© 2019 IJSRSET | Volume 6 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology
DOI : https://doi.org/10.32628/IJSRSET196195

Price Prediction System

Kalyani Wagh, Rinkal Mendhe, Payal Bhoyar, Karishma Siriya, Akhil Anjikar
Information Technology, RGCER, Nagpur, Maharashtra, India

ABSTRACT

Nowadays house prices tend to increase frequently. This is due to the demand for residential sector every year, especially in urban areas. Prediction of house prices is important, especially for property investors and property buyers. While buying a piece of house, a person may be reported prices much higher than their actual values. While selling their house, the person may be reported prices lower than their values. The current house negotiation process in the rural areas involves unauthorized officials carry out the house transactions with traditionally defined parameters that lack clarity, and hence the buyers/sellers are tricked easily. The pricing prediction system uses concepts of machine learning as well as model evaluation and validation techniques to establish a system capable to predict the price of house accurately. The system uses algorithms that consider relevant parameters which affect the suitability of a piece of house, hence ultimately affecting the price. The system is a useful tool for the lesser educated people living in the rural areas who are unaware of the legitimate pricing of house and are tricked into being informed the price much higher that it's actual value.

Keywords : House Price, Python, Machine Learning, Validation Techniques, Notebook Server.

I. INTRODUCTION

The pricing prediction system uses concepts of machine learning as well as model evaluation and validation techniques to establish a system capable to predict the price of house accurately. The system uses algorithms that consider relevant parameters which affect the suitability of a piece of house, hence ultimately affecting the price. The system is a useful tool for the lesser educated people living in the rural areas who are unaware of the legitimate pricing of house and are tricked into being informed the price much higher that it's actual value. It is well considered to make the system a tool which is hassle-free and easy to understand for its users. It uses a Jupyter notebook server. The Jupyter notebook web application is based on a server-client structure. The pricing prediction system aims to provide people with the facility to accurately predict the price of a piece of house. It does so using the concepts of machine learning and where in a set of defined parameters that affect the price of house are considered for the said area, hence generating an intelligent prediction about the value of the house.

One of its prime features is to provide an easy to use interface which the lesser-educated individuals can understand without much trouble. Recognizing the fact that it is this lesser-educated sector of people that are usually the victim of illegitimate transactions which involve inaccurate reporting of house prices, the pricing prediction system aims to eliminate the possibility of such business chicanery.

The system provides an accurate prediction of the price of house. In doing so, it reasons the interested parties for the valuation and hence is self-explanatory of its estimated value.

By the nature of the user-friendly interface of the prediction system, it can be used by people who are less-educated and familiar with digital devices.
It can help in two common rural scenarios:

1. While negotiating for a piece of house to purchase, a person can use the predictor to offer an accurate price for the area in negotiation.
2. While negotiating for a piece of house to be sold, a person can use the predictor to decide the accurate price for the area in negotiation.

Hence in any transaction, the pricing prediction system can be used for a fair negotiation in both the parties

II. RELATED WORK

Price of a house can be influenced by several factors, including economic factors, physical attributes of a house, location, and the concept offered by a house. Some of the economic factors that can affect house prices include: inflation, monetary policy, gross domestic product (GDP), interest rates, and the provision of credit and loans from banks.

Physical attributes are all things possessed by a house which can be observed using the sense of human vision. The existing papers evaluate the performance of the prediction using validation techniques to facility sections facility to accurately predict the price of a piece of house. The age of a house can also be categorized as a physical attribute of a house. Meanwhile, concept is all the ideas offered by the environment where the house is located, which can attract potential buyers. For example, some house offers a minimalist home concept that targets potential buyers who are young families. On the other hand, some house projects offers a healthy environmental concept that provides Green Open Spaces.

Location is also an important factor in determining house prices. Locations may affect prevailing house prices. Location can be a factor affecting house prices because the location offers easy access to some public facilities, such as schools, campuses, hospitals, even family recreational facilities, such as malls or culinary tours.

The data is collected for the actual housing prices in the area, which is verified by checking if the price is legitimate by official standards. This will yield a data set for the system to work on.

The data will be properly split into testing and training subsets, and a suitable performance metric will be determined.

Then we analyze the performance graphs for a learning algorithm with varying parameters and training set sizes which will enable picking the optimal model that best generalizes the unseen data. Finally, this optimal model is tested on a new sample and the predicted selling price is compared to our statistics.

