2021 International Conference on Culture-oriented Science & Technology (ICCST) | 978-1-6654-4254-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCST53801.2021.00108

A Movie Score Prediction Model Based on XGBoost Algorithm

Yiying Wang[1,2],Zhe Yan[1],Lidong Xing[2]
[1] School of Automation, Harbin University of Science and Technology, Heilongjiang, Harbin 150080, China
[2] Institute of Automation, Chinese Academy of Sciences, China
retsbol0214@163.com, yanzhehrb@163.com, lidong.xing@ia.ac.cn

Abstract—**With the improvement of people's quality of life, filmmaking has become an increasingly important part of the business industry. The score prediction before the movie is released has great reference value for the research of the related recommendation work. Although several studies have been done in this field, it is still necessary to improve the prediction performance of the movie scoring model. In this paper, an effective model is developed to predict movie score by constructing the feature engineering and using XGBoost for prediction. The whole experimental procedure strictly complies with the real-world scenario. The empirical evaluation results demonstrate the outperformance of our approach has good prediction ability on the real data.**

Index Terms—**XGBoost; Prediction; Data Mining; Feature Engineering; Movie Score**

I. Introduction

Online movie communities and review aggregator sites such as Douban movie, Maoyan movie, IMDB and Rotten Tomatoes provide the estimated movie score of mass audience. The good forecast performance is able to produce a valuable advice in choosing artists for the movie producers. The improvement of audiences' expectation to the movies, in turn, can definitely increase the revenue. Movie market dynamics are also closely related to the evaluate local movie communities and review sites. Therefore,accurate prediction of movie score is informative and beneficial for audiences,movie producers, online platforms and even economic study.

According to the prediction index system adopted by researchers, the research on movie prediction can be divided into the research on movie value prediction based on Internet data and the research on movie score prediction based on movie features.

The research on movie value prediction based on Internet data aims to estimate the market value of movie, given the online reputation. However, it is traditionally difficult to screen out neutral comments for prediction from social network platform. Users' comments may be biased. Besides, as compared to the features of movie, it is trickier to quantify the features of network reputation. For example, there are no unified metrics to measure the relevance between the user's emotions expressed in the comments of different platforms and the corresponding evaluated movies, which are crucial for screening out the highly relevant comments to the movies. Furthermore, these online reputation are usually dynamic rather than constant. Thus, we do not consider using data containing users strong subjective emotions for score prediction.

The research on movie score prediction based on movie features aims to estimate the movie score by using machine learning method to make score prediction after acquiring movie feature data. Formally, movie feature data are composed of the movie attributes feature and the production participants feature. The movie attributes feature here can be regarded as the related feature of the movie content such as theme, release year, soundtrack and so forth. The production participants feature here can be regarded as the aggregated level of participants such as director, screenwriter and the main actors. Usually, the movies produced by high level teams will have higher market popularity.

Within this perspective of score prediction, it is necessary to consider all possible feature to make a higher accurate rate in scoring model. In the existing research, due to the poor performance of the selected algorithm, the insufficient data screening and feature engineering processing of the collection, the model usually can not achieve a good prediction performance. In this paper, we propose a model which combines the processing of feature engineering to ensure the accuracy of prediction. We evaluate the linear correlation strength between two features by calculating the value of Pearson coefficient, and combine or eliminate features according to the calculation results. This strategy ensures that the features used in the training model have a high correlation with the movie score. In addition, the XGBoost algorithm, which shows a strong generalization ability and excellent regression performance in the field of predictive regression, is used to predict the score of movies.

We conduct comprehensive experiments on a real-world movie score dataset. We demonstrate that the proposed model can achieve a good performance in prediction.

II. Related Theory and Model

A. Feature Engineering

Feature engineering is a very important concept in the field of machine learning. Usually, it can be regarded as the work of machine learning application design feature set, and the quality of Feature Engineering essentially determines the upper performance limit of the whole model [1]. One of the most important problem is to extract effective and high correlation features from existing historical data. The quality of feature engineering will directly affect the prediction effect of the model,and has a significant impact on the space-time complexity and convergence rate of the model.

Data is the carrier of information. The original data usually contains lots of noise with the insufficiently concise information presentation. In order to obtain better training data, the purpose of feature engineering is to use a series of engineering processes to obtain data which can most effectively represent the information of the feature. In addition, the new coding method needs to minimize the impact of uncertainty in the original data.

