2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE) | 979-8-3503-5317-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CISCE62493.2024.10653289

Commodity demand forecasting of e-commerce merchants based on the Stacking fusion model

Jiayao Guo*
Northwestern Polytechnical University, Xi'an, Shaanxi, China
17835878818@mail.nwpu.edu.cn

Abstract: **With the rise of e-commerce platforms and the continuous expansion of market scale, how to supply and distribute and manage the goods in e-commerce supporting warehouses is a factor that needs to be considered by every merchant. In order to reduce merchants' inventory costs and fulfill commitments on time, this paper establishes a forecasting model based on the Stacking fusion model combined with grid search. With "merchant - warehouse - commodity" as a time series about commodity demand on the time axis, the Stacking fusion model is established. It integrates XGBoost, LishtGBM and Catboost machine learning models, and adopts grid search and cross-validation methods for parameter optimization. Prediction accuracy, recall rate, goodness of fit coefficient, root-mean-square error and 5% error accuracy were used to evaluate the prediction performance.**

Keywords: Stacking fusion model; XGBoost; LishtGBM; Catboost; Grid search; Cross verification;

I. INTRODUCTION

A. Background introduction

With the increase of business in the e-commerce industry, the number of merchants is also increasing year by year. The e-commerce platform will unify the management of the goods of many merchants, so reasonable management and planning will save more costs. The knowledge of big data can be used to carry out reasonable planning and management, which mainly includes demand forecasting and inventory optimization.

Demand forecasting enables managers to predict local needs in advance and make corresponding decisions. However, many problems need to be considered in the corresponding prediction of people, which will bring great difficulty to the demand prediction. Therefore, it is necessary to design certain algorithms to find out the rule of historical data and find out similar situations, so as to achieve accurate prediction.

Inventory optimization is a method commonly used by enterprises to control the frequency and quantity of goods replenishment, which can save inventory costs and improve the efficiency of inventory management.

B. Research introduction

We have collected the demand quantity of various commodities in various warehouses of some e-commerce companies in Henan, China from 2022-5-15 to 2023-5-15, and the data set is arranged according to "merchant number - warehouse number - product number".

Based on the historical data of the e-commerce company, a model was built to forecast the demand of the goods of each merchant in each warehouse from 2023-05-16 to 2023-05-30, and the performance of the established model was evaluated after the prediction.

We first perform feature engineering processing on data, build a Stacking fusion model, select optimal parameters using grid search and cross-validation methods, and fit the pre-processed data sets as training sets. Finally, accurate prediction results are obtained, and indicators are judged by model fitting. Analyze the predictive performance of different machine learning models and fusion models.

C. Algorithm introduction

The Stacking model itself is a hierarchical model that can be divided into multi-level fusion. The principle is that the output results of the upper layer are used as new data training sets to transport the learners of the lower layer. In general, in order to prevent overfitting, a simple model with strong structural stability is selected as a two-layer classifier [1]. In order to prevent the overfitting of the model and make the final prediction result more accurate, we use Internet search for parameter selection and crossover validation for analysis, which can usually be selected.

XGBoost is a powerful Gradient Boosting framework for solving a variety of machine learning problems, including classification and regression. XGBoost adopts the idea of gradient lift algorithm, and gradually optimizes the loss function to reduce the prediction error by integrating multiple decision tree carts (weak learners). It has a good effect on accuracy, interpretability and performance, mainly reflected in the use of the first and second derivative of the loss function when solving the optimal solution of the objective function, and also added the regular term, so that the approximate optimization of the objective function is closer to the actual value.

II. DATA PREPROCESSING AND FEATURE ENGINEERING

(1) Division of sub-data

According to "merchant-warehouse-goods" as the division unit of the data set, the total data set is divided into the prediction sub-data set of the goods in each warehouse corresponding to each merchant.

(2) Detection and elimination of outliers

In the original data set, due to the long time span, missing values and outliers appear in the original data set, and there is a large gap between some purchase quantity data, so this paper uses $3\hat{\partial}$ criteria to test outliers and remove them.

A. *Feature engineering*

Based on the features of the data set, such as purchase quantity, date, commodity primary classification, commodity secondary classification, commodity tertiary classification, merchant scale, warehouse type, merchant code, commodity code and warehouse code, this paper carries out feature engineering on the basis of features, as some features have different effects on the final purchase quantity and these features will affect the final prediction accuracy. The original features are extracted and reconstructed.

(1) Timing characteristics

Since the purchase time of each commodity already exists in the data set, this paper reconstructs the timing of each subdata set divided above on the basis of it, assigns the starting time as "1", and then adds one incrementally.

(2) Characteristics of commodity purchase proportion

This paper measures the proportion and importance of each commodity in the stock of the warehouse and merchants according to the proportion of the goods purchased in the previous year.

