FLIP ROBO

HOUSING PRICE PREDICTION

Submitted by:

SHREYASI SAHA

(INTERNSHIP 32)

ACKNOWLEDGMENT

I would like to thank the Khusboo Garg my SME in Internship from Flip-Robo Technology for providing me with Train and Test Dataset ,

Sample Documentation and Data Description without which working in this project would have been very challenging .

I would also like to mention I received help and resource from various websites like Scikit Learn , Datatrained , Towards DataScience and Medium etc .

INTRODUCTION

Problem Statement:

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.

- Conceptual Background of the Domain Problem

  Observation Based on the above data, we can drop the following columns - LotFrontage - Alley - FireplaceQu - PoolQC - Fence - MiscFeature - Id (dropping this not because of count, irrelevant) - MoSold (dropping this not because of count, irrelevant) - Street (dropping this not because of count, irrelevant) - Utilities (dropping this not because of count, irrelevant)

- Review of Literature

The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

Which variables are important to predict the price of variable?

How do these variables describe the price of the house?

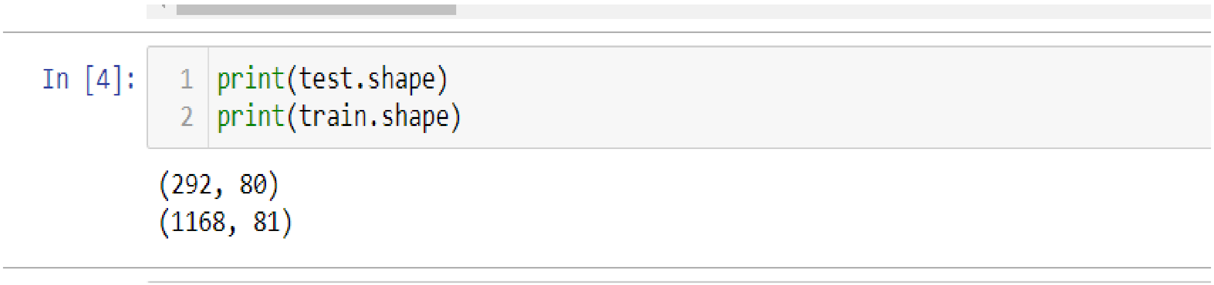
- ## Motivation for the Problem Undertaken

  It is 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 Modeling of the Problem

Data contains 1460 entries each having 81 variables.
Data contains Null values. We need to treat them using the domain knowledge and our own understanding.
Extensive EDA has to be performed to gain relationships of important variable and price.
Data contains numerical as well as categorical variable. We need to handle them accordingly.
We have to build Machine Learning models, apply regularization and determine the optimal values of Hyper Parameters.
We need to find important features which affect the price positively or negatively.
Two datasets are being provided to you (test.csv, train.csv). We will train on train.csv dataset and predict on test.csv file.



In [4]:

```
print(test.shape)
print(train.shape)
```

```
(292, 80)
(1168, 81)
```