B. XGBoost Algorithm

XGBoost is an efficient system implementation of gradient promotion. Based on the idea of random forest algorithm, XGBoost will not only pre-ordering the features, but also supports column sampling to reduce over fitting and computational complexity. In the process of tree splitting, the feature with the largest gain in the current target is selected to split in each time, and the minimum value of loss function is calculated. When the gain of splitting is less than the threshold, the splitting can be removed. When splitting reaches the maximum depth of the tree, the tree stops splitting [2]. The algorithm establishes the model according to the distribution and selects the direction of gradient descent in the continuous updating iteration to ensure the optimal prediction result [3].

a) Tree model: In the process of tree model construction, each layer greedily selects a feature segmentation point as the leaf node, which makes the gain of the whole tree maximum after splitting. The more times a feature is split, the more benefit it brings to the whole tree, and the more important the feature is [4]. Similarly, the larger the average gain of each feature is, the more important the feature is. During the split process, the weight of each leaf node can be expressed as $w(gi, hi).g_i$ and h_i are defined as follows.

$$g_i = \partial \hat{y}_{(t-1)} l(y_i, \hat{y}^{(t-1)}) \tag{1}$$

$$h_i = \partial^2 \hat{y}_{(t-1)} l(y_i, \hat{y}^{(t-1)}) \tag{2}$$

Where $l(y_i, \hat{y_i})$ is used to calculate the difference between the target values y_i and predicted values $\hat{y}$. For each split point, the gain can be expressed as the total weight after splitting minus the total weight of the leaf node before the split. The gain is defined as follows.

$$Gain = \sum_{newleft} w + \sum_{newright} w - \sum_{nosplit} w \tag{3}$$

b) The combination of decision trees: As a non-parametric supervised learning model, decision tree model is often used for classification and regression. The model can quickly find decision rules according to the characteristics of the data without any prior assumptions on the data. XGBoost adopts the integration strategy on the basis of decision tree. By using the gradient lifting algorithm, XGBoost continuously reduces the loss of the previous decision tree and produces a new tree composition model to ensure the reliability of the final decision [5]. XGBoost will add a tree in each iteration, and the linear combination expression of n trees is defined as follows.

$$\widehat{y_i}^{(t)} = \sum_{k=1}^{n} f_k(x_i) = \widehat{y_i}^{(t-1)} + f_t(x_i), f_k \in F \tag{4}$$

Where F is the function space containing all trees, and $f_k(x_i)$ is the weight of the i-th sample to the leaf node of the k-th tree. In order to minimize the cost of the segmented tree, each feature is considered as the gain of the split point according to the weight of all leaf nodes.

c) Importance measure: The importance measurement index is a way to evaluate the importance of each feature in its feature set. XGBoost constructs the decision tree according to the number of the feature splits(FS), average gain(AVG) and average cover(AVC) to complete the classification task accurately. For the above important indicators, the expressions are defined as follows.

$$FS = |X| \tag{5}$$

$$AVG = \frac{\sum Gain_{\bar{X}}}{FS} \tag{6}$$

$$AVC = \frac{\sum Cover_{\bar{X}}}{FS} \tag{7}$$

X is the set of features to be classified into leaf nodes. In equation(6),Gain is the gain value of every leaf node in X which is divided by equation(3) 2. In equation(7), Cover is the number of samples falling on each node in X.

d) Objective function: The objective function constructed by XGBoost is defined as follow [6].

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

Function L is a loss function and a differentiable concave function.Function Ω is a regular term, which is used to reduce the complexity of the model and prevent over fitting.The Ω expression is defined as follows.

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

T is the number of leaf nodes in the tree model, which provides control over the number of leaf nodes. The larger the γ is, the smaller the number of leaf nodes. $\frac{1}{2}\lambda\|w\|^2$ is expressed as a regular term, w is the weight of leaf nodes, which ensures that the model will not be over fitted due to excessive weight.

