Ensemble Learning Based Housing Price Prediction Model

Conference Paper · November 2018

2 authors:

Bowen Yang
Tongji University

Buyang Cao
Tongji University

Ensemble Learning Based Housing Price Prediction Model

Bowen Yang

Tongji Univeristy
School of Software Engineering,
Shanghai, China
bowenyang@yeah.net

Buyang Cao

Tongji Univeristy
School of Software Engineering,
Shanghai, China
Caobuyang@tongji.edu.cn

***Abstract*---Housing price is influenced by multiple factors· The most existing housing price forecasting models usualyl belong to so called single predictor model, whose prediction accuracy is not ideal and the over-fitting phenomenon happens often due to the data noise. To resolve these issues, we propose an ensembe lerning based housing price prediction model, which incorporates various predictors. To evaluate the effectiveness of the proposed model, extra trees, random forest, GBDT and XGB algorithms are selected for the benchmarks. The dataset used is the California housing price available over the web. The results demonstrate that the proposed method can improve the predicting accuracy and stability comparing to these four single prediction models.**

I. INTRODUCTION

Real estate is not only the key sector of the national economy, but also one of the citizen's major concerns. Due to the housing demands, people's attention to the housing price continues increasing. It is critical to provide accurate predictions of housing prices. Housing price is impacted by multiple factors ([2], [10]) including time and space, house ages, surrounding conditions, communities, transportation, etc. Existing prediction models are usually single predictor ones, i.e., a single forecasting model is applied for the prediction. The prediction accuracy of this type model is not satisfactory when datasets are noisy [4]. Some simple ensemble models such as random forest would encounter over-fitting phenomenon when data contains more noise. To address these issues, we propose an ensemble learning ([1], [11]) based housing price prediction model in this paper. The model is built upon multiple single predictors (they will be called *base predictors* in the following discussions) including random forest (RF), extra trees (ET), GBDT, and XGB.

Random forest [7], whose basic unit is a decision tree, is an ensemble algorithm/model employing multiple trees. It shows its superiority in many application areas. It is capable of handling high dimensional data without feature selection. It can get an unbiased estimation of the internal generation error during the forest generating process, and the generalization capability is good. Nevertheless, random forest may suffer overfitting in some classification or regression problems where noise occurs often.

Extra trees, also known as Extremely Randomized Trees, is the combination of decision trees. Similar to the random forest, it randomly selects partial features to construct a tree. Extra trees directly use training samples to construct random trees and modify the way of bagging. Therefore, when data is noisier or the dataset is large, this methodology performs better than the standard random forest. However, due to more randomly sampled data, some selections are not satisfactory and the quality of prediction results fluctuates greatly.

GBDT (Gradient Boosting Decision Tree) [8] is an iterative decision tree algorithm. This algorithm consists of multiple decision trees whose conclusions form the final answer, and GBDT is considered to have a strong generalization capability. The core of a GBDT is composed by regression trees. Therefore, most GBDTs are used for regression predictions. Although a GDBT does not need to perform complex feature engineering and transformation, it is not quite suitable for the problems with high-dimensional features.

XGBoost (XGBT) [6] is an open-source software library including the gradient boosting framework aiming to provide a "scalable, portable and distributed gradient boosting library". Other than running on a single machine, it also supports the distributed frameworks. Using XGBoost, we can train models more efficiently and get better prediction results.

Based on the above discussions, the existing methods/predictors cannot provide very satisfactory and stable prediction results if they are applied individually. They are impacted negatively by the noise in datasets. Hence, we will develop an ensemble learning based prediction model by incorporating the above mentioned four predictors attempting to obtain better prediction outcomes. The procedure of creating the proposed model or method may be described briefly as follows.

The stacking method ([5], [9], [11]) of ensemble learning is applied to construct our proposed model. It first partitions the data sets (see the details in the following sections), and then uses each base predictor to conduct the predictions based on the extract features related to housing prices. Specifically, the first part of the dataset is used for training, and the second part is employed for testing these base predictors. At the end, taking the testing results as the inputs, the high-level (ensemble) model is finally trained as our prediction model.

In the next section we are going to present our model, the training process, and some computational experiments. The paper concludes with remarks.

II. Ensemble learning based prediction model

The following picture (figure 1) depicts the overall model training and ensembling process for the proposed model, where the dataset is the California housing price data.

