P-ISSN: 2986-4178
E-ISSN: 2988-4004

Emerging Statistics and Data Science Journal
Vol 2, No.3, Tahun 2024

Decision Tree-Based Boosting Method with An Application in House Sale Price Prediction

Intan Lisnawati[1]

[1]National Central University, Berbah, Sleman, 55573 Indonesia

*Corresponding author: intanlisnawati17@gmail.com

P-ISSN: 2986-4178
E-ISSN: 2988-4004

Riwayat Artikel

Dikirim: 30 Agustus 2024
Direvisi: 24 Oktober 2024
Diterima: 27 Oktober 2024

ABSTRAK

Metode *boosting* kembali memberikan inovasi dalam langkahnya, seperti XGBoost yang baru 'lahir' pada 2016 lalu. Metode yang nampak powerful ini melatarbelakangi pemilihan metode untuk memberikan prediksi yang dalam artikel ini adalah harga rumah. Dalam penulisan ini, keefektifannya akan diujicobakan kemudian dibandingkan dengan pendahulunya, *gradient boosting*. Melalui aplikasi beberapa data, nantinya akan memberikan sebuah prediksi berdasarkan data yang dimasukkan. Untuk menghasilkan prediksi yang lebih baik berdasarkan estimator tunggal, metode *ensemble* mengkombinasikan berbagai estimator tunggal dalam memberikan prediksi. Parameter setiap metode juga dapat diatur sedemikian rupa untuk memperkecil nilai error. Dalam penulisan ini, disajikan data percobaan yang kemudian memberikan prediksi harga rumah. Data *testing* digunakan untuk menilai metode yang paling rendah memberikan nilai error. Diantara metode yang diterapkan, *gradient boosting* menunjukkan nilai error terkecil US$ 22,766, disusul XGBoost US$ 24,069, sedangkan error terbesar oleh *decision tree* US$ 35,637.

Kata Kunci: estimator tunggal, metode *ensemble*, prediksi.

ABSTRACT

Given some input, we want to make a prediction for the corresponding output. In order to make a better prediction over a single estimator, ensemble methods combine the predictions of several base estimators built with a given learning algorithm. As XGBoost is newer than gradient boosting, it is known to be more powerful than gradient boosting. This apparently effective method is the background for choosing boosting methods to provide predictions, which in this article is house prices. Each method's parameter also can be adjusted in order to get a closer gap between the real and the predicted value. By using the House Sale Price training data set, we apply some ensemble methods to predict the unseen House Sale Price data set and see the accuracy based on their Root Mean Squared Error (RMSE) value. It shows that Gradient Boosting gives the smallest RMSE, US$ 22,766, meanwhile XGBoost US$ 24,069, and Decision Tree US$ 35,637.

Keywords: base learner, ensemble method, prediction.

1. Introduction

One of the basic necessities of life is a place to live, namely a house [1]. The features of a house will ultimately affect the price of the house. The price also tends to increase each year. To find out how algorithms forecast home selling prices, some researchers used a variety of algorithms, including random forest [2], general regression neural network [3], and multiple linear regression [4].

Researchers employed a variety of machine learning techniques to assess the algorithms' efficacy. Gradient boosting is one of the most often used machine learning techniques since it provides an estimate by taking into account earlier steps. Boosting itself is categorized as an ensemble learning method, as it also has another type called bagging [5].

Ensemble method trains multiple learners to solve the same problem [6]. It tries to construct a set of learners and combine them. The learners combined in an ensemble are called base learners or weak learners. By combining learners, it will produce a model which provides more accurate predictions from the base learner. It builds the model in a sequential manner such that the new model is built to correct the errors made by its predecessor [5]. This combined model is usually called as a strong learner.

In boosting, the base learners can be any machine learning algorithm that performs slightly better than random guessing [7]. Gareth mentioned that boosting is able to use decision trees as its base learner [8]. Decision trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [9]. Thus, in this study we consider decision trees to be the base learners.