III. ANALYSIS AND MODELLING

A. Data Collection

Property data is available from a variety of private and public sources. The study involves both primary and secondary data. Primary data has been collected through interviews and personal visits to the various companies to know the present situation of the market and the secondary data is collected mainly through various newspapers, magazines, Internet and Reserve Bank of India review. The data between January 1997 and December 2013 are used in the analysis. The data is useful for assessing the performance of property as a key to future investment.

B. Regression technique:

Regression analysis is widely used for prediction and forecasting. Regression analysis is also used to understand which among the independent variables are related to the dependent variable and to explore the forms of these relationships. If more independent variables are added, it is able to determine an estimating equation that describes the relationship with greater accuracy. Multiple regression looks at each individual independent variable and test whether it contributes significantly to the way the regression describes the data.

C. Linear Regression :

A model in statistics which helps predict the future based upon past relationship of variables is a linear regression.

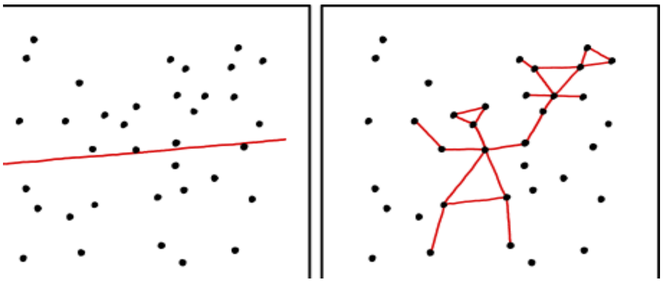


Fig. 1. Linear regression

Regression works on the line equation, y=mx+c. Trend line is set through the data points to predict the outcome. The variable we are predicting is called the criterion variable and the variable we are basing our predictions on is called the predictor variable. When there is only one predictor variable, the prediction method is called Simple Regression .and if multiple predictor variable are present then it is a case of multiple regression.

We use train data and test data. Train data to train our machine and test data to see if it has learnt the data well or not.

D. Libraries and Construction:

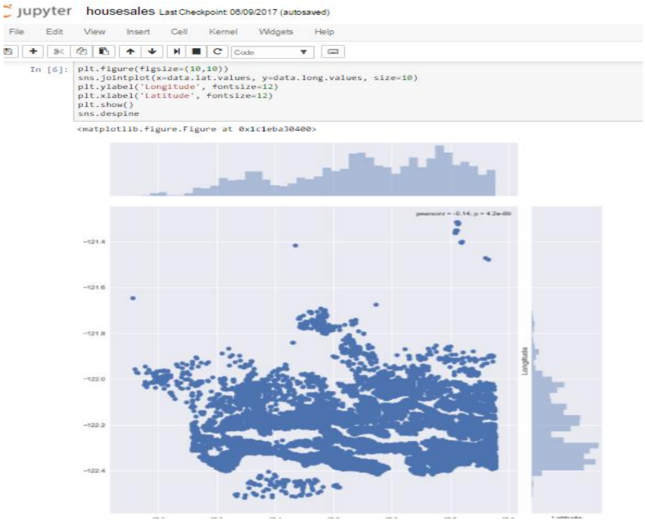
- Dependencies are imported. For linear regression we use sklearn (built in python library) and import linear regression from it.
- First import the library from sklearn
- Then a variable is created where gradient boosting regressor is defined and set parameters to it , here:

1. n_estimator—The number of boosting stages to perform.
2. We should not set it too high which would overfit our model.max_depth—The depth of the tree node.
3. learning_rate—Rate of learning the data

- Initialize Linear Regression to a variable reg.
- It is now known that the prices are to be predicted , hence labels (output) are set as price columns and dates are converted to 1's and 0's so that it doesn't influence the data much .0 is used for houses which are new that is built after 2014.
- Again import another dependency to split the data into train and test.
- The train data has been made as 90% and 10% of the data to be is the test data , and randomized the splitting of data by using random state.
- Hence train, test data and labels for both labels has been acquires as to fit the train and test data into linear regression model.
- After fitting the data to the model the score of data prediction. can be checked.
- For building a prediction model , many experts use **gradient boosting regression ,** a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

E. Visualizing the location of the houses based on latitude and longitude.

- According to the dataset, latitude and longitude on the dataset for each house is present. the common location and how the houses are placed is to be identified.



```
In [6]: plt.figure(figsize=(10,10))
        sns.jointplot(x=data.lat.values, y=data.long.values, size=10)
        plt.ylabel('Longitude', fontsize=12)
        plt.xlabel('Latitude', fontsize=12)
        plt.show()
        sns.despine
        <matplotlib.figure.Figure at 0x1c1eba30400>
```

Fig. 2. Visualization using Sea born

Feature	Client 1	Client 2	Client 3
Total number of rooms in home	5 rooms	4 rooms	8 rooms
Neighbourhood poverty level (as %)	17%	32%	3%

Table I : A sample scenario for house suitability.

