A Music Recommendation System Based on logistic regression and eXtreme Gradient Boosting

Haoye Tian
School of Big Data and Software Engineering,Chongqing University
Chongqing, China
haoyetian@cqu.edu.cn

Haini Cai
School of Big Data and Software Engineering,Chongqing University
Chongqing, China
hainicai@cqu.edu.cn

Junhao Wen
School of Big Data and Software Engineering,Chongqing University)
Chongqing, China
jhwen@cqu.edu.cn

Shun Li
School of Big Data and Software Engineering,Chongqing University
Chongqing, China
lishun@cqu.edu.cn

Yingqiao Li
School of Big Data and Software Engineering,Chongqing University
Chongqing, China
yingqiaoli@cqu.edu.cn

***Abstract*—With the rapid growth of music industry data, it is difficult for people to find their favorite songs in the music library. Therefore, people urgently need an efficient music recommendation system to help them retrieve music. Traditional collaborative filtering algorithms are applied to the field of music recommendation. However, collaborative filtering does not handle data sparse problems very well when new items are introduced. To solve this problem, some people use the logistic regression method as a classifier to predict the user's music preferences to recommend songs. Logistic regression is a linear model that does not handle complex non-linear data features. In this paper, we propose a hybrid LX recommendation algorithm by integrating logistic regression and eXtreme Gradient Boosting(xgboost). A series of experiments are conducted on a real music dataset to evaluate the effectiveness of our proposed LX model. Our results show that the error and AUC of our LX model are better than other methods.**

Keywords—recommendation system, logistic regress, xgboost, music preferences

I. Introduction

With the rapid development of big data on the Internet, the problem of information overload has become increasingly serious. For example, the Spotify music library already has 30 million songs and the number of songs grows at a rate of 20,000 songs per day. Therefore, it is difficult for people to find their favorite songs from such a rich library. In order to solve this problem, a personalized music recommendation is designed to discover suitable music for users.

Traditional collaborative filtering algorithm is widely used in music recommendation [1]. For example, Last.fm, a professional music site, uses collaborative filtering to recommend songs to their users. In the music field, a large amount of new music is added to the music library every day. Therefore, cold start problems can occur frequently. However, collaborative filtering is not good at handling cold start issues. To address this problem, some content-based recommendation methods are proposed by extracting music content features [2], [3]. Although these methods solve the cold start problem to some extent, they still fail to make full use of these high-dimensional data.

In order to extract powerful features from high-dimensional data, some advanced recommendation models are designed by exploiting the user and item information. Bartz et al. applied logistic regression(LR) to recommend a sponsored search term [4]. Their work shows that the comprehensive performance of LR is slightly better than traditional collaborative filtering. LR model is a simple linear model and does not handle non-linear features very well. To solve this problem, the tree model was introduced because of its non-linear property. For example, the eXtreme Gradient Boosting(xgboost) proposed by Chen and Guestrin is widely used [5]. Xu and Liu et al. improved Chen's work to recommend products to users on the shopping site. Their results have a better performance with higher f1-score than the baseline and have a high time efficiency [6]. However, this tree model is easy to overfit if there are few samples or too many features.

In this paper, we design an improved tree and LR fusion model to avoid the above problems. Our approach enhances LR expression capabilities by extracting non-linear features with a tree model. In the tree structure, we choose xgboost model because of its low complexity. In addition, in order to reduce over-fitting of xgboost and improve the generalization ability of the fusion model, we optimize the fusion part of the model structure by shrinking the dimensions of output data of xgboost. At the feature level, we construct the user profile from his/her historical listening behavior and cross the profile with song features to learn more potential information. Finally, we use these features to train the xgboost and LR fusion model to predict the user's music preferences.

II. Related Work

A. Logistic regression

Since the LR method was proposed, it has been applied to many fields. Bender et. employed LR models to analyze real

978-1-7281-2009-6/$31.00 ©2019 IEEE

retinal lesion data in medical research [7]. The work in [8] reduces the empirical risk of SVM output by using the results of logistic regression analysis and successfully predicts corporate financial distress in the financial domain. Maranzato et al. exercised logistic regression and step-by-step optimization for fraud detection in reputation systems [9].