Knowing the advantage of ensemble learning, we are interested to complement the previous research in the field of boosting works to give house price prediction and investigate the precision for the prediction with the real price aimed with parameters included on the dataset. The result then can be compared to see the effectiveness of the algorithm giving predictions. In this study, our concern is housing in the USA. But basically the method can be run into any data set as long as it has variables to explore further making predictions. The model is first trained by using a training data set until it reaches the desired model, through the smallest loss between observed and predicted value. Comparison among the models is done to see the effectiveness of the ensemble method rather than a single decision tree.

2. Research Method

In this section we simulate data training by using House Price data set from Kaggle competition. This dataset contains house prices in Ames, Iowa, USA which was collected by Dean De Cock in Data Science education [10]. In total, it has 1,460 sale price observations which has 79 features. The house's sale price data consists of 43 categorical and 36 numerical features. We split the data into 90% training and 10% testing, that is 1,314 data training and 146 data testing.

Before going further, we put a note of gradient boosting equation [11] on the equation below:

$$F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m) \quad (1)$$

where

$$\rho_m = arg\ \min_{\rho} \sum_{i=1}^{N} \quad l[y_i, F_{m-1}(x_i) + \rho h(x_i; \alpha_m)] \tag{2}$$

and $M = 1, \ldots, M$ is the iteration.

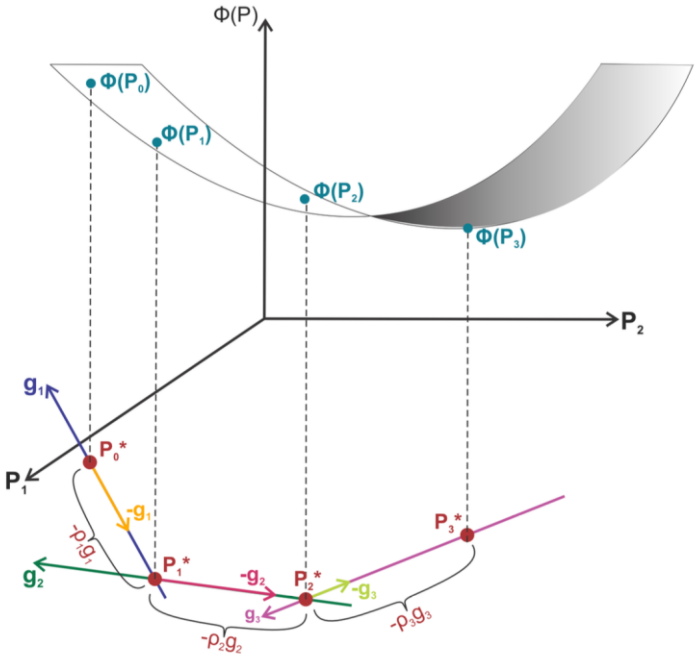

Figure 1. ρ_m controls how far we step along g_m direction and will stop after finding smallest loss between previous prediction, P_m, and current prediction, P_{m-1} [12]

Meanwhile, which makes gradient boosting different with XGBoost is that XGBoost applies weight w_j^* which affects the L in the XGBoost algorithm. Below is the equation, see [13].

$$w_j^* = -\frac{\Sigma_{x_i \epsilon\, I_j} g_i}{\Sigma_{x_i \epsilon\, I_j} h_i + \lambda} \tag{3}$$

$$L^{(m)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \quad \frac{\left(\Sigma_{i \in I_j} g_i\right)^2}{\Sigma_{i \in I_j} h_i + \lambda} + \gamma T \tag{4}$$

with

$$g_i = \left[\frac{\partial l(y,z)}{\partial z}\right]_{y=y_i, z=\hat{y}^{(m-1)}(x_i)}$$

$$h_i = \left[\frac{\partial^2 l(y,z)}{\partial z^2}\right]_{y=y_i, z=\hat{y}^{(m-1)}(x_i)}$$

w = leaf weight

L = the scoring function to measure the quality of a tree structure q

T = the number of leaves in the tree

If the tree is too big, then both λ and λ will play a role in controlling how big the tree is.

Preprocessing data in Exploratory Data Analysis is also done to know the characteristics of the data we simulate. The accuracy of this method is based on Root Mean Squared Error (RMSE) value. Therefore, the smaller RMSE then the better the prediction which is given by:

$$RMSE = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} f(x_i) - f(\underline{x})\right)^2} \quad (5)$$

where $f(x_i)$ is the target data and $f(\underline{x})$ is the mean.

