An optimized prediction algorithm based on XGBoost

Cheng Sheng
College of Mathematics and System Sciences
Xinjiang University
Urumqi Xinjiang, China
Email:1130196160@qq.com

Haizheng Yu
College of Mathematics and System Sciences
Xinjiang University
Urumqi Xinjiang, China
Email:yuhaizheng@xju.edu.cn

2022 International Conference on Networking and Network Applications (NaNA) | 978-1-6654-6131-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/NANA56854.2022.00082

***Abstract*—The real estate market is closely related to people's life. It is very important to accurately predict the future real estate price. Traditional methods are difficult to describe the nonlinear characteristics of house price prediction. XGBoost algorithm can effectively represent the nonlinear relationship in house price prediction. However, the selection of parameters determines the learning and generalization ability of XGBoost, and it is very important to determine the parameters of XGBoost. Particle swarm optimization algorithm can select the training parameters of XGBoost more quickly and accurately. Therefore, this paper studies the house price prediction based on the hybrid model of particle swarm optimization XGBoost algorithm, namely PSO-XGBoost model. Using the collected sample data of houses in Ames, Iowa, five different machine learning algorithms including PSO-XGBoost are used to predict house prices. Finally, the results of five algorithms are compared and analyzed. The experimental results show that PSO-XGBoost model has the highest prediction accuracy and the best effect, and the prediction effect of integrated learning algorithm is better than that of linear regression model.**

***Index Terms*—Machine learning; PSO; XGBoost; Housing price predicion**

I. Introduction

The fluctuation of housing prices has an important impact on the economic development of the country. Accurate prediction of house prices is of great importance to the national economy and has become a focus of scholars' attention [1]. Nowadays, machine learning has been used for regression prediction for some time and has achieved better and better results, especially in the problem of house price prediction, where many scholars have built models to predict house prices by machine learning algorithms. Some scholars focus on using specific machine learning models for house price prediction, for example, Rahman et al.[2][3][4][5] used artificial neural networks to build house price prediction models for different regions. The results of the study show that neural networks have superior prediction performance. Some scholars have compared several different machine learning models. For example, Ho et al.[6] used three models, support vector machine, random forest, and gradient augmentation machine, to study a sample data of about 40,000 housing transactions in Hong Kong over 18 years and compared the effectiveness of these models. Both random forest and gradient augmentation machines were found to outperform support vector machines.Embaye et al.[7] used a range of machine learning methods such as Ridge, LASSO, Tree, Bagging, Random Forest, Boosting and Ordinary Least Squares (OLS) to predict housing rental values It was found that all machine learning algorithms outperformed OLS in prediction compared to various OLS models. Huang et al.[8] used the real estate data of three counties in Los Angeles to predict the house value by using linear and nonlinear machine learning methods.Finally, the effects of models such as linear regression, decision tree, boosting, random forest and support vector machine were compared and the study found that the support vector machine method had the best performance.

However, most of the machine learning algorithms have complex parameter combinations, and the parameter settings will directly affect the final performance of the model, The traditional manual adjustment of parameters is not only time-consuming but also may not achieve optimal results. Therefore, some researches have combined machine learning algorithms with intelligent optimization algorithms to achieve the purpose of automatic optimization model. For example, Xu et al.[9] proposed a PSO-SVM model that combined with particle swarm optimization algorithm and support vector machine to automatically recognize and classify the surface defect images of lithium battery electrodes. The experimental results show that the average recognition rate of PSO-SVM for defects is 98.3%, which is an effective and feasible automatic detection and recognition method. Song et al.[10] proposed an optimization model that combines particle swarm optimization algorithms and the XGBoost algorithm to explore the relationship between steel's performance and its composition and manufacturing parameters, and compare it with other machine learning models.The experimental results show that the PSO-XGBoost model is better than other models. GuJirong et al.[11] proposed a hybrid housing price prediction method (G-SVM) by combining genetic algorithm (GA) and support vector machine, and validated the house price prediction ability of G-SVM method using domestic data cases. The experimental results show that the prediction accuracy of the G-SVM model is better than that of the gray model (GM). Wang et al.[12] proposed a house price prediction method based on the combination of particle swarm optimization algorithm and support vector machine, and investigated the prediction performance of PSO-SVM model and BP neural

978-1-6654-6131-3/22/$31.00 ©2022 IEEE
DOI 10.1109/NaNA56854.2022.00082

network using the real estate sample data of Chongqing city. The experimental results show that PSO-SVM has higher prediction accuracy than BP neural network.

