Prediction of Overall Ratings on rateBeer Dataset

CSE 258 Assignment 2

Abstract

Various recommender system techniques have been proposed since the mid-1990s, and many sorts of recommender system software have been developed recently for a variety of applications. Researchers and managers recognize that recommender systems offer great opportunities and challenges for various domains, with more recent successful developments of recommender systems for real-world applications becoming apparent. We could learn from the information of users on the Internet, which is users' history behavior to predict their future behavior or to recommend items they possibly like. Therefore, this paper will explore several models with different strategies on predicting a customer's overall rating of a given beer. And the result will be evaluated and compared through RMSE to find the best model for this recommendation task.

Key Words: recommender system, beer rating, random forest, XGBoost, latent factor models

I. Introduction

Nowadays, more and more people start purchasing products on e-commerce websites, listening to music, and reading news online. Our Internet is overwhelming with thousands of items and services that ones can choose. Therefore, developing tools for the decision making process and filtering irrelevant online information for customers is crucial for companies to have better user-experience. Recommender systems are efficient tools for prediction users preference. It provides suggestions for items that are most relevant to users from the rating and reviews data.

In this paper, we analyze the beer data with ratings. The beer rating data has about 1.5 millions records and their 16 characteristics, including review of taste, appearance, aroma , etc. We want to predict the overall ratings using the features from the dataset for individual beers. First, in Section III, we explain the analysis of this dataset to have a basic understanding of each factor and how they relate to the overall rating. We will also do some data cleaning work to obtain new factors that may be useful for the predictive task. In Section IV, We introduce our models, which are the basic mean rating model, random forest, XGBoost and latent factor models for the beer rating prediction task. The error metrics we use here is the Root Mean Square Error (RMSE). Later, in Section V, we will conclude the results by applying the methods to the beer rating dataset, and we will find out that the latent factor model behaves the best. Section VI is the conclusion, where we will discuss the best model and future improvements.

II.Related Work

In this project, we have used an existing dataset Ratebeer[1], which is an independent community of enthusiasts and professionals dedicated to supporting and promoting better beer. In addition to this dataset, we found that BeerAdvocate was also used to do similar research. The data span a period of more than 10 years, including all ~3 million reviews up to November 2011. Each review includes ratings in terms of five "aspects": appearance, aroma , palate, taste, and overall impression. Reviews include product and user information, followed by each of these five ratings, and a plaintext review.

For this paper, we will use previous review data to predict users' overall rating for a given beer, using some models learned in class CSE258.

For the regression problem, at present, most applications in this field are multiple linear regression models that assign different weights to multiple features. Current industry and academic research use deep learning networks as well, which leverage deep learning to send the input (the data of images) through different layers of the network work, with each network hierarchically defining specific features. In addition, current state-of-art methods studying type of regression review data are the following: Simple Linear Regression, Polynomial Regression, Support Vector Machine, Random Decision Tree, Random Forest and Neural Network. In this paper, we will select some of the above-mentioned models to conduct user behavior research.

For the latent factor model, a novel study proposes a latent factor recommendation system that explicitly accounts for each user's level of experience. This paper acknowledges that their tastes may have changed over time, and may change again in the future. Thus, J. McAuley et al. [2] put the latent factor into consideration, and build an advanced recommender system. Research shows that Shifting trends in the community, the arrival of new products, and even changes in users' social networks may influence their rating behavior. Being inspired by this paper, this paper also uses the latent factor model to study user behavior based on the data of ratebeer among 11 years, so as to study users' perception of beer overall score.

The conclusion we came up with in this project is similar to existing works. By analyzing customer reviews, we found that taste and aroma features are the most important features in determining the overall rating of a beer.

III.Exploratory Analysis

In this section, we perform some data analysis on the RateBeer dataset to better understand the relationship of features and ratings.

Table 1 Beer Rating dataset statistics

Statistical Feature	Amount
Number of Reviews	1,500,000
Number of Users	22,769
Number of Beers	62,370
Number of Brewers	4,186
Number of Beer Styles	89

3.1 Data Description

The dataset we choose for our project is rateBeer provided by Professor Mcauley. It has around 1.5 millions user reviews with 13 features, which are beer name, beer ID, brewer ID, beer alcohol by volume(ABV), beer style, appearance rating, aroma rating, palate rating, taste review, overall review, review time, review profile name, and review text.