The flow diagram below shows how we do the numerical simulation in the boosting methods. By using training data set we sequentially train the data in $m = 1, \ldots, M$ iteration to get a model which gives us the smallest loss. Then we use that model to predict the output of the information given in the testing data set. Here the base learner is the decision tree.

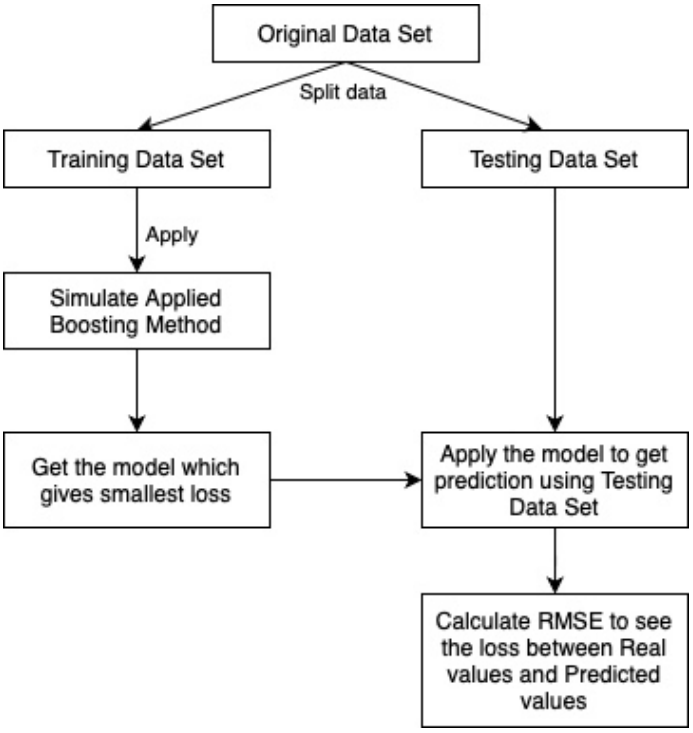

Figure 2. Flow Diagram of Boosting Numerical Simulation

The original data set was split into 90% training and 10% testing so that we trained the training data set until it gives a model which gives the smallest error among its predictions. The desired model we got then is used to give a prediction in the testing data set. In this step, compute the error between the real values and the predicted values to see the accuracy of the model we use.

3. Result and Discussion

In this section, we barely discuss the result after running the data we got from open sources data, *Kaggle*. We serve the result divided in 3 parts which explain each method we used here.

3.1. Decision Tree

The first method we apply for numerical simulation is decision trees as it is the base learner that we will use for the other methods. We apply the CART algorithm in the training data set until we finally get a tree which considers the highest gain every time splitting the region. As a note, we did not set the tree's depth, thus the nodes are expanded until all leaves are pure, that is only contains exactly the same features in a region.

After getting the tree, then it can be used to predict the output of the unseen data set by inputting the house's features to the corresponding tree. We also compare the prediction of house sale price and to the real value of 146 testing data sets. To see the accuracy of the tree, Figure 6 already displayed the RMSE value as well as compared it to the benchmark we set in the beginning.

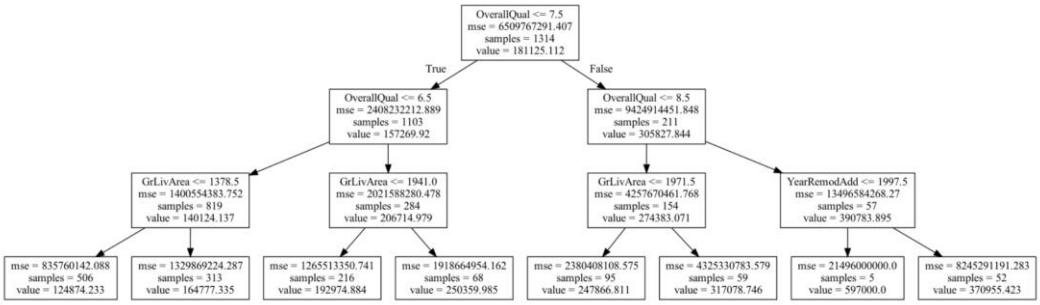

Figure 3. Some Parts of The Tree in Decision Tree Formed after Running The Data

As can be seen in Figure 3 above, it provides the tree's appearance in gradient boosting. It keeps updating the data set to be classified into areas which navigate into smaller errors at each step. Thus if we take a look closer, the error given would be smaller and smaller as it gains to the next branch of the tree.