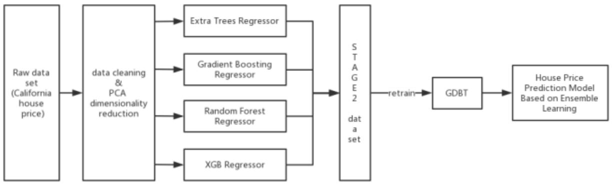

Figure 1. Ensemble training process of the proposed model

A. *Data processing*

This paper uses California housing price data for training and evaluating the model. According to the analysis of the original dataset, it contains numeric and non-numeric features and some records miss data contents. The dataset needs to be cleaned before being used for training and testing. We analyzed the original dataset and found that the feature 'ocean_proximity' is expressed with String format, it's not a friendly way for machine learning, so we decided to transform the ocean_proximity feature data to numeric data. The ocean_proximity data contains five types data such as NEAR BAY, INLAND, NEAR OCEAN.etc. So we add five features depend on this five data type into the original dataset and mark the corresponding data as 1, the others as 0, for example, if the house is near bar, we mark the Ocean_proximity_NEAR_BAR as 1, the other four features as 0, the detail refer TABLE I and TABLE II. Finally, the non-numeric data of the selected features is transformed to numeric ones and the missing data will be replaced properly. The samples of both original and processed data are shown in the follows tables respectively:

TABLE I. ORIGINAL DATA SAMPLE

longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
-122.23	37.88	41	880	129	322	126	8.3252	452600	NEAR BAY
-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500	NEAR BAY
-122.24	37.85	52	1467	190	496	177	7.2574	352100	NEAR BAY
-122.25	37.85	52	1274	235	558	219	5.6431	341300	NEAR BAY
-122.25	37.85	52	1627	280	565	259	3.8462	342200	NEAR BAY
-122.25	37.85	52	919	213	413	193	4.0368	269700	NEAR BAY
-122.25	37.84	52	2535	489	1094	514	3.6591	299200	NEAR BAY
-122.25	37.84	52	3104	687	1157	647	3.12	241400	NEAR BAY
-122.26	37.84	42	2555	665	1206	595	2.0804	226700	NEAR BAY

TABLE II. PROCESSED DATA SAMPLE

	longitude	latitude	housing_me	total_rooms	total_bedroc	population	households	median_incor	median_h	Ocean_proxin	Ocean_proximity_	Ocean_proximity_ISI	Ocean_proximity_N	Ocean_proximity_NEAR OCEAN
0	-122.23	37.88	41	880	129	322	126	8.3252	452600	0	0	0	1	0
1	-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500	0	0	0	1	0
2	-122.24	37.85	52	1467	190	496	177	7.2574	352100	0	0	0	1	0
3	-122.25	37.85	52	1274	235	558	219	5.6431	341300	0	0	0	1	0
4	-122.25	37.85	52	1627	280	565	259	3.8462	342200	0	0	0	1	0
5	-122.25	37.85	52	919	213	413	193	4.0368	269700	0	0	0	1	0
6	-122.25	37.84	52	2535	489	1094	514	3.6591	299200	0	0	0	1	0
7	-122.25	37.84	52	3104	687	1157	647	3.12	241400	0	0	0	1	0
8	-122.26	37.84	42	2555	665	1206	595	2.0804	226700	0	0	0	1	0
9	-122.25	37.84	52	3549	707	1551	714	3.6912	261100	0	0	0	1	0
10	-122.26	37.85	52	2202	434	910	402	3.2031	281500	0	0	0	1	0
11	-122.26	37.85	52	3503	752	1504	734	3.2705	241800	0	0	0	1	0

Each record of the dataset contains 15 attributes, and the dataset consists of 20,000 records. We construct a sample matrix (20,000 x 15) $X = [x_1, x_2, ..., x_n]^T$ based on the number of records and features where n is the number of records or samples. However, some features are not necessarily contribute to the variable, i.e., the housing price, to be predicted and they even act as noise. Therefore, we apply the principal component analysis (PCA) method to reduce the dimension for better results. Particularly, The Karhunen-Loeve Transform (KLT) [12] is applied to perform the PCA task. A new sample matrix or data collection L will be generated after the PCA process. In the following training and testing procedures we are using the dataset L after the dimensional reduction by PCA.

B. *Training base predictors*

As we mentioned earlier, the ensemble learning based prediction model proposed in this paper is created based on the stacking ensemble learning method ([5], [11]), where ET, RF, GBDT, and XGB are base predictors. In this case we need first to train these selected base predictors. For each base predictor, we apply the associated regression model and use the new sample data L after the dimensional reduction described above as the input training data. As usual, the dataset is divided into two parts, whereas 99.5% of it is for training and 0.5% of it is used for testing. During the testing process the model parameters are adjusted to achieve the more satisfactory results of the underlying models. The parameters to be set for the individual base predictors are listed as follows:

max_features: max features allowed to use in each predictor;

n_estimators: the number of trees of each predictor;

colsample_bytree: specifying the number of columns per random sample

max_depth: the max depth of a node in one tree of a predictor

subsample: the ratio of the input data to be sampled.

TABLE III. MODEL PARAMETERS AFTER TRAINING

ET	max_features=6	n_estimators=100		
RF	max_features=6	n_estimators=100		
GBDT	max_features=12	n_estimators=500	max_depth=8	subsample=0.8
XGB	max_features=12	n_estimators=200 Colsample_bytree=0.8	max_depth=8	subsample=0.8

After having tuned the parameters, the resultant models/predictors can be used for conducting predictions. In figure 2 we present the prediction results of all four base predictors and the (sampled) real housing prices for comparisons.

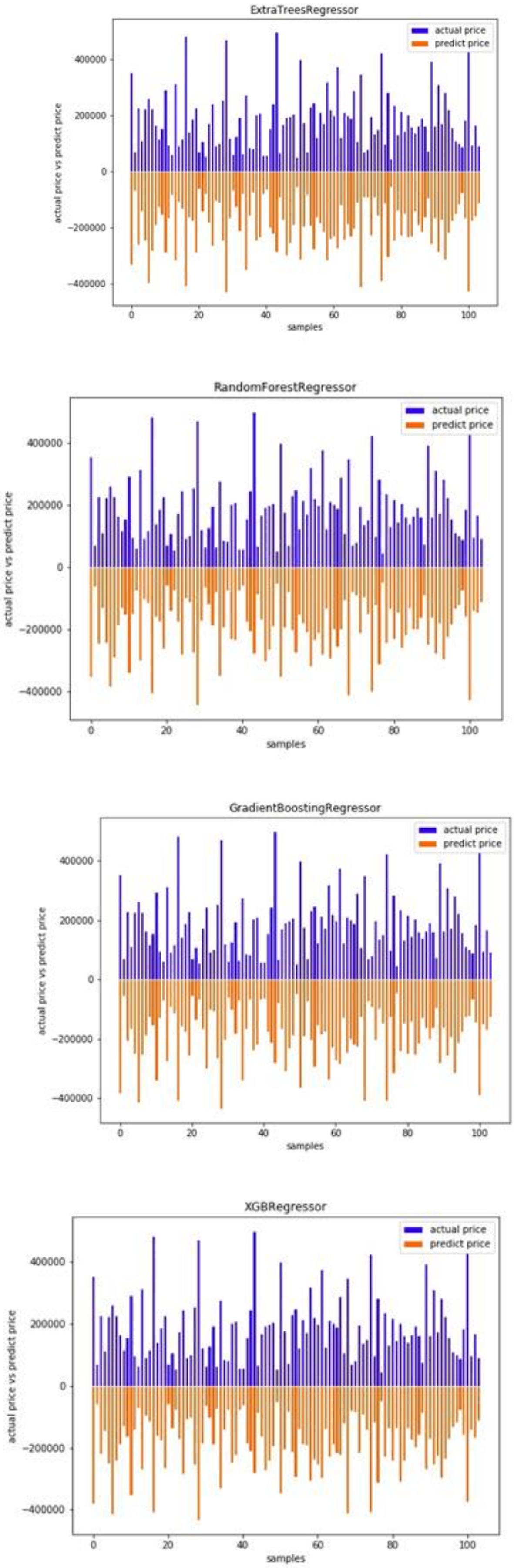

Figure 2. The results obtained by four basic predictors

The following table (table 4) lists the loss function values measured by mean square errors, which can be applied to evaluate the performance of each predictor.

TABLE IV. MEAN SQAURE ERRORS OF FOUR BASE MODELS

Predict model	Mean Squared Error
Extra Trees	44216.900081
Random Forest	44625.093537
Gradient Boosting	43764.930335
XGB	43279.231065

Based on the outcomes, it is not difficult to find out that both GBDT and XGB produce better prediction results than ET and RF do in terms of MSE (mean sqaure error). In the next section, we will present more details on how to apply these four base predictors to construct an ensemble model. Further computational experiments and benchmarks are carried out to evaluate the effectiveness/performance of the resulted ensemble model.

C. *Model ensemble and training*

In the previous section, we have trained four base predictors and obtained the corresponding forecasting models and results. These four base predictors will be employed to create the final ensemble model. The entire training process for the ensemble model is performed in the following two stages:

- Assuming that the given sample dataset $L= \{\{x_i, y_i\}, i = 1, 2, ..., n\}$ contains n tuples (samples), where x_i is the feature vector of the i-th sample after the dimensional reduction or PCA, y_i is the i-th target or real value. Specifically in our case, there are 20,000 samples with each having a certain number of features and y_i is the true housing price associated with ith sample.

  In order to prevent the over-fitting situation from being happening, the principle of cross-validation is applied to construct the second-level dataset. As we use Stacking ensemble learning method to predict our house price data, we need to use the four basic predictor to predict once, and get the prediction result, then we merge the result and a part of original dataset we get before as the second-level dataset using a well-perform predictor to predict again, so that we can avoid some one of the predictors' decision, and ensemble the four predictors' predicting result as our final predicting result. The original dataset L is randomly divided into k parts (they are called subsamples in the following discussion) $L_1, L_2, ..., L_k$. Furthermore, we define L_i and $L^\wedge_i = L - L_i$, for $i = 1, 2, ..., k$ to be the i-th cross-validation training and testing datasets respectively.

  Four base predictors will be trained separately using the training datasets and four resultant base predictors are obtained. The prediction result achieved by the j-th predictor on i-th sample in the testing dataset is denoted by Z_{ij}. We have k subsamples and thus the training process repeats k times, whereas each subsample will be used for performing t predictions and obtaining the corresponding predicting results. These predictions together with the target values of the corresponding samples form the dataset are used for the second-stage, namely, $L_{cv} = \{(Z_{i1}, Z_{i2}, \ldots, Z_{it}, y_i), i = 1, 2, \ldots, n\}$. Through this process, the training dataset of the ensemble training is a new dataset consisting of all prediction results and the corresponding target values (housing prices in our case). At the end, the final ensemble prediction model is obtained upon L_{cv}.

- According to the results presented in the previous section, it is found that GDBT model is better choice for the ensemble model because it possesses relatively low MSE. Hence we utilize GDBT as the training model again for the final ensemble model. Similarly, the parameters such as *n_estimators*, *learning_rate*, *subsample*, etc. listed in table 3 are adjusted as well based on the best solution obtained during this training process. Our computational experiements reveal that the setting of *n_estimators* = 100, *subsample* = 0.75 can achieve the best solution. At the end, the ensemble learning based housing price prediction model is trained and constructed. The effectivness of this model will be evaluated through a series of computational experiements presented in the next section.

D. *Results analysis*

Similar to the evaluating processes for four base predictors mentioned above, we compare the predicted housing prices from the ensemble model to the actul ones to verify the accuracy or performance of the ensemble model. Figure 3 depicts the predicted houing prices against the actual ones.

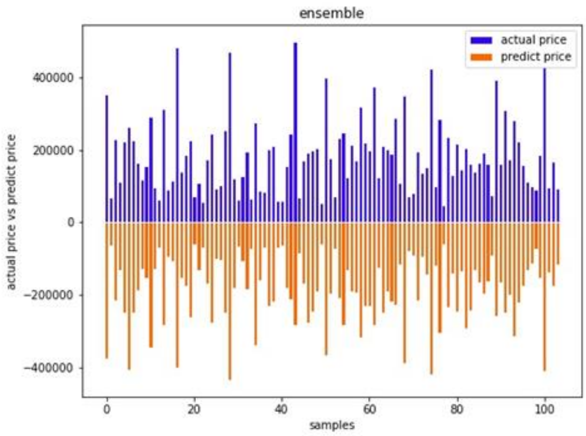

Figure 3. Prediction results obtained by the ensemble model

The horizontal axisin represents the data samples, and the vertical axis shows the housing prices where the blue is for the actual prices while the orange is for the predicted ones. It is very hard to validate the accuracy and effectiveness of the ensemble model visually, again we use the loss function (mean square error) to evaluate the predicting accuracy (effectiveness). The complete results are shown in table 5. For comparison purpose, the MSE (mean of square error) of four base predictors are also listed in the table.