3.2 Data Cleaning & Transformation

To better perform the data analysis and the predictive task, we cleaned the data and added new features to the dataset. We first labeled the string features such as **beer/style** and **review/profileName** with index for vector normalization. And then we converted timestamp **review/time** into several features: **review_year**, **review_month**, **review_day**, **review_weekday** and **review_hour** to analyze the relationship between review time and overall rating in a detailed time dimension. We also built a new feature **review_wc**, the number of characters for each review. Here're the 16 features we have for our predictive task:

brewer_i d	beer_id	user_id	review_l en
review_o verall	review_a roma	review_a ppearance	review_p alate
review_ta ste	beer_AB V	beer_styl eid	review_y ear
review_ month	review_d ay	review_w eekday	review_h our

3.3 Data Analysis

3.3.1 Distribution of Reviews

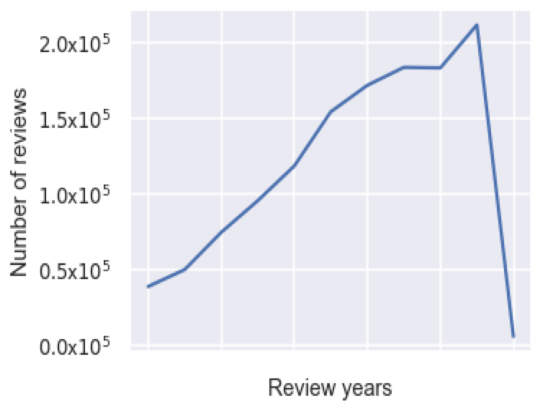


Figure 1 Number of Reviews per Year

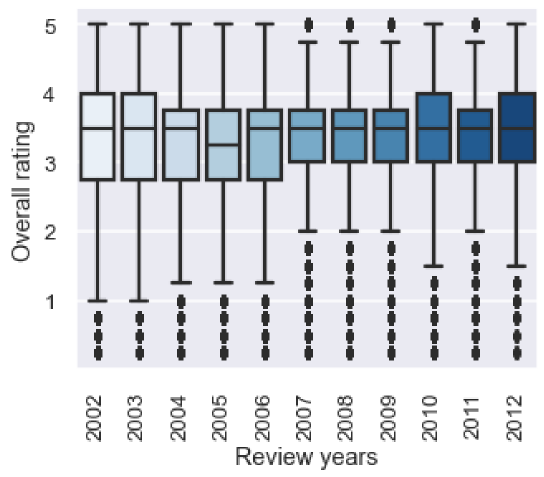


Figure2 Distribution of Ratings per Year

Figure 1 & 2 shows the distribution of reviews and ratings per year. From Figure 1, we can see that the number of reviews has been increasing steadily from 2002 to 2011, and 2012 is a special year, because the data records only end at the beginning of the year, so the data collection is not sufficient. Figure 2 shows the distribution of overall ratings in each year. For most of the years, the median of overall ratings is 3.7 and the majorities are located between 2.9 and 4.1. This demonstrates that the ratings given to beers are highly centralized and skewed and customers are generally satisfied with the beer.