3.1.1. Gradient Boosting

Since the gradient boosting method does an iteration step and gives an updated prediction value every time we iterate, we run in some m iterations to approach the outcome. Early simulations give higher values of RMSE meaning that the distance between prediction and real value is still too far. It keeps producing lower values as the iteration is more frequent which means that the distance of prediction is getting closer to the real value.

Table 1 Loss Value in US$ between Observed and Prediction of The Sale Price at Each Iteration

Sale Price	r_0	$y - r_0$	r_1	$y - r_1$	r_2	$y - r_2$	r_3	$y - r_3$
91,000	181,125	-90,125	175,500	-84,500	173,204	-82,204	168,908	-77,908
197,900	181,125	16,774	182,310	15,589	186,907	10,992	188,150	9,749
315,500	181,125	134,374	200,108	115,391	207,138	108,361	222,835	92,664
167,000	181,125	-14,125	179,490	-12,490	177,194	-10,194	172,899	-5,899
179,900	181,125	-1,225	18,779	-7,899	194,829	-14,929	199,797	-19,897
174,000	181,125	-7,125	179,490	-5,490	179,331	-5,331	178,913	-4,913
175,500	181,125	-5,625	179,490	-3,990	179,331	-3,831	178,913	-3,413
172,500	181,125	-8,625	179,490	-6,990	173,796	-1,296	169,501	2,998
392,500	181,125	211,374	187,799	204,700	194,829	197,670	199,797	192,702
170,000	181,125	-11,125	182,310	-12,310	182,159	-12,159	183,394	-13,394

Table 1 compares that mostly all of the loss at each iteration is getting smaller, which means that the prediction is getting closer to the real value, y. Since the accuracy of our simulation is based on RMSE, then after simulating gradient boosting to some value in the testing data set, we compare its value to the benchmark of Top 4% and Top 10%. The comparison among those RMSE is shown in Figure 4.

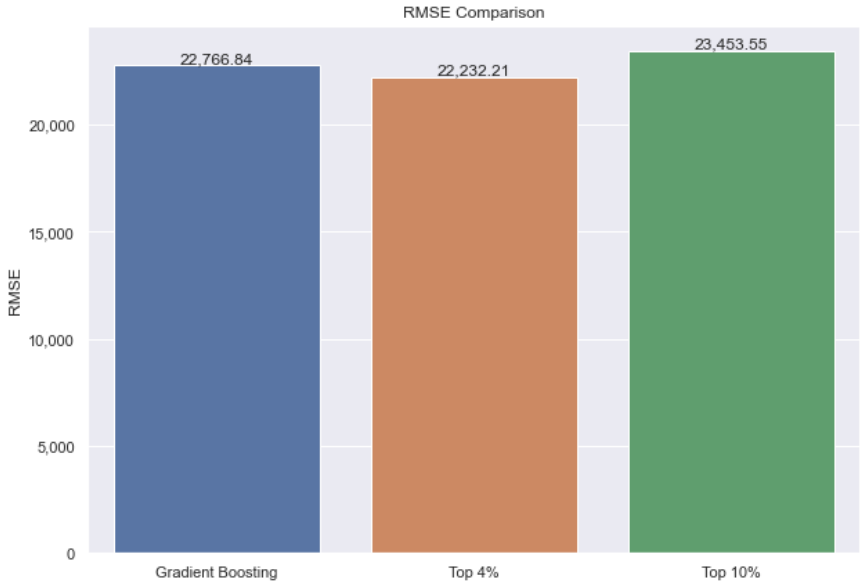

Figure 4. RMSE comparison between gradient boosting and the benchmark

As can be seen above, RMSE value of gradient boosting actually is good enough since US$22,766 is relatively small compared to the original house's sale price which ranges from US$34,100 to US$795,500 with US$188,000 on average. Also, it only gives a slightly different RMSE value compared to the Top 4% which is only around US$ 500 difference. Thus, it can be said that Gradient Boosting is able to make a plausible prediction as well as one of the best top 10% which gives a small difference between prediction and real value.

