-null    int64
51   BedroomAbvGr    1168 non-null    int64
52   KitchenAbvGr    1168 non-null    int64
53   KitchenQual     1168 non-null    object
54   TotRmsAbvGrd    1168 non-null    int64
55   Functional      1168 non-null    object
56   Fireplaces      1168 non-null    int64
57   FireplaceQu     617 non-null     object
58   GarageType      1104 non-null    object
59   GarageYrBlt     1104 non-null    float64
60   GarageFinish    1104 non-null    object
61   GarageCars      1168 non-null    int64
62   GarageArea      1168 non-null    int64
```



```
64   GarageCond      1104 non-null    object
65   PavedDrive      1168 non-null    object
66   WoodDeckSF      1168 non-null    int64
67   OpenPorchSF     1168 non-null    int64
68   EnclosedPorch   1168 non-null    int64
69   3SsnPorch       1168 non-null    int64
70   ScreenPorch     1168 non-null    int64
71   PoolArea        1168 non-null    int64
72   PoolQC          7 non-null       object
73   Fence           237 non-null     object
74   MiscFeature     44 non-null      object
75   MiscVal         1168 non-null    int64
76   MoSold          1168 non-null    int64
77   YrSold          1168 non-null    int64
78   SaleType        1168 non-null    object
79   SaleCondition   1168 non-null    object
80   SalePrice       1168 non-null    int64
.types: float64(3), int64(35), object(43)
```

With the above screenshots, we can see that there are 3 columns with the float data type, 35 columns with an integer data type, and 43 columns with the object data type. Also, we can see that there are null values in the given data.

- Data Preprocessing Done

The data set contains null values and handled those null values with sicikt-learn imputation pre-processing. Dropped columns 'Id'(unique number),in columns where the data set contains nearly 80% null values 'PoolQC','Fence','MiscFeature','Alley', and the column 'Utilities' because of only one value in all the rows.

- Data Inputs- Logic- Output Relationships

Mainly the material used for the total constructions(exterior, foundation,interior), Garage (quality, basement height)and the age(year built) of the house are the main things decides the house sale price.

- Hardware and Software Requirements and Tools Used

The tools/libraries used for the project are Pandas, NumPy,Matplotlib,Seaborn and Sickit-learn.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  I used seaborn to analyze the categorical and numerical data to check skewness and outliers. With the function df.describe() we can check the descriptive statistics of all numerical columns.

- Testing of Identified Approaches (Algorithms)

  The algorithms used for the project:

  *Linear Regression

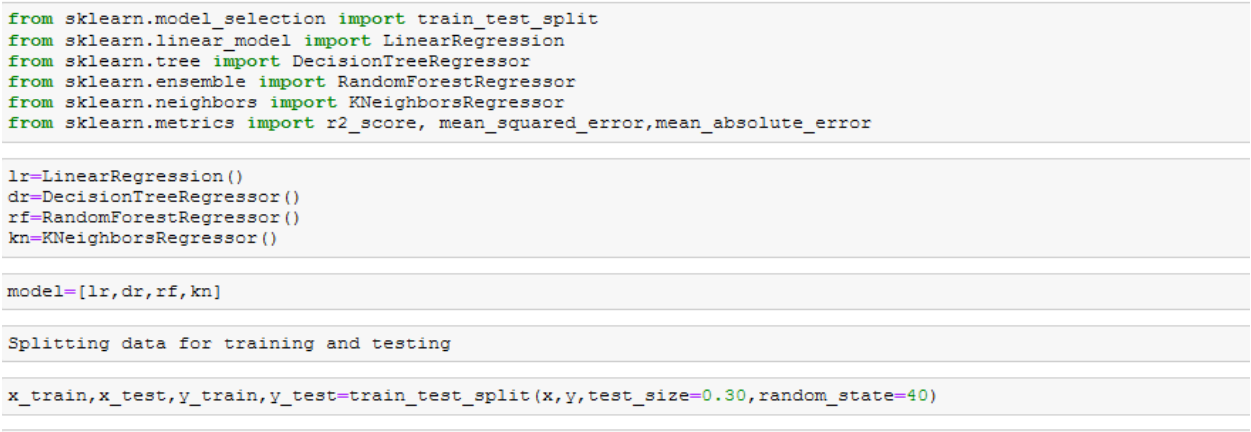
  *DecisionTree Regressor

  *RandomForestRegressor

  *KnearestneighbrosRegressor

- Run and Evaluate selected models

  I used “for loop” to train the data with all 4 algorithms at one go. Here are the screen shots of the code and result:



```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import r2_score, mean_squared_error,mean_absolute_error
```

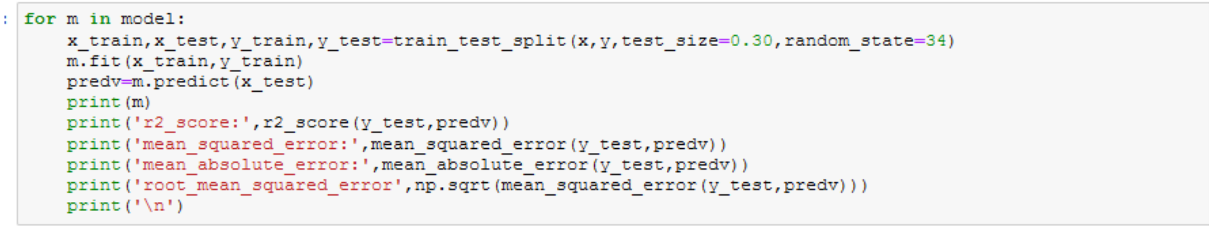
```
lr=LinearRegression()
dr=DecisionTreeRegressor()
rf=RandomForestRegressor()
kn=KNeighborsRegressor()
```

```
model=[lr,dr,rf,kn]
```

```
Splitting data for training and testing
```

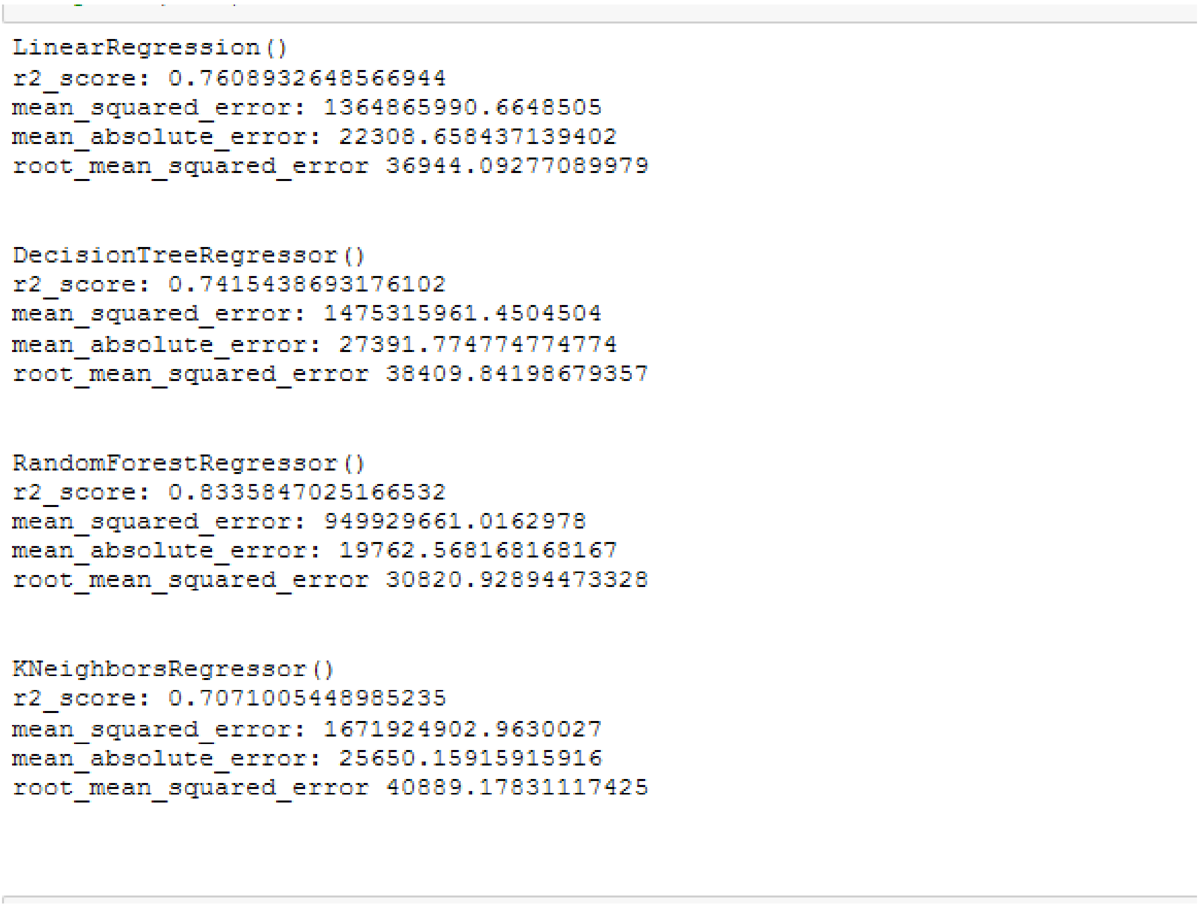
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=40)
```

