International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

HOUSE PRICE PREDICTION

Swaroop Karmankar[1], Anurag Thakur[2], Mehul Paigwar[3]

Ashwini Lokhande[4], Tushar Ughade[5], Kajal Sharma[6]

[1.2.3.4.5.6] *Department of Computer Science and Engineering,*
G.H Raisoni Academy of Engineering and Techology, Nagpur, Shradha Park, B-37/39
Hingna-wadi road,Nagpur,440028

(Professor Priyanka Gonnade, Department of CSE, GHRAET, Nagpur, 440028)

Abstract - Food, water and shelter are three of the most basic needs for every living beings. Due to increase in civilization and livelihood there is exponential growth in buying houses in certain localities. Buying a house is a hassle now-a-days. So, the objective of this paper is to predict house prices with more accuracy considering various factors which affects the price, using different algorithms and techniques in machine learning. Along with these, we are also going to use some supervised algorithm such as SVR (Support Vector Regression) and Random Forest. Implementation of bagging and boosting will also be done. Thus after implying various techniques and algorithm on data set, this paper focuses on that related factors which may affect price variations are also taken in considerations, this will help us to predict prices more accurately. This paper provides vague knowledge on various techniques and machine learning algorithm used to predict house prices on any data set. After every implementation we will also study about cross validation and hyper tuning of parameters.

Key Words: Multiple Linear Regression, Ridge regression, LASSO regression, Support Vector Machine (SVM), Random Forest,Hyper tuning.

1. INTRODUCTION

As mentioned shelter is one of the basic need of every living beings. Now-a-days it is called as House which is pretty difficult to own due to affordability. Due to many circumstances house prices are changing rapidly, mostly due to economic growth but also due to many physical factors such as locality, reachability and etc. Either it increases or decreases gradually or exponentially. For any family who wish to expand their generation a house is a must. For any individual who is unaware about the prices of houses will get an idea about the price and can decide keeping affordability in mind. It is not an easy task to predict accurate prices of the estate. Our model will make this possible with more precision. This will also help the seller to get fair price for his\her property.

We began with the collection of data. For an instance we are using data set of Bangalore in which we have following parameters : area type, availability, location, size, society, total sq. ft. bath and balcony. Now the preprocessing of the data needs to be done. The next step will of exploratory data analysis, moving forward will be feature engineering, this will determine the independent and dependent variable. Once all this is completed, we will split the data and perform training of data. The data set will be trained according to different algorithms and techniques implemented. We try to model predictions by performing training and splitting of data and predict it using various machine learning techniques like different forms of Regression. We have analyzed each regression technique and calculated its score. Now we implement our various techniques and algorithm using tools like Python and IPython Jupyter Notebook.

2. Methodology

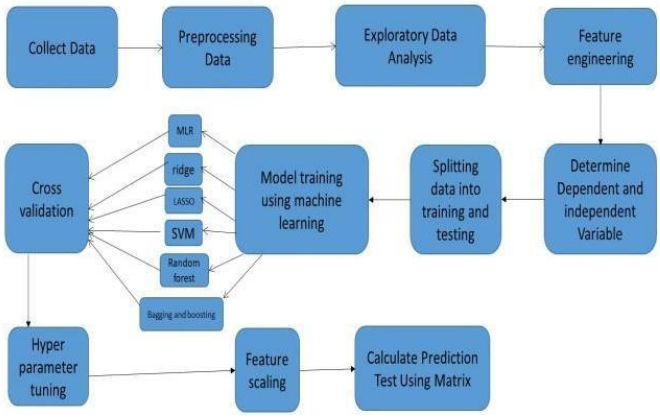


Fig -1: flow chart

The two data sets-train set and test data considered in theproject is taken from Machine Hackathon platform. It consists of features of house-property in Bengaluru. There are 9 features in both the info sets. The features can be explained as follows:

1. Area type-describes the area
2. Availability-when it's possesses or when it's ready.
3. Price- Value of the property in lakhs.
4. Size- in BHK or Bedroom (1-10 or more)
5. Society- to which it belongs.
6. Total_sqft - total sqft of the property.
7. Bath-No of bathrooms
8. Balcony- No of balcony
9. Location – where it is located

With 9 features available, we attempt to build regression models to predict house price. The data set consists of 11320 records with 9 explanatory variables. After pre-processing the dataset, train data set consistt of 5788 records with 250 variables. In test dataset. there were around 1448 records amd 250 variable. While building regression models we are often required to convert the specific i.e. text features to its numeric representation. Thetwo most common ways to do this is to use label encoder or one hot encoder. Label encoding in python are often achieved by using sklearn library.

But Label Encoder are not used when we have very large dataset. Because it comapare between two numerical values therefore we have used One Hot Encoding. One hot encoding refers to splitting the column that contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains "0" or "1" corresponding to which column it has been placed.

This dataset includes quite a few categorical variables (both train and test data set) for which we will need to create dummy variables or use label encoding to convert into numerical form. These would be fake/dummy variables because they are placeholders for actual variable and are created by ourselves. Also, there are a lot of null values present as well, so we will need to treat them accordingly. The features bath, price, balcony are numerical variables. Features like area_type, total_sqft, location, society, availability, and size appears as categorical variables.

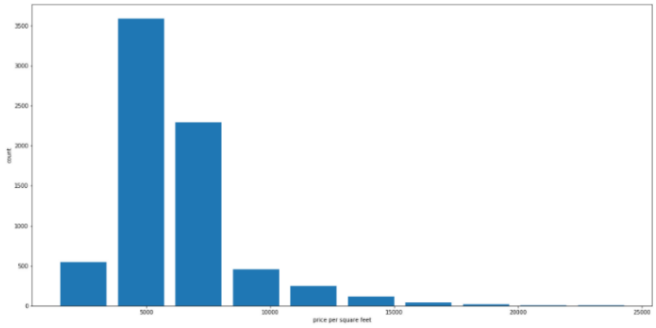
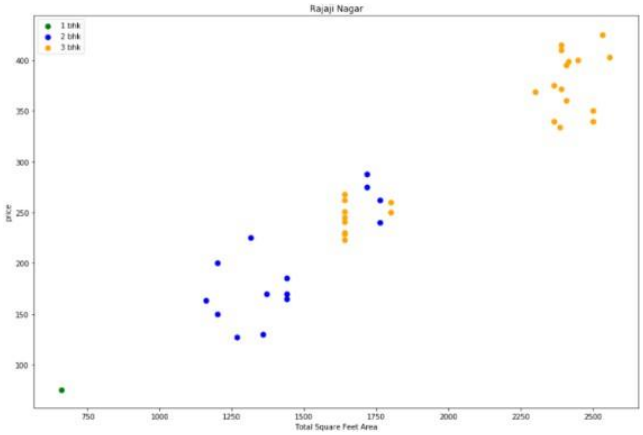


Fig -2 : count vs price per square feet

It could be seen that price per square feet is highly skewed from 5000sqft to 10000sqft. It shows that number of houses having price per square feet is highest in 5000sqft.



This scatter plot shows the relation between price and total sqft in the area "Rajaji Nagar". We can clearly see there is a cluster of BHK having sqft area of about 2500.

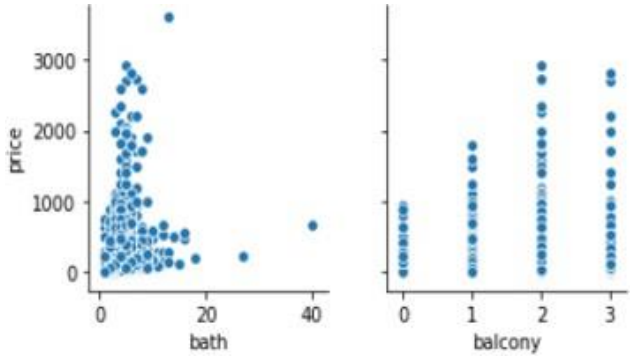


Fig -4: Scatterplot between quantitative variables

The general steps in data pre-processing are:

- Converting categorical features into numerical variables in order to fit Linear Regression model.
- Scaling of data
- Spliting data into train-test sets.