Data Sources and their formats

A US-based housing company named **Surprise Housing** has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file .

```
: [  1 train

        Id  MSSubClass MSZoning Lotfrontage LotArea Street  Alley LotShape LandContour  Utilities LotConfig LandSlopeeighborhood Condition1 C(

   0  127        120        RL         NaN    4928  Pave  NaN       IR1         Lvl  AllPub     Inside       GU      NPk\111    Norm
   1  889         20        RL        95.0   15865  Pave  NaN       IR1         Lvl  AllPub     Inside      Mod        NAmes    Norm
   2  793         60        RL        92.0    9920  Pave  NaN       IR1         Lvl  AllPub    CulDSac       Gli     NoRidge    Norm
   3  110         20        RL       105.0   11751  Pave  NaN       IR1         Lvl  AllPub     Inside       Gli      NWAmes    Norm
   4  422         20        RL         NaN   16635  Pave  NaN       IR1         Lvl  AllPub        FR2       GU      NWAmes    Norm

1163  289         20        RL         NaN    9819  Pave  NaN       IR1         Lvl  AllPub     Inside       Gli      Sawyer    Norm
1164  554         20        RL        67.0    8777  Pave  NaN       Reg         Lvl  AllPub     Inside       Gli     Edwards    Feedr
1165  196        160        RL        24.0    2280  Pave  NaN       Reg         Lvl  AllPub        FR2       GU      NPk\111    Norm
1166   31         70     C (all)      50.0    8500  Pave Pave       Reg         Lvl  AllPub     Inside       GU      IDOTRR     Feedr
1167  617         60        RL         NaN    7861  Pave  NaN       IR1         Lvl  AllPub     Inside       Gli     Gilbert     Norm

1168rowsx 81 columns
```

```
n [5]:  1 test
ut[s]:
          Id MSSubClass  MSZoning LotFrontage  LotArea  Street  Alley  LotShape  LandContour  Utilities  LotConfig  LandSlope  Neighborhood  Condition1  C1

         337        20        R        86.0     14157    Pave   NaN       IR1          HLS     AllPub     Corner        Gtl       StoneBr        Norm
        1018       120       RL         NaN      5814    Pave   NaN       IR1          Lvl     AllPub    CulDSac        Gtl       StoneBr        Norm
         929        20       RL         NaN     11838    Pave   NaN       Reg          Lvl     AllPub     Inside        Gtl       CollgCr        Norm
        1148        70        R        75.0     12000    Pave   NaN       Reg          Bnk     AllPub     Inside        Gtl       Crawfor        Norm
     4  1227        60       RL        86.0     14598    Pave   NaN       IR1          Lvl     AllPub    CulDSac        Gtl       Somerst        Feedr

   287    83        20        R        78.0     10206    Pave   NaN       Reg          Lvl     AllPub     Inside        Gtl       Somerst        Norm
   288  1048        20       RL         570      9245    Pave   NaN       IR2          Lvl     AllPub     Inside        Gtl       CollgCr        Norm
   289    17        20       RL         NaN     11241    Pave   NaN       IR1          Lvl     AllPub    CulDSac        Gtl         NAmes        Norm
   290   523        50       RM        50.0      5000    Pave   NaN       Reg          Lvl     AllPub     Corner        Gtl       BrkSide        Feedr
   291  1379       160       RM        21.0      1953    Pave   NaN       Reg          Lvl     AllPub     Inside        Gtl        BrDale        Norm

292 rowsx 80 columns
```

Data Pre-Processing

Storing null values in train , then printing columns with more than 0 null values

Impute missing values

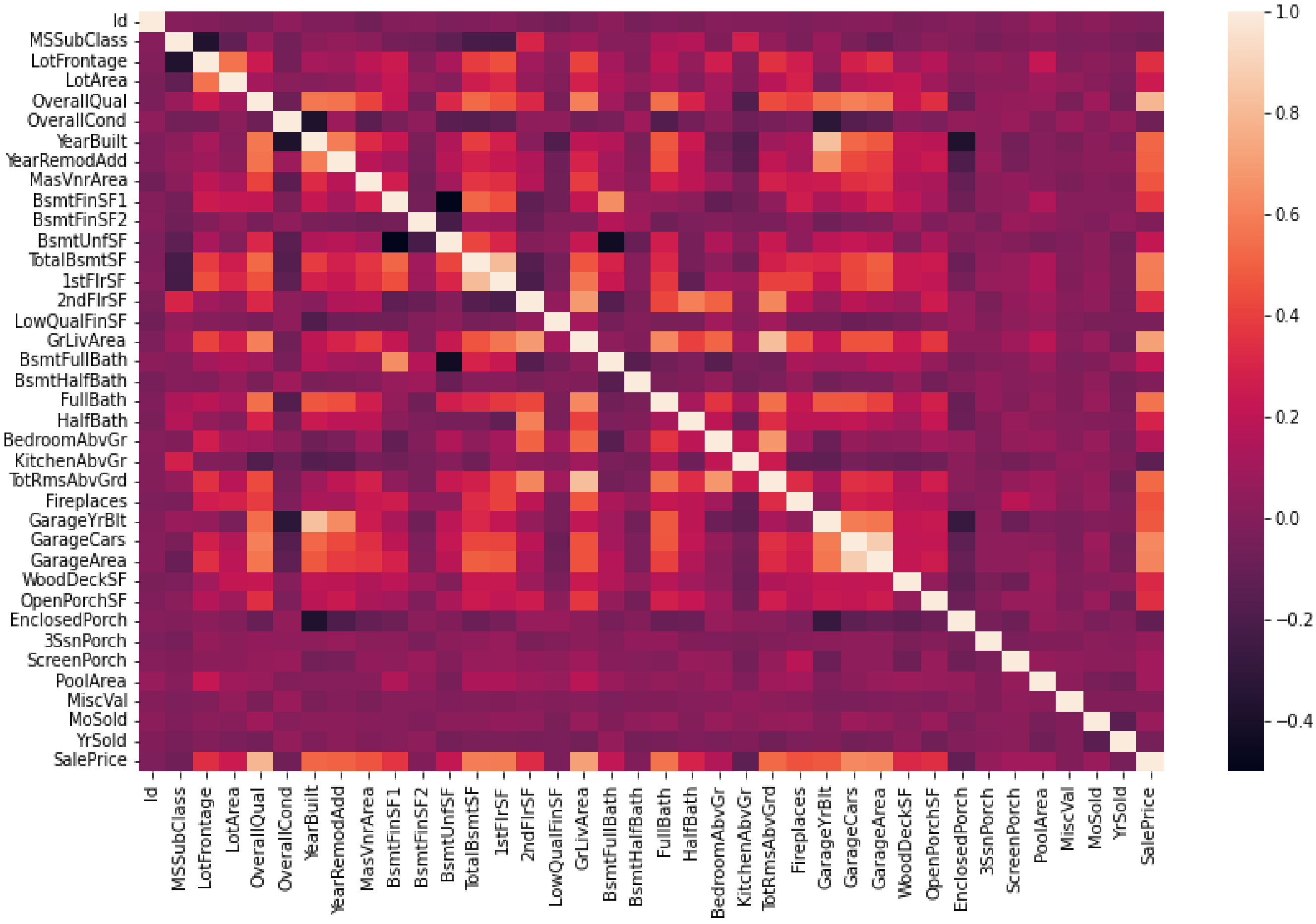
Dropping columns which has around 50 percent missing values