In Euclidean space, the above objective functions are more difficult to optimize than traditional methods. Therefore, the model is trained by superposition, and a base classifier is added in each iteration. The added objective function expression is defined as follows.

$$L^{(t)} = \sum_{i}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{10}$$

After deleting the constant term in equation(10) and combining equation(1) and equation(2), equation(11) is obtained as follow.

$$L^{(t)} \approx [\sum_{i}^{n} l(g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{11}$$

Set the loss function l be the square difference function. Function l is defined as follows.

$$l = \frac{1}{2}(y - f(x))^2 \tag{12}$$

From equation(1),equation(2) and equation(12), we can get equation(13)and equation(14).

$$g_i = \frac{\partial l}{\partial f} = f(x_i) - y_i \tag{13}$$

$$h_i = \frac{\partial^2 l}{\partial f^2} = 1 \tag{14}$$

Set the set of leaf nodes j as $I_j = \{i|q(x_i)\}$, set w_i as the weight of leaf nodes, and the objective function is defined as follows.

$$\tilde{L}^{(t)} = \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_i + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \tag{15}$$

When $q(x)$ is given, the optimal node weight w_j can be calculated by formula(16).

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{16}$$

After obtaining the optimal node weights, the optimal objective function calculation expression can be be calculated by formula(17).

$$\hat{L}^{(t)}(q) = -\frac{1}{2} \sum_{i=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{17}$$

III. Method and Material

A. Experimental Data

The audience's evaluation of movies in different countries is inevitably influenced by the culture, values and other factors of their region. Thus, choosing a domestic movie scoring platform can effectively reflect the Chinese people's preferences and evaluation criteria of watching and selecting movies. Douban Movies is currently the most neutral movie review website reflecting the evaluation of Chinese audiences, less affected by the screeners or fans. In this paper, we use Douban's movie rating data as the data source to analyze and predict the score. Besides, we construct the web crawler through Python to achieve the acquisition of data.

Initial data collected includes the basic information data of the film and the data of the creators. The basic information of the movie includes the following data: Douban ID, movie name, list of creators, release year, movie category, number of raters and score. Specific data characteristics are shown in TABLE I.

TABLE I
Movie Basic Information

Data Item	***Data Meanings***
mov_dbid	uniquely identifies the movie
mov_name	the name of the movie
main_actor	the list of actors
director	Name of director
screenwriter	Name of screenwriter
mov_type	The theme list of the movie
score	Movie score on Douban
scorepeople	The number of people who scored the movie
year	the year of movie release time

Besides, basic information of the creators mainly includes the average score of total work and the average score of latest five data works from director, screenwriter and the top three leading actors in the film. Specific data are shown in TABLE II.

TABLE II
Data of main participants in film production

Data Item	***Data Meanings***
p_dbid	ID of Douban's artist information
p_name	the name of the artist
p_nn	the average score of historical works
p_tn	average score of nearly five participating works

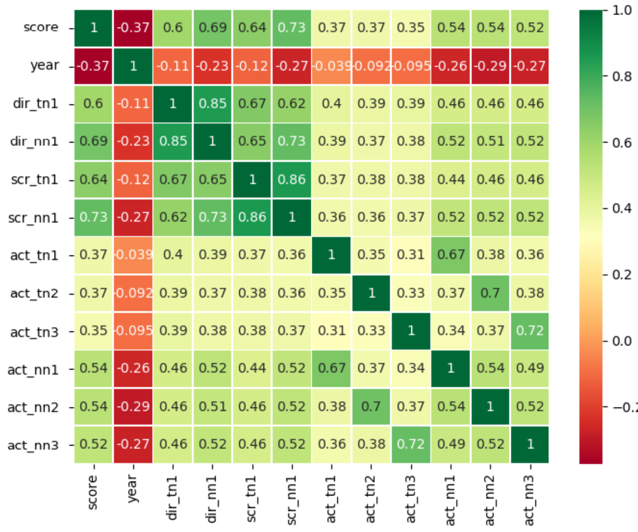
B. Data Preprocessing and Feature Engineering

a) Feature Cleaning: In this process, data items with missing values are deleted. Only the data items with sound data are kept for model training to ensure good model training results. At the same time, we delete the data with less than 550 ratings to ensure that the rating data is more neutral to the audience's evaluation and reduce the impact of uncertainty in the data.

b) Correlation analysis: Due to the data obtained from the Internet often contains a lot of noise, in order to improve the prediction model, it is necessary to extract the most effective and highly correlated features for prediction [7]. During using a tree model, several highly correlated features together may squeeze out other better features when the tree splits [8]. Redundant prediction features will not improve the prediction performance of the model, but reduce the accuracy of the model. Therefore, before training, we should find the potential relationship between the data by using correlation analysis. Meanwhile, further preparation and prediction should be made based on the selection of the data with the highest correlation of the prediction target. In this process, Pearson correlation coefficient is selected as the index of correlation analysis to measure the degree of correlation. Pearson correlation coefficient is derived by dividing the covariance by the standard deviation of two variables, and its formula is expressed as follow.