Compared with ant colony algorithm, grid algorithm, and genetic algorithm, particle swarm optimization algorithm is powerful and easy to implement[12]. Many studies use particle swarm optimization algorithms to optimize machine learning algorithms. In order to improve the prediction accuracy, this paper is based on PSO-XGBoost model for house price prediction study. XGBoost has good performance because of the second-order Taylor expansion and explicit regularization terms, which can effectively represent hidden non-linear relationships in housing prediction problems. The particle swarm optimization algorithm has a unique advantage in optimizing the XGBoost parameters, which can automatically optimize the parameter adjustment process of the XGBoost model and overcome the disadvantages of adjusting the XGBoost model parameters by empirical method or trial-and-error method. The PSO-XGBoost model combines the advantages of the two algorithms, and using this model for house price prediction can significantly improve the timeliness and accuracy of the prediction.

II. ALGORITHM CONSTRUCTION

A. Particle Swarm Optimization Algorithm

The particle swarm optimization algorithm (PSO) is a population-based heuristic algorithm, which is inspired by the foraging behavior of the bird group[13].

Suppose that in a D-dimensional search space, there are n particles form a group, where the position of i-th particle is represented as a D-dimensional vector: $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, and the speed of i-th particle is $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$.Corresponding individual extremum is $P_i = (P_{i1}, P_{i2}, \ldots, P_{iD})$, where $i = 1, 2, \ldots, n$. During each iteration, the particle updates its speed and position through the following formula:

$$\begin{aligned} V_{id}^{(t+1)} =& \omega \times V_{id}^{(t)} + c_1 \times r_1 \times \left(P_{id}^{(t)} - X_{id}^{(t)}\right) \\ &+ c_2 \times r_2 \times \left(P_{gd}^{(t)} - x_{id}^{(t)}\right), \end{aligned} \tag{1}$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)}, \tag{2}$$

where $d = 1, 2, \ldots, D$; t is the current iteration number; V_{id} is the velocity of the particle; P_{id} is the individual optimum; P_{gd} is the global optimum. c_1 and c_2 are the individual and social learning factors; r_1 and r_2 are positive random number in the range $(0, 1)$ under normal distribution. ω is the inertia weight, its decreasing form can be expressed as

$$\omega = \frac{\omega_{\max} + (iter - iter_i) \times (\omega_{\max} - \omega_{\min})}{iter}, \tag{3}$$

where $iter$ is the maximum number of iterations, iter i is the current number of iterations, $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of ω, respectively.

B. XGBoost Algorithm

XGBoost is a highly scalable end-to-end tree lifting system, which is widely used by data scientists and provides the most advanced results on many issues[14] The objective function of XGBoost is:

$$obj = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right), \tag{4}$$

where K is the number of trees, $L\left(y_i, \hat{y}_i\right)$ is loss function, $\Omega(f)$ is the regularization term, which is used to control the complexity of the model and prevent the model from overfitting.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2, \tag{5}$$

where T is the number of leaves, ω is the score of the leaf node. The predicted value of the i-th sample after t-th iteration is:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t\left(X_i\right), \tag{6}$$

also

$$\sum_{i=1}^{t} \Omega\left(f_i\right) = \sum_{i=1}^{t-1} \Omega\left(f_i\right) + \Omega\left(f_i\right), \tag{7}$$

$\sum\limits_{i=1}^{t-1} \Omega\left(f_i\right)$ is a constant. From (5), (6) and (7):

$$\begin{aligned} \mathrm{obj}^{(t)} =& \sum_{j=1}^{T} \left[\sum_{i \in \mathrm{I}_j} L\left(y_i, \hat{y}_i^{(t-1)} + \omega_j\right) + \frac{1}{2}\lambda \omega_j^2\right] \\ &+ \gamma T + \text{ constant }. \end{aligned} \tag{8}$$

Perform a second-order Taylor expansion on $L\left(y_i, \hat{y}_i^{(t-1)} + \omega_j\right)$,

$$\begin{aligned} L\left(y_i, \hat{y}_i^{(t-1)} + \omega_j\right) &\simeq L\left(y_i, \hat{y}_i^{(t-1)}\right) \\ &+ L'\left(y_i, \hat{y}_i^{(t-1)}\right)\omega_j + \frac{1}{2}L''\left(y_i, \hat{y}_i^{(t-1)}\right)\omega_j^2. \end{aligned} \tag{9}$$

