Predicting Rating of Amazon Fine Food from Reviews
CSE 190 Assignment 2

Yenni Chu	Yen Pham	Nghi Duong	Yuqin Wu
A91427052	A11780046	A11758655	A99027167
yechu@ucsd.edu	ytpham@ucsd.edu	naduong@ucsd.edu	yuw017@ucsd.edu

ABSTRACT:

The development of the Internet changed the way people eat and responding for food. Amazon is a biggest website where users can easily purchase all kind of food they need with only a mouse-click as well as there are many supportive ways for decision making. Most of the food have the ratings and reviews from other customers who have different backgrounds, cultures, eating habits, personalities, and prefer flavors. We, as the data analysis scientists, aim to create a predictive model, which is designed for predicting the rating of different food based on the data given, such as the time when the review was written, number of helpfulness voting, as well as the length of the review. In order to achieve the goal, we utilize the Amazon Fine Food Reviews dataset, which has some necessary features for us to analyze. We mine and analyze the data using the models, such as linear regression, ridge regression, and Latent-factor models, all of which we have learned from our CSE 190 Data Mining course. Eventually we want to build an acceptable model which helps us better understand how customers rate the food they purchased.

1 INTRODUCTION

We've learned many models for predicting a task in data mining; however, using the right model is very important for a good result and also reducing the running time. In this assignment, we try to accomplish a predictive task using a supervised learning model; however, in the end we believe we still achieve the goal with an acceptable error. Latent-factor model is satisfied the requirements even using limited features. In this report, we analyze the predictive process from the beginning to the final results using this model and comparing with linear regression model and ridge regression which two we have used most of the time in previous work. Ideally, we hope we can find out how this kind of special dataset we got helps us build the model correctly using necessary features and finally obtain a satisfied result.

2 EXPLORATORY ANALYSIS

2.1 Review example:

product/productId: B00813GRG4
review/userId: A1D87F6ZCVE5NK
review/profileName: dll pa
review/helpfulness: 0/0
review/score: 1.0
review/time: 1346976000
review/summary: Not as Advertised
review/text: Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represent the product as "Jumbo".

2.1.1 Data Fields Explanation:

product/productId : the ID of the product
review/userID : the ID of the user
review/profileName: name of the user
review/helpfulness: fraction of users who found the review helpful
review/score: rating of the product
review/time: time of the review (unix time)
review/summary: review summary
review/text: text of the review

2.2 Dataset statistics

Number of reviews	568,454
Number of users	256,059
Number of products	74,258
Users with > 50 reviews	260
Median no. of words per review	56
Timespan	Oct 1999 - Oct 2012

This dataset consists of reviews of fine foods from Amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plaintext review. Be more specific, there are 74,257 of unique users, and 6,055 of unique items in all the ratings by using the the data field "review/userId" and "product/productId".

Dataset Address: http://snap.stanford.edu/data/web-FineFoods.html.

2.3 Analysis

In order to find the relations between score and and features in reviews, we graph the distribution of score according different features. We have tried several features such as length of text, length of summary, and ratio of helpfulness, but we cannot find obvious pattern and relation of them. However, the figure 7 with distribution of score according the time in year has some interesting pattern. As time passed by, there are more and more high score rating received. For example, in 2012, there are more than 105,000 of reviews with score 5, but only around 15,000 of reviews with score 1. It might indicate that the change in lifestyle and eating behavior, users just become more generous on rating food, or there are other underlying reasons. Figure 1.1 to figure 4 are the distributions of rating score over features in reviews in binhex heap map or scatter plot.

