Housing Price Prediction Model

Submitted by:

Nihal R. Singh

ACKNOWLEDGMENT

This Project includes the details about housing company, which uses data analytics to get house details and sale the houses and make the profit and increases there revenue.

US Based Housing company Surprising Housing uses the data analytics to purchases houses at lower price of their actual price and sell them to higher price to make profit out of them.

The Housing price prediction model is a regression model as the target column is having continuous numerical data.

INTRODUCTION

- Business Problem Framing

  The Housing price prediction model is basically predict the Housing price after buying them on lower price.

- Conceptual Background of the Domain Problem

  The model is basically related to the Housing domain and consist of different details related to houses and after analysing all the data predict the price of the house.

- Review of Literature

  A US Based housing company Surprise Housing collect multiple data and done the data analysis to purchase the houses at a lower price of their actual values and sell them to higher price in Australia.

  Surprise Housing is basically uses the model to predict the price of the house and doing the analysis on the data.

- Motivation for the Problem Undertaken

  The Housing price prediction model consist of consist of house related data and do the analysis and then predict the price of the house using model.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The Dataset is in csv format which consist of train and test data. Performing all the data analysis and EDA on the test and train data but building the model only on train data and test data is only used to predict the model. Plotting the multiple plots like bar plot, dist plot, box plot, heat map, cat plot to analyse the data and which features are important and related to the target and delete the non-important features. Using z-score and iqr method to find the outliers and deleting it. Scaling the data to build model.

- Data Sources and their formats

  The Dataset is in csv format which consist of train and test data in different csv files so merging both the file into single data frame to perform the analysis and then separate them for the model building.

  The data having the different datatypes i.e. int, object and float.

```
#Loading the test and train dataset
train=pd.read_csv('train.csv')
test=pd.read_csv('test.csv')
```

```
#combining train and test dataset
train['Source']='train'
test['Source']='test'

#Combing train and test dataset for the EDA and preprocessning
df=pd.concat([train,test],ignore_index=True)
```

```
#printing shape of train,test and df dataset
print(train.shape, test.shape, df.shape)
```

```
(1168, 82) (292, 81) (1460, 82)
```

```
#printing the dataframe
df.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSc
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	NaN	0	2	20
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	NaN	0	10	20
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	NaN	0	6	20
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	NaN	MnPrv	NaN	0	1	20
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	NaN	0	6	20

- Data Pre-processing Done

  The Data Pre-processing consist of below mentioned points:

  1. Merge the train and test data for the Data pre-processing.
  2. Checking the null values form the dataset, if null value is available them replace them with the suitable values.
  3. Describing the data.
  4. Using the encoding method to encode the categorical data into the numerical data.
  5. Checking all the features and checking the impact with target if no impact is there then deleting the features.
  6. Plotting the multiple plots like cat plot, dist plot, box plot to check the data distribution and outliers from the datasets.
  7. Using IQR method to check the outliers and remove the outliers from the data frame.
  8. Checking the skewness after removing the outliers from the dataset and skewness is present in the features so using the power transform method to remove the skewness.
  9. Using the heat map and variance inflation factor to check the multicollliniarity issue between the features.
  10. If issue is there then delete the feature which having such issue.
  11. Scaling the data using the standard scalar method.
  12. Separating the train and test data from the data frame to build the model on train data frame.
  13. Separating the features and target to build the model.

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

  Deleted the multiple columns based on the analysis which does not having any relation with target and the final outcome of the model.

- Hardware and Software Requirements and Tools Used

  The Tool used to build the model is anaconda jupyter.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Multiple problem faced during the model building. All are listed bellows.

  1. Checking for the unwanted columns analyse the relation with target and removal of that features.
  2. Checking the null values and replace it.
  3. Checking of zeros and replace it.
  4. Managing the categorical columns using multiple methods.
  5. Removal of outliers form the features using the IQR.
  6. Removal of skewness using the power transform function.
  7. Removal of multicolliniarity issue from the features.

- Testing of Identified Approaches (Algorithms)

  This is a Regression problem as the target column has continuous data so multiple regression model has been used for the model building and basically training and testing the data on multiple models to select the best model.

  The multiple models are used as given below:

  1. Linear Regression
  2. KNeighbors Regressor
  3. Random Forest Regressor
  4. AdaBoost Regressor
  5. SVR

- Run and Evaluate selected models

  The model is a classification model so all the classification model building algorithm has been used to build the model and across all the model the best one has been selected as final model.

  The different models which has been used has been listed below:

  1. Linear Regression : Linear Regression is a supervised machine learning algorithm where the predicted output is continuous. It's used to predict values within a continuous range.

```
In [148]: #seprating the train and test datasets with the best find random state
          x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=179)
```

```
In [149]: #model Evaluation for LR:
          print('\n==========Model Evaluaton and Accuracy check using LinearRegression==========')
          lr=LinearRegression()
          lr.fit(x_train,y_train)
          y_predlr=lr.predict(x_test)
          print(f"The accuracy of the model using LinearRegression is: {r2_score(y_test,y_predlr)*100:.2f}%\n")



          #model Evaluation for KNN:
          print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========')
          knn=KNeighborsRegressor()
          knn.fit(x_train,y_train)
          y_predknn=knn.predict(x_test)
          print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")



          #model Evaluation for RandomForestRegressor:
          print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor==========')
          rfr=RandomForestRegressor()
          rfr.fit(x_train,y_train)
          y_predrfr=rfr.predict(x_test)
          print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")



          #model Evaluation for AdaBoostRegressor:
          print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor==========')
          ada=AdaBoostRegressor()
          ada.fit(x_train,y_train)
          y_predada=ada.predict(x_test)
          print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")



          #model Evaluation for SVR:
          print('\n===============Model Evaluaton and Accuracy check using SVR==================')
          svr=SVR()
          svr.fit(x_train,y_train)
          y_predsvr=svr.predict(x_test)
          print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

```
==========Model Evaluaton and Accuracy check using LinearRegression==========
The accuracy of the model using LinearRegression is: 88.46%
```

  After training the data in Linear Regression the r2 score of the model is 88.46%.