(3) Commodity binary coding reconstruction

In Table 1, we performed binary processing on the data.Because the commodity is coded in the form of "first-second-third level", this paper converts the codes of the three levels into three 8-bit binary numbers respectively, and then converts them into decimal, to obtain new commodity coding features.

Table 1: Coding refactoring table

	Decimal coded	Binary coding		Reconstructed decimal
Primary	0~19	00000000~ 00010011	⇒	0~255
Secondary	0~61	00000000~ 00111101		
Three-level	0~209	00000000~ 11010001		

III. STACKING FUSION PREDICTION MODEL

Considering that merchants, warehouses and commodities each have different characteristics in three dimensions, four machine learning models, XGBoost, LishtGBM, catboost and Stacking fusion model, are selected to highlight their characteristics and improve the prediction results of the model. Various machine learning models will be discussed and established below.

When machine learning models are used alone, they may have poor generalization performance and be easily trapped in local minimums in the face of large data Spaces. Therefore, integrated learning can be used to improve the models. The integrated Stacking algorithm meets this requirement. Like other machine learning machine, it needs to use the initial data for corresponding training, parameter adjustment, fitting and other work, and then use the predicted results of the initial learning machine to carry out the learning and training process of the secondary layer.

LightGBM is an efficient gradient lifting method that delivers high performance and scalability while finding a balance between accuracy and efficiency. It uses a gradient lifting algorithm and a decision tree algorithm based on histogram to speed up the training process.

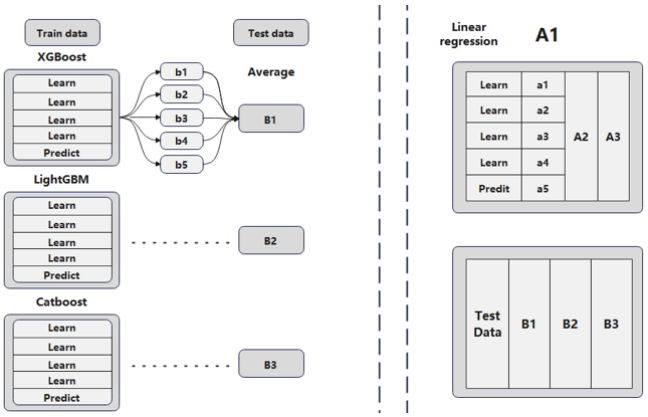

Figure 1: Cross-validated Stacking fusion model diagram

In Figure 1, A formula is used to describe its operation mechanism, assuming that the data set is $I=\{a_i,b_i\}(i=1,2,\cdots,n)$, where the a_i table is used to input the sample characteristics, and b_i represents the prediction label. I is now divided into two parts, one for the training data set and one for the test data set. The base layer model D_K is trained through the training set divided by 50% to get the base learner, using $Y_i,(i=1,2,\cdots,n)$ expression, [2] and in this case, we choose three single learners, then n=3. The test set and the original test set are input for prediction, and the new data result sample is obtained. Next, the new sample data set is input into the second level, that is, secondary learning training, which can adopt regression model or a more stable learner, and finally get the final prediction result, which is expressed by the following formula:

$$Y_{last}=Y_i(Y_1(a_i),Y_2(a_i),Y_3(a_i)) \tag{1}$$

A. *The XGBoost model*

Given the data set $I=\{(c_i,d_i)\}(|I|\in N,c_i\in R,d_i\in R)$, a certain prediction sample is represented by c_i, and the results obtained are expressed by the following formula:

$$\hat{d_i}=\sum_{t=1}^{T}g_t(c_i),(i=1,2,\cdots.n,g_t(c_i)\in G) \tag{2}$$

Where T is how many trees there are; g_t stands for regression tree; G represents a function space formed by the regression tree, which has the following relation:

$$G = \{g_t(c_i) = \sigma_{e(c)}\}, (\sigma \in R^T) \tag{3}$$

Among them, $g_t(c_i)$ represents the prediction score of T classifiers in the model for the predicted sample, σ represents the label of each leaf in the regression tree, and $e(c)$ represents a mapping relationship that maps the sample to the corresponding leaf.

The predicted score of leaf nodes on the tree is calculated by the following formula:

$$\alpha = \sum_{i=1}^{n} s(d_i, \hat{d}_i) + \sum_{t=1}^{T} \varepsilon(g_t) \tag{4}$$

Where $s(d_i, \hat{d}_i)$ represents the loss function and g represents the regularized object, which is the function that sums the complexity of the entire tree.