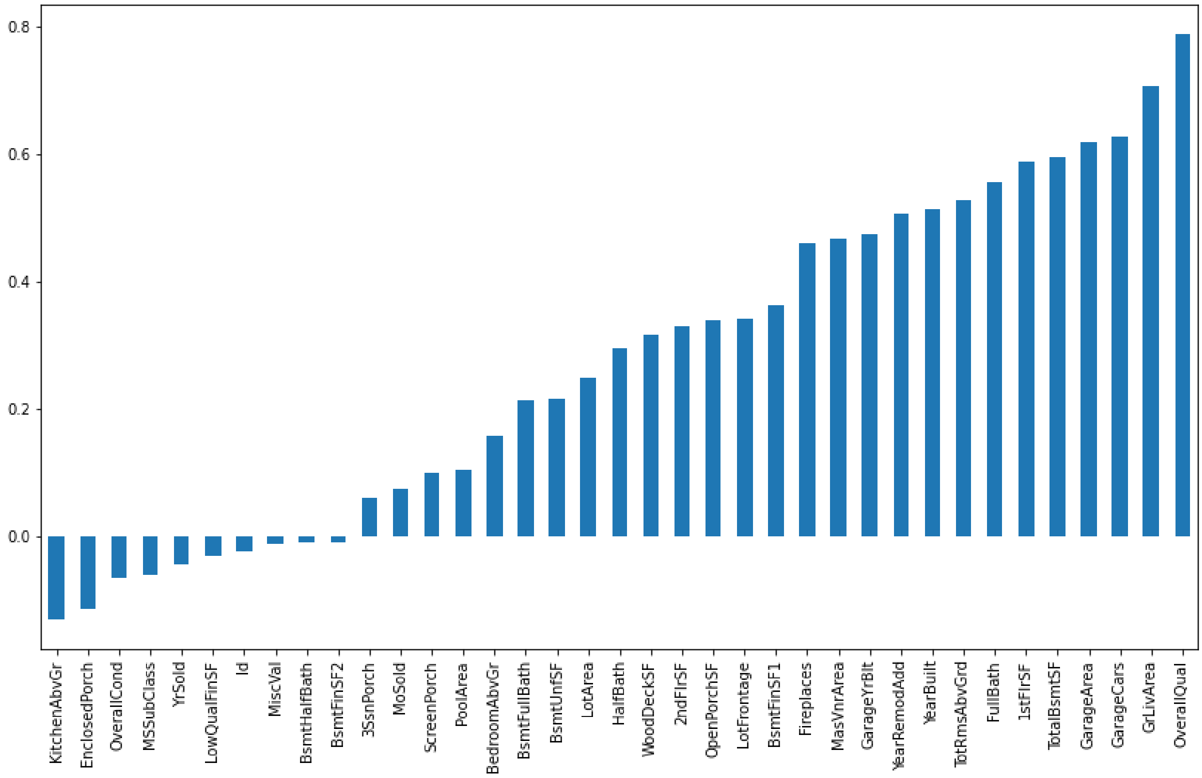
Selecting categorical features and level encoding

Label encoding of input features

Scaling input and test data using StandardScaler module.

Data Inputs- Logic- Output Relationships





- Hardware and Software Requirements and Tools Used

The General Hardware used for this project is :-

8 GB RAM

512GB SSD

Intel i5 processor

The Software and tools used for this project is :-

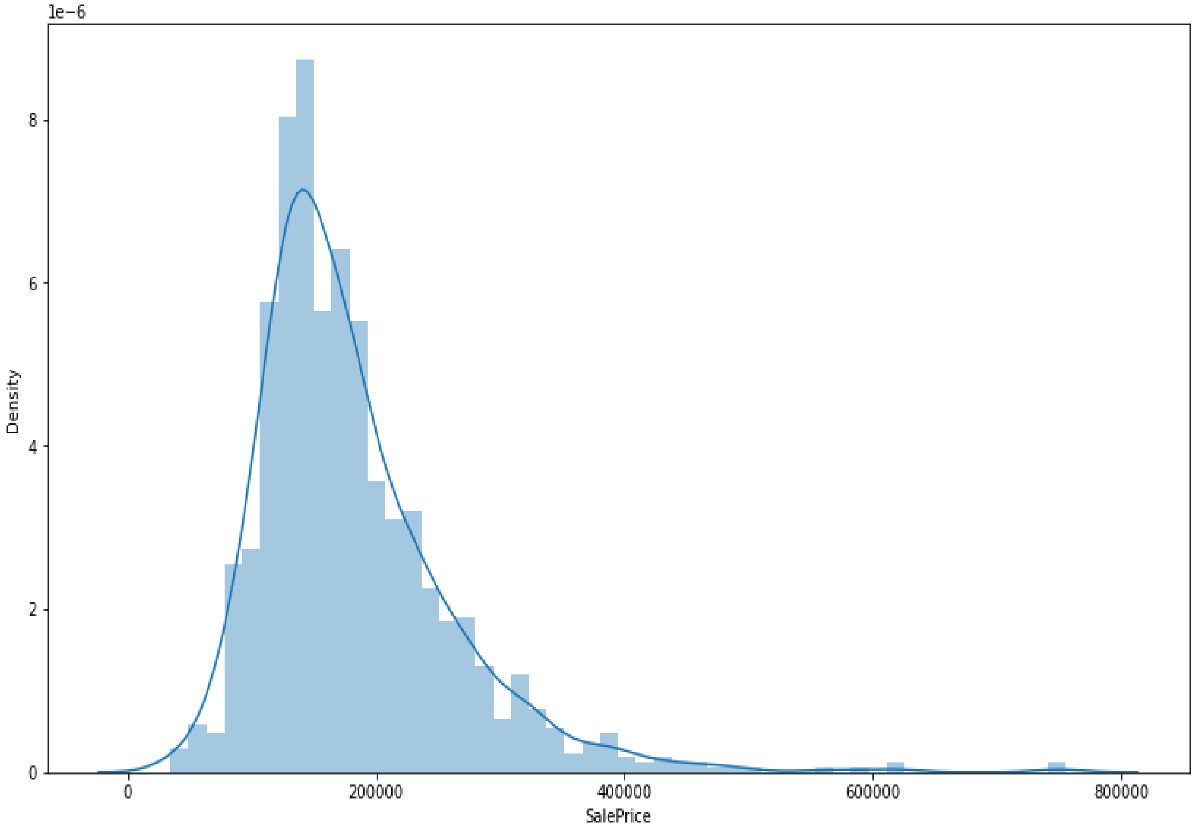
Python (Jupyter Notebook)

Scikit Learn

Various tools :- Pandas , Matplotlib , NumPy , Seaborn etc

Model/s Development and Evaluation

Following is the distribution plot of MarketValue which is our Target.



The Different Models which are used are :-

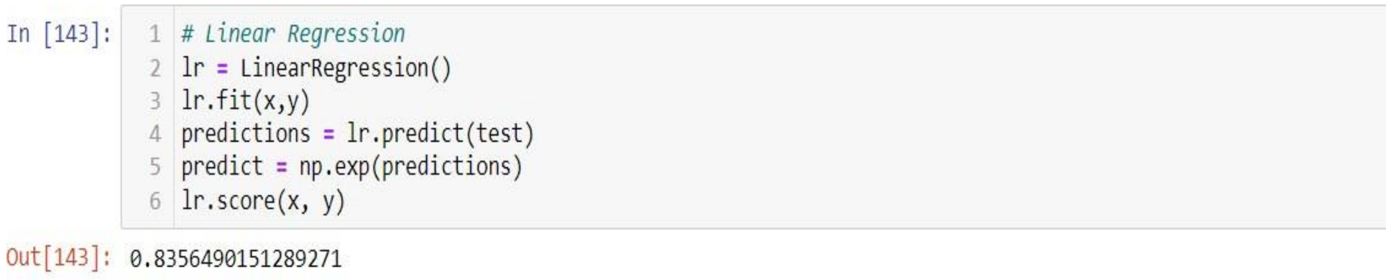
1. Linear Regression
2. Ridge Regression
3. XGB Regressor
4. Random forest model
5. Decision Tree Regressor
6. Ada AdaBoost Regressor

CONCLUSION

We used various different types of models and used RSME score to determine which model is best.

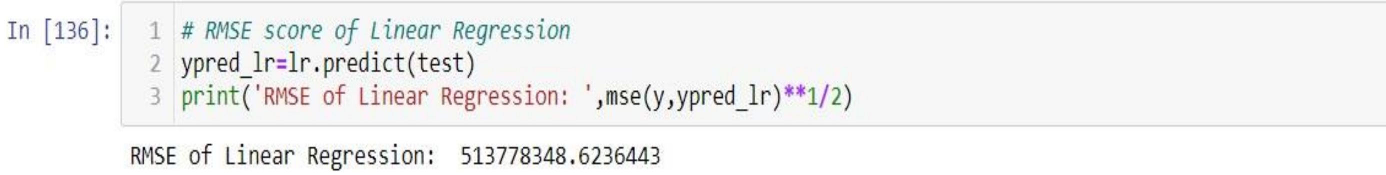
Root mean squared error (RMSE) is the square root of the mean of the square of all of the error.

1) Linear Regression



```
In [143]:
# Linear Regression
lr = LinearRegression()
lr.fit(x,y)
predictions = lr.predict(test)
predict = np.exp(predictions)
lr.score(x, y)
```

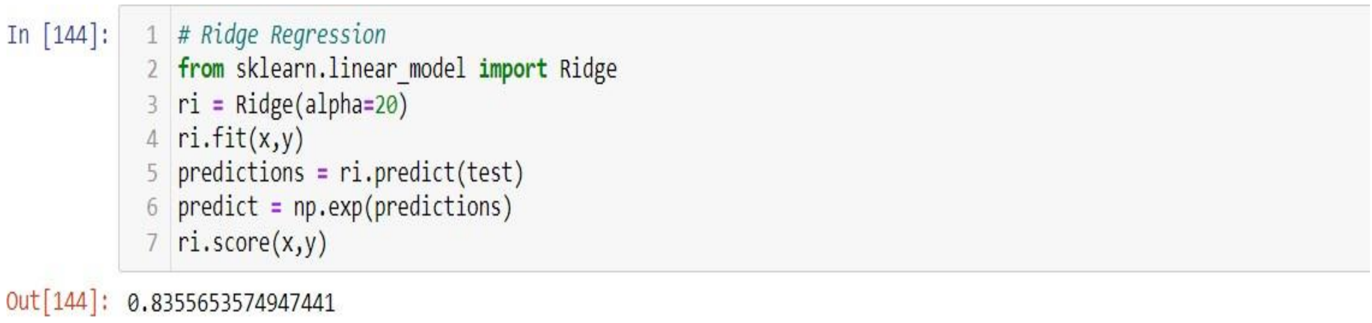
```
Out[143]: 0.8356490151289271
```



```
In [136]:
# RMSE score of Linear Regression
ypred_lr=lr.predict(test)
print('RMSE of Linear Regression: ',mse(y,ypred_lr)**1/2)
```

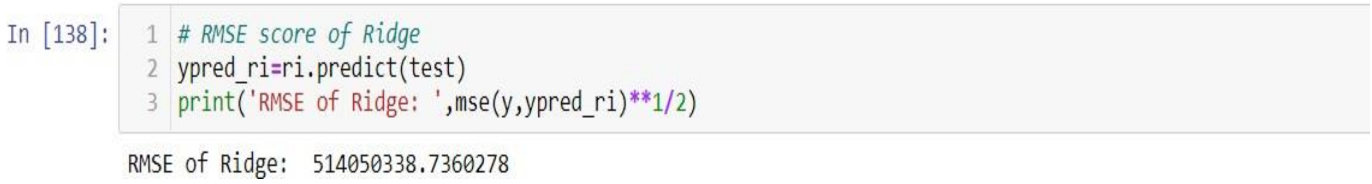
```
RMSE of Linear Regression:  513778348.6236443
```

2) Ridge Regression



```
In [144]:
# Ridge Regression
from sklearn.linear_model import Ridge
ri = Ridge(alpha=20)
ri.fit(x,y)
predictions = ri.predict(test)
predict = np.exp(predictions)
ri.score(x,y)
```

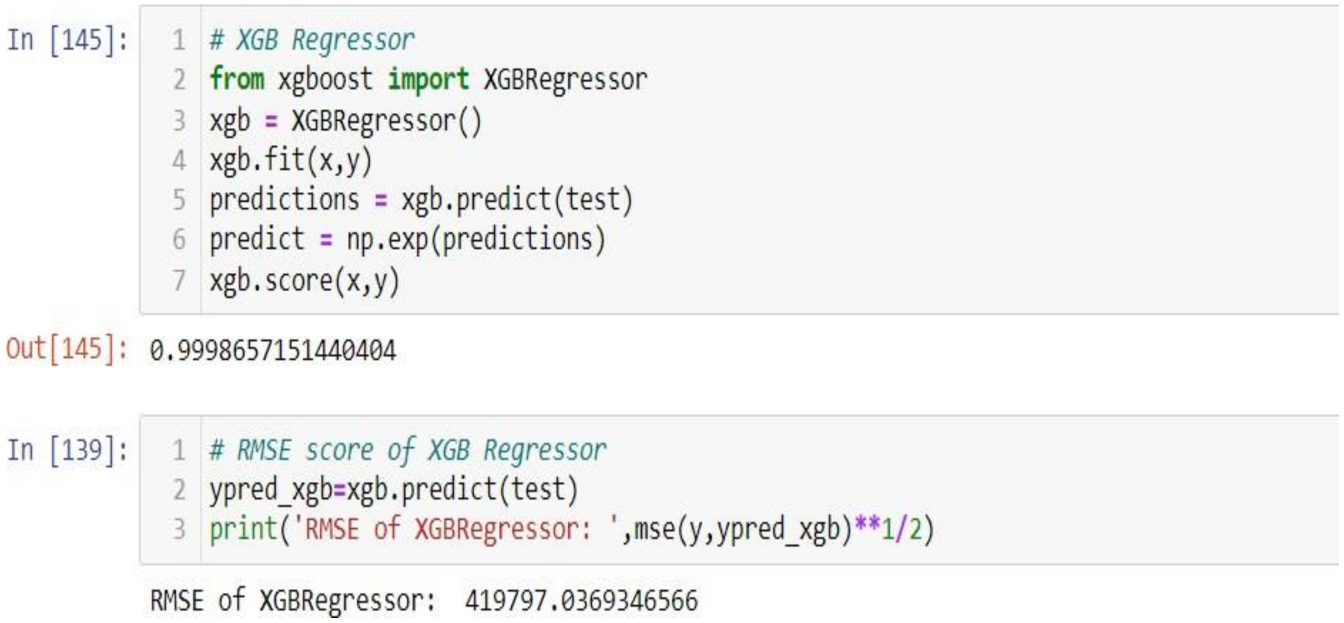
```
Out[144]: 0.8355653574947441
```



```
In [138]:
# RMSE score of Ridge
ypred_ri=ri.predict(test)
print('RMSE of Ridge: ',mse(y,ypred_ri)**1/2)
```

```
RMSE of Ridge:  514050338.7360278
```

3) XGB Regressor



In [145]:

```
# XGB Regressor
from xgboost import XGBRegressor
xgb = XGBRegressor()
xgb.fit(x,y)
predictions = xgb.predict(test)
predict = np.exp(predictions)
xgb.score(x,y)
```

Out[145]:

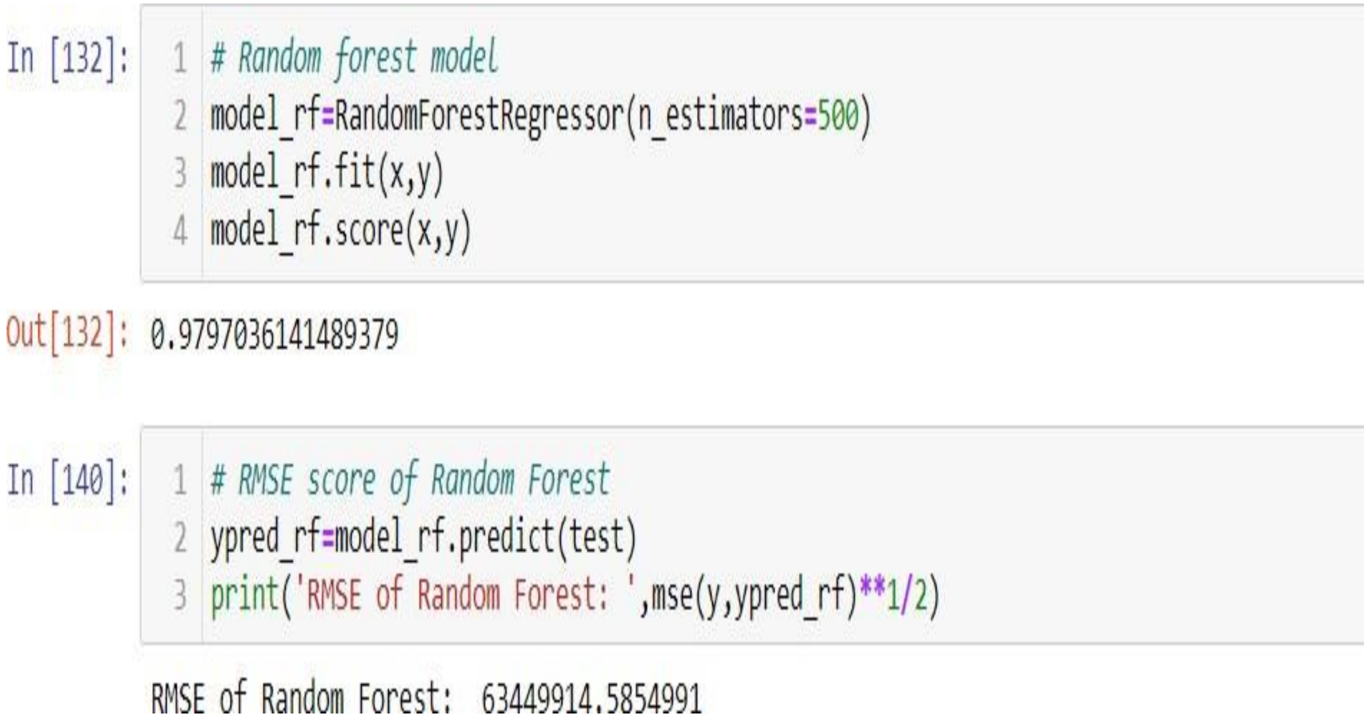
```
0.9998657151440404
```

In [139]:

```
# RMSE score of XGB Regressor
ypred_xgb=xgb.predict(test)
print('RMSE of XGBRegressor: ',mse(y,ypred_xgb)**1/2)
```

```
RMSE of XGBRegressor:  419797.0369346566
```

4) Random forest model



In [132]:

```
# Random forest model
model_rf=RandomForestRegressor(n_estimators=500)
model_rf.fit(x,y)
model_rf.score(x,y)
```

Out[132]:

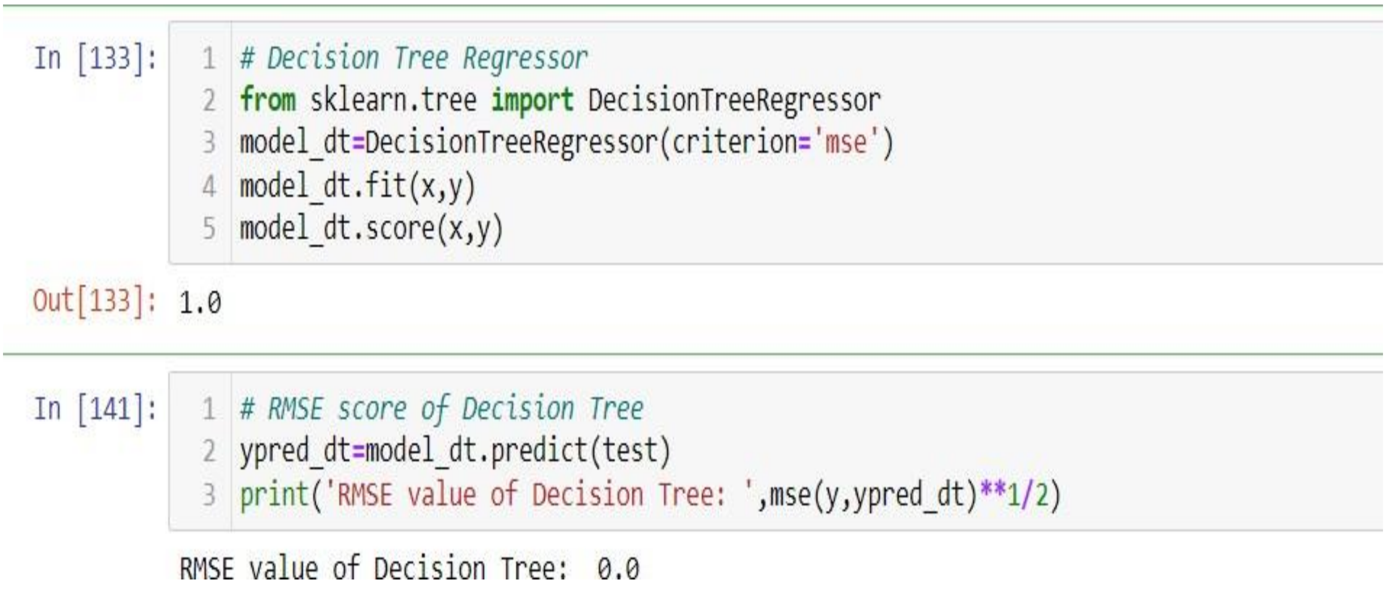
```
0.9797036141489379
```

In [140]:

```
# RMSE score of Random Forest
ypred_rf=model_rf.predict(test)
print('RMSE of Random Forest: ',mse(y,ypred_rf)**1/2)
```

```
RMSE of Random Forest:  63449914.5854991
```

5) Decision Tree Regressor



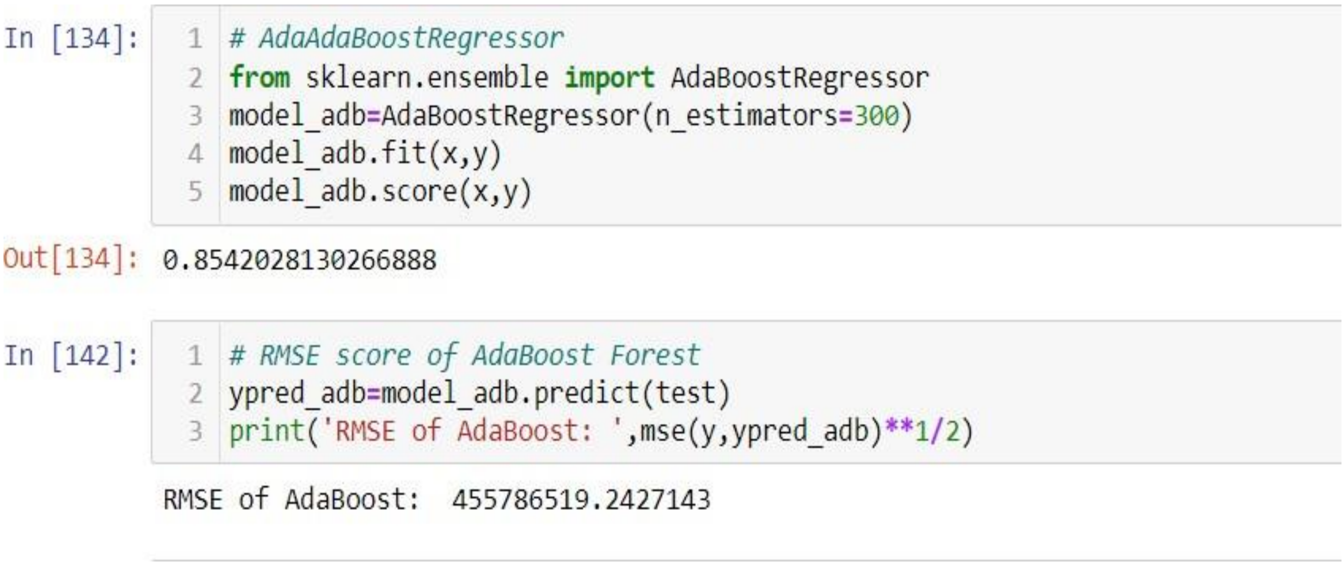
```
In [133]: # Decision Tree Regressor
          from sklearn.tree import DecisionTreeRegressor
          model_dt=DecisionTreeRegressor(criterion='mse')
          model_dt.fit(x,y)
          model_dt.score(x,y)

Out[133]: 1.0
```

```
In [141]: # RMSE score of Decision Tree
          ypred_dt=model_dt.predict(test)
          print('RMSE value of Decision Tree: ',mse(y,ypred_dt)**1/2)

          RMSE value of Decision Tree:  0.0
```

6) Ada AdaBoost Regressor



```
In [134]: # AdaAdaBoostRegressor
          from sklearn.ensemble import AdaBoostRegressor
          model_adb=AdaBoostRegressor(n_estimators=300)
          model_adb.fit(x,y)
          model_adb.score(x,y)

Out[134]: 0.8542028130266888
```

```
In [142]: # RMSE score of AdaBoost Forest
          ypred_adb=model_adb.predict(test)
          print('RMSE of AdaBoost: ',mse(y,ypred_adb)**1/2)

          RMSE of AdaBoost:  455786519.2427143
```

We can see that the Decision tree is the best model, with best RSME score .