In the field of the recommendation system, LR is also one of the most popular models. Quevedo et al. proposed a tag candidate set ranking method based on LR model. The results illustrate that it performs significantly better than previous methods [10]. Wang et al. merged LR and another model to increase f1-score by 2%-36% on the dataset provided by Alibaba's mobile shopping department [11].

B. Xgboost

The Tree models are often applied to overcome weaknesses in linear models. Some people solved some predictive problems by building a model with the decision tree [12], [13]. In 2001, a gradient boosting decision tree(GBDT) proposed by Friedman caused widespread interest [14]. After that, some people have achieved good results in many areas by using or improving GBDT. For example, Xie and Coggeshall made GBDT as a classifier in data mining competition to predict hospital transfer and mortality, and finally, his team ranked second in two tasks [15]. Zhang et al. proposed a two-set gradient-enhanced decision tree model to predict the gap between taxi demand and supply [16]. Experiments on real large-scale dataset prove that this method is effective on sparse data and superior to other state-of-the-art methods.

Tianqi Chen made some improvements based on GBDT. In terms of algorithms, xgboost improves the generalization ability of the model by optimizing the loss function and regularization. In engineering, xgboost supports parallel computing, which improves the speed of the model. These advantages made xgboost shine in many recent competitions.

III. The Dataset

In this paper, our dataset is provided in The Million Song Dataset(MSD) [17]. In the field of music recommendation, MSD is a very professional data website created by the National Science Foundation. It collects a lot of audio features and metadata of the million contemporary popular music tracks. Moreover, it also includes some supplementary datasets, such as Last.fm dataset, Taste Profile subset, etc. Therefore, this dataset has rich music data, such as audio, lyrics, tag, user behavior and so on.

Because of the copyright issues related to music, we can't directly get the original audio of music. So MSD provided a releasable version by extracting the relevant features from audio. And, Some other contributors also extracted some of the more plentiful features as a compliment. For example, the musiXmatch dataset has a feature mapping on lyrics so that we can get important information about the song text. Because of the contributions of these researchers, we can do a lot of interesting experiments on MSD. The datasets used in this paper is as follows:

- The Echo Nest Taste Profile Subset: 1 million users, 380,000 songs, 1.4 million user play record.
- The Last.fm Dataset: 500,000 song tags.
- The Millionsongsubset_full: 20,000 audio features.

In order to reduce the impact of the lack of audio features on the model, we use more than 20,000 samples containing audio features to train.

IV. Improve LR with Xgboost

A. LR model

As mentioned earlier, LR was widely used as a classic classifier in the field of recommendation system. In this article, we first preprocess the data to get a sample format that can be trained by our method. Then, we extract the user and song features from the dataset and perform feature engineering such as transformation and intersection on these features. Finally, we use LR model to predict the user's music preferences and compare results with our method.

The LR model is a generalized linear model. The continuous random variable X is subject to the logistic distribution, i.e. the distribution function F(x) and density function f(x) are as follows:

$$F(x) = P(X \le x) = \frac{1}{1+e^{-(x-\mu)/\gamma}} \tag{1}$$

$$f(x) = F'(x) = \frac{e^{-(x-\mu)/\gamma}}{\gamma(1+e^{-(x-\mu)/\gamma})^2} \tag{2}$$

where μ is the positional parameter and $\gamma > 0$ is the shape parameter. The F(x) and f(x) are shown in Figure 1 and Figure 2.