$$P_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (18)$$

A correlation analysis is now performed on numeric data items, and a correlation thermogram is constructed as shown in Figure 1.



	score	year	dir_tn1	dir_nn1	scr_tn1	scr_nn1	act_tn1	act_tn2	act_tn3	act_nn1	act_nn2	act_nn3
score	1	-0.37	0.6	0.69	0.64	0.73	0.37	0.37	0.35	0.54	0.54	0.52
year	-0.37	1	-0.11	-0.23	-0.12	-0.27	-0.039	-0.092	-0.095	-0.26	-0.29	-0.27
dir_tn1	0.6	-0.11	1	0.85	0.67	0.62	0.4	0.39	0.39	0.46	0.46	0.46
dir_nn1	0.69	-0.23	0.85	1	0.65	0.73	0.39	0.37	0.38	0.52	0.51	0.52
scr_tn1	0.64	-0.12	0.67	0.65	1	0.86	0.37	0.38	0.38	0.44	0.46	0.46
scr_nn1	0.73	-0.27	0.62	0.73	0.86	1	0.36	0.36	0.37	0.52	0.52	0.52
act_tn1	0.37	-0.039	0.4	0.39	0.37	0.36	1	0.35	0.31	0.67	0.38	0.36
act_tn2	0.37	-0.092	0.39	0.37	0.38	0.36	0.35	1	0.33	0.37	0.7	0.38
act_tn3	0.35	-0.095	0.39	0.38	0.38	0.37	0.31	0.33	1	0.34	0.37	0.72
act_nn1	0.54	-0.26	0.46	0.52	0.44	0.52	0.67	0.37	0.34	1	0.54	0.49
act_nn2	0.54	-0.29	0.46	0.51	0.46	0.52	0.38	0.7	0.37	0.54	1	0.52
act_nn3	0.52	-0.27	0.46	0.52	0.46	0.52	0.36	0.38	0.72	0.49	0.52	1

Figure 1. Feature correlation heat map

- If a pair of data items Pearson coefficient is greater than 0.6, which means the high correlation between the features, the features are merged. Among them, the pair of features with the largest Pearson coefficient is the priority choice to perform feature merging. For feature A and feature B that perform feature merging, merge rules is defined as follows.

$$NF = FA\frac{P_{A,T}}{P_{A,T}+P_{B,T}} + FB\frac{P_{B,T}}{P_{A,T}+P_{B,T}} \quad (19)$$

T is the target item. FA is the data item of feature A, FB is the data item of feature B. $P_{A,T}$ is the Pearson coefficient for A and the target item, and $P_{B,T}$ is the Pearson coefficient for B and the target item. According to Figure 2, the data groups that are highly related to score and require feature merging are listed in TABLE III.

TABLE III
HIGH CORRELATION FEATURE TABLE

Feature1	*Feature2*	*Correlation coefficient*
dir_tn1	dir_nn1	0.85
scr_tn1	scr_nn1	0.86
act_tn1	act_nn1	0.67
act_tn2	act_nn2	0.70
act_tn3	act_nn3	0.72

The data items in TABLE III are combined into new data items according to the equation(19) to quantify the feature level of the corresponding object.

- The correlation coefficient between features that not shown in Figure 1 and score is less than 0.3. In this process, we can consider these features to be almost irrelevant to the movie score. Therefore, these features that not shown in Figure(2) were not considered in subsequent experiments.

c) Converting the categorical to numerical: In machine learning tasks, features are not always continuous values, it may be discrete classification values. Features represented by numbers will be more effective [8]. Since classifiers often default to continuous and ordered data, data cannot be used directly in classifiers when converted to a numerical representation unique heat encoding [9]. In this process, data items of the movie category are uniquely encoded by using the get_dummies function that comes from Scikit-learn library.

C. Basicline Model

In this section, we will introduce alternative algorithms as baseline model to demonstrate the effectiveness of the proposed model.

