Proceedings of the International Conference on Inventive Computation Technologies (ICICT-2018)
IEEE Xplore Part Number: CFP18F70-ART; ISBN:978-1-5386-4985-5

Comparison of Ensemble Methods for Real Estate Appraisal

Prathamesh Kumkar, Ishan Madan, Ashutosh Kale, Omkaar Khanvilkar
Student, Department of Computer Engineering
Thadomal Shahani Engineering College
Mumbai, India

Mr Aejazul Khan
Assistant Professor, Department of Computer Engineering
Thadomal Shahani Engineering College
Mumbai, India

***Abstract*—Real estate appraisal is necessary for a range of endeavours, including property insurance, investment analysis, taxation and sales listing. Along with prospective buyers and sellers, estimating the prices of real estates can benefit investors as well. This paper makes a comparative analysis of four ensemble methods, viz. Bootstrap Aggregating, Random Forest, Gradient Boosting and Extreme Gradient Boosting for real estate appraisal. The comparison is based on estimating real estate prices in Mumbai. The data for this study was obtained by web scraping from the real estate website 99acres. The collected data was utilized for training and testing the above-mentioned ensemble models and comparing their performances. Grid search was used for fine-tuning the hyperparameters of the learning models. Also, preprocessing techniques such as dimensionality reduction and one-hot encoding were used for improving the accuracy of the models. Analysis of the results shows that using ensemble methods for estimating real estate prices is a realistic prospect.**

Keywords—ensemble methods; bootstrap aggregating; random forest; gradient boosting; extreme gradient boosting; real estate appraisal

I. INTRODUCTION

Real estate appraisal is an integral part of the property buying process. Traditionally, the appraisal is performed by professional appraisers specially trained for real estate valuation. For the buyers of real estate properties, an automated price estimation system can be useful to estimate prices of properties currently on the market. Such a system can be particularly helpful for novice buyers who are buying a property for the first time, with little to no experience.

The real estate website 99acres.com has over 45 thousand listings for residential properties in Mumbai. The prices of properties are dependent on many variables such as the location of the property, area, number of bedrooms, type of furnishing, hence establishing the need for developing an estimation model. A variety of approaches for estimating real estate prices have been developed over the years. These range from Multiple Regression Analysis [1] through Artificial Neural Networks [2, 3], Fuzzy Logic [4] and hybrid techniques including ANFIS [5] and Fuzzy Neural Networks [6]. Among these approaches, ensemble methods are found to perform consistently well [7].

In this paper, four ensemble methods: Bootstrap Aggregating, Random Forest, Gradient Boosting and Extreme Gradient Boosting are compared based on their effectiveness in estimating the real estate prices in Mumbai. First, the ensemble methods that were used in this study are described. Then, the data cleaning and preprocessing activities are discussed along with hyperparameter tuning for the models. Finally, we present our results followed by the conclusions that can be drawn from them.

II. ENSEMBLE METHODS

Ensemble methods is a technique in machine learning which produces an optimal predictive model by combining several base models [8]. In this technique, the overall accuracy of prediction is improved by combining the results of various individual models developed using a particular machine learning algorithm. An overview of the working of ensemble methods is shown in Fig. 1.

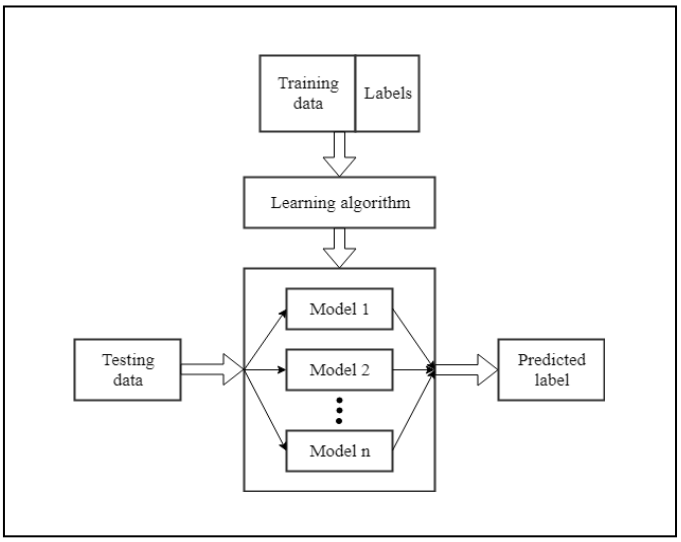

Fig. 1. The basic working of ensemble methods

Ensemble methods are broadly classified into two categories:

1) Averaging: The methods in this category build several models independently, and the predictions of each of

the models are then averaged with an aim to reduce the variance of the combined model. Variance quantifies how the predictions made on the same observation are different from each other. A model with a high variance will overfit on the training data and perform poorly on any new observations beyond the training set.

2) Boosting: The idea behind boosting is to sequentially build several weak models to produce a powerful resultant model eventually. As a result, the bias of the combined model is reduced. Bias quantifies how much the predicted values are distinct from the actual values. A model with a high bias can be said to be underperforming and failing to capture essential trends in the data.

A. Bootstrap Aggregating (Bagging)

Bagging is a category of algorithms that develop multiple individual estimators on the original data by taking a random subsample of the data each time [9]. The average of all the predictions from different estimators is used which is more robust than each of the separate estimators. In bagging, each estimator in the ensemble has the same weight as that of the others. These methods reduce the variance of the estimators and also overcome overfitting by introducing randomization in the way each estimator is constructed. Thus, bagging provides an efficient way to enhance the effectiveness of an elementary model.

B. Random Forest (RF)

A random forest is an aggregation of multiple decision trees [10]. It is an extension over Bagging, in which, in addition to taking a random subset of the training set to build a decision tree, it also takes a random subset of features rather than using all the features to build the trees. Due to this, the bias of the entire forest slightly increases as compared to a single decision tree. However, as the predictions of all the decision trees are averaged to produce the final result (as in Bagging), the variance of the forest reduces which compensates for the increased bias.

C. Gradient Boosting (GB)

Boosting is a group of algorithms in ensemble strategy that consecutively builds on weak learners in order to generate one final strong learner. A weak learner is a model that may not be very accurate or may not take many estimators into account. Unlike Bagging that builds each model independently and then aggregates the predictions of the models at the end without giving preference to any model, in Boosting, each model that is built dictates on what features the next model will focus. Boosting can effectively convert weak learners into a strong learner by building a weak model, making conclusions about the importance of various features and parameters, and then using those conclusions to build a new, stronger model.

Gradient Boosting is a generalization of Boosting and aims to reduce the error with each consecutive model until one final model is produced [11, 12]. In GB, a new weak learning model is added in each phase, and the previous models also are left unchanged. GB has three main components:

1) Loss function: The loss function must be differentiable and is dependent on the problem. The GB framework is general, and there is no need to develop a new boosting model for every loss function used. Equation (1) shows the least squared error loss function used for this study.

$$\mathrm{L} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (1)$$

n : Total number of observations in the training set
y_i : Actual Sale Price for the i^{th} observation
$\hat{y}_i$: Predicted Sale Price for the i^{th} observation

2) Weak Learning models: GB uses decision trees as weak learning models. To minimize the loss, a greedy approach is used to construct the trees. In order to constrain the weak learning models, different parameters such as the maximum number of levels, leaf nodes and splits are used.

3) Additive model: During each step, trees are added one at a time without changing the existing trees. In order to minimize the loss while adding trees, gradient descent is used. The loss is reduced by varying tree parameters and moving in the direction of lesser residual loss. In order to improve the final output, the newly calculated output is then added to the existing sequence of outputs. The training process is stopped when the loss achieves the desired minimum value or shows no improvement in the existing validation set.

D. Extreme Gradient Boosting (XGBoost)

XGBoost aims to improve the scalability and accuracy of Gradient Boosting [13]. It makes maximum utilization of computing power through parallel processing and built-in cross-validation. It is designed in such a way that it reduces the computing time and ensures optimal use of memory and hardware resources. XGBoost also permits users to define their optimization objective and evaluation criteria. The three primary forms of gradient boosting: GB, Stochastic GB and Regularized GB, can be implemented using XGBoost. The main feature that makes XGBoost superior and unique compared to other algorithms is the inclusion of regularization. The objective function of XGBoost comprises of two parts: the loss function and the regularization function as shown in (2).

$$\mathrm{obj} = \sum_{i=1}^{n} \mathrm{L}\left(y_i\, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_i) \quad (2)$$

n : Total number of observations in the training set
t : Total number of steps required for optimization
y_i : Actual Sale Price for the i^{th} observation
$\hat{y}_i$: Predicted Sale Price for the i^{th} observation
L : Training loss function
Ω : Regularization term

III. METHODOLOGY

Fig. 2 shows an overview of the methodology adopted for estimating the prices of real estate properties using the various ensemble models.

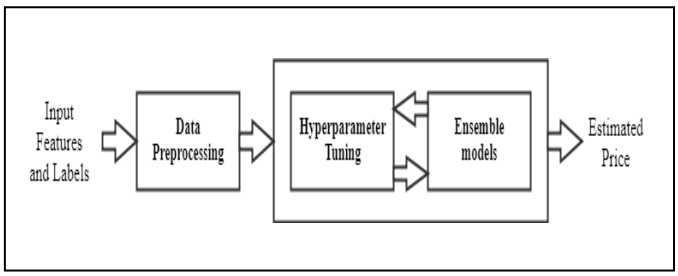

Fig. 2. Estimating real estate prices using ensemble models

A. Data

The analysis was done using data acquired from the 99acres dataset which included 45297 property listings for Mumbai. As shown in Table I, the initial dataset consisted of thirteen attributes.

TABLE I. DEFINITION OF ATTRIBUTES IN THE DATASET

Attribute	Definition
Sale Price	The selling price of the property (Float; lakhs of INR)
Carpet area	The total usable floor area (Float; square feet)
Built-up area	The carpet area plus the areas covered by inside and outside walls and balconies (Float; square feet)
Super built-up area	The built-up area plus a portion of areas such as lobbies, staircases, etc. (Float; square feet)
Location	The locality of the house within Mumbai (String)
No. of bedrooms	The number of bedrooms (Integer)
No. of bathrooms	The number of bathrooms (Integer)
No. of balconies	The number of balconies (Integer)
No. of parking slots	The number of parking slots (Integer)
Floor number	The floor number of the property (Integer)
Furnishing type	The type of furnishing (String; furnished, semi-furnished, unfurnished)
Flooring type	The type of flooring (String; marble, ceramic, vitrified, mosaic, cement, wood, granite, vinyl, stone)
Property type	The type of property (String; residential apartment, independent house or villa, independent or builder floor, farmhouse)

The dataset was cleaned by removing any observation that had missing values for any of the attributes. Further, outliers were eliminated from the training data. For example, listings that had Locations with a frequency count of less than ten were removed. Table II shows the correlation coefficients of different numeric attributes concerning Sale Price.

TABLE II. CORRELATION COEFFICIENTS WITH RESPECT TO SALE PRICE

Attribute	Correlation coefficient
Carpet area	0.813459
Built-up area	0.791755
Super built-up area	0.831567
No. of bedrooms	0.640829
No. of bathrooms	0.676433
No. of balconies	0.014330
No. of parking slots	0.221469
Floor number	0.390168

As seen in Table II, the attributes 'Super built-up area', 'Built-up area' and 'Carpet area' are strongly correlated with Sale Price. However, the Real Estate (Regulation and Development) Act, 2016, made it mandatory that the Sale Price should be quoted based on the Carpet Area. Hence, only Carpet area was used for the analysis. Also, since the attribute 'No. of balconies' has a minimal correlation coefficient, it was not taken into account for the analysis. Thus, the features considered for the study included Location, Carpet area, Number of bedrooms and bathrooms, Number of parking slots, Floor number, Furnishing type, Flooring Type and Property type.

The cleaned dataset included 16194 observations out of the initial 45297 observations. The cleaned dataset was then used to form the training and the testing sets for evaluating the various models. The testing set was randomly selected from this dataset and included 3239, i.e., 20 per cent of the total observations. The remaining 12955, i.e., 80 per cent of the observations constituted the training set.

B. Preprocessing of data

The dataset considered in this paper contains categorical features like Location, Property type, Furnishing type and Flooring type which can have string values. For example, the 'Location' feature can have values such as 'Ghatkopar', 'Santacruz', 'Byculla' and so on. All the categorical features in the dataset were converted into binary ones with values 1 for true and 0 for false. For example, if a property had 'Location' as 'Ghatkopar', then the sub-feature (a new feature created) 'Location_Ghatkopar' would have a value of 1, while the remaining sub-features like 'Location_Santacruz', 'Location_Byculla', etc. would each have a value of 0. Further, to increase the variance between the features and to reduce the number of dimensions used to train the models, Principal Component Analysis was used.

C. Hyperparameter Tuning

Hyperparameters are those parameters of an algorithm whose values cannot be estimated from the data and need to be set before training a model with the given algorithm. Grid search was implemented to adjust the hyperparameters of the models. Grid search exhaustively considers all possible combinations of the hyperparameters given to it as the input, along with the training dataset, and builds a model using each

of these hyperparameter combinations. Finally, it returns the set of parameters that produced the best model which can be further used for testing the model using the testing dataset.

D. Evaluation Criterion

Mean Absolute Percentage Error (MAPE) was used to compare the relative performance of the models. It quantifies the error in terms of percentage and is defined by (3).

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3)$$

n : Total number of observations in the training set
y_i : Actual Sale Price for the i^{th} observation
$\hat{y}_i$: Predicted Sale Price for the i^{th} observation

MAPE was used as the evaluation criterion as it is relatively intuitive and it is easily interpreted in terms of relative error. Moreover, it is found to scale well with large datasets and does not suffer due to the large volume of the dataset.

IV. Experimental Results

The training and testing datasets were used to evaluate the aforementioned ensemble models. Before training any of the models, the different preprocessing techniques described before were applied to the data. The models were built on the training dataset and then evaluated based on their accuracy in estimating the prices for observations in the testing dataset. Fig. 3 shows each model and its corresponding MAPE.

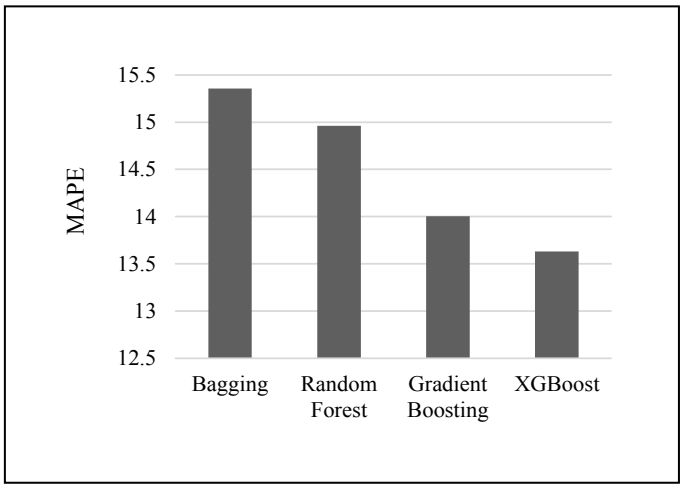

Fig. 3. MAPE for the ensemble models

It can be seen that XGBoost had the lowest MAPE, i.e., 13.63% and performed the best among the four ensemble models. This can be attributed to the addition of the regularization term in the objective function of the XGBoost model. Also, XGBoost can learn efficient structures as compared to other algorithms because it uses Newton boosting. In addition to this, it performs column subsampling, which introduces more randomization to reduce correlation further. Gradient Boosting, Random Forest and Bagging models followed XGBoost with a MAPE value of 14.00%, 14.96% and 15.36% respectively. The Bagging model had the lowest performance which was expected as it just aims to minimize the variance without improving the model's predictive capability.

V. Conclusion

In this paper, four ensemble methods, namely Bagging, Random Forest, Gradient Boosting and Extreme Gradient Boosting were analysed and compared in terms of their efficiency in the appraisal of real estates in Mumbai. The property listings available on the real estate website 99acres were used as the data source for this study. The features of the properties used for estimating the prices were Location, Carpet area, Number of bedrooms and bathrooms, Number of parking slots, Floor number, Furnishing type, Flooring Type and Property type. Mean Absolute Percentage Error was the performance measure employed for comparing the models. The analysis showed that Extreme Gradient Boosting (XGBoost) model performed the best as compared to the rest of the ensemble models. The results confirm that ensemble models can be useful for estimating real estate prices.

References

[1] J. Zurada, A. Levitan, and J. Guan, "A comparison of regression and artificial intelligence methods in a mass appraisal context," Journal of Real Estate Research, vol. 33, no. 3, pp. 349-387, 2011.

[2] S. Peterson and A.B. Flanagan, "Neural network hedonic pricing models in mass real estate appraisal," Journal of Real Estate Research, vol. 31, no. 2, pp. 147-164, 2009.

[3] H. Selim, "Determinants of house prices in Turkey: Hedonic regression versus artificial neural network," Expert Systems with Applications, vol. 36, no. 2, pp. 2843-2852, 2009.

[4] C. Bagnoli and H. Smith, "The theory of fuzzy logic and its application to real estate valuation," Journal of Real Estate Research, vol. 16, no. 2, pp. 169-200, 2009.

[5] J. Guan, J. Zurada, and A. Levitan, "An adaptive neuro-fuzzy inference system based approach to real estate property assessment," Journal of Real Estate Research, vol. 30, no. 4, pp. 395-422, 2008.

[6] Z. X. li, "Using fuzzy neural network in real estate prices prediction," 2007 Chinese Control Conference, Hunan, pp. 399-402, 2007.

[7] E.A. Antipov and E.B. Pokryshevskaya, "Mass appraisal of residential apartments: An application of random forest for valuation and a CART-based approach for model diagnostics," Expert Systems with Applications, vol. 39, no. 2, pp. 1772-1778, 2012.

[8] Zhi-Hua Zhou, *Ensemble methods: foundations and algorithms*, Boca Raton, FL: Taylor & Francis, 2012.

[9] L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123-140, 1996.

[10] T.K. Ho, "The random subspace method for constructing decision forests," IEEE transactions on pattern analysis and machine intelligence, vol. 20, no. 8, pp. 832-844, 1998.

[11] J.H. Friedman, "Greedy function approximation: a gradient boosting machine," The Annals of Statistics, vol. 29, no. 5, pp. 1189-232, 2001.

[12] J.H. Friedman, "Stochastic gradient boosting," computational statistics & data analysis, vol. 38, no. 4, pp. 367-378, 2002.

[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.