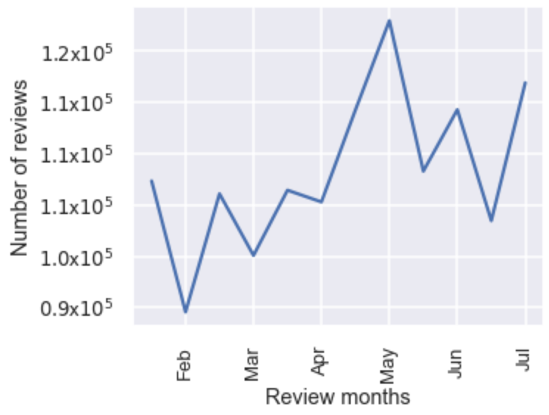


Figure 3 Number of Reviews per Months

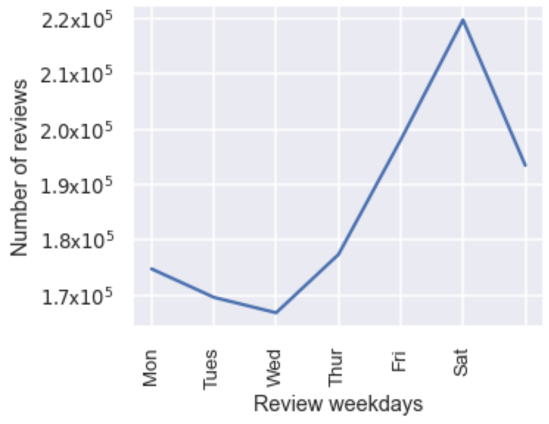
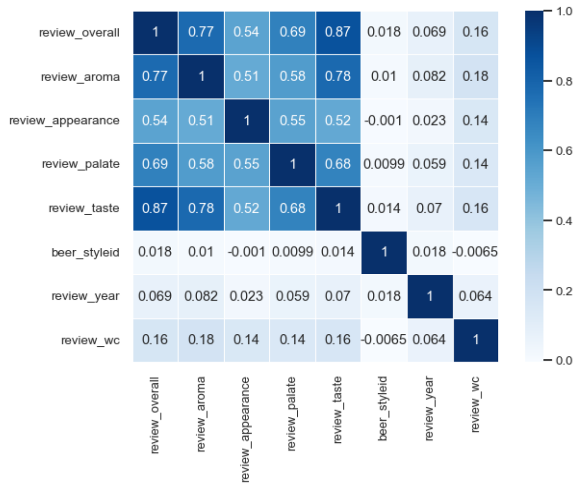


Figure 4 Number of Reviews per Weekdays

In Figure 3 & 4, we study the distribution of reviews per month and weekday. We found that in Figure 3, the peak is in August. This may be due to the summer holiday of the students. A relatively large student group may have an impact on the number of comments on Beer. In the rest of the time, The higher number of reviews is the beginning of the year and the end of the year, which can easily be explained because of various festivals (e.g. Thanksgiving, Spring Festival, Christmas Day). We found that in Figure 4, Friday and Saturday have the highest user ratings. This conclusion is also very consistent with our daily life, because at the end of the week, one way for everyone to entertain is to drink.

3.3.2 Collection between features

It is important to formulate data schema and extract features. But more important is to know the relationship between features, which is a guideline to help us choose useful features. By calculating the correlation coefficient between features, we can know which feature is highly correlated with the overall score. Figure 5 is a correlation matrix between overall ratings and ratings for aroma, appearance, mouthfeel, beer volume alcohol, beer style, and year of review. In this heatmap, brighter background colors indicate higher correlation coefficients and vice versa.



	review_overall	review_aroma	review_appearance	review_palate	review_taste	beer_styleid	review_year	review_wc
review_overall	1	0.77	0.54	0.69	0.87	0.018	0.069	0.16
review_aroma	0.77	1	0.51	0.58	0.78	0.01	0.082	0.18
review_appearance	0.54	0.51	1	0.55	0.52	-0.001	0.023	0.14
review_palate	0.69	0.58	0.55	1	0.68	0.0099	0.059	0.14
review_taste	0.87	0.78	0.52	0.68	1	0.014	0.07	0.16
beer_styleid	0.018	0.01	-0.001	0.0099	0.014	1	0.018	-0.0065
review_year	0.069	0.082	0.023	0.059	0.07	0.018	1	0.064
review_wc	0.16	0.18	0.14	0.14	0.16	-0.0065	0.064	1

Figure 5 Correlation matrix of features

Figure 5 shows that among the multiple features, we found that the user is most concerned about taste first, with a correlation of 0.87, followed by aroma, with a correlation of 0.77, and finally palate and appearance, with a correlation of 0.69 and 0.54, respectively. We found that users basically don't care about the year, style and description of beer, and the correlation degrees are all below 0.1.