- **Random Forest:** Random forest is an ensemble algorithm. Ensemble algorithms are those which combines more than one algorithms of same or different kind for classifying objects [9]. Random forest classifier creates a set of decision trees from randomly selected subset of training set and then aggregates the votes from different decision trees to decide the final class of the test object.
- **Adaboost:** Adaboost combine multiple classifiers with selection or training set at every iteration and assigning right amount of weight in final voting to have good accuracy score for overall classifier retrains the algorithm iteratively by choosing the training set based on accuracy of previous training. The weight of each trained classifier at any iteration depends on the accuracy achieved. Each weak classifier is trained using a random subset of overall training set [10].
- **K-Nearest Neighbors:** K-Nearest Neighbors is one of the most basic essential classification algorithms in Machine Learning. It find the set of K nearest neighbors in the training set to the predictor vector. Nearest neithbors is an extremely flexible classification scheme, and dose not involve any preprocessing of the training data [11]. It is widely disposable in real-life scenarios since and it is non-parametric, which meaning it does not take any underlying assumptions about the distribution of data. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data.

D. Model Building

This experiment uses Python language programming and builds a model based on the Scikit-learn machine learning framework. Before the training, we used the multiparameter grid search to determine the i parameters that need to be determined. Set k as the current adjustment parameter. The steps of using grid search method to determine model parameters are described as follows [12]. Firstly, we need to determine the feasible region of the k parameter and select several equidistant or non-equidistant feasible values while other parameters are unchanged. Secondly, the optimal values of k parameters are determined by Cross-validation. Finally, the k value is replaced by the k+1 parameter to optimize the determination process. Optimal parameters are detailed in TABLE IV below.

TABLE IV
MODEL PARAMETER TABLE

Parameter Name	*parameter values*
learning_rate1	0.02
max_depth	5
max_delta_step	2
n_estimators	699
subsample	0.7
colsample_bytree	0.9
reg_alpha	1
reg_lambda	1
scale_pos_weight	0.2

After the model parameters are determined, all the clean dataset were divided to training set and testing set with Cross-validation for classification and prediction.

E. Evaluation index

This study mainly evaluates the model effect by using MSE and goodness of fit as evaluation index.

MSE is a convenient method to measure the average error and it can evaluate the degree of change of the data. The smaller the value of MSE, the more accurate the prediction model can be to describe the experimental data. The calculation formula of MSE is defined as follows.

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

In statistics, the value of goodness of fit reflects the relative degree of regression contribution and the percentage of the total variation of the dependent variable Y that the regression relationship can account for. This parameter can be used to judge the ability of statistical models to fit data. Its formula is defined as follows.

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(\bar{y}_i - y_i)^2} \tag{21}$$

The closer the R^2 value is to 1, the better the regression line fits the observed values. Conversely, the smaller the R^2 value is, the worse the regression line fits the observed values.

IV. ANALYSIS OF RESULTS

The experimental results are shown in TABLE V.

From the above comparison experiment on prediction MSE and goodness of fit, it can be concluded that the XGBoost algorithm has significantly better prediction performance than other algorithm. The MSE in predicting movie scores is much lower than other algorithms, and the goodness of fit reaches 0.7165, which shows that the model performs best in data fitting compared with other algorithm models.

TABLE V
EVALUATION OF REGRESSION MODEL

Algorithm	*MSE*	*Goodness of fit*
XGBoost	0.6238	0.7165
Random forest	0.6922	0.6854
Adaboost	0.8117	0.6314
KNN	0.8244	0.6253

In order to further verify the effect of the model, we use the model to predict the scores of the latest movies. In the process of using the model to predict the actual world data, we found that for some artists, there is no historical data in the database due to they are the first time to participate in film production. In order to solve the problem of data vacancy, we propose the following methods to fill the data.

Set the movies that need to be predicted contain N types of movie theme, and movie participants who lack initial data are listed as P_{work}. P_{init} is the initialization value of the data to be filled. For participant data items that lack initial data, the initialized score data is calculated based on the equation(22).

$$P_{init} = \frac{1}{N}\sum_{i=1}^{N} Score_{P_{work}} \tag{22}$$

We used the trained model to predict the latest movies, the results are shown in TABLE VI. Besides,the data missing during the movie prediction process is shown in TABLE VII.