Let $g_i = L'\left(y_i, \hat{y}_i^{(t-1)}\right), h_i = L''\left(y_i, \hat{y}_i^{(t-1)}\right)$, so:

$$\begin{aligned} obj^{(t)} &\simeq \sum_{j=1}^{T} \left[\sum_{i = \mathrm{I}_j} \left(g_i \omega_j + \frac{1}{2} h_i \omega_j^2\right) + \frac{1}{2}\lambda \omega_j^2\right] + \gamma T \\ &= \sum_{j=1}^{T} \left[\left(\sum_{i = \mathrm{I}_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i \in \mathrm{I}_j} h_i + \lambda\right)\omega_j^2\right] + \gamma T. \end{aligned} \tag{10}$$

Let $G_j = \sum\limits_{i \in \mathrm{I}_j} g_i, H_j = \sum\limits_{i = \mathrm{I}_j} h_i$, so:

$$obj^{(t)} = \sum_{j=1}^{T} \left[G_j \omega_j + \frac{1}{2}\left(H_j + \lambda\right)\omega_j^2\right] + \gamma T. \tag{11}$$

since ω_j is independent of each other, and $G_j \omega_j +$

$\frac{1}{2}\left(H_j+\lambda\right)\omega_j^2$ is twice, the optimal solution is:

$$\omega_j^* = -\frac{G_j}{H_j+\lambda}, \tag{12}$$

$$\text{obj}^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T. \tag{13}$$

C. PSO optimized XGBoost algorithm

The training process of XGBoost involves the setting of many parameters. In this study, seven parameters that have a significant impact on XGBoost algorithm are optimized by PSO. The information of each parameter is given in TABLE I. The position vector in particle swarm optimization algorithm is assigned to these parameters, and the prediction accuracy of the model is taken as the fitness value. According to TABLE I, the velocity vector and the position vector of the i-th particle at the t-th iteration can be expressed as:

$$\begin{aligned}(V)_i^{(t)} = \Big[&V_{i,eta}^{(t)}, V_{i,colsample\_bytree}^{(t)}, V_{i,max\_depth}^{(t)},\\ &V_{i,gamma}^{(t)}, V_{i,subsample}^{(t)}, V_{i,alpha}^{(t)}, V_{i,lambda}^{(t)}\Big],\end{aligned} \tag{14}$$

$$\begin{aligned}(P)_i^{(t)} = \Big[&P_{i,eta}^{(t)}, P_{i,colsample\_bytree}^{(t)}, P_{i,max\_depth}^{(t)},\\ &P_{i,gamma}^{(t)}, P_{i,subsample}^{(t)}, P_{i,alpha}^{(t)}, P_{i,lambda}^{(t)}\Big].\end{aligned} \tag{15}$$

The fitness value of the particle is:

$$(F)_i^{(t)} = \left((P)_i^{(t)} \rightarrow \text{XGBoost}\Big|_{\text{training set}}\right)_{[\text{metric }=RMSE]}. \tag{16}$$

The individual optimal value is:

$$P_{id}^{(t)} = \min\left(F_i^{(t)}\right), 0 \leq j \leq t. \tag{17}$$

The global optimal value is:

$$P_{gd}^{(t)} = \min\left(P_{kd}^{(t)}\right), 1 \leq k \leq n. \tag{18}$$

Update the position, velocity and inertia weight of each particle according to (1) to (3), and finally the optimal fitness value is found.

PSO-XGBoost algorithm are shown in Fig. 1, This figure shows the process of particle swarm optimization XGBoost algorithm. The optimal result can be obtained by running PSO-XGBoost once, while XGBoost needs to be adjusted manually many times, and the optimal result may not be obtained[15].

As can be seen from Fig. 1, the steps of PSO-XGBoost algorithm are as follows:

a) Initialize particle swarm optimization parameters, including particle number, learning factor, weight coefficient and maximum number of iterations;

b) Train XGBoost model, and the parameters to be optimized change with the flight of particles;

c) Calculate and evaluate fitness values. The fitness value comes from the output value of XGBoost model and is used to evaluate the performance of PSO. The smaller the fitness value, the better the performance;

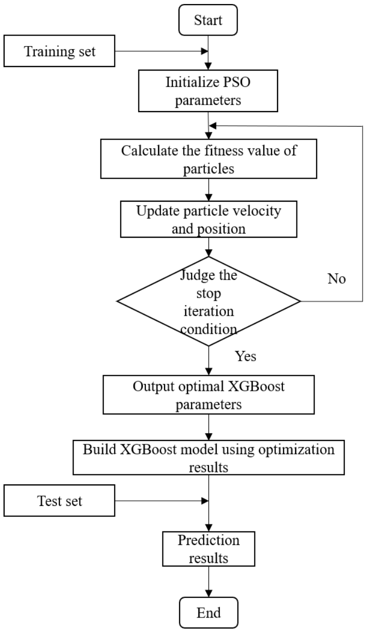
Start
Training set
Initialize PSO parameters
Calculate the fitness value of particles
Update particle velocity and position
Judge the stop iteration condition
No
Yes
Output optimal XGBoost parameters
Build XGBoost model using optimization results
Test set
Prediction results
End

Fig. 1. PSO-XGBoost algorithm flow chart

TABLE I. the main parameters of XGBoost to be optimized

Parameter	Default value	Range	Explain
eta	0.3	[0, 1]	Learning rate.
colsample_bytree	1	(0, 1]	Subsample ration of columns for each tree
max_depth	6	[1, ∞)	Maximum depth of a tree
gamma	0	[0, ∞)	Minimum loss required for splitting
subsample	1	(0, 1]	Subsample ratio of the training instances.
alpha	0	[0, ∞)	L1 regularization term
lambda	1	[0, ∞)	L2 regularization term

d) Judge whether the stop iteration condition is met. If satisfied, the iterative process is terminated and the optimal parameters of XGBoost model are obtained. Otherwise, continue iterative calculation;

e) Validate the regression model. Use the optimization results to establish the oost model and output the results of house price prediction.

III. EXPERIMENT ANALYSIS

A. Data source and preprocessing

The dataset in this article is derived from the Kaggle[16]. This dataset is composed of historical data of residences in Ames, Iowa, including training sets and test sets. The training set includes classification variables, numerical variables and some missing values. There are a total of 1461 observations, including a total of 79 features with the SalePrice. The test set has 1460 observation values and this set is less than the SalePrice of the training set. We use the model of training set to fill the SalePrice of the test set to achieve the aim of predicting house prices.

In this study, we adopt a mean interpolation method for variables with more than 50% data, while variables with missing data exceeding 50% are deleted directly. The one-hot encoding is used to convert the classification variables to numerical variables. Finally, the principal component analysis is adopted for dimension reduction processing because there are so many features in original datasets.

B. Exploratory data analysis

Exploratory data analysis is an essential step before building a regression model. In this way, researchers can discover the implicit patterns of the data, which in turn helps choose appropriate machine learning approaches[17]. The histogram of house price is shown in Fig. 2, which indicates that it is tilted and the logarithmic conversion is required to normalize. After conversion, the result is shown in Fig. 3

The heat map is the best way to understand the correlation between variables. As shown in Fig. 4, the deeper the grid color indicates the higher the correlation between the two variables. Therefore, independent variables that are highly related to sales prices include OverallQual, TotRmsAbvGrd, GrLiveArea, GarageArea and GarageCars.

Fig. 5 is a box plot between the community and the sales price. You can observe the characteristic of the community that affects house prices. The most expensive houses are located in Northridge Heights and Stone Brook, whose house prices are between \$ 250,000 and \$ 350,000, and the price of several high -end houses can reach up to \$ 750,000. Meanwhile, the houses with the lowest prices are distributed in Old Town, Brook Side, Sawyer, North Ames, Edwards, Iowa DOT, Rail Road, Meadow Village and Briardale.

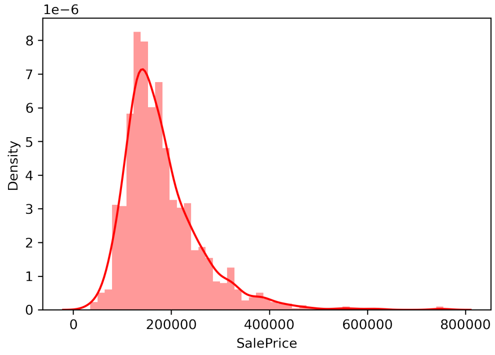

Fig. 2. SalePrice distribution

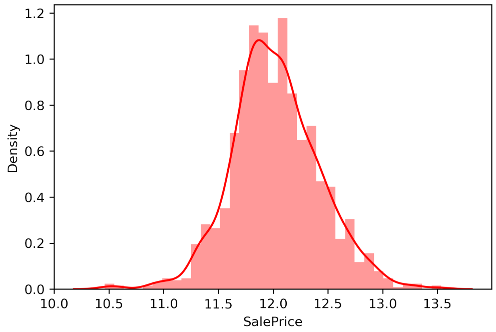

Fig. 3. Log transformation SalePrice distribution

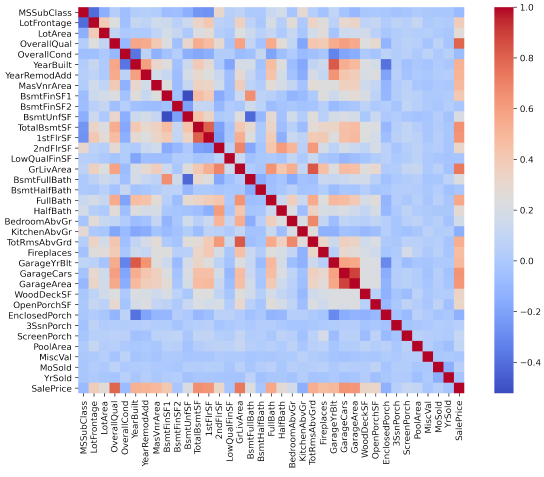

Fig. 4. Variable correlation heat map

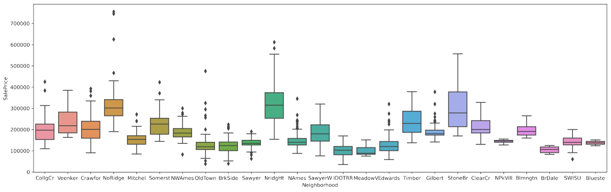

Fig. 5. Boxplot for Neighborhood

C. Evaluation Index

In this paper, we select five different regression models to predict house prices and compares their performance. The evaluation indexes of each model are root mean square error ($RMSE$) and goodness of fit (R^2), which can be expressed as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{19}$$

$$R^2 = \frac{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \tag{20}$$

Where y_i, $\hat{y}_i$ and $\bar{y}$ are the logarithmically processed house prices (Saleprice), the prediction value of the house price and the average value of the house price respectively.

D. Results and analysis

Firstly, Ridge regression and Lasso regression are used to predict in this paper. The loss function of Ridge regression is a linear least squares function, and the regularization is given by L2 norm. Different from Ridge regression, Lasso regression changes the regularization term from L2 norm to L1 norm. The model parameter settings are $alpha = 1$ and $alpha = 0.0005$ respectively. The scatter diagram between the predicted value of house price and the real value of the two models is shown in Fig. 6 and Fig. 7. In the two figures, the X axis is the predicted value of Ridge regression and Lasso regression respectively, and the Y axis is the real value of house price.

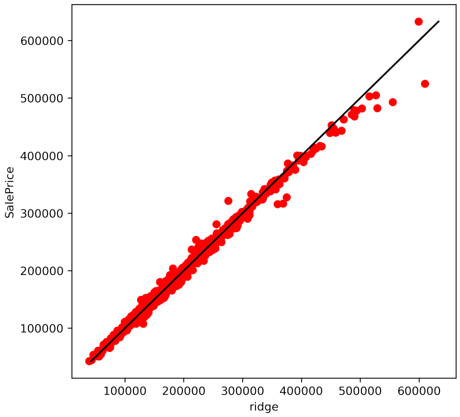

Fig. 6. Ridge Regression

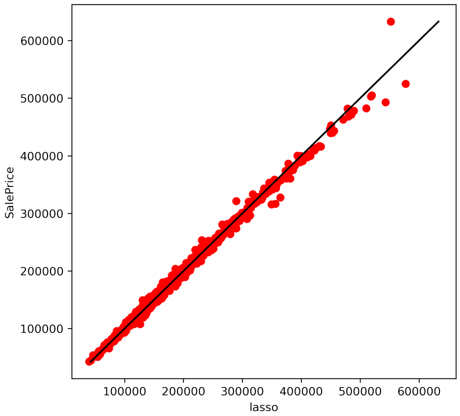

Fig. 7. Lasso Regression

It can be seen that Ridge regression and Lasso regression are better for lower-priced house prediction, but when the price of the house exceeds $ 400,000, the scatter chart is roughly below the line, indicating all house prices are higher than the real value.

Considering the nonlinear relationship between house prices and influencing factors, we also consider two integrated learning algorithms, LightGBM and XGBoost. LightGBM is a gradient enhancement framework based on tree learning algorithms. It is similar to XGBOOST, and is also a gradient improved algorithm, but LightGBM usually has a faster training speed and a lower memory usage rate[18]. Fig. 8 and Fig. 9 are house prices predictive values and real values of LightGBM and XGBoost respectively. In the two figures, the X axis is LightGBM regression and XGBOOST regression prediction values respectively, and the Y axis is real values.

It can be seen that the model has a good prediction effect whether the house price is high and low. Then, as shown in Fig. 1, particle swarm optimization algorithm is used to optimize XGBoost, and the optimized PSO-XGBoost model is used for regression prediction. The scatter chart of the predicted value and actual value of the model is shown in Fig. 10. In this figure, the X axis is PSO-XGBoost regression forecast value, and the Y axis is the real value. Finally, the evaluation indicators of all models are shown in TABLE II.

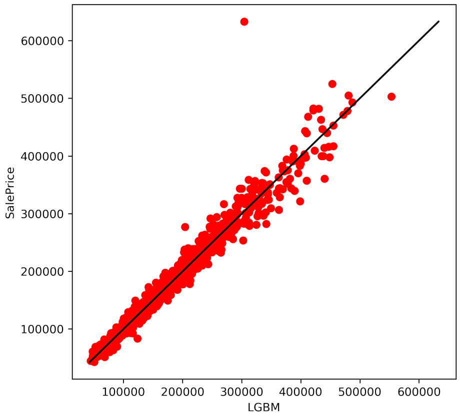

Fig. 8. LGBM Regression

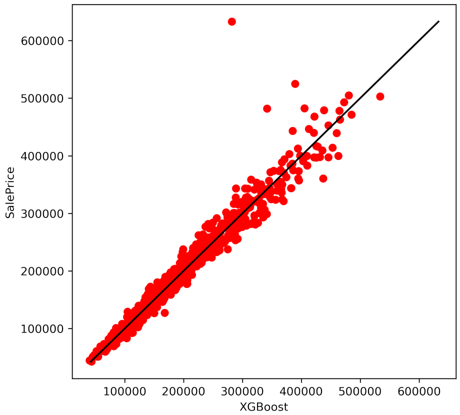

Fig. 9. XGBoost Regression

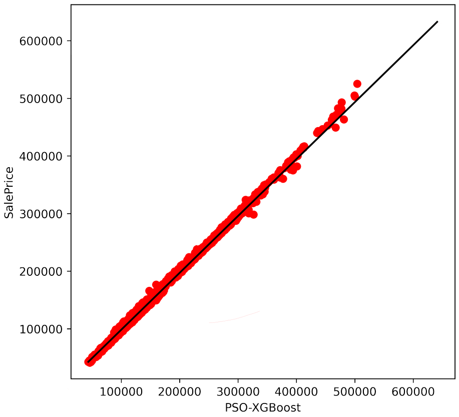

Fig. 10. PSO-XGBoost Regression

Fig. 10 shows the prediction performance of PSO-XGBoost.

TABLE II. Comprehensive model evaluation indicators

Model	R^2	RMSE
Ridge	0.9375	0.1314
Lasso	0.9386	0.1307
LightGBM	0.9605	0.1230
XGBoost	0.9576	0.1225
PSO-XGBoost	0.9887	0.1015

We can see that all the points in the figure are very close to which indicates that the prediction value is very close to the actual value and prediction performance of the model is very good.

It can be seen from TABLE II that the performance of the LightGBM and the XGBoost is not so diverse, but both models have better performance than Ridge regression and Lasso regression. Compared with XGBoost, the R^2 of PSO-XGBoost increases from 0.9517 to 0.9887 and the RMSE decreases from 0.1251 to 0.1015. The result shows that the performance of XGBoost model has been significantly improved after particle swarm optimization. Therefore, among the five models studied in this paper, PSO-XGBoost has the best prediction performance. In addition, the prediction accuracy of ensemble learning model is better than that of linear regression model.

IV. Conclusion

In this study, PSO-XGBoost algorithm is used to predict housing prices. Aiming at the problem of low accuracy and efficiency of adjusting parameters of XGBoost model by empirical method, particle swarm optimization algorithm is used to solve the problem of difficult parameter adjustment when predicting house prices by XGBoost model, and improve the speed and accuracy of house price prediction. Based on the sample data of houses in Ames, Iowa, the prediction performance of PSO-XGBoost model and five models including Ridge, Lasso, LightGBM and XGBoost are studied. The experimental results show that the prediction effect of XGBoost is significantly improved after particle swarm optimization. The R^2 and RMSE of PSO-XGBoost model reach 0.9887 and 0.1105 respectively, which has higher prediction accuracy than the other four models. The prediction effect is the best. In addition, the integrated learning models of LightGBM and XGBoost have better prediction effect than the linear regression model.

V. Acknowledge

This work is supported in part by the National Natural Science Foundation of China (61662079, 11761070, U1703262) and Xinjiang Natural Science Foundation of China (2021D01C078).

References

[1] Li, D., Liu, L., and Lv, H. (2021). Prediction of Chinas Housing Price Based on a Novel Grey Seasonal Model. Mathematical Problems in Engineering, 2021, 1-11.

[2] Rahman, S. N. A., Maimun, N. H. A., Najib, M., Razali, M., and Ismail, S. (2019). The artificial neural network model (ANN) for Malaysian housing market analysis. PLANNING MALAYSIA, 17(1), 1-9.

[3] Kitapci, O., Tosun, ., Tuna, M. F., and Turk, T. (2017). The use of artificial neural networks (ANN) in forecasting housing prices in Ankara, Turkey. Journal of Marketing and Consumer Behaviour in Emerging Markets, 1(5), 4-14.

[4] Chiarazzo, V., Caggiani, L., Marinelli, M., and Ottomanelli, M. (2014). A neural network based model for real estate price estimation considering environmental quality of property location. Transportation Research Procedia, 100(3), 810-817.

[5] Kang, J., Lee, H. J., Jeong, S. H., Lee, H. S., and Oh, K. J. (2020). Developing a forecasting model for real estate auction prices using artificial intelligence. Sustainability, 12(7), 1-19.

[6] Ho, W. K., Tang, B. S., and Wong, S. W. (2021). Predicting property prices with machine learning algorithms. Journal of Property Research, 38(1), 48-70.

[7] Embaye, W. T., Zereyesus, Y. A., and Chen, B. (2021). Predicting the rental value of houses in household surveys in Tanzania, Uganda and Malawi: Evaluations of hedonic pricing and machine learning approaches. Plos one, 16(2), e0244953-e0244953.

[8] Huang, Y. (2019). Predicting home value in California, United States via machine learning modeling. Statistics, optimization and information computing, 7(1), 66-74.

[9] Xu, C., Li, L., Li, J., and Wen, C. (2021). Surface defects detection and identification of lithium battery pole piece based on multi-feature fusion and PSO-SVM. IEEE Access, 9, 85232-85239.

[10] Song, K., Yan, F., Ding, T., Gao, L., and Lu, S. (2020). A steel property optimization model based on the XGBoost algorithm and improved PSO. Computational Materials Science, 174, 109472.

[11] Gu, J., Zhu, M., and Jiang, L. (2011). Housing price forecasting based on genetic algorithm and support vector machine. Expert Systems with Applications, 38(4), 3383-3386.

[12] Wang, X., Wen, J., Zhang, Y., and Wang, Y. (2014). Real estate price forecasting based on SVM optimized by PSO. Optik, 125(3), 1439-1443.

[13] Eberhart, R., and Kennedy, J. (1995, November). Particle swarm optimization. In proceedings of the IEEE international conference on neural networks (Vol. 4, pp. 1942-1948).

[14] Chen, T., and Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[15] Gu, K., Wang, J., Qian, H., Su, X., and Rajinikanth, V. (2021). Study on intelligent diagnosis of rotor fault causes with the PSO-XGBoost algorithm. Mathematical Problems in Engineering, 2021, 1-17.

[16] https://www.kaggle.com/c/house-prices-advanced-regression-techniques

[17] Truong, Q., Nguyen, M., Dang, H., and Mei, B. (2020). Housing price prediction via improved machine learning techniques. Procedia Computer Science, 174, 433-442.

[18] Ke G, Meng Q, Finley T, et al. (2017, December). LightGBM: a highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 3149-3157).