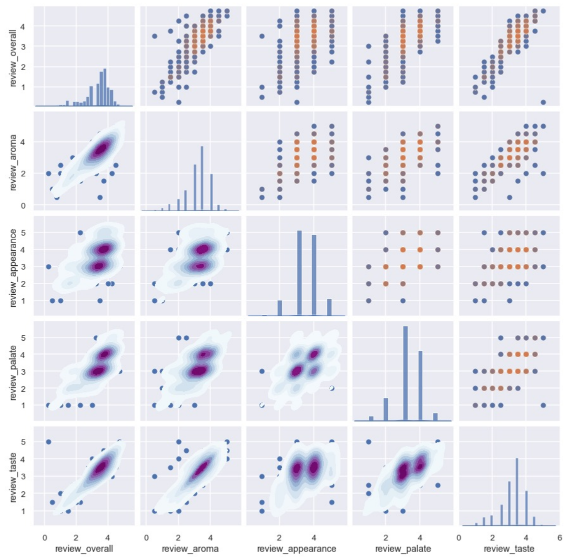


Figure 6 Joint distribution of reviews

We further analyze the joint distribution of overall reviews and partial reviews, and present the results in Figure 6. Diagonal numbers are probability distribution functions (PDF) of reviews. The upper triangle contains the scatterplot and the lower triangle contains the Kernel Density Estimator (KDE). This observation tells us that it is not enough to only use partial ratings to predict overall ratings. In other words, other features like beer taste, styles and review word count may also be important factors.

IV. Approach

In this part, we will introduce our model for solving the prediction problem. We use a latent factor model to predict the overall beer ratings. We will introduce different baseline models for the predictive task as well.

For each model, we split the dataset into training, validation and testing sets. We randomly shuffled the dataset to get a better accurate result for the models.

The evaluation metrics we use to differentiate the models' performance is the Root Means Square Error (RMSE).

The baseline model we chose for evaluation is the Mean Rating model mentioned in the class. For users who haven't rate any similar (Jaccard similarity) beers of the target one, we predict the user's overall rating with average rating of the beer or the mean rating of the train dataset.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y}_i)^2}$$

$$Mean\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

$$Variance = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2$$

We use two feature-based supervised models to compete with our latent factor model. Fifteen features of the beer rating dataset are used for the random forest model and XGBoost model. The beer review, user review, style review is the average rating for each beer, user and beer style from the data records. For each beerId, userId, styleId, we have the taste, aroma, palate, and appearance ratings to calculate the average of them. The review/wc and review/year represent the word count and publish year of the review. The detailed way to process the data is in the previous section.

Beer_review_ taste	Beer_review_ aroma	Beer_review_ palate	Beer_ review_ appearance
User_ review_ taste	User_ review_ aroma	User_ review_ palate	User_ review_ appearance
Styel_ review_ taste	Style_ review_aroma	Style_ review_palate	Style_ review_ appearance
review/wc	beer/ABV	review/year	

For the latent factor model, it assumes that each user and beer is associated with some K-dimensional latent factor vector. We do not need to specify each feature, since the feature is given in the beer rating dataset.

4.1 Random Forest

Random Forest is a supervised machine learning algorithm that is largely used in solving classification and regression. It constructs decision trees on different samples and takes most of the vote for classification and average in case of regression[3]. Because Random Forest uses the bootstrap aggregating concept, each decision tree in the forest will choose a collection of features at random to use for training. The advantage of using Random Forest is that it can correct the decision trees characteristic of overfitting to the training set[4]. It can also effectively handle large-scale datasets such as the beer rating dataset.

Unlike many deep learning models, random forest is stable even if the new data is introduced to the dataset. The overall model is not affected much because the data affect one tree, not all the trees. Compared with the regression models, it also examines the impact between features. In particular, trees grown deep tend to overfit training sets with low bias but high variance. Random forests are a choice to get an average multiple deep decision tree and this generally greatly boosts the performance of the model.

4.2 XGBoost

XGBoost (eXtreme Gradient Boosting) is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It initially started as a research project by Tianqi Chen [5] as part of the DMLC group. It was soon integrated with some other packages to make it easier to use, such as scikit-learn for Python users. The advantage of the XGBoost model is that it can often achieve higher accuracy than a single decision tree. This is because the XGBoost model includes many salient features which are different from other gradient-boosting algorithms. For example, it includes [6] the Clever penalization of trees, proportional shrinking of leaf nodes, Newton Boosting, extra randomization parameter, and automatic Feature selection. Those features make XGBoost outstanding from many other supervised learning algorithms. Since the gradient of the data is considered for each tree, XGBoost's calculation is faster and the precision is more accurate than Random Forest. We want to use the XGBoost to improve the previous result from Random Forest methods.

In the paper, we will use the XGBoost to predict the beer rating from the rateBeer dataset. As above, we have 15 features related to the beer rating process. And we employ gradient boosting as the supervised learning technique which attempts to accurately predict the target variable beer rating by combining the estimates of models.