TABLE VI
PREDICITION OF THE LATEST MOVIES

Order	*Movie*	*Prediction*	*Real score*
1	Cliff walkers	7.95	7.6
2	On your Wedding Day	6.41	5.1
3	Home Sweet Home	5.91	5.7
4	Tiger Robbers	6.13	4.3
5	Dynasty Warriors	5.75	4.1
6	Break Through the Darkness	6.82	6.5
7	Once Upon a Time in Hong Kong	5.81	5.5
8	Miss Mom	7.25	6.3

TABLE VII
INFORMATION OF DATA MISSING

Order	*Missing item*
1	-
2	Screenwriter
3	Screenwriter
4	-
5	-
6	Screenwriter
7	Screenwriter
8	The third main actor

According to the standard that movie score in Douban less than 6.0 is regarded as rotten movie, we can see from TABLE V that the rating model misclassifies movies No.2 and No.4. To find the reason for this bias, we went to the movie reviews for the corresponding movie. A lot of reviews number of movie reviews showed that the poor evaluation of the two movies were caused by their awful movie plots. In Movie No.2, since there is no historical data for the screenwriter in the database, the initial calculation of the screenwriter data based on equation(22) is needed to fill in the gap before the predictive score. The initialization of data comes from the calculation of the whole movie data, which is a reasonable performance due to the deviation from the actual performance of the screenwriter.

In Movie No.4, although the data is complete, there is still a significant score bias. From the historical data of the No.4 movie screenwriter, it is found that the overall score of his work has a distinct downward trend over time. However, in this experiment, the feature quantification process does not take into account the influence of the score trend, but only takes the average value as the feature quantification value for all kinds of historical data. It is a difficult problem to find out a better quantitative indicator to reflect the trend of creative performance based on the participants' historical score sequence, which needs to be studied in the future.For the above problems, this prediction model still needs further research and improvement.

In addition, other movies in TABLE VI have achieved the correct movie classification with a small deviation. This result shows that the model has a good prediction performance for most movies. Overall, the prediction model studied in this experiment has a good prediction performance for the actual data, and its bias is within the acceptable range.

V. Conclusion

In this study, a movie scoring model is implemented based on XGBoost algorithm by selecting high correlation features as prediction variables and combining with feature engineering processing.

The experimental result shows that the model performs well on the test set and has good fitting ability to the real data. This model can provide valuable reference for rating prediction for the general public. In addition, the way to construct this prediction model can be generalized to other movie scoring websites.

VI. Acknowledgements

This work was supported by the National Key R&D Program of China (2020YFB1406900) and Technology Program of Beijing (Z201100001820002). It was also the research achievement of the Key Laboratory of Digital Rights Services.

References

[1] M. Oyamada, "Extracting Feature Engineering Knowledge from Data Science Notebooks," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 6172-6173.

[2] Z. Wen, J. Shi, B. He, J. Chen, K. Ramamohanarao and Q. Li, "Exploiting GPUs for Efficient Gradient Boosting Decision Tree Training," in IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 12, pp. 2706-2717.

[3] Y. Sun and G. Yang, "Feature Engineering for Search Advertising Recognition," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 1859-1864.

[4] G. Duan and X. Ma, "A Coupon Usage Prediction Algorithm Based On XGBoost," 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2018, pp. 178-183.

[5] M. Gumus and M. S. Kiran, "Crude oil price forecasting using XGBoost," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 1100-1103.

[6] Chen, T. , and C. Guestrin . "XGBoost: A Scalable Tree Boosting System." the 22nd ACM SIGKDD International Conference ACM, 2016.

[7] R. Dhir and A. Raj, "Movie Success Prediction using Machine Learning Algorithms and their Comparison," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 385-390.

[8] R. Punmiya and S. Choe, "Energy Theft Detection Using Gradient Boosting Theft Detector With Feature Engineering-Based Preprocessing," in IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 2326-2329.

[9] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[10] P. Peng, Y. Zhang, Y. Wu and H. Zhang, "An Effective Fault Diagnosis Approach Based On Gentle AdaBoost and AdaBoost.MH," 2018 IEEE International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2018, pp. 8-12.

[11] Jing Peng, D. R. Heisterkamp and H. K. Dai, "LDA/SVM driven nearest neighbor classification," in IEEE Transactions on Neural Networks, vol. 14, no. 4, pp. 940-942.

[12] HuaZhou Chen, Fu Chen and LiLi Xu, Near-infrared LSSVM quantitative analysis of fish meal ash based on parameter optimization method based on grid search, Journal of Analytical Science, vol:32(2), 2016, pp:198-202.