For loop used for all 4 algorithms to train the data at one go.



```
for m in model:
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=34)
    m.fit(x_train,y_train)
    predv=m.predict(x_test)
    print(m)
    print('r2_score:',r2_score(y_test,predv))
    print('mean_squared_error:',mean_squared_error(y_test,predv))
    print('mean_absolute_error:',mean_absolute_error(y_test,predv))
    print('root_mean_squared_error',np.sqrt(mean_squared_error(y_test,predv)))
    print('\n')
```

Result of the all 4 algorithms.



```
LinearRegression()
r2_score: 0.7608932648566944
mean_squared_error: 1364865990.6648505
mean_absolute_error: 22308.658437139402
root_mean_squared_error 36944.09277089979


DecisionTreeRegressor()
r2_score: 0.7415438693176102
mean_squared_error: 1475315961.4504504
mean_absolute_error: 27391.774774774774
root_mean_squared_error 38409.84198679357


RandomForestRegressor()
r2_score: 0.8335847025166532
mean_squared_error: 949929661.0162978
mean_absolute_error: 19762.568168168167
root_mean_squared_error 30820.92894473328


KNeighborsRegressor()
r2_score: 0.7071005448985235
mean_squared_error: 1671924902.9630027
mean_absolute_error: 25650.15915915916
root_mean_squared_error 40889.17831117425
```

- Key Metrics for success in solving problem under consideration

  Since this is regression model RMSE (Root mean squared error) value is considered to check how good the model is performing. Less RMSE model is good. From the above mentioned algorithms result we can see that RandomForestRegressor is working well.

- Visualizations

*For the categorical data I used bar plot to check from each column which value is making the house price high.

*For the Numerical data I used correlation matrix and plot them with the help of heatmap to check which column is correlation with the sale price (target variable)

*Main observations found through visualization: Mainly the material used for the total constructions, Garage (quality, basement height) and the age(year built) of the house are the main things decides the house sale price.

- Interpretation of the Results

  With the help of visualization we can see sales price depends on which column mainly. From pre-processing we filled null values. So there are no null values before we train the data. Encoding done for categorical

CONCLUSION

Key Findings and Conclusions of the Study

**Key Findings: Checked which column have null values,skewness.

**Observations: Sales price depends on mainly which columns.

- ## Learning Outcomes of the Study in respect of Data Science

Visualization is key thing to check which independent variable is correlated with target variable. While doing this problem I came to know that the data is not cleaned, there are null values, skewness and outliers in the data. I did pre-processing to remove null values, skewness and outliers. I used 4 algorithms to train the data. From this RandomForestRegressor is working good I did hyper parameter tuning to get better accuracy so it raised from 83 – 84% r2 score with reduced RMSE.

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

If there is skewness in the given data first we have to apply log then square root transformation taking the threshold value as +/-0.5, if these both transformations are not working good, then the distribution of the data points in that column is not uniform. For such cases we have to build model keeping skewed data.