4.3 Latent Factor

A latent variable model is a statistical model that relates a set of observable variables (also called manifest variables or indicators) to a set of latent variables. We start with the 'standard' latent-factor recommender system [7], which predicts ratings for user/item pairs (u, i) according to

$$res(u,i) = \alpha + \beta_u + \beta_b + Y_u \times Y_b$$

Here, α is a global offset, β_u and $\beta_i$are user and item biases(respectively) and γ_u and γ_i are user and item features. Although this simple model can capture rich interactions between users and products, we are not able to do so with limited computation resources because of the huge size of this matrix, thus optimization function should be proposed by using gradient descent.

$$argmin_{a,b} \sum_{u,b} (a + \beta_u + \beta_b - R_{u,b})^2 + \lambda(\sum_u \beta_u^2 + \sum_b \beta_u^2)$$

Here, λ is the regulation coefficient, and we choose λ=0.00005 in this case, and the dimension of vector γ_u and γ_b is set as 5.

V. Experiment

In this section, we will discuss the settings of our experiment and the results of different models. Figure 7 shows RMSE of the four models.

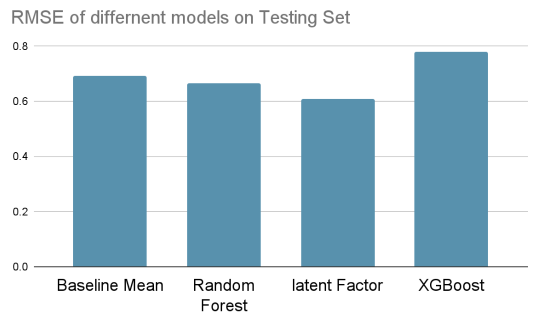


Figure 7 RMSE of different models

5.1 Choosing hyper-parameters

Hyperparameters are certain values or weights that determine the learning process of an algorithm. Hyperparameter tuning is crucial for supervised learning problems. For tree-based models like XGBoost, the learnable parameters are often the choice of decision variables at each node, which include maximum depth of the tree, the number of trees to grow, the number of variables to consider when building each tree, the minimum number of samples on a leaf and the fraction of observations used to build a tree.

We use bayesian optimization with HYPEROPT to implement the search of hyperparameter space. HYPEROPT is a powerful python library, which is capable of finding the best values of hyperparameters according to the minimum of loss function.

After fine-tuning the model on validation dataset, following are the the hyper parameters for RandomForest model:
max_depth=2,
random_state=0,
n_estimators=100

As for XGBoost model, we have:
colsample_bytree= 0.7425468394306338,
gamma = 1.0191348756684306,
max_depth =18,
min_child_weight = 10.0,
reg_alpha = 144.0,
reg_lambda = 0.5912161401594015

5.2 Random Forest

The RMSE for the random forest model is 0.67 as shown in Figure 7.

Figure 8 indicates the importance of different factors of the random forest model. From the figure, we can make some observations. The first one is that the average rating for the taste of the beer is the most important factor in terms of prediction. Second, the average ratings for users are more important than the average ratings for beer and average ratings for style, except for the beer taste factor. The third observation is the style of beer factors (style_review_appearance, style_review_aroma, style_review_, palate, styel_review_taste) have the lowest feature importance score among all features. Lastly, ignoring the style factors, beer ABV and review years are not good features compared to review word counts. The importance of the review word counts factor is much greater than the other two features.

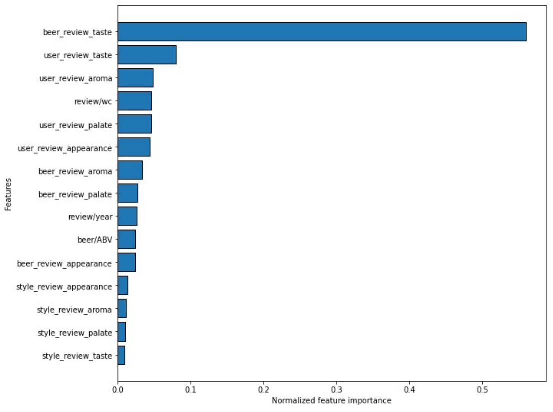


Figure 8 Normalized feature importance of Random Forest

We can make assumptions from those observations that: for further study, using the factor of beer taste may improve the prediction accuracy, which is consistent with our commonsense as taste is often considered as the most important factor while rating. In addition, in most cases, user preferences are more important than the biases of beer. The observations also indicate that style of beer may not be important, and it may not be a good predictor of beer ratings. In the future study, we may avoid using beer style factors for predictive use.