For the representation of complexity, the XGBoost regularization object can be measured, there is the following formula:

$$\varepsilon(g_t) = \mu N + \frac{1}{2}\beta \sum_{k-1}^{N} w_k^2 \tag{5}$$

Where μ and β are constants, and N represents the total number of leaf nodes. Bring it into the above objective function, after a series of transformation simplification process, and XGBoost is a forward distribution algorithm, so the greedy algorithm can be used to find the local optimal solution. [3] The optimal features and partition nodes of the model can be calculated by the following formula:

$$Best = \frac{1}{2}\left[\frac{(\sum_{i\in L} O_i)^2}{\sum_{i\in L} P_i + \beta} + \frac{(\sum_{i\in R} O_i)^2}{\sum_{i\in R} P_i + \beta} - \frac{(\sum_{i\in (L+R)} O_i)^2}{\sum_{i\in (L+R)} P_i + \beta}\right] - \mu \tag{6}$$

B. Overview of the LightGBM model

LightGBM optimizes the branch growth of the growing tree, carries out targeted search on the same layer of leaves in the traversal process, and splits the leaves with the largest split gain in the layer by judging that the single branch growth is too deep, which also prevents overfitting, and makes up the cost loss of the original XGBoost algorithm for traversing the same layer of leaves. It also improves speed and accuracy.

C. Catboost model

The advantage of catboost is that it can automatically process the encoding and transformation of class features and effectively utilize this information during training. It can automatically process the coding of category features, without the need for manual thermal coding or label coding. In addition, catboost also uses a sort-based method when generating trees to better deal with category features, thereby improving the accuracy of prediction [5]. The weight calculation method is as follows:

$$\frac{\sum_{l=1}^{p-1}[h_{ol}, R = h_{op}] \cdot U_{ol} + c \cdot P}{\sum_{l=1}^{p-1}[h_{ol}, R = h_{oP^*} R] + c} \tag{7}$$

The principle is as follows: Through category feature processing, the feature values are replaced by unique thermal coding conversion or average label value, but the phenomenon of overfitting may occur in subsequent computational training. Therefore, the prior probability method is adopted to solve this problem for catboost, which is similar to adding the priority weight coefficient of the feature value and sorting it. Thus, the error caused by some low eigenvalues in the feature is reduced, and the prediction offset problem is solved [5].

D. Construction based on the fusion of Stacking models

Stacking is an integrated machine learning technology that fuses multiple classification or regression models through a meta-classifier. [4] This question will train three single machine learning models, XGBoost, LightGBM and Catboost, and take their predicted results as a new training set, and then send them to the XGBoost model for the final training prediction in the secondary training [6].

Work layer division and application training:

(1) The first layer

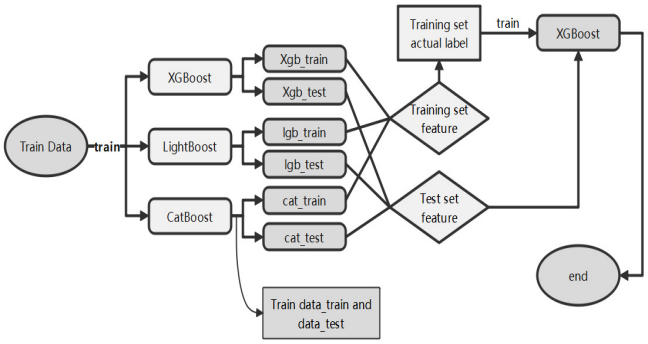

Figure 2: Training flow chart

In Figure 2, for the first layer, we sorted the data set into a training set according to the ratio of 4:1 and split it into "data_train" and "data_fest ", which contained 265,068 data pieces and 66,268 data pieces respectively. Then the "data__train" is trained by XGBoost algorithm, LightGBM algorithm and CatBoost algorithm. Finally, the three trained models were used to predict the fused training set, and the results were saved as "y_ xgb"," y_ lgb "and" y_ _cat "respectively [7].

(2) The second layer

Step1: The prediction results of three separate learners at the first layer are taken as a new data set and combined with the features of the training data set. Here, we use XGBoost algorithm as the second layer metamodels to train and predict

the results at the second layer. The reasons for choosing XGBoost are as follows: Its evaluation criteria are the best among the three single-learner tools and conform to the second layer of the Stacking model (the model needs to be stable and perform well).

Step2: Use the trained XGBoost algorithm to predict the required prediction samples and get the final result.

E. Hyperparameter selection based on grid search and cross validation

In order to optimize the Stacking fusion model, the performance of this class of machine learning classifiers also needs to be optimal. Therefore, the model needs to be constantly adjusted during the training process to select optimal parameters to improve the overall prediction performance of the model.In this paper, we use grid search and cross-validation methods for parameter analysis and screening.

(1) Grid search

Grid search is a method used for hyperparameter tuning to select the best combination of hyperparameters for a model. By exhaustively searching for combinations of a given hyperparameter and evaluating the performance of each combination on the verification set, the best performing hyperparameter combination is found [8].