  The cross validation score for the Linear Regression model is 76.05%

2. SVR : Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

```
#model Evaluation for SVR:
print('\n==========Model Evaluaton and Accuracy check using SVR==========')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

```
==========Model Evaluaton and Accuracy check using LinearRegression==========
The accuracy of the model using LinearRegression is: 88.46%


==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========
The accuracy of the model for using KNeighborsRegressor is: 32.51%


==========Model Evaluaton and Accuracy check using RandomForestRegressor==========
The accuracy of the model using RandomForestRegressor is: 86.73%


==========Model Evaluaton and Accuracy check using AdaBoostRegressor==========
The accuracy of the model for using AdaBoostRegressor is: 82.41%


==========Model Evaluaton and Accuracy check using SVR==========
The accuracy of the model for using SVR is: -9.37%
```

After training the data in SVR the r2 score of the model is -9.37%.

The cross validation score for the SVR Regression model is -5.78%

3. Ada boost Regressor: AdaBoost Regressor, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method.

```
#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")



#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor==========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")



#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor==========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")



#model Evaluation for SVR:
print('\n================Model Evaluaton and Accuracy check using SVR====================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

```
==========Model Evaluaton and Accuracy check using LinearRegression==========
The accuracy of the model using LinearRegression is: 88.46%


==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========
The accuracy of the model for using KNeighborsRegressor is: 32.51%


==========Model Evaluaton and Accuracy check using RandomForestRegressor==========
The accuracy of the model using RandomForestRegressor is: 86.73%


==========Model Evaluaton and Accuracy check using AdaBoostRegressor==========
The accuracy of the model for using AdaBoostRegressor is: 82.41%
```

After training the data in Ada boost Regressor the r2 score of the model is 82.41%.

The cross validation score for the Ada boost Regression model is 76.36%

4. Random forest Regressor: Random forests is a supervised learning algorithm. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.

```
#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor==========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")


#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor==========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")


#model Evaluation for SVR:
print('\n===============Model Evaluaton and Accuracy check using SVR==================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

```
==========Model Evaluaton and Accuracy check using LinearRegression==========
The accuracy of the model using LinearRegression is: 88.46%


==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========
The accuracy of the model for using KNeighborsRegressor is: 32.51%


==========Model Evaluaton and Accuracy check using RandomForestRegressor==========
The accuracy of the model using RandomForestRegressor is: 86.73%
```

After training the data in Random forest Regressor the r2 score of the model is 86.73%.

The cross validation score for the Random forest Regression model is 82.96%

5. KNN Regressor: KNN also known as K-nearest neighbour is a supervised and pattern classification learning algorithm which helps us find which class the new input(test value) belongs to when k nearest neighbours are chosen and distance is calculated between them.

```
#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")


#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor==========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")


#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor==========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")


#model Evaluation for SVR:
print('\n================Model Evaluaton and Accuracy check using SVR====================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

```
==========Model Evaluaton and Accuracy check using LinearRegression==========
The accuracy of the model using LinearRegression is: 88.46%


==========Model Evaluaton and Accuracy check using KNeighborsRegressor==========
The accuracy of the model for using KNeighborsRegressor is: 32.51%
```

After training the data in KNN Regressor the r2 score of the model is 32.51%.

The cross validation score for the KNN Regression model is 32.86%

- Key Metrics for success in solving problem under consideration

  There are multiple points which impact the final outcome for the model.

  1. Deleting the non related features from the data frame, which is not shows impact on final model.
  2. Checking the multicolliniarity issue and removing it by deleting the features from the data frame.
  3. Checking the skewness and removing the skewness using the power transform method.
  4. First merging the train and test data set and performing the complete EDA and visualization then again separate the train and test data set and perform model building on train data set ans test data set is used to predict the model.

- Visualizations

  Visualization is basically finding some outcomes after visualizing the data in form of some graph or some plots. Different visualizing methods has been use the to do the analysis on the data. In this model multiple plots has been used to do analysis on the provided data.

  1. Bar Plot – To check the relation between feature and target.
  2. Dist plot – To check the data distribution and checking the linearity of data distribution.
  3. Heat Map – To check the null count and multicolliniarity issue between the features.
  4. Cat Plot – To check the relationship of the target variable with the multiple features.
  5. Box Plot – To check the outliers in the features.
  6. Count Plot – Use to check the count of 0 values in features.

- Interpretation of the Results

  The Housing Price Prediction Model is used to do the prediction of the house price based on the multiple conditions (features).

CONCLUSION

- Key Findings and Conclusions of the Study

  Main finding in the models are not all the provided features are relevant to the model building. Some of them are not useful. The data consist of lots of null values, which can be replaced with mean and mod values.

- Learning Outcomes of the Study in respect of Data Science

  There are multiple features available in the data frame but all are not useful with respect to the final outcome. Many of the unwanted features and all the features has the outliers which has been taken care of using the iqr method. There are multicolliniarity issue is also available in the features that is also managed by deleting the features having multicolliniarity issues. The model is build on the train data set and it has been tested on the test data set.