5.3 XGBoost

The RMSE we obtained with the XGBoost model is 0.781 as shown in Table X.

Compared to the random forest model, the behavior of the XGBoost model is poorer.

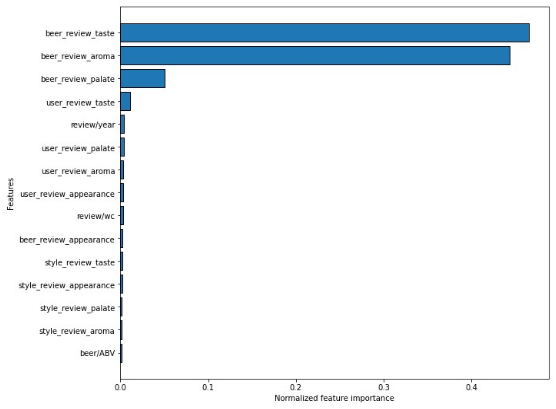


Figure 9 Sorted feature importance of XGBoost

Figure 9 indicates the importance of different factors of the XGBoost model. We observe that: first, the average ratings for beer are more important than the average ratings for users and styles, except for the ABV features. Second, the beers' average ratings of taste, aroma, and palate are the most important factors among all. From the figure, we can see that these three factors have the highest importance which outstrips other factors. In addition, the figure indicates that the style of beer factors and beer ABV factor may not be good features in terms of predicting beer ratings.

From those observations, we can make assumptions for further study that, using the factor of beers may improve the prediction accuracy since they are the important factors. In most cases, the biases of beer are more important than the style preference. The observations also show that beer ABV and style of beers may not be good predictors of ratings. In further study, we may not use those factors in predicting beer.

5.4 Latent Factor

The RMSE we obtained with the latent factor model is 0.61 as shown in Figure 7. Latent factor model behaves the best among the four predicting models. Figure 10 indicates the loss of training and validation in the gradient descent of the latent factor models. The training and validation loss both converge for the latent factor model, and the validation loss is slightly higher than the training loss. So far, the latent factor model surpasses the performance of other models. Although the latent factor models achieve good accuracy in rating prediction, it has the drawback of a large amount of time cost and uses of computation resources. The latent factor model may not be the only model that has the accurate result, we believe that there may be other models that are accurate and use fewer computation resources. In future study, it is valuable to look for methods that are accurate and not sacrificing the computation resources.

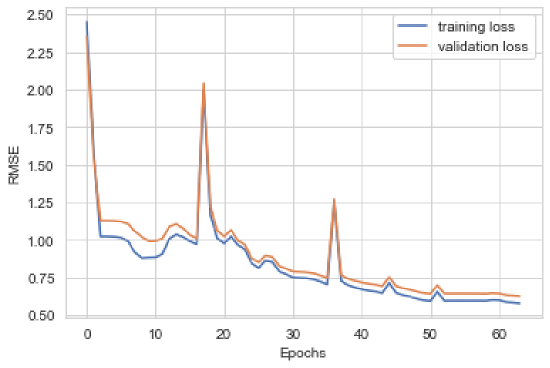


Figure 10 Training and validation loss for latent factor models

VI.Conclusion

In this paper, we predicted overall ratings of a known user on the RateBeer dataset. First, we did a comprehensive exploratory data analysis on the whole dataset with statistical features, features distribution and correlation. In the predictive task, we applied the Random

Forest, Latent Factor, and XGBoost model. Compared with the baseline mean model, both Random Forest and Latent Factor model achieved better performances, and the latter has the lowest RMSE of 0.61 among the three models, this is the one we'll recommend. And XGBoost with the boosting strategy may not be suitable for our predictive regression task. For further study, we suggest improving the time complexity problems of the Latent Factor model or looking for alternative models that have high accuracy without sacrificing the computation resources.

Reference

[1] Julian McAuley, Jure Leskovec, Dan Jurafsky Learning attitudes and attributes from multi-aspect reviews International Conference on Data Mining (ICDM), 2012

[2] J. L. Julian McAuley. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013, 2012.

[3] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[4] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016.

[5] "Story and Lessons behind the evolution of XGBoost". Retrieved 2016-08-01.

[6] Gandhi, Rohith (2019-05-24). "Gradient Boosting and XGBoost". Medium. Retrieved 2020-01-04.

[7] F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors. Recommender Systems Handbook. Springer, 2011.