3.1.2. XGBoost

In this setup, we only vary the value of λ and γ, and other parameters remain the same as a default of XGBoostRegressor. Iterations follow the default $m = 1$ and maximum depth is 6. Thus the tree will stop being built after it reaches the depth 6. We discover that the prediction outcome is only affected by the λ value. When we modify λ, the outcome varies. Figure 5 shows that the most minimum RMSE corresponds to the λ value being 49. On the other hand, based on our dataset, γ does not affect the predictions. The γ values vary from 1 to 100, and the outcome shows no improvement or degradation. So in the comparison shown in Figure 5 when we set the λ value in XGBoost equal to 49.

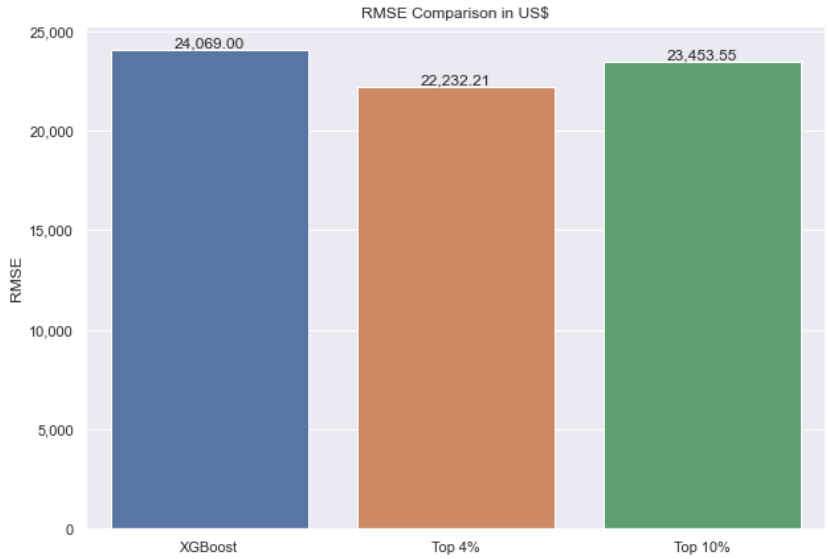

Figure 5. RMSE comparison between XGBoost and the benchmark

Based on the pie chart above, compared to gradient boosting, XGBoost does not show a better prediction. It may happen since we only set the λ value and do not consider advanced parameters. According to the official XGBoost documentation on *https://xgboost.readthedocs.io/en/stable/index.html*, there are 3 different categories of parameters for XGBoost: general parameters, booster parameters, and task parameters. There are a lot of parameters available to be set up in order to get the smaller loss prediction. However, keep in mind that the complexity of the model will rise as we set more parameters. In the meantime, since the regularized objective function was established, we merely take different values of γ and λ into consideration. This suggests that a sophisticated method would not always yield a superior result because it also depends on the input of the data set and the parameters applied.

