HOUSING PROJECT

Submitted by:

Rijul Kumar

ACKNOWLEDGMENT

Reference that i have used are:

- Data Trained Education online video
- Materials provided by Flip Robo
- Geeks for Geeks
- Stackoverflow

INTRODUCTION

- ## Business Problem Framing

    1. 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.
    2. We are supposed to build a model using Machine Learning in order to predict the actual value of the prospective properties.

- ## Conceptual Background of the Domain Problem

    The prices of houses depends on many factors. Such as:

    - Lot size
    - Type of road access to property
    - Type of utilities available
    - Physical locations within city limits
    - Rates the overall material and finish of the house
    - Heating quality and condition, Central air conditioning
    - Kitchen quality
    - And so on…

- Motivation for the Problem Undertaken
  - We are supposed to build a model using Machine Learning in order to predict the actual value of the prospective properties.
  - This model will then be used to understand how exactly the prices vary with the variables. This can be used to concentrate on areas that will yield high returns.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
  - First of all i imported data from train and test csv files to DataFrames using pandas.
  - After that i used .describe() to know the statistical information (such as max, min value,etc) of train dataframe.
  - Then i used .shape to know shape of train and test dataframe.
  - Afterwards i used .dtypes to know data type of each column of train and test dataframe.

- Data Sources and their formats
  - test.csv and train.csv provided by FlipRobo
  - DataDescription.txt provided by FlipRobo

- Data Preprocessing Done
  - First of all for data preprocessing i checked whether there is a NULL value or not in dataframe using heatmap as well as .isnull()

- After that NULL values were filled with mode of columns for categorical columns and mean of columns for continuous columns.
- After that i used count plots from seaborn library to plot all categorical columns for visualisation.
- Next i used Density plots from seaborn library to plot all continuous columns for visualisation.
- Then i did encoded the dataframe using Ordinal Encoder.
- After that i checked for correlations using heatmaps, correlation matrix and BAR plot.
- Finally i confirmed high correlations using VIF and dropped highly correlated columns.
- After that i removed skewness and checked outliers.

- ## Data Inputs- Logic- Output Relationships

  - Data Input :

    These are basically the factors (such as lot size, type of road access to property, type of utilities available, etc) which affects the prices of house.

  - Data Output :

    Our Target variable is SalesPrice which is the price of the houses and we are supposed to predict it with the help of data input (i.e factors affecting house prices).

- ## Set of assumptions related to the problem under consideration

    - First assumption that i took was that all ‘object’ data type columns are categorical data and hence plotted count plots.
    - Next assumption that i took was that all ‘float64’ and ‘int64’ data type columns are continuous data and hence plotted density plots.

- ## Hardware and Software Requirements and Tools Used

    - Hardware used:

      Laptop with intel core i5 7th gen

    - Software used:

        i. Jupyter notebook
        ii. Required python libraries such as numpy, pandas, seaborn, matplotlib, etc
        iii. Required libraries for model such as sklearn, etc

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
    - After preprocessing data (removing NULL, encoding, removing high correlations, removing skewness and outliers) i separated columns into features and target.
    - As this is a regression problem so we tried 4 models - LinearRegression, XGBRegressor, RandomForestRegressor and SVR
    - I also tried 4 metrics method - r2_score, mse, rms, mae
    - Then i used Lasso for regularization.
    - Finally i used Ensemble Technique.

- Testing of Identified Approaches (Algorithms)

  As this is a regression problem so we tried following 4 models -
    - LinearRegression
    - XGBRegressor
    - RandomForestRegressor
    - SVR

- ## Run and Evaluate selected models

  I defined a function model and then tried 4 different models using it

In [293]:
```
def model_selection(algorithm_instance,features_train,target_train,features_test,target_test):
    algorithm_instance.fit(features_train,target_train)
    model_1_pred_train = algorithm_instance.predict(features_train)
    model_1_pred_test = algorithm_instance.predict(features_test)
    print("Accuracy for the training model : ",r2_score(target_train,model_1_pred_train))
    print("Accuracy for the testing model : ",r2_score(target_test,model_1_pred_test))

    Train_accuracy = r2_score(target_train,model_1_pred_train)
    Test_accuracy = r2_score(target_test,model_1_pred_test)

    for j in range(2,10):
        cv_score = cross_val_score(algorithm_instance,feature,target,cv=j)
        cv_mean = cv_score.mean()
        print("At cross fold " + str(j) + " the cv score is " + str(cv_mean) + " and accuracy score for training is
        print("\n")
```

Result that i got for each model :

- LinearRegression :
  At random state 1 the training accuracy is : 0.8258075894920763
  At random state 1 the testing accuracy is : 0.7944252079267548

  At cross fold 2 the cv score is 0.6755921993594929 and accuracy score for training is - 0.6501706537893246 and accuracy score for testing is 0.7944252079267548

- XGBRegressor :
  Accuracy for the training model : 0.999928181520936
  Accuracy for the testing model : 0.8173283235415273

  At cross fold 2 the cv score is 0.7937274760605788 and accuracy score for training is 0.999928181520936 and accuracy score for testing is 0.8173283235415273

- RandomForestRegressor :
  Accuracy for the training model :
  0.9729581383527576
  Accuracy for the testing model :
  0.8475074393604315
  At cross fold 2 the cv score is 0.8040774368386595 and accuracy score for training is 0.9729581383527576 and accuracy score for testing is 0.8475074393604315

- SVR :
  Accuracy for the training model : -0.0497743509077444
  Accuracy for the testing model : -0.024558376537638482
  At cross fold 2 the cv score is -0.0579295586606845 and accuracy score for training is -0.0497743509077444 and accuracy score for testing is -0.024558376537638482

Finally i concluded that RandomForestRegressor() gives best accuracy and hence i took it as main model.

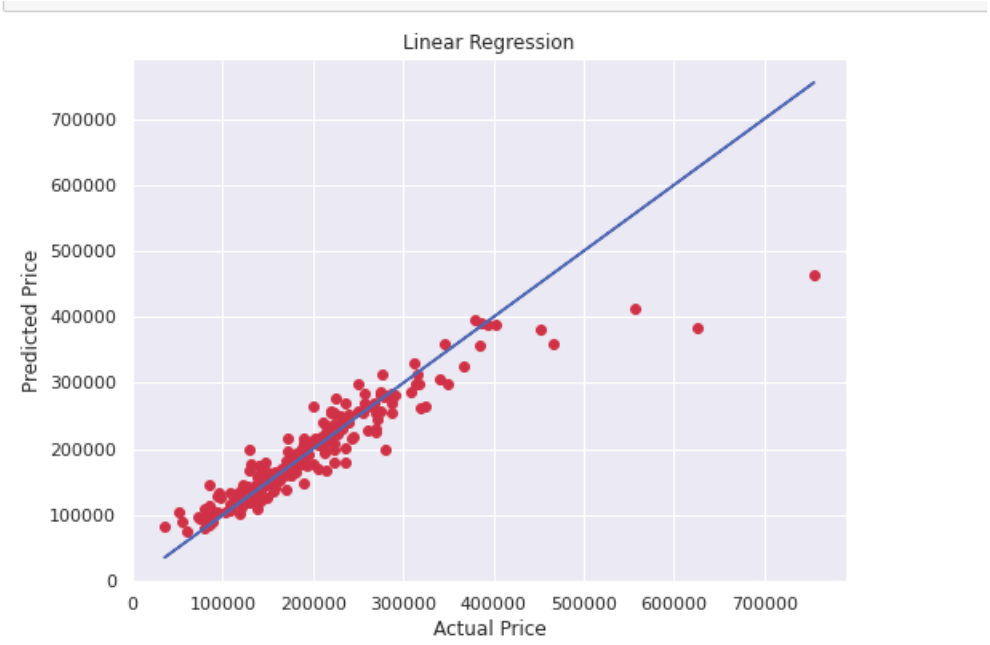
- ## Key Metrics for success in solving problem under consideration

  I tried 4 different metrics method:
  - r2_score
  - mse
  - rms
  - mae

  I got best results from r2_score and hence used it in final model

- ## Visualizations

  Got a good best fit curve for linear regression as most of the points lie on it



- ## Interpretation of the Results

  So the results which i got were:

  - Although RandomForestRegressor() is best model but LinearRegression() and XGBRegressor() can also be used as they are also good models.
  - Above point can be seen through best fit curve as well as r2_score which is the also best among 4 metrics tried.
  - We got our final accuracy as 82.69%

CONCLUSION

- Key Findings and Conclusions of the Study

    - From this study i learnt that sometimes when purchasing houses people may have some very strict requirement for some amenities like in land contour factor almost all wanted Leveled one while in some factors like Foundation they are willing to accept more options.

    - House prices are highly dependent on Overall Quality and Ground Living area as they were highly correlated in the dataframe.

- Learning Outcomes of the Study in respect of Data Science

  Some problems faced and their solution (using visualisation and algorithm) used were:

    - Removing outliers was leading to very high data loss (57.45%). So in place of removing it i used RandomForestRegressor() which is not sensitive to outliers.
    - Too many columns which made me unable to check which column has all NULL values. So here i used heatmaps which helped me to find column with all NULL values.

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

  Some limitations are :

  - There are many other factors which are not in the data which may play major role in prices of some houses.
  - Unrelated factors such as global warming may indirectly also affect price of houses.
  - With evolving technology, prices of houses made with new technology increases while those made with old technologies decreases.

  Scope for future work :

  - This can be made further accurate by taking more and more factors into account