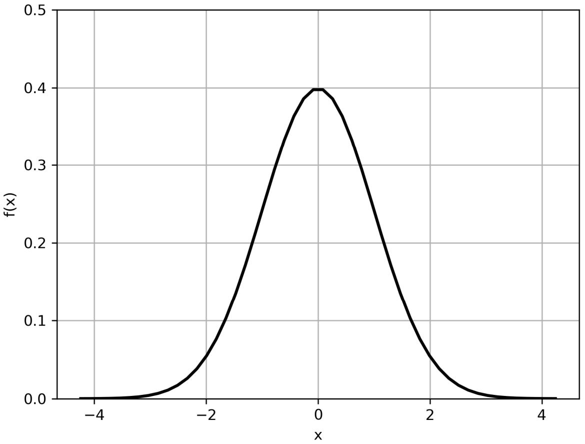

Fig. 1. The density function of LR

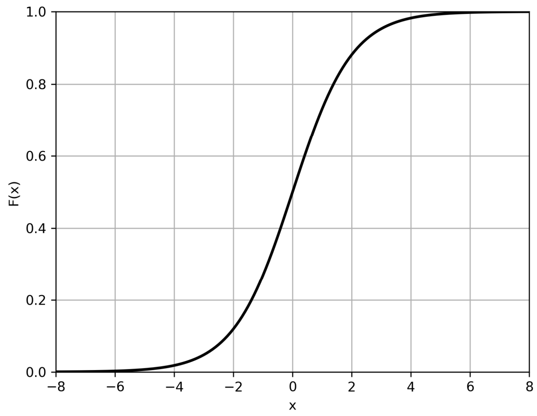

Fig. 2. The distribution function of LR

We define a vector with n user or song features $x=(x_1,x_2,\ldots,x_n)$. Let the conditional probability $p=P(y=1|x)$ be the probability of an event(y=1) occurring under x condition. Then the LR model can be expressed as

$$p=\frac{1}{1+e^{-g(x)}} \tag{3}$$

where $g(x)=w_0+w_1x_1+\ldots w_nx_n$, w is the weight of x. Now, suppose there are m samples, the labels are $y_1,y_2,\ldots,y_m$. The probability that the user like the song ($y_i=1$) under given conditions is defined as

$$p_i=P(y_i=1|x_i) \tag{4}$$

Similarly, the probability that the user does not like the song ($y_i=0$) is $1-p_i$, So the label probability of the sample can be defined as

$$P(y_i)=p_i^{y_i}(1-p_i)^{1-y_i} \tag{5}$$

Because each sample is independent each other, their joint distribution is the product of the distribution of the edges, which is defined as

$$L(w)=\prod_{i=1}^{m}p_i^{y_i}(1-p_i)^{1-y_i} \tag{6}$$

Then we apply the maximum likelihood estimation method to estimate the model parameters. We redefine (7) by employ logarithm and get

$$InL(w)=\sum_{i=1}^{m}\left(y_iInp_i+(1-y_i)In(1-p_i)\right) \tag{7}$$

The opposite of the above formula is our objective function. The derivation process also demonstrates that the LR model is a linear expression of the data. In the next section, we try to use our method to improve the non-linear expression ability of the LR model.

B. LX model

In order to exploit non-linear features in the data, people often use more complex model such as decision tree. However, only one decision tree cannot reach a very high level. So, some tree-based boosting algorithms were developed. Although these algorithms can learn the non-linear characteristics of the data, they are also subject to the over-fitting problem of the tree model. To take advantage of both the tree model and the LR model, a fusion model was proposed [18]. This fusion model utilizes tree-based boosting algorithms to extract non-linear features and then feeds these features to the LR model for training. The fusion structure is shown in Figure 3.

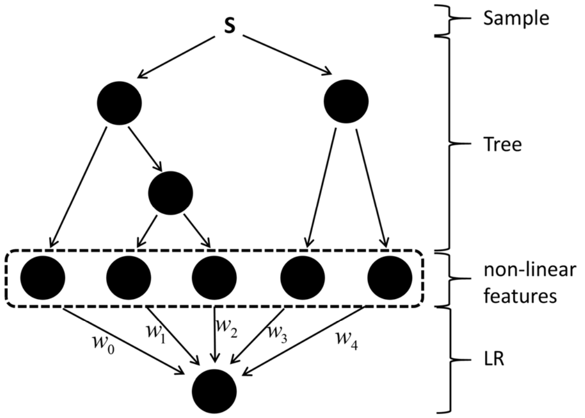

Fig. 3. Fusion model structure

There are two trees in Figure 3. S is an input sample. After traversing two trees, sample s falls in the leaf nodes of two trees. Each leaf node corresponds to the LR one-dimensional feature. The new vector value of the structure is 0 or 1. For example: for input s, suppose it falls in the second node of the left tree, coded as [0,1,0], and the first node in the right tree encodes [1,0], so the overall encoding is [0,1,0,1,0].