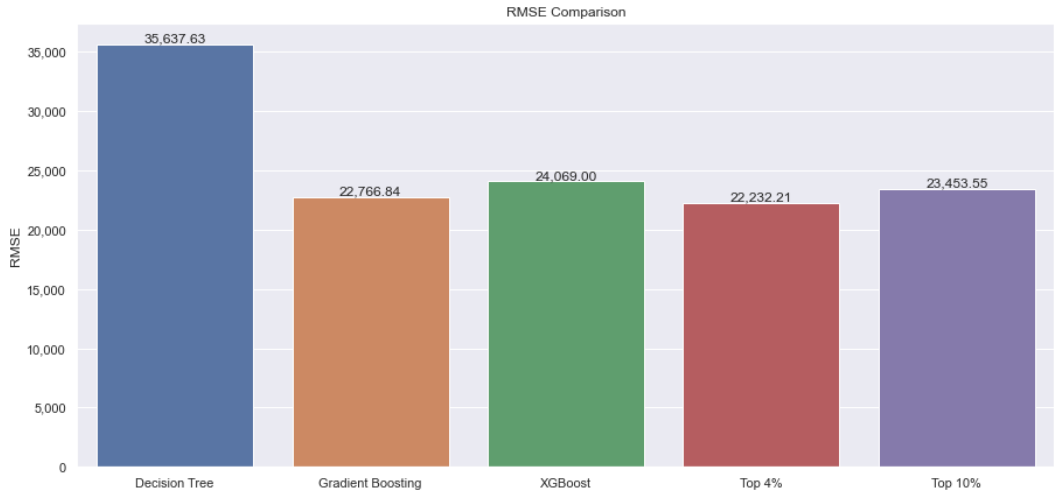

Figure 6. RMSE Comparison between Decision Tree, Gradient Boosting, XGBoost, Benchmark Top 4% and Top 10%

Table 2 Execution Time Comparison Between Each Method

Method Name	Execution time (in seconds)
Decision Tree	0.011
XGBoost	0.157
Gradient Boosting	0.208

In execution time, we try to compare the execution time for a prediction of the unseen data. As can be seen, Decision Tree is at the top due to its simple way among others. Meanwhile, the slowest one to give a prediction is gradient boosting and XGBoost is in the middle among others.

4. Conclusion

Overall, it can be said that the ensemble methods we applied in this paper, they are gradient boosting and XGBoost, perform about 35% better than a single decision tree. With the exception of the decision tree, we are aware that the RMSE value is approximately US$23,000. Also, gradient boosting outperforms the XGBoost. Under this setup, gradient boosting is fairly simple compared to the top 4% which was set up with more parameters, even though it still cannot beat the top 4% outcome. Meanwhile, to run the algorithms, the decision tree is the fastest due to its simplicity.

Based on the result, we could suggest the user to apply XGBoost in order to get the most accurate house price prediction. By this research, a buyer can figure out the house based on their budget and needs, on the other hand a seller can make a range in the market price. Further developments are welcomed to deepen the study in order to enrich the research fields especially in Indonesia.

5. References

[1] F.A. Sitanggang and P.A. Sitanggang, Buku Ajar Perilaku Konsumen. Jawa Tengah: PT. Nasya Expanding Management, 2021.

[2] N. Hadi and J. Benedict, "Implementasi *Machine Learning* untuk Prediksi Harga Rumah Menggunakan Algoritma *Random Forest*," *Computatio: Journal of Computer Science and Information Systems,* vol. 8, no. 1, pp. 50-61, April 2024.

[3] E.F. Rahayuningtyas, F.N. Rahayu and Y. Azhar, "Prediksi Harga Rumah Menggunakan *General Regression Neural Network*," *Jurnal Informatika,* vol. 8, no. 1, pp. 59-66, April 2021.

[4] M.L. Mu'tashim, S.A. Damayanti, H.N. Zaki, T. Muhayat and R. Wirawan, "Analisis Prediksi Harga Rumah Sesuai Spesifikasi Menggunakan *Multiple Linear Regression*," *Jurnal Informatik edisi ke-17,* no. 3, pp. 238-245, Dec 2021.

[5] H. Wu, J.M. Yamal, A. Yaseen, and V. Maroufy, Statistics and Machine Learning Methods for EHR Data: From Data Extraction to Data Analytics, Boca Raton, Florida: CRC Press, 2021.

[6] Z. H. Zhou, "Ensemble Methods: Foundations and Algorithms", CRC Press, 2012.

[7] Dishant Kharkar. (2023, July 16). About Boosting and Gradient Boosting Algorithm [Online]. Available: https://www.linkedin.com/pulse/boosting-gradient-algorithm-dishant-kharkar/.

[8] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, 2nd ed. New York: Springer, 2017.

[9] Scikit Learn. 1.10 Decision Trees [Online]. Available: https://scikit-learn.org/stable/modules/tree.html.

[10] D. D. Cock, "House Prices - Advanced Regression Techniques," https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques.

[11] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", Annals of Statistics, pp. 1189-1232, 1999.

[12] I. Lisnawati, "Tree-Based Ensemble Methods with An Application in House Sale Price Prediction," M.Sc. theses, Dept. Mathematics, National Central Univ., Taoyuan, Taiwan, 2022.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.