- Sea born gets this beautiful visualization. Join plot function helps see the concentration of data and placement of data and can be really useful. The inference of the visualization is that latitude between -47.7 and -48.8 has many houses , which means it is an ideal location.
- Considering a scenario of a real estate agent in the Mumbai area looking to use the model to help price homes owned by clients that they wish to sell. The following data is fetched by the three clie
- The statistics calculated in the Data Exploration section are used to help justify the response. Of the three clients, client 3 has the biggest house, in the best public school neighbourhood with the lowest poverty level; while client 2 has the smallest house, in a neighbourhood with a relatively high poverty rate and not the best public schools.
- Predictions for each client's home.

```
# Produce a matrix for client data
client_data = [[5, 17, 15], # Client 1
              [4, 32, 22], # Client 2
              [8, 3, 12]]  # Client 3

# Show predictions
for i, price in enumerate(reg.predict(client_data)):
   print("Predicted selling price for Client {}'s home: ${:,.2f}".format(i+1, price))
Predicted selling price for Client 1's home: $345,707.55
Predicted selling price for Client 2's home: $260,645.00
Predicted selling price for Client 3's home: $903,393.75
```

The recommendation prices should follow the computed results as above:

Predicted selling price for Client 1's home: $345,707.55

Predicted selling price for Client 2's home: $260,645.00

Predicted selling price for Client 3's home: $903,393.75

Client 3 has largest [Total number of rooms in home] and lowest [Neighborhood poverty level] therefore the price should be the highest one. On the contrary, client 2 has lowest [Total number of rooms in home] and highest [Neighborhood poverty level] and hence the price should be lowest among them.

The prediction is reasonable with the result of C2 < C1 < C3 where Cn stands for nth client.

E. Defining a Performance Metric

It is difficult to measure the quality of a given model without quantifying its performance over training and testing. This is typically done using some type of performance metric, whether it is through calculating some type of error, the goodness of fit, or some other useful measurement. For this project, the calculation of the coefficient of determination for a model is a useful statistic in regression analysis, as it often describes how "good" that model is at making predictions.

The values for R^2 range from 0 to 1, which captures the percentage of squared correlation between the predicted and actual values of the target variable. A model with an R^2 of 0 is no better than a model that always predicts the mean of the target variable, whereas a model with an R^2 of 1 perfectly predicts the target variable. Any value between 0 and 1 indicates what percentage of the target variable, using this model, can be explained by the features. *A* model can be given a negative R^2 as well, which indicates that the model is arbitrarily worse than one that always predicts the mean of the target variable.

For the performance_metric function in the code cell below, the following needs to be implemented

- Use r2_score from sklearn.metrics to perform a performance calculation between y_true and y_predict.
- Assign the performance score to the score variable.

```
from sklearn. metrics import r2_score
def performance_metric(y_true, y_predict):
 """ Calculates and returns the performance score between
 true and predicted values based on the metric chosen.
"""
# TODO: Calculate the performance score between 'y_true' and 'y_predict'
 score = r2_score(y_true, y_predict)

# Return the score
   return score
```

variable:

Assume that a dataset contains five data points and a model made the following predictions for the target variable:

F. Calculation of Statistics:

As the primary implementation, the descriptive price predictions is used for the city of Nagpur. Since numpy has already been imported to use this library to perform the necessary calculations, These statistics will be extremely important later on to analyse various prediction results from the constructed model. We will need to implement the following:

1. Calculate the minimum, maximum, mean, median, and standard deviation of 'MEDV', which is stored in prices.
2. Store each calculation in their respective variable.

G. Goodness of Fit

Assume that a dataset contains five data points and a model made the following predictions for the target variable:

Table II: True value prediction.

True Value	Prediction
3.0	**2.5**
-0.5	**0.0**
2.0	**2.1**
7.0	**7.8**
4.2	**5.3**

```
# Calculate the performance of this model
score = performance_metric([3, -0.5, 2, 7, 4.2], [2.5, 0.0, 2.1, 7.8, 5.3])
print("Model has a coefficient of determination, R^2, of {:.3f}.".format(score))
```

H. Fitting a Model

- The final implementation requires that everything is brought together and training a model using the **decision tree algorithm**. To ensure that an optimized model is produced, the model is trained using the grid search technique to optimize the 'max_depth' parameter for the decision tree. The 'max_depth' parameter can be thought of as how many questions the decision tree algorithm is allowed to ask about the data before making a prediction..