In this structure, the author uses the gradient boosting decision tree (GBDT) to extract non-linear features. However, in the tree-based boosting algorithms, Tianqi Chen's xgboost model is considered to be more effective. The main improvements of xgboost to GBDT are as follows:

- Xgboost adds an explicit regularization to the objective function to avoid over-fitting. In the following formula, L is the objective function of xgboost, which is defined as

$$L=\sum_i l(\hat{y}_i,y_i)+\sum_k\Omega(f_k) \tag{8}$$

where $l(\hat{y}_i,y_i)$ is the loss of the model. Ω is the regularization term and is defined as

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\|w\|^2 \tag{9}$$

where T is the number of trees and w is the weight of the leaf nodes. When we minimize the objective function, Ω reduces the complexity of the model by reducing the number of trees T and weakening the weight of the leaf nodes w.

- The first derivative of the objective function is used in the GBDT to calculate the pseudo-residue for generating the next tree model. Xgboost uses not only the first derivative but also the second derivative on the objective function by Taylor expansion. This allows the model to optimize the objective more quickly.
- The metric for finding the best segmentation point in the CART regression tree is to minimize the mean square error. The criterion for xgboost is to maximize the information gain. Xgboost will split the leaf nodes if information gain is greater than a certain threshold, which further reduces the over-fitting problem.

We propose the LX model by integrating LR and xgboost. And the framework of LX model is as follows. First of all, based on the advantages of xgboost mentioned before, we choose xgboost instead of GBDT to extract the non-linear features of the data. Salgado et al. proposed Logistic regression and Takagi-Sugeno fuzzy models, which use the method of separating binary from numerical features in the modeling process to increase the performance of a single model using both types of features together [19]. So we use a similar approach when training the model. The decision tree in xgboost is the regression tree and it is more suitable for processing continuous data. So we input the continuous features of the data into xgboost, and then splice the output of xgboost with the discrete features of the original data. Then, the output layer of xgboost is a sparse matrix, which is not conducive to the training of the LR model. Therefore, after splicing the data features, we discard the dimension with coverage below 90%. At the same time, in order to learn the information obtained by xgboost from the data, we add the predicted probability score of xgboost to the feature set. Finally, we use the feature set as the input of the LR model. The process is shown in Figure 4.

V. Experiments

In this section, we conduct some experiments on the MSD dataset to evaluate our method. We compare our LX model with the LR and xgboost models. In section A and B, in order to show the best performance of these models, we do some feature engineering and study the hyper parameters. In the last section, we compare the experimental results of the three models and draw our conclusions.

A. Feature engineering

In order to extract the non-linear feature in the data, we construct the user's preference profile which contains some important features. First, we obtain the user's profile that represents the user's preference for the artist and the genre of songs. For example, we define user u_i 'preference for rock as

$$Preference(u_i, rock) = \sum_{s_j \in S, g_j = rock} T(u_i, s_j) * C(g_j, rock) \quad (10)$$

where s_j is the jth song of the song set S, g_j is the genre of the song s_j, $T(u_i, s_j)$ is the times that u_i listens to s_j, and $C(g_j, rock)$ is the confidence of $g_i = rock$. Then, the user profile feature is crossed with the artist or genre of the current song. In this way, we get the user's artist and genre preference features for the current song: prefer_artist, prefer_genre.

We analyze all the features using the feature importance tool in xgboost. The result proves that the features we construct play an important role in the model. The result is shown in Figure 5. The y-axis is the feature name and the x-axis is the feature importance score. In the future, we will try to dig more user personalized features, such as user age, gender, and user preference for rhythm.