The data preprocessing of each feature in train and test datasets is summarized as below:

- There are around 1305 different locations.

- The null values present in balcony records around 609 data points were imputed with mode (most occurring value) „2" where null values in test data which were around 69 have been imputed with 2.
- The null values present in bath records have been inputed with mode (most occurring value) „2" BHK" in both sets.
- We observe that, all total-sqft records are not in square-feet in both the data sets. Some of them are in square-yards, acres, perch, guntha and grounds.Every data point with respect to total-sqft has been converted into square-feet by carrying out necessary transformations.
- The area_type has four categories: Super built-up area, plot area, carpet area and built-up area. We have converted into dummy variables in both sets.
- The column size has records in BHK, bedroom and RK. The numerical part associated with BHK and Bedroom has been extracted and two separate features BHK and Bedroom have been created. The feature size has been excluded from data.

• Likewise,pre-processing steps in the data test set were performed. All these data preprocessing steps have been carried out in Jupyter notebook python with necessary packages.

3. Results

We have evaluated model's performance using metrics: the coefficient of determination R2 , adjusted R2 and RMSE (Root Means Square Error).

RMSE: It can be defined as the standard sample deviation between the predicted values and the observed ones.

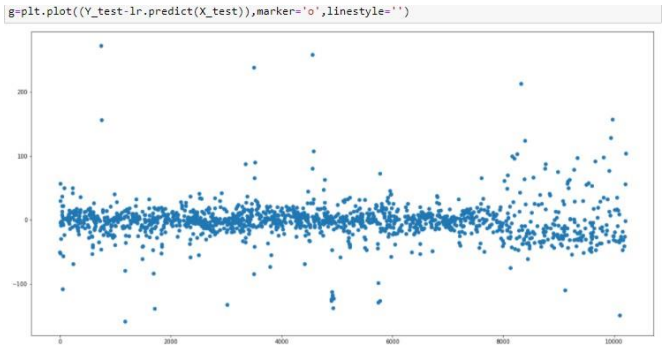
The R-square value provides a measure of how much the model replicates the actual results.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_{predicted} - y_{observed})^2}{\sum_{i=1}^{n} (y_{predicted} - \bar{y}_{observed})^2}$$

The higher the R-squared, the higher the model fitsthe data given. The R-squared value ranges from 0 to 1, representing the percentage of a squared correlation between the target variable's expected and real values. As our dataset is large, we have kept 80% as the training data and 20% as the testing data in our model. For higher accuracy we have removed outliers from our dataset.

3.1 Linear Regression

R-squared value for model developed is **0.8501** for test data and 0.8534 for the training dataset using linear regression. Yet we go for other regression algorithms and machine learning algorithms for better accuracy



By this plot, it is clear that there is very less error between training and testing data. The regression only tries to minimize the error and it does not account for model complexity.

Lasso Regression

LASSO means least absolute shrinkage, and the selection operator is an LR technique that also regularizes functionality. It is identical to ridge regression, except that it varies in the values of regularisation. The absolute values of the sum of regression coefficients are taken into consideration.

R-squared value for model developed R-squared value for model developed is **0.7261** for test data and 0.7054 for the training dataset using Lasso regression.

3.2 SVR(Support Vector Regression)

In simple linear regression we attempt to minimize the error, whereas in SVR we try to fit he error within a certain threshold. It is a regression algorithm and uses a similar Support Vector Machines (SVM) methodology for regression Analysis.

R-squared value for model developed is **0.6246** for test data and 0.5037 for the training dataset using SVR.

3.3 Random Forest

Random forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.
Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features.

R-squared value for model developed is **0.8120** for test data and 0.9608 for the training dataset using Random Forest Algorithm.