(2) Cross verification

All combinations traversed through the grid are searched, and for each combination, the model performance is evaluated on the validation set using methods such as cross-validation or set-aside validation. Finally, the combination of hyperparameters with the best performance is selected as the hyperparameters of the optimal model [9][10].

In this paper, the above indicators are calculated and the predictive performance of the model is evaluated from these perspectives.

Table 2: Performance table of each model

	Accuracy	Recall	*RMSE*	$Accuracy_5$	R^2
LightBGM	0.601	0.609	318.57	0.458	0.617
CatBoost	0.592	0.598	289.63	0.412	0.608
XGBoost	0.621	0.634	257.15	0.378	0.637
Stracking	**0.674**	**0.677**	**214.59**	**0.349**	**0.679**

In Table 2, the performance of the model is shown.By comparing and analyzing other models in ***Accuracy***, ***Recall, RMSE, Accuracy5,*** and ***R2***, the fusion model is significantly superior to separate models in terms of performance indicators and in the Stacking process. It also has higher stability and generalization ability. It can more accurately predict the demand for goods in the warehouse corresponding to each merchant in the future.

The forecast results are as follows:

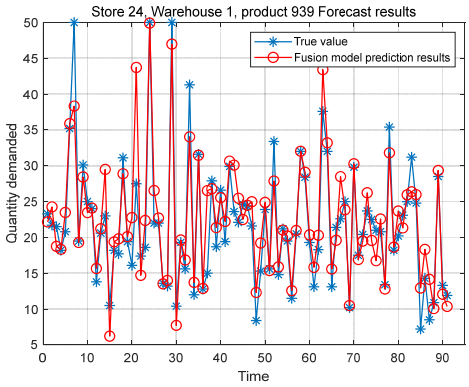

Figure 3: Forecast result

In Figure 3, the distribution of the real value and predicted value of item No. 1, Warehouse No. 24, and item No. 939 on the test set is shown.

IV. Conclusion

This paper establishes the Stacking fusion model, which integrates three machine learning models: XGBoost, LishtGBM, and Catboost. Grid search and cross-verification methods are adopted to optimize parameters and finally predict the commodity demand in each merchant's warehouse in the next 15 days. The prediction performance of the XGBoost, LishtGBM, Catboost, and Stacking fusion models was evaluated with prediction accuracy, recall rate, goodness of fit coefficient, root mean square error, and 5% error accuracy.

The novelty of the Stacking fusion model is reflected in two aspects: First, it combines a variety of different machine learning algorithms and utilizes their respective advantages. Secondly, these models are integrated in the Stacking mode to further improve the forecasting performance. This approach of integration and superposition is more complex and efficient than a single model or simple combination approach. Compared with machine learning model alone, the combined model has better prediction effect and more stable performance.

By using multiple different base models, superimposed fusion models can alleviate overfitting problems. Reduce the over-sensitivity to specific data structures and improve the generalization ability of the model. By optimizing the parameters of grid search, the performance and stability of the model can be further improved, and the model can be more adaptable to the characteristics of data and tasks.

Based on the use scenarios of model performance, stability, explainability, and e-commerce demand, we can more comprehensively evaluate the pros and cons of different models by combining comparative analysis and demonstration of these aspects, and select the model with the best Stacking effect.

References

[1] Wang Changxia. Bank Credit card Customer Churn Forecasting based on the Stacking model integration [D]. Lanzhou university,

[2] Wang Yan, Guo Yuankai. Application of Improved XGBoost Model in Stock Forecasting [J]. Computer Engineering

[3] Shi Jiaqi, Zhang Jianhua. Load forecasting Method Based on integrated learning of Multi-model Fusion Stacking [J]. Proceedings of the CSEE,2019,39(14):4032-4042.

[4] Xie Yong, Ji Mengzhong et al. Application analysis of Monthly Housing Rent Prediction based on Xgboost and LightGBM algorithm [J].

[5] Miao Fengshun, Li Yan, Gao Cen et al. Diabetes prediction method based on CatBoost algorithm [J]. Computer system application

[6] Zhu Yuxiao, Lv Linyuan. Review of recommendation system evaluation indicators [J]. Journal of University of Electronic Science and Technology of China,2012,41(02):163-175. (in Chinese)

[7] Zhong Xi, Sun Xiange. Research on naive Bayesian integration method based on KMeis ++ clustering [J]. Computer Science

[8] HANJW, KAMBERM. Concept and technology of data mining [M]. Fan Ming, Meng Xiaofeng, Trans. Beijing: China Machine Press

[9] Jiang Hua, Ji Feng, Wang Huijiao et al. Improved Kmeans algorithm of ocean data anomaly detection [J]. Computer engineering and design

[10] Lin L, Li S, Wang K, et al. A new FCM-XGBoost system for predicting Pavement Condition Index[J]. Expert Systems With Applications