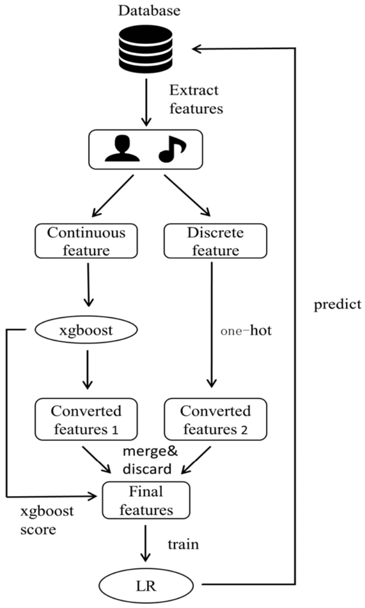

Fig. 4. The framework of LX model

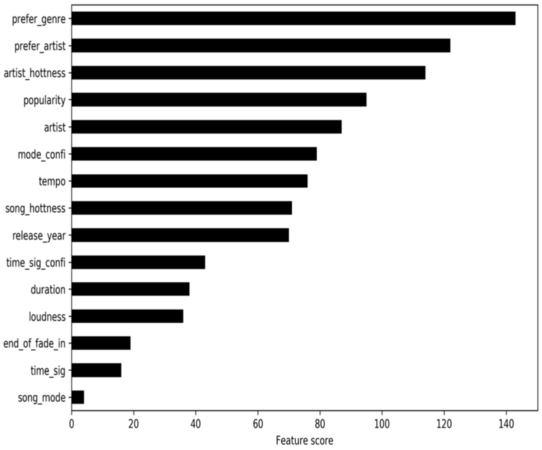

Fig. 5. Xgboost feature importance

B. Model hyper parameter

The setting of the hyper parameter may have a relatively large impact on the performance of the model, so we apply the grid-search method to search the optimal hyper parameter. In the logistic regression model, the hyper parameter C is the reciprocal of the regularization coefficient that is used to control the complexity of the model. We first observe the impact of the inverse of regularization strength C on the LR model. As shown in Figure 6.

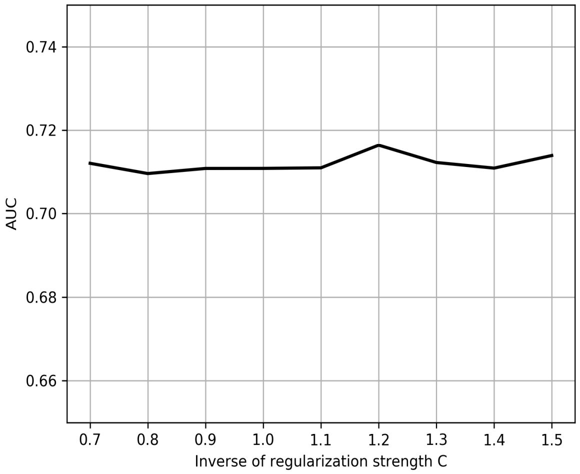

Fig. 6. Hyper parameter C on AUC of the LR

It can be seen that with the change of the hyper parameters C, the AUC of the LR model does not change significantly. Therefore, we can infer that the impact of the hyper parameters C on the LR model is small. We also observe the effect of other hyper parameters on LR, and the results are similar. Therefore, the performance of the LR model cannot be greatly improved by adjusting the hyper parameters.

In addition, we view the impact of the maximum depth of the tree on the xgboost model. As shown in Figure 7.