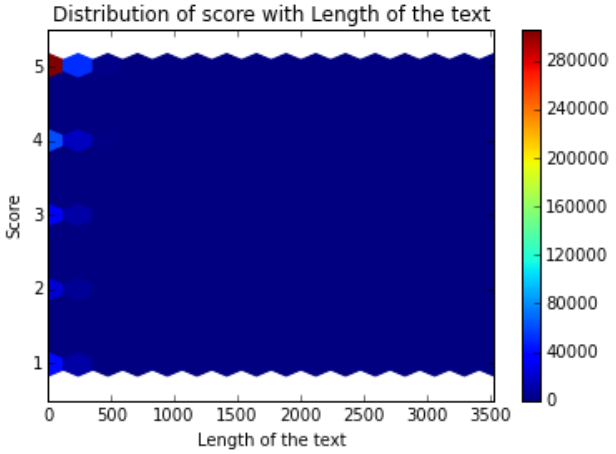

Figure 1.1

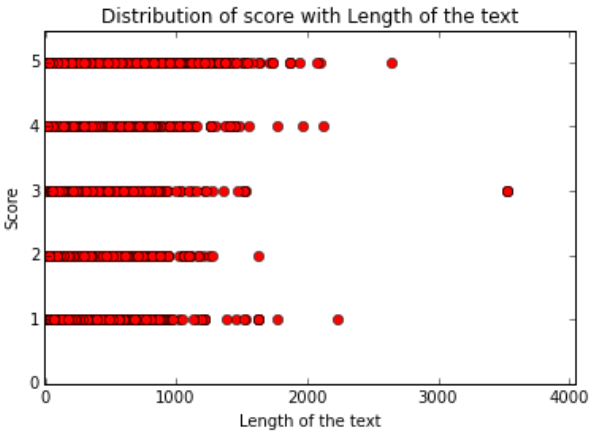

Figure 1.2

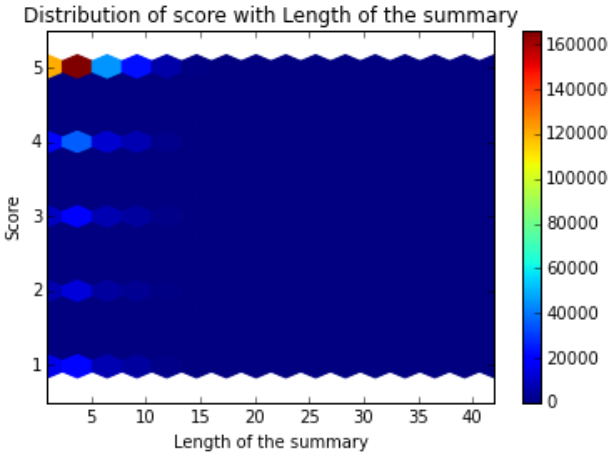

Figure 2.1

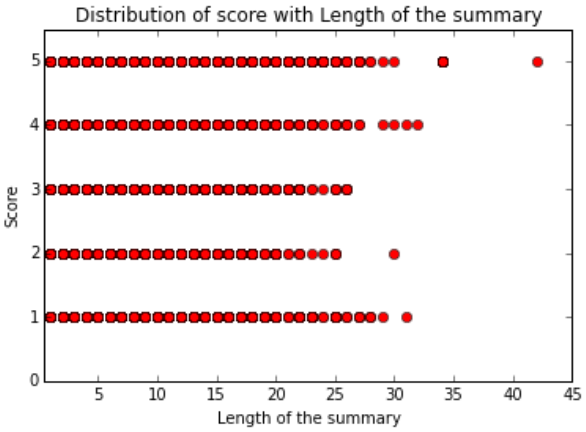

Figure 2.2

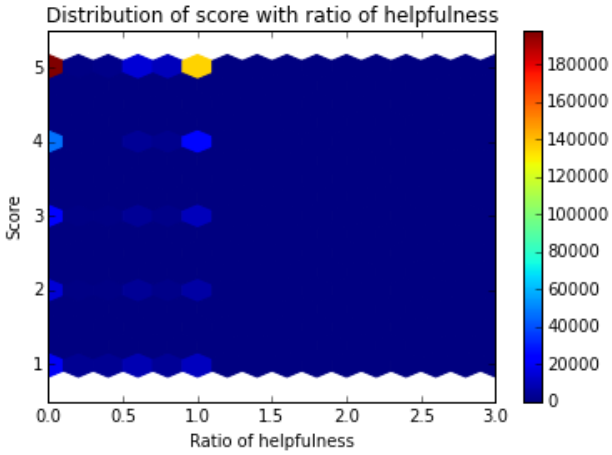

Figure 3.1

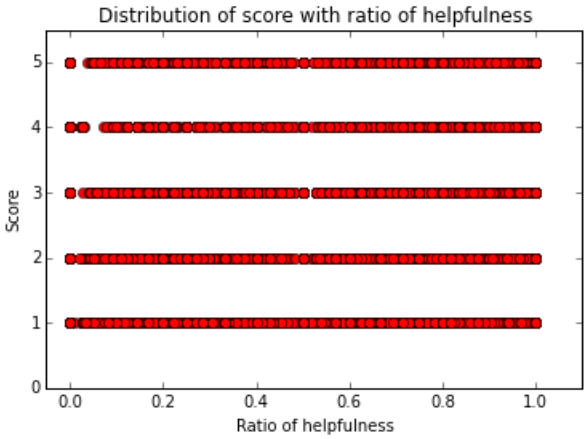

Figure 3.2

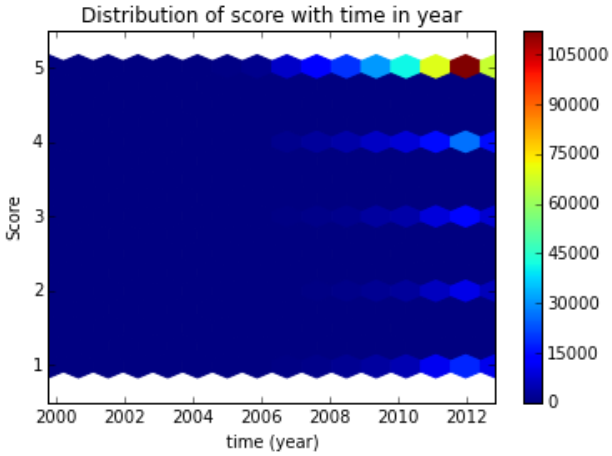

Figure 4.1

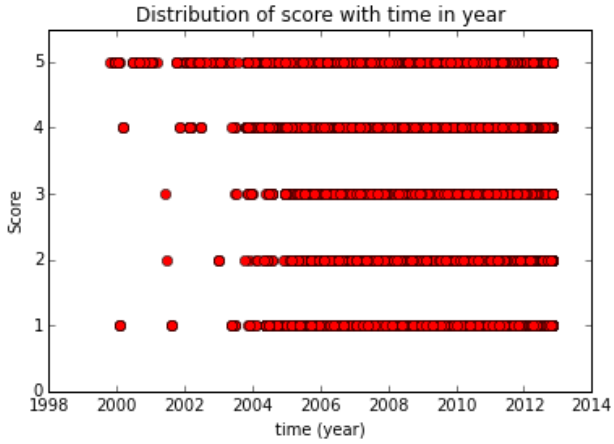

Figure 4.2

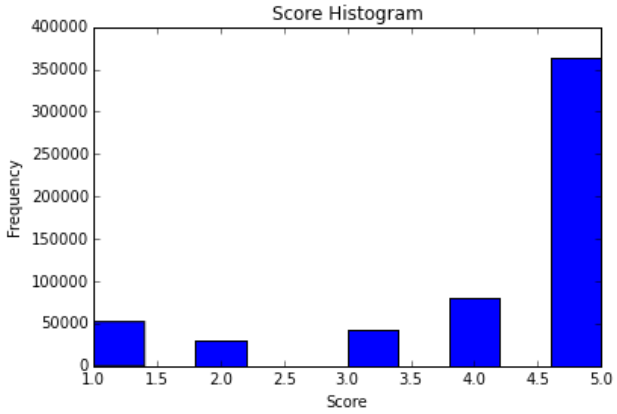

Figure 5

Another interesting aspect of dataset is that, there is an enormous number of review with score 5, and relatively less reviews of the rest scores according to the figure 5. People are tend to be generous and give higher rating scores on the food reviews.

Figure 6.1 Negative words

Figure 6.1 Positive words

We also do some sentiment analysis of text review and see if this will help us to predict the rating scores. Firstly, we separate the all dataset into two category, higher score and lower score group. The higher score group consists of dataset that have rating score more and equal than 4, and lower score group consists of dataset that have rating score less and equal than 2. We count all the words that appear in text review of those two dataset and decide which words are positive and which words are negative. We also remove all the stop words to make sure the positive and negative words are more accurate.

The items in this dataset are fine foods which most of them are good and healthy. Based on the goodness of the items, we could think that the scores of them are high. In addition, because the products are food, the score could high rely on user's' eating habits and preference. Depend on their emotion, feeling and different type of personalities, users could give various score on one single items. It could help us to understand why most of the features in the dataset have less effect to the score because the score may be related to the users and the items which can not be graphed and shown like other features above.

We also think about using the review text to analyze the negative and positive words. However, the most popular words in the review text for high score items and low score items are very similar. For example, users could review the high score items as "good" and also the low score items as "not good". Although "good" is a positive word, it does not matter in this case. Just a few words really means something such as "best" or "worst". It is really hard to approach the problem this way, but it is an interesting way to increase the accuracy of the prediction.

3 LITERATURE

Rating prediction system is actually a pretty mature field where many algorithms already exist for a long time. However, the accuracy goes drastically low when the rating is twisted with individually unrelated, unique and random features. In our case, for the same product, two users can give exactly opposite rating based on the same reason. For instance, a user would probably give 5.0 to a dessert because it is so sweet but another user would probably give 1.0 also because it is too sweet. Therefore, this kind of dataset is really difficult to predict using traditional algorithm.