TABLE V. MSE FOR ALL MODELS

Predict model	Mean Squared Error
Extra Trees	44216.900081
Random Forest	44625.093537
Gradient Boosting	43764.930335
XGB	43279.231065
Ensemble Model	41811.422310

It is not difficult to recognize the prediction results obtained by the ensemble model possesses the lowest MSE, which is reduced by 6.7% overall on average. The computation results indicate that the ensemble model is able to provide the most accurate predictions in general.

A base predictor would have its pros and cons, and it might not be able to work on all datasets with the universal superiority. By applying the ensemble techniques, we can strengthen the advantages of the underlying base predictors or models while suppressing the shortcomings of these base models. The ensemble model demonstrates its effectiveness in dealing with datasets with noise and overfitting problems.

III. CONCLUSION

In this paper we present a housing price prediction model built upon compose ET, RF, GBDT and XGB by applying the stacking ensemble learning methodology technique. The process of building an ensemble model includes extracting relevant features from California housing price data, performing the dimensional reduction, and training the model respectively. During the ensemble model construction, the individual prediction results are used as the inputs for training the ensemble predictor, which leads the final prediction model. The advantage of this model is that it can improve the prediction accuracy and effectively avoid the overfitting phenomenon when the dataset contains noise or too many features. At the same, the ensemble model is able to produce more stable results. Although we cannot claim the proposed ensemble model functions better than each base predictor consistently for all scenarios, the outcomes obtained by the ensemble model are very promising. It also encourages us to apply the similar technology to other machine learning problems.

ACKNOWLEDGMENT

This work was partially supported by the China Intelligent Urbanization Co-Creation Center [grant number CIUC20150011].

REFERENCES

[1] Oza N C. Online Ensemble Learning. *The AAAI conference on artificial intelligence*, 2000.

[2] Park B. and Bae J K. Using machine learning algorithms for housing price prediction. *Expert Systems With Applications*, 42: 2928 – 2934, 2015.

[3] Bourassa S C, Cantoni E, and Hoesli M. Predicting House Prices with Spatial Dependence: A Comparison of Alternative Methods. *Journal*

of Real Estate Research, 32: 139-160, 2010.

[4] Rajan U, Seru A, and Vig V. The failure of models that predict failure: Distance, incentives, and defaults. *Journal of Financial Economics*, 115: 237 – 260, 2015.

[5] Burger E M and Moura S J. Building Electricity Load Forecasting via Stacking Ensemble Learning Method with Moving Horizon Optimization. 2015.

[6] Rodriguez J J,, Kuncheva L I, and Alonso C J. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28: 1619 – 1630, 2006.

[7] Liu J and Wu C. A gradient-boosting decision-tree approach for firm failure prediction: an empirical model evaluation of Chinese listed companies. *Journal of Risk Model Validation*, 11: 43 – 64, 2017.

[8] Sikora R. A modified stacking ensemble machine learning algorithm using genetic algorithms. *Handbook of Research on Organizational Transformations through Big Data Analytics*. IGI Global, 43 – 53, 2015.

[9] James Brown and John Smith. How not to cite papers. In John Smith, editor, *Proceedings of the First International Conference on Modern Bibliometrics (MODBIB 2009)*, pages 20–30, Pasadena (CA), USA, July 25–28 2009.

[10] Glaeser E L and Nathanson C G. An extrapolative model of house price dynamics. *Journal of Financial Economics*, 2017.

[11] Zhou Z H. Ensemble learning. Encyclopedia of biometrics, 2015: 411-416.

[12] https://en.wikipedia.org/wiki/Karhunen%E2%80%93Lo%C3%A8ve_theorem