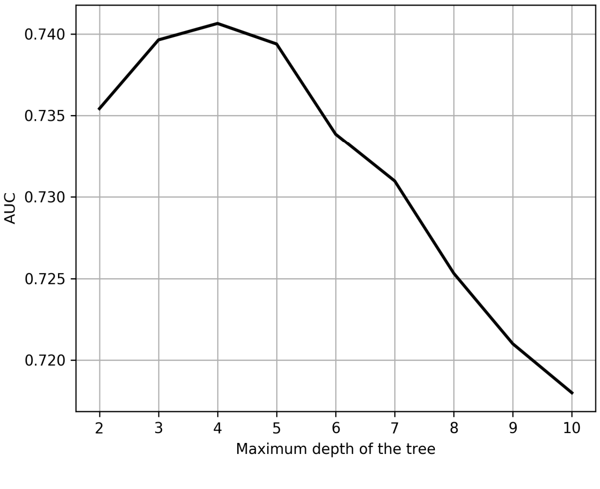

Fig. 7. The maximum depth of the tree on AUC of the xgboost

From the figure, we can clearly see that when the maximum depth of the tree is less than 5, as the maximum depth of the tree increases, the AUC of the model increases as well. However, when the tree depth is greater than 5, the AUC starts to drop very quickly. This proves that the grid-search method is helpful for the xgboost model. We employ this method to traverse the various hyper parameters of the model in the open source xgboost package of the Chen Tianqi team and find the optimal set. Some important hyper parameters results are shown in Table I.

TABLE I. OPTIMAL HYPER PARAMETERS OF XGBOOST MODEL

Parameters	**Value**
objective	binary:logistic
n_estimators	140
learning_rate	0.1
max_depth	5
gamma	0.3

After that, we compare the error rate of xgboost on the training and test set. As shown in Figure 8.

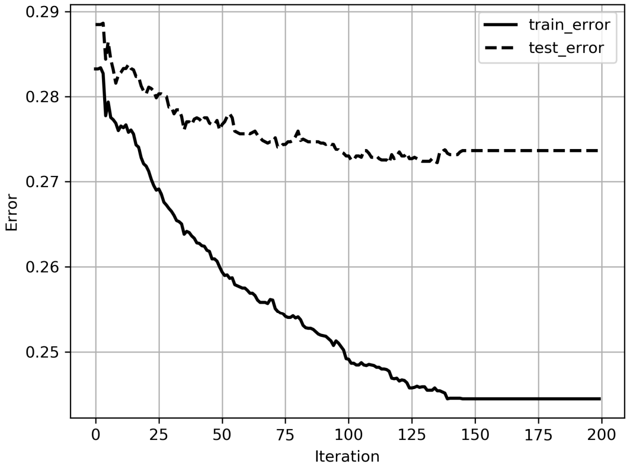

Fig. 8. The error of xgboost on the training and test set

As the iterations (the number of trees) increase, the error rate of xgboost on training and testing steadily decline. When the iterations reach around 140, the model converges and the test error no longer drops. Xgboost's error rate achieves good results by applying the early-stopping method. In addition, the gap between the training set and the test set error is small at the beginning of the training. However, as the iterations increase, the gap becomes very large. This proves that the xgboost is over-fitting and its generalization ability is weak.

C. Model evaluation

Our experimental metrics are error and AUC. The results are shown in the following Table II.

TABLE II. EXPERIMENTAL RESULTS OF THREE MODELS

Model	Error	AUC
LR	0.3062	0.7268
xgboost	0.2723	0.7663
LX	**0.2376**	**0.8087**

From the table, we can see that the AUC score of LR is lower than the scores of the other two models, which means the LR model achieve a poor prediction result. The reason behind this phenomenon is that it's difficult for LR to extract non-linear features without manual construction of large-scale feature engineering. The error and AUC of xgboost are obviously better than the LR model, which shows that xgboost has good performance under the current simple feature engineering. This proves that xgboost can handle non-linear data very well. However, as shown in Figure 7, xgboost is over-fitting, and the generalization ability on the test set does not perform well. Among these three models, the performance of the LX model is optimal, and the AUC even reaches a high score of 0.80. This demonstrates that the LX model not only extracts the non-linear features of the data but also overcomes the over-fitting problem of xgboost. Therefore, our LX model combines the advantages of the xgboost and LR models and is a high-level classifier.

VI. Conclusion and Future Work

This paper studies the prediction of users' music preferences in the music recommendation field. We adopt the fusion model of xgboost and LR as our classifier and optimize the fusion part. In terms of features, we do some feature engineering on the user's profile and proved effective by xgboost. Comparing the comprehensive performance of different models on the test set, we find that our LX model has the best results, which proves that our method has strong practicability in the field of music recommendation.

This article only studies the user's behavior information and does not combine the user's social network information. Currently, social network-based methods have proven to be effective to improve prediction accuracy in recommendation systems. So, In the future, we will try to mine the user's social relationships and combine it with our LX model to recommend music to users.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 6167060382, 61602070, in part by the New Academic Seedling Cultivation and Exploration Innovation Project under Grant [2017]5789-21.

References

[1] Sánchez-Moreno D., González A. B. G., Vicente M. D. M., Batista, V. F. L., and García M. N. M., "A collaborative filtering method for music recommendation using playing coefficients for artists and users," Expert Systems with Applications, 2016, vol. 66, pp. 234-244.

[2] Li Q., Myaeng S. H., and Kim B. M., "A probabilistic music recommender considering user opinions and audio features," Information processing & management, 2007, vol. 43, no. 2, pp. 473-487.

[3] Van den Oord A., Dieleman S., and Schrauwen B., "Deep content-based music recommendation," Advances in neural information processing systems. 2013, pp. 2643-2651.

[4] Bartz K., Murthi V., and Sebastian S., "Logistic regression and collaborative filtering for sponsored search term recommendation," Second workshop on sponsored search auctions. 2006, pp. 5.

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

[6] Xu L., Liu J., and Gu Y., "A Recommendation System Based on Extreme Gradient Boosting Classifier," 2018 10th International Conference on Modelling, Identification and Control (ICMIC). IEEE, 2018, pp. 1-5.

[7] Bender R., Grouven U., "Ordinal logistic regression in medical research," Journal of the Royal College of physicians of London, 1997, vol. 31, no. 5, pp. 546-551.

[8] Hua Z., Wang Y., Xu X., Zhang B., and Liang L., "Predicting corporate financial distress based on integration of support vector machine and logistic regression," Expert Systems with Applications, 2007, vol. 33, no. 2, pp. 434-440.

[9] Maranzato R., Pereira A., do Lago A. P., and Neubert M., "Fraud detection in reputation systems in e-markets using logistic regression," Proceedings of the 2010 ACM symposium on applied computing. ACM, 2010, pp. 1454-1455.

[10] Quevedo J. R., Montañés E., Ranilla J., and Díaz I., "Ranked tag recommendation systems based on logistic regression," International Conference on Hybrid Artificial Intelligence Systems. Springer, Berlin, Heidelberg, 2010, pp. 237-244.

[11] Wang Y., Feng D., Li D., et al. "A mobile recommendation system based on logistic regression and Gradient Boosting Decision Trees," IJCNN. 2016, pp. 1896-1902.

[12] Medina F., Aguila S., Baratto M. C., et al. "Prediction model based on decision tree analysis for laccase mediators," Enzyme and microbial technology, 2013, vol. 52, no. 1, pp. 68-76.

[13] Lee S., Lee S., and Park Y. "A prediction model for success of services in e-commerce using decision tree: E-customer's attitude towards online service," Expert Systems with Applications, 2007, vol. 33, no. 3, pp. 572-581.

[14] Friedman J. H. "Greedy function approximation: a gradient boosting machine," Annals of statistics, 2001, pp. 1189-1232.

[15] Xie J., Coggeshall S., "Prediction of transfers to tertiary care and hospital mortality: A gradient boosting decision tree approach," Statistical Analysis and Data Mining: The ASA Data Science Journal, 2010, vol. 3, no. 4, pp. 253-258.

[16] Zhang X., Wang X., Chen W., et al. "A Taxi Gap Prediction Method via Double Ensemble Gradient Boosting Decision Tree," Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2017 IEEE 3rd International Conference on. IEEE, 2017, pp. 255-260.

[17] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR), 2011.

[18] He X., Pan J., Jin O., et al. "Practical lessons from predicting clicks on ads at facebook," Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. ACM, 2014, pp. 1-9.

[19] Salgado C. M., Fernandes M. P., Horta A, et al. "Multistage modeling for the classification of numerical and categorical datasets," IEEE International Conference on Fuzzy Systems. IEEE, 2017, pp. 1-6.