1. Use Decision Tree Regressor from sklearn.tree to create a decision tree regressor object.
2. Assign this object to the 'regressor' variable.
3. Create a dictionary for 'max_depth' with the values from 1 to 10, and assign this to the 'params' variable.
4. Use make_scorer from sklearn metrics to create a scoring function object.
5. Pass the performance_metric function as a parameter to the object.
6. Assign this scoring function to the 'scoring_fnc' variable.
7. Use GridSearchCV from sklearn.grid_search to create a grid search object.
8. Pass the variables 'regressor', 'params', 'scoring_fnc', and 'cv_sets' as parameters to the object.
9. Assign the GridSearchCV object to the 'grid' variable.

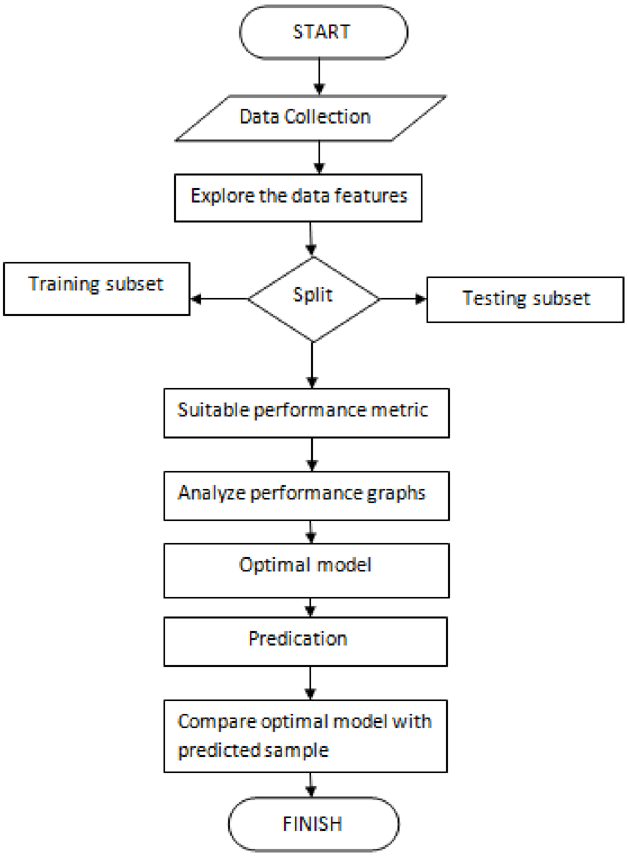

START
Data Collection
Explore the data features
Training subset
Split
Testing subset
Suitable performance metric
Analyze performance graphs
Optimal model
Predication
Compare optimal model with predicted sample
FINISH

Fig. 3. Working of prediction Process

IV. VISUALIZATION

- Below output shows the training and testing of prediction with maximum depth variance.

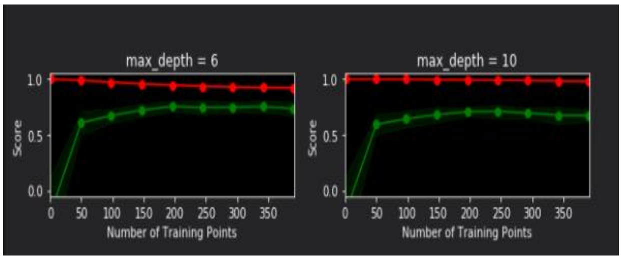
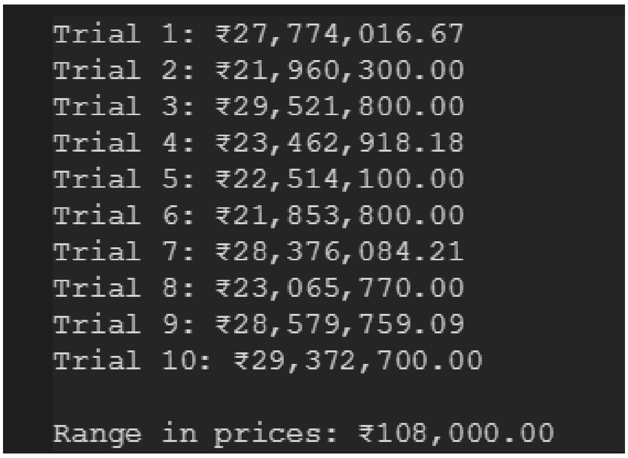


Fig. 4. Train and test visual.

- Test and train data with respect to maximum depth

Fig. 5. The parameters.