For this dataset, since it is an existing dataset collected by professor Julian McAuley and his colleagues, there indeed is a report[3] written by professor McAuley also using it. In the report, they use the dataset to study how users experience affects their taste. In order to understand how different experience levels affect user's taste, they, actually, introduce latent factor model to analyze the dataset. The only difference is, they tend to focus on temporal effects but ignore the user's own development and personality. On the contrary, in our model, we are focusing on the inner connection and relationship among users and foods by letting it learn user's' habit and what they really purchased. Consequently, it can adjust those factor to minimize the MSE. In professor's report, they also utilize beer rating from BeerAdvocate, movies from Amazon, and wine from CellarTracker. Those dataset are all rated mainly based on customers' own tastes and background. As a result, to some extent, these dataset are really similar to the one we are using now. We all try to observe a way to convert the inner relationship into math equations.

Finally, in both our report and professor's report we chose latent factor model as the best model to deal with such complicate problem. We have found that latent factor model has a really good performance on data that has chaotic and unpredictable feature so far.

4 PREDICTIVE TASK

After analyzing the dataset, we still decided to build a fine model to predict the rating of the food from each user's review. Considering this is a relatively challenging task due to the difficulty for finding a organized and predictable pattern, we try to find out the most important, if there is any, feature from the given data, such as user's id, items id, timestamp, helpfulness rate, review text, and its summary, that can be used to improve the accuracy of the prediction. If there is any important feature indeed existing, this report will give readers an answer of what to focus on when predicting a relatively unorganized and unpredictable dataset. Otherwise, this report will verify our assumption ---- the rating fluctuates based on many abstract elements such as user's taste, habit, background, and personality which are difficult to convert to digital data and therefore affect the accuracy of the prediction task. Therefore, only supervised learning approach will work fine for such a task.

For performance evaluation, since in this dataset, we already have the real rating for each review, we can extract them from our dataset and then compare them with our predictive results. We use MSE (mean-squared error) to indicate and assess the performance as well as the validity of each of our models and ultimately find the best model with lowest MSE we got.

Obviously, the baseline for comparison is using the average rating to predict each review's rating despite of any other feature. For example, if the average rating finally we get is 2.8, then we predict that all reviews will give a 2.8 as their rating. Since this is a mature task that we have already done in previous homework, even though there are definitely many approaches to predict the rating, we still tend to start with some regression models as what we did in previous homework to see if these models are what we desire for such a special kind of product(food). If not, we will try Latent-factor model, a supervised learning approach invented by professor McAuley and his colleagues that perform so well on the relationship between each individual user and item.

For the features, we tend to use all the data it provides to see if they will drastically enhance the accuracy of our prediction. We use the length of the review text as well as the length of its summary, which we add into the feature matrix for training and validation.

5 MODEL

• 5.1 Regression

Regression is known as the simple supervised learning approaches to learn the relationships between the input variables (features) and the output variables (predictions). In this assignment, we use two kinds of regression including linear regression and ridge regression.

• 5.1.1 Linear regression

In linear regression, we assume that a predictor has the form

$$X\Theta = y$$

with y is the vector of outputs (label), X is the matrix of features (data) and Θ is the unknown (which features are relevant).

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. Before attempting to fit a linear model to observed data, a model should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other but that there is some significant association between the two variables.

When applying linear regression model to the dataset, using right features will help a lot to reduce the errors in the predictions. Because the prediction task is predicting the score of the fine foods, based on all the features we have in the dataset. However, from the figure 1 to figure 8 that we show earlier, there are no too much obvious relationship with score and the rest of the review information. Yet, we still try to train the model with those feature and find out how much they are relevant to the rating scores.

Firstly, we fit the linear regression model only with feature unix time, and we obtain 1.705057 for MSE which is pretty big, but this is what we expect since we already see the pattern in the figures mentioned earlier. Similarly, we add other features such as length of text, length of summary, and ratio of helpfulness to the model and obtain around 1.69 for MSE. In conclusion, the linear regression with some basic review information as feature probably not a good idea. It is because, there are not so much

helpful and intuitive features that are strongly related to the rating scores, so we have to try other models to make better predictions of rating score.

• 5.1.2 Ridge regression (or Tikhonov regularization)

This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm.

Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares:

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2$$

$\alpha \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of α, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity.