3.4 XGBoost Regression Model

XGBoost stands for extreme gradient boosting which is most efficient technique for either regression or classification problem. It is decision tree based algorithm that make use of gradient boosting framework.
It provides the features that greatly have impact on performance of model. This technique helps in developing a model that have less variance and more stability.

R-Squared R-squared value for model developed is **0.8707** for test data using XGBooost.

We have used Hyperparameter optimization for higher accuracy and cross validation for accessing the effectiveness of our models, particularly in cases where we needed to mitigate over fitting. It is also of used in determining the hyper parameters of our model, that result in lowest test error.

We have used Hyper Parameter technique called as RandomizedSearchCV with fold cross validation and defined grid for hyperparameter optimization.

```
n_estimators = [100,500,900,1200,1500]
max_depth =[2,3,5,10,15]
booster=['gbtree','gblinear']
learning_rate=[0.05,0.1,0.01,0.15,0.20]
min_child_weight=[1,2,3,4,5]

#define a grid for hyperparameter optimization
hyperparameter_grid = {
    'n_estimators':n_estimators,
    'max_depth':max_depth,
    'learning_rate':learning_rate,
    'min_child_weight':min_child_weight,
    'booster':booster,
    'base_score':base_score
}
```

RandomSearch generally sets up a grid of hyper parameter values and selects random combinations to train the model and score. "cross_val_score" splits the data into say 5 folds. Then for every fold it fits the info on 4 folds and scores the 5th fold. Then it gives you the 5 scores from which you'll calculate a mean and variance for the score. You crossval to tune parameters and obtain an estimate of the score.

After hyperparameter tuning,

R-squared value for model developed is 0.8888 for test data. Cross validation score for Random Forest algorithm is **0.7772** which is a better than normal score. Because cross_val_score allows us to ascertain how well our model generalizes.

```
#Random Forest algorithm
from sklearn.model_selection import cross_val_score
score=cross_val_score(rfr,X_test,Y_test,cv=10)
score
```

```
array([0.64543543, 0.75759645, 0.84209475, 0.85950464, 0.78707071,
       0.76159369, 0.73585693, 0.85903546, 0.70801556, 0.81675644])
```

This will give us mean of 0.77 and highest score of 0.8594 for correctness of our model.

```
#Lasso regression model
from sklearn.model_selection import cross_val_score
score=cross_val_score(lr_lasso,X_train,Y_train,cv=10)
score
```

```
array([0.77446627, 0.68589333, 0.55014279, 0.67361855, 0.60657671,
       0.75796864, 0.7475168 , 0.6685101 , 0.66110712, 0.76274213])
```

Here, for Lasso regression highest score is 0.7744 but mean score is 0.6888.

4. CONCLUSIONS

An optimal model does not necessarily represent a robust model. Sometimes the data itself might be too noisy or it could contain too few samples to enable a model to accurately capture the target variable which implies that the model remains fit. By above, we have come to one conclusion that hyper parameter optimization does increases the score and reduces difference between actual price and predicted price.
Our model have given accuracy of 88.88% by performing Hyperparameter optimization and using XGBoost Regression model. We can build models through advanced techniques namely neural networks, and particle swarm optimization to improve the accuracy of predictions.

5.ACKNOWLEDGEMENT

We would like to thank and acknowledge our Professor Priyanka Gonnade, Department of CSE who helped in research work. Without her contribution and guidance, this project will not be this successful.

6. REFERENCES

1. Machine Learning introduction. http://medium.com/@ageitge/machine-learning-is-fun-80ea3ec3c471

2. Banglore House Price dataset https://www.kaggle.com/chetantram/bangalore-house-price-analysis

3. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056

4. International Conference on Electronics and Sustainable Communication Systems (ICESC 2020) IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

5. ICIMIA 2020, IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1

6. 2019 IEEE International Conference on Data Mining (ICDM)

7. The 2019 6th International Conference on Systems and Informatics (ICSAI 2019)

8. Proceedings of the 2017 IEEE IEEM IEEM.2017.8289904