- First trial of prediction with client's requirement prediction system gives a variance in many prediction and try to be give a accurate value of the housing price



```
Trial 1: ₹27,774,016.67
Trial 2: ₹21,960,300.00
Trial 3: ₹29,521,800.00
Trial 4: ₹23,462,918.18
Trial 5: ₹22,514,100.00
Trial 6: ₹21,853,800.00
Trial 7: ₹28,376,084.21
Trial 8: ₹23,065,770.00
Trial 9: ₹28,579,759.09
Trial 10: ₹29,372,700.00

Range in prices: ₹108,000.00
```

Fig. 6. The output values as predicted.

V. CONCLUSION

In this paper, several tests have been performed using linear regression and particle swarm optimization methods to perform house price prediction. In this project basic concepts are used on data collected for housing prices in a Nagpur,Maharashtra area to predict the selling price of a new home. Firstly data is explored to obtain important features and descriptive statistics about the dataset. Next, the data is split into testing and training subsets,and a suitable performance metric is determined for the problem. Then the performance graphs are analysed for a learning algorithm with varying parameters and training set sizes. These enable to pick the optimal model that based generalizes for unseen data. Finally, this optimal model is tested on a new sample and compared for the predicted selling price to our statistics.

VI. FUTURE SCOPE

For further research, the prediction system can be optimized by using evolutionary algorithms or machine learning to determine the boundaries of the membership function or the selection of rules to achieve better accuracy. In addition, the number of each object variable can be included in the calculation of house prices by using hedonic pricing method. There are quite a few things that can be polished or add in the future work.

Though,we are able to idetify most of the residential areas.There may be some more places that have housing complexes,multi-story apartments which are located in coomercial areas.Such apartments were not included in this paper and can be counted in future to give a more accurate result.with more and more demand for housing in metropolitan cities,ther eis definate increase in the number of private builders that provide real estate with additional amenitis to attract more customers.There are several other models available that can be implemented for prediction.Data given as input to such model should be compatible with the tool used and the operators involved in the process.Also,more number of dataset can be used to increase the accuracy of the model.The main Objective of using a different model should be to reduce the calculation time and carry out the whole process in ease.

VII. ACKNOWLEDGEMENT

This research was supported by our project guide **Mr.Akhil Anjikar** (Assistant Professor), We thank him for providing us an opportunity to do the project research on Pricing Pridiction System at Rajiv Gandhi College Of Engineering & Research (RGCER), Hingna,Nagpur and by giving us the support and guidence.

VIII. REFERENCES

[1]. Lipo Wang; Fung Foong Chan; Yaoli Wang; Qing Chang "Predicting public housing prices using delayed neural networks" 2016; 2016 IEEE Region 10 Conference (TENCON).

[2]. Data-driven Fuzzy Rule Extraction for Housing Price Prediction in Malang, East Java.

[3]. https://rapidminer.com/resource/correct-model-validation/

[4]. Yulin Wang; Jiaxi Han; Xuhang Zhang; Wei Luo "Pricing research of 'photo-earning' task based on logistic regression and BP neural network" 2018; 2018 Chinese Control And Decision Conference (CCDC).

[5]. Wan Teng Lim; Lipo Wang; Yaoli Wang; Qing Chang "Housing price prediction using neural networks" 2016; 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).

[6]. Yingyu Feng; Kelvyn Jones "Comparing multilevel modelling and artificial neural networks in house price prediction" 2015; 2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM).

[7]. Muhammad Fahmi Mukhlishin; Ragil Saputra; Adi Wibowo "Predicting house sale price using fuzzy logic, Artificial Neural Network and K-Nearest Neighbor" 2017 2017 1st International Conference on Informatics and Computational Sciences (ICICoS).

[8]. Seçkin Karasu; Aytaç Altan; Zehra Saraç; Rifat Hacioğlu "Prediction of Bitcoin prices with machine learning methods using time series data" 2018; 2018 26th Signal Processing and Communications Applications Conference (SIU).

[9]. Yingyu Feng; Kelvyn Jones "Comparing multilevel modelling and artificial neural networks in house price prediction" 2015; 2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM).

[10]. Lipo Wang; Fung Foong Chan; Yaoli Wang; Qing Chang "Predicting public housing prices using delayed neural networks" 2016; 2016 IEEE Region 10 Conference (TENCON).

Cite this article as :

Kalyani Wagh, Rinkal Mendhe, Payal Bhoyar, Karishma Siriya, Akhil Anjikar, "Price Prediction System", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), ISSN : 2456-3307, Volume 6 Issue 2, pp. 36-43, March-April 2019. Available at doi : https://doi.org/10.32628/IJSRSET196195
Journal URL : http://ijsrset.com/IJSRSET196195