For comparison, we use similar features for both linear regression and ridge regression. However, in ridge regression, changing α could make the model more fitted with the data and reduce the risk of overfitting. The range of α we used is multiple by 10 such as 0.001, 0.01, 0.1, 1, 10, 100, 1000. There will be many different values of α but we will pick the one with the least error and it will be displayed in the result section below.

In order to verify the accuracy of the prediction, we calculate the Mean-Squared error following with the formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2.$$

We will compute the sum of squares of the difference between the prediction values and the actual value in the dataset and dividing by the number of the values in the test set. The MSEs are really high because most of the features we are using to predict are not important enough. It reinforces the theory above that the users and items are the most important features which affect the accuracy much more.

• 5.2 Latent-factor model:

This model is the idea from the professor McAuley and his colleague, and this is probably the best one we have tried so far. Since scores are given by independent user on a single product, so the model needs to learn about user's reference and product reference.

First, a very basic model (baseline) that we tried to train is

$$\text{score} = \alpha$$

with α is the average score that a particular user gives on any product.

It turns out that the model works pretty well with the data whose MSE = 0.707226318382 (Mean Squared Error) is not as big as the model we have been trying above. In order to be more precise in prediction, we now want the model to learn more about user's behavior and product's reference. So, we try with model

$$score_{u,i} = \alpha + \beta_u + \beta_i$$

The objective for this model is

$$argmin_{\alpha,\beta} \sum_{u,i}(\alpha + \beta_u + \beta_i) + \lambda[\sum_u \beta_u^2 + \sum_i \beta_i^2]$$

(*)

To achieve this objective function, the procedure that we follow is is repeating updating α, β_u andβ_i using 3 update functions:

$$\alpha = \frac{\sum_{u,i}(score_{u,i} - (\beta_u + \beta_i))}{length\ of\ data}$$

$$\beta_u = \frac{\sum_{i \in I_u}(score_{u,i} - (\alpha + \beta_i))}{\lambda + |I_u|}$$

$$\beta_u = \frac{\sum_{u \in U_i}(score_{u,i} - (\alpha + \beta_u))}{\lambda + |U_i|}$$

Note that these update equation come from deriving equation (*) respectively to α, and β_i. For hyperparameter λ, we pick λ from set {0,001, 0.01, 0.1, 1, 10, 100, 1000} and pick the one which give a smallest MSE on validation data.

6 RESULTS

With all the experiments that we have tried with different models, the results we obtained are understandable. The first model is linear regression with respect to different features. We have tried different combinations of the feature in reviews and mean squared errors sticked around 1.7. This is what we expect since there is no strong connection between those features and rating scores . The table

1 are some examples of feature combinations and their mean squared errors.

Model	Features used	MSE
1	**Unix time**	**1.705057**
2	**Unix time + length of text**	**1.699669**
3	**Unix time + length of text + length of summary**	**1.694953**

Table 1. Result from linear regression model

The second model we built is ridges regression, the reason that we use this model is we can add penalty into the model to regularize the process and reduce overfitting from linear regression. The table below is the result of ridge regression and its mean squared errors with respect to different combinations of feature. However, the mean squared errors are still pretty high which are similar to linear regression.

Model	Feature used	α = 0.001	α = 1.0	α = 100
1	Unix Time	1.72706474851	1.72678490898	1.72250156861
2	Unix Time + Length of the review text	1.71734637187	1.7173198107	1.74324581195
3	Unix Time + Length of the review text + Length of the review summary	1.71864990407	1.7185560259	1.75882192602

Table 2. Result from ridge regression model

The reason why these outputs are all not really ideally is obvious. These features do not necessarily have strong connections with users nor items. Therefore, we turned to build a brand new model which can learn the behavior of each user and item so it would probably have a better performance.

Last model we built is the latent-factor model. Instead of just using the basic features from reviews, we let the model itself tune its parameters by treating them individually and learning the behavior of users and items. It turns out that the MSE on this model has improved significantly on validation data when λ= 0.1 (MSE = 0.613884226289).

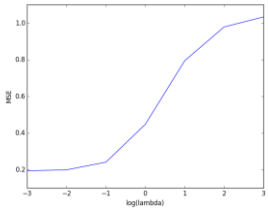

Figure 7.1 MSE on training data

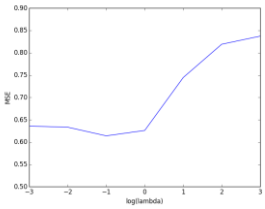

Figure 7.2 MSE on validation data

7 CONCLUSION

Users' rating behavior change overