FLIP ROBO

Housing Price Prediction Project

Submitted by:

Atish Kalangutkar

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my SME (Subject Matter Expert) Shwetank Mishra as well as Flip Robo Technologies who gave me the opportunity to do this project on Housing Price Prediction, which also helped me in doing lots of research wherein I came to know about so many new things.

Also, I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement. All the external resources that were used in creating this project are listed below:

1) https://www.google.com/

2) https://www.youtube.com/

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 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 below.

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?

Conceptual Background of the Domain Problem:

The project will require knowledge and practice in building Graphs /plots and analyzing them to get the relationship between dataset, Knowledge of Different Learning Models to build and predict the required output. Basic Data science concepts to increase the quality of the dataset and Python Knowledge (Coding Language) which will be used to solve the complete Micro Credit Defaulter project. Understanding of calculating F2 score, accuracy, skewness and basic mathematics/statistical approaches will help to build an accurate model for this project.

Review of Literature:

Market price is what a willing, ready and bank-qualified buyer will pay for a property and what the seller will accept for it. The transaction that takes place determines the market price, which will then influence the market value of future sales.
Price is determined by local supply and demand, the property’s condition and what other similar properties have sold for without adding in the value component.
Market value is an opinion of what a property would sell for in a competitive market based on the features and benefits of that property (the value), the overall real estate market, supply and demand, and what other similar properties have sold for in the same condition.
The major difference between market value and market price is that the market value, in the eyes of the seller, might be much more than what a buyer will pay for the property or it’s true market price. Value can create demand, which can influence price. But, without the demand function, value alone cannot influence price. As supply increases and demand decreases, price goes down, and value is not influential. As

supply decreases and demand increases, the price will rise,and value will influence price. Market value and market price

can be equal in a balanced market.

However, buyers and sellers can view value differently. A

seller might feel that their in-ground pool is a benefit, but the

buyer could see it as a negative and place less value on the

property. Or the seller could feel the new roof they put on the

house has great value; however, the buyer places no value on

this because they expect the property to have a roof in good

condition. Or a builder might feel he has superior quality and

demand a higher price, but the buyer places less value on

quality and more value on the lot, neighborhood and floor plan

of the property.

Motivation for the Problem Undertaken:

I wanted to solve the real-life problem using the Technical skills

gathered during the course of being a Data Scientist and improving

the skill set

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem:

We are building a model in Machine Learning to predict the actual value of the prospective properties and decide whether to invest in them or not. So, this model will help us to determine which variables are important to predict the price of variables & also how these variables describe the price of the house. This will help to determine the price of houses with the available independent variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns.
Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The most common

form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion. For specific mathematical reasons this allows the researcher to estimate the
conditional expectation of the dependent variable when the independent variables take on a given set of values.
Regression analysis is also a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

Data Sources and their formats

Data set provided by Flip Robo was in the format of CSV (Comma Separated Values). The dimension of data is 1168 rows and 81 columns. There are 2 data sets that are given. One is training data
and one is testing data.
1) Train file will be used for training the model, i.e., the model will learn from this file. It contains all the independent variables and the target variable. Size of training set: 1168 records.
2) Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data. Size of test set: 292 records.

Data Preprocessing Done:

Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre- process our data before feeding it into our model. Therefore, it is the first and crucial step while creating a machine learning model. I have used some following pre-processing steps:

a. Loading the training dataset as a dataframe

b. Used pandas to set display I ensuring we do not see any truncated information

c. Checked the number of rows and columns present in our training dataset

d. Checked for missing data and the number of rows with null values

e. Verified the percentage of missing data in each column and decided to discard the ones that have more than 50% of null values

f. Dropped all the unwanted columns and duplicate data

present in our data frame. Separated categorical data and numeric data

names in separate list variables for ease in visualization

h. Checked the unique values information in each column to get

a list for categorical data

i. Performed imputation to fill missing data using mean on

numeric data and mode for categorical data columns

j. Used Pandas Profiling during the visualization phase along

with pie plot, count plot, scatter plot, Bar plot and line plot.And also checked the top 10 highest sale price in the dataset.Than made separate data frame for it and than I plotted count plot and pie plot for that data.

k. With the help of Label encoder converted all

object data type columns to numeric datatype .

l.With the help of heatmap, correlation bar graph was able to understand the Feature vs Label relativity and insights on multicollinearity amongst the feature columns
m.Than I checked for outliers for continuous data features by plotting box plot and I found that lot of outliers were present.But I was not able to remove the outliers as there was almost 30% data loss.So I didn't removed the outliers.
n.Than I checked for skewness for continuous data features.And I treated the skewness with the help of Power transform method.
o. Separated feature and label data

p.Than scaled the data and used vif to check the multicollinearity.And if the value was more than 5 than I have dropped the columns from the dataset.

q.Than I have used feature selection method to get the best features to train the model.
r.Than used different models to select the best model

S.Checked for the best random state to be used on our Regression Machine Learning model pertaining to the feature importance details
t. Finally created a regression model function along with evaluation metrics to pass through various model formats

Data Inputs- Logic- Output Relationships:

When we loaded the training dataset, we had to go through various data pre processing steps to understand what was given to us and what we were expected to predict for the project. When it comes to the logical part, the domain expertise of understanding how real estate works and how we are supposed to cater to the customers came in handy to train the model with the modified input data. In the Data Science community there is a saying “Garbage In Garbage Out” therefore we had to be very cautious and spent almost 80% of our project building time in understanding each and every aspect of the data and how they were related to each other as well as our target label.With the objective of predicting housing sale prices accurately we
had to make sure that a model was built that understood the customer priorities trending in the market imposing those norms when a relevant price tag was generated. I tried my best to retain as much data possible that was collected but I feel discarding columns that had lots of missing data was good. I did not want to impute

data and then cause biases in the machine learning model from values that did not come from real people.

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

The assumption part for me was relying strictly on the data provided to me and taking into consideration that the separate training and testing datasets were obtained from real people surveyed for their preferences and how reasonable a price for a house with various features inclined to them was.

Hardware and Software Requirements and Tools Used:

Hardware Used:

i.

RAM: 8 GB

ii.

CPU: AMD Ryzen 5 4600H

iii.

GPU: AMD Radeon ™ Vega 8 Graphics and NVIDIA GeForce GTX 1650 Ti

Software Used:

i.

Programming language: Python

ii.

Distribution: Anaconda Navigator

iii.

Browser based language shell: Jupyter Notebook

Libraries/Packages Used:

Pandas, NumPy, matplotlib, seaborn, scikit-learn.

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods):

I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data and EDA to check the correlation of independent and dependent features. Also, before building the model, I made sure that the input data was cleaned and scaled before it was fed into the machine learning models.

For this project we need to predict the sale price of houses, meaning our target column is continuous so this is a regression problem. I have used various regression algorithms and tested for the prediction. By doing various evaluations I have selected Gradient Boosting Regressor as the best suitable algorithm for our final model as it is giving a good r2-score and Root mean square error is less as compared to other algorithms used. Other regression algorithms are also giving me good accuracy but some are over-fitting and some are under-fitting the results which may be because of less amount of data.

In order to get good performance as well as accuracy and to check my model from overfitting and under-fitting I have made use of the K-Fold cross validation and then hyper parameter tuned the final model.

But after doing hyperparameter tuning the training testing and cross validation score reduced.so I saved the model on which hyperparameter tuning was not done.

Before I loaded the testing data and started performing the data pre-processing as the training dataset and obtaining the predicted sale price values out of the Regression Machine Learning Model.

Testing of Identified Approaches (Algorithms)

The algorithms used on training and test data are as follows:

A. Xtreme Gradient Boosting Regression Model

B. Gradient Boosting Regression Model

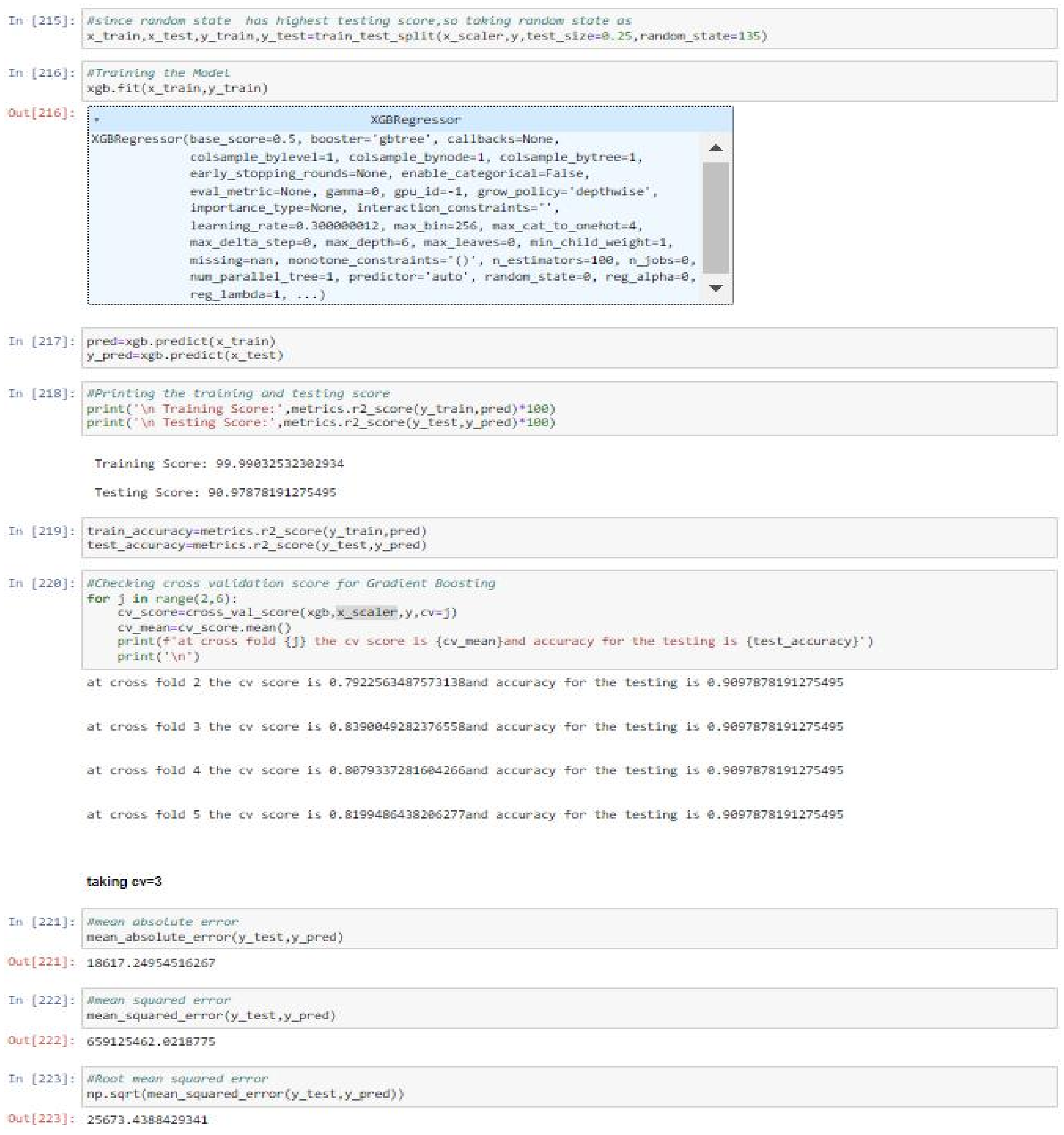
C. AdaBoost Regression Model

D. Random Forest Regression Model

Run and Evaluate selected models:

I used a total of 4 Regression Models after choosing the random state amongst 0-200 numbers. The code for the models is listed below.

XGB



```
In [215]: #since random state  has highest testing score,so taking random state as
          x_train,x_test,y_train,y_test=train_test_split(x_scaler,y,test_size=0.25,random_state=135)

In [216]: #Training the Model
          xgb.fit(x_train,y_train)

Out[216]:                         XGBRegressor
          XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
                       colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                       early_stopping_rounds=None, enable_categorical=False,
                       eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
                       importance_type=None, interaction_constraints='',
                       learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
                       max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
                       missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
                       num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
                       reg_lambda=1, ...)

In [217]: pred=xgb.predict(x_train)
          y_pred=xgb.predict(x_test)

In [218]: #Printing the training and testing score
          print('\n Training Score:',metrics.r2_score(y_train,pred)*100)
          print('\n Testing Score:',metrics.r2_score(y_test,y_pred)*100)

           Training Score: 99.99032532302934

           Testing Score: 90.97878191275495

In [219]: train_accuracy=metrics.r2_score(y_train,pred)
          test_accuracy=metrics.r2_score(y_test,y_pred)

In [220]: #Checking cross validation score for Gradient Boosting
          for j in range(2,6):
              cv_score=cross_val_score(xgb,x_scaler,y,cv=j)
              cv_mean=cv_score.mean()
              print(f'at cross fold {j} the cv score is {cv_mean}and accuracy for the testing is {test_accuracy}')
              print('\n')
          at cross fold 2 the cv score is 0.7922563487573138and accuracy for the testing is 0.9097878191275495


          at cross fold 3 the cv score is 0.8390049282376558and accuracy for the testing is 0.9097878191275495


          at cross fold 4 the cv score is 0.8079337281604266and accuracy for the testing is 0.9097878191275495


          at cross fold 5 the cv score is 0.8199486438206277and accuracy for the testing is 0.9097878191275495
```

taking cv=3

```
In [221]: #mean absolute error
          mean_absolute_error(y_test,y_pred)

Out[221]: 18617.24954516267

In [222]: #mean squared error
          mean_squared_error(y_test,y_pred)

Out[222]: 659125462.0218775

In [223]: #Root mean squared error
          np.sqrt(mean_squared_error(y_test,y_pred))

Out[223]: 25673.4388429341
```

Gradient Boosting

```
gb=GradientBoostingRegressor()
```

In [258]:
```
#since random state  has highest testing score,so taking random state as
x_train,x_test,y_train,y_test=train_test_split(x_scaler,y,test_size=0.25,random_state=135)
```

In [259]:
```
#Training the Model
gb.fit(x_train,y_train)
```

Out[259]:
```
▾ GradientBoostingRegressor
GradientBoostingRegressor()
```

In [260]:
```
pred=gb.predict(x_train)
y_pred=gb.predict(x_test)
```

In [261]:
```
#Printing Training and testing score
print('\n Training Score:',metrics.r2_score(y_train,pred)*100)
print('\n Testing score:',metrics.r2_score(y_test,y_pred)*100)
```

```

 Training Score: 95.56101098754579

 Testing score: 91.2862131362734
```

In [262]:
```
train_accuracy=metrics.r2_score(y_train,pred)
test_accuracy=metrics.r2_score(y_test,y_pred)
```

In [263]:
```
#Checking cross validation score for XGBoost
cv_score=cross_val_score(gb,x_scaler,y,cv=3).mean()
cv_score
```

Out[263]:
```
0.8492358888670711
```

In [264]:
```
#Mean Absolute Error
mean_absolute_error(y_test,y_pred)
```

Out[264]:
```
18300.582066591644
```

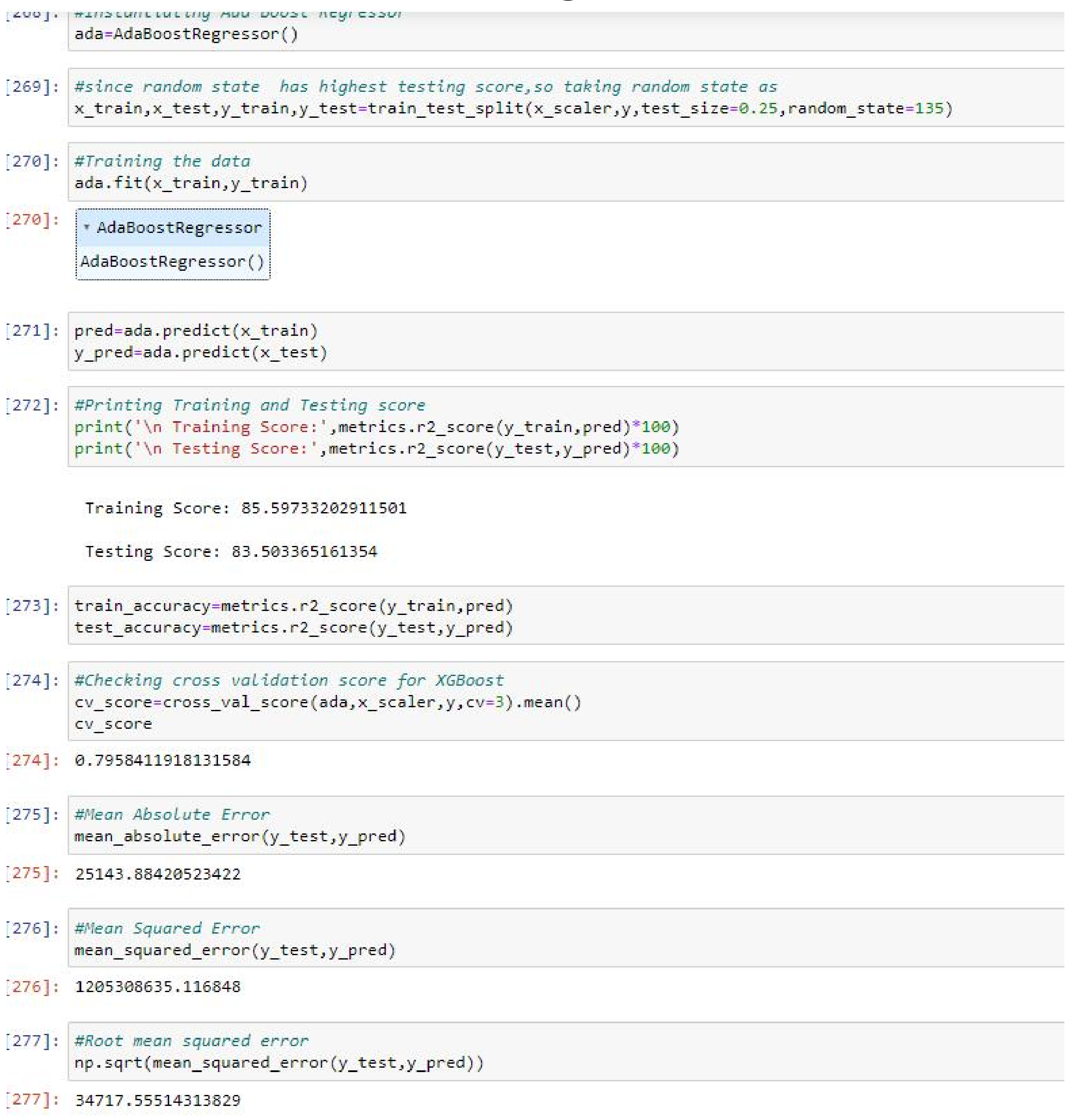
In [265]:
```
#Mean Squared Error
mean_squared_error(y_test,y_pred)
```

Out[265]:
```
636663334.9253095
```

In [266]:
```
#Root mean squared error
np.sqrt(mean_squared_error(y_test,y_pred))
```

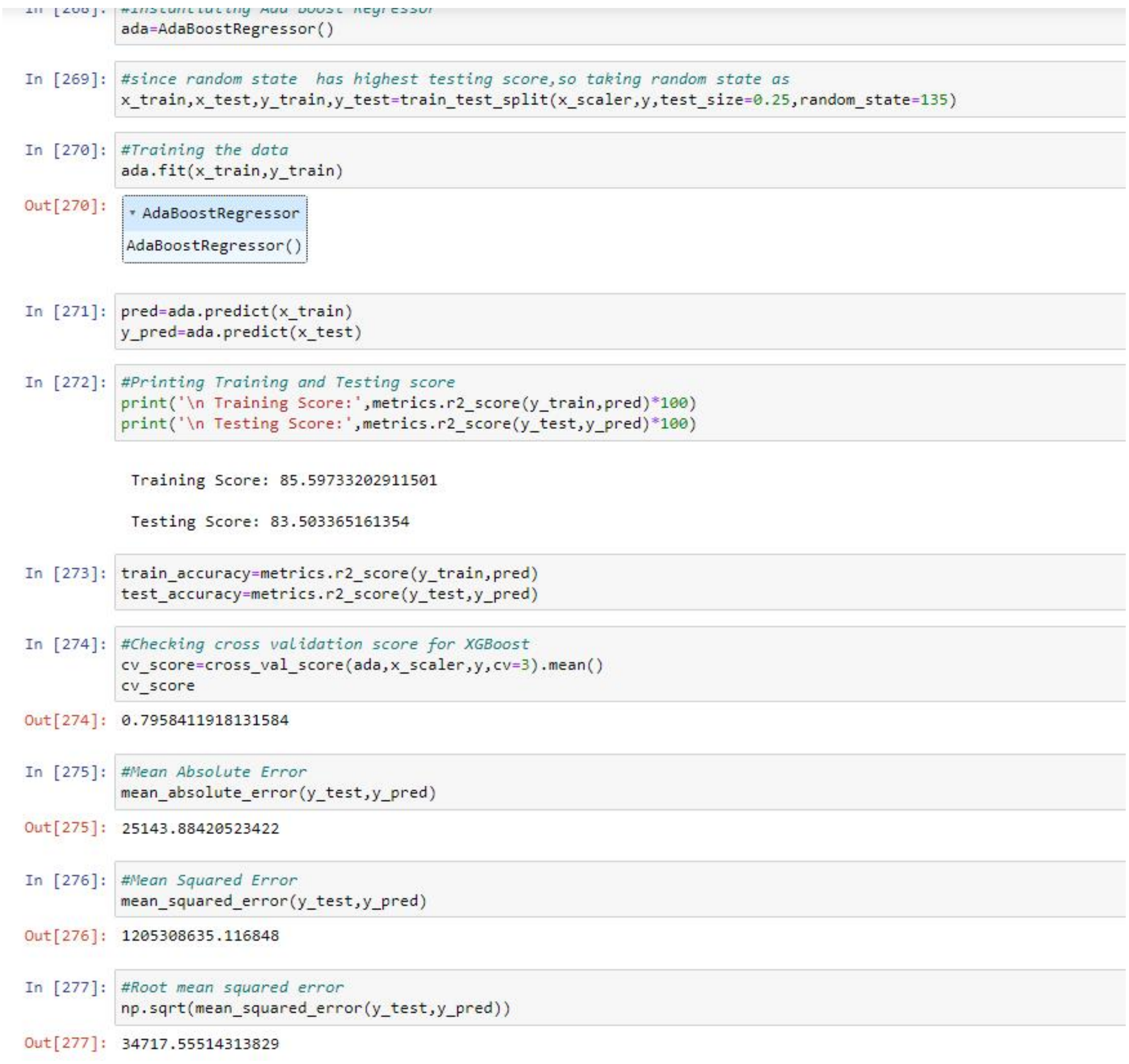
Out[266]:
```
25232.188468805267
```

Ada Boosting



```
[268]: #Instantiating Ada Boost Regressor
       ada=AdaBoostRegressor()

[269]: #since random state  has highest testing score,so taking random state as
       x_train,x_test,y_train,y_test=train_test_split(x_scaler,y,test_size=0.25,random_state=135)

[270]: #Training the data
       ada.fit(x_train,y_train)

[270]: ▾ AdaBoostRegressor
       AdaBoostRegressor()

[271]: pred=ada.predict(x_train)
       y_pred=ada.predict(x_test)

[272]: #Printing Training and Testing score
       print('\n Training Score:',metrics.r2_score(y_train,pred)*100)
       print('\n Testing Score:',metrics.r2_score(y_test,y_pred)*100)

        Training Score: 85.59733202911501

        Testing Score: 83.503365161354

[273]: train_accuracy=metrics.r2_score(y_train,pred)
       test_accuracy=metrics.r2_score(y_test,y_pred)

[274]: #Checking cross validation score for XGBoost
       cv_score=cross_val_score(ada,x_scaler,y,cv=3).mean()
       cv_score

[274]: 0.7958411918131584

[275]: #Mean Absolute Error
       mean_absolute_error(y_test,y_pred)

[275]: 25143.88420523422

[276]: #Mean Squared Error
       mean_squared_error(y_test,y_pred)

[276]: 1205308635.116848

[277]: #Root mean squared error
       np.sqrt(mean_squared_error(y_test,y_pred))

[277]: 34717.55514313829
```

Random Forest



```
In [268]: #instantiating Ada Boost Regressor
          ada=AdaBoostRegressor()

In [269]: #since random state  has highest testing score,so taking random state as
          x_train,x_test,y_train,y_test=train_test_split(x_scaler,y,test_size=0.25,random_state=135)

In [270]: #Training the data
          ada.fit(x_train,y_train)

Out[270]: ▾ AdaBoostRegressor
          AdaBoostRegressor()

In [271]: pred=ada.predict(x_train)
          y_pred=ada.predict(x_test)

In [272]: #Printing Training and Testing score
          print('\n Training Score:',metrics.r2_score(y_train,pred)*100)
          print('\n Testing Score:',metrics.r2_score(y_test,y_pred)*100)

           Training Score: 85.59733202911501

           Testing Score: 83.503365161354

In [273]: train_accuracy=metrics.r2_score(y_train,pred)
          test_accuracy=metrics.r2_score(y_test,y_pred)

In [274]: #Checking cross validation score for XGBoost
          cv_score=cross_val_score(ada,x_scaler,y,cv=3).mean()
          cv_score

Out[274]: 0.7958411918131584

In [275]: #Mean Absolute Error
          mean_absolute_error(y_test,y_pred)

Out[275]: 25143.88420523422

In [276]: #Mean Squared Error
          mean_squared_error(y_test,y_pred)

Out[276]: 1205308635.116848

In [277]: #Root mean squared error
          np.sqrt(mean_squared_error(y_test,y_pred))

Out[277]: 34717.55514313829
```

Key Metrics for success in solving problem under Consideration:

The key metrics used here were r2_score, cross_val_score, MAE, MSE and RMSE. We tried to find out the best parameters and also to increase our scores by using Hyperparameter Tuning and we will be using the GridSearchCV method.

1. Cross Validation:

Cross-validation helps to find out the overfitting and underfitting of the model. In the cross validation the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part being 20% of the full dataset. While running the Cross-validation the 1st part (20%) of the 5 parts will be kept out as a holdout set for validation and everything else is used for trainingdata. This way we will get the first estimate of the model quality of the dataset.

In the similar way further iterations are made for the second 20% of the dataset is held as a holdout set and remaining 4 parts are used for training data during the process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross-validation process to get the remaining estimate of the model quality.

2. R2 Score:

It is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.

3. Mean Squared Error (MSE):

MSE of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss. RMSE is the Root Mean Squared Error.

4. Mean Absolute Error (MAE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

5. Hyperparameter Tuning:

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. You must select from a specific list of hyperparameters for a given model as it varies from model to model.We are not aware of optimal values for hyperparameters which would generate the best model output. So, what we tell the model is to explore and select the optimal model architecture automatically. This selection procedure for hyperparameters is known as Hyperparameter Tuning. We can do tuning by using GridSearchCV.

GridSearchCV is a function that comes in the Scikit-learn (or SK-learn) model selection package. An important point here to note is that we need to have the Scikit-learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

Visualizations:

I used pandas profiling to get the over viewed visualization on the pre-processed data. pandas-profiling is an open-source Python module with which we can quickly do an exploratory data analysis with just a few lines of code. It generates interactive reports in web format that can be presented to any person, even if they don’t know programming. It also offers report generation for the dataset with lots of features and customizations for the report generated.
I then created pie plots, count plots,scatter plots,line plots anf bar plots to get further visual insights on our training dataset feature values.

Interpretation of the Results

Results:

1) Large amount of null values are present in the dataset

2) Data Set is not normally distributed

3) Dataset have outliers in most of the variables

4) Dataset is not normalized

5) Dataset is highly skewed

6) Gradient Boosting Algorithm is best suited for the current dataset

CONCLUSION

Key Findings and Conclusions of the Study:

We found that to predict the House price using Data Science the best way after performing Data Cleaning is to use Gradient Boosting Algorithm it provides 91% accuracy which is better than other Regression algorithms.

Learning Outcomes of the Study in respect of Data Science:

The above study helps one to understand the business of real estate. How the price is changing across the properties. With the Study we can tell how multiple real estate amenities like swimming pool, garage, pavement and lawn size of Lot Area, and type of Building raise decides the cost. With the help of the above analysis, one can sketch the needs of a property buyer and according to need we can project the price of the property.

Limitations of this work and Scope for Future Work:

During this project I have faced a problem of low amount of data. Many columns are with the same entries in more than 80% of rows which lead to reduction in our model performance. One more issue is there are a large number of missing values present in this data set, so we have to fill those missing values in the correct manner. We can still improve our model accuracy with some feature engineering and by doing some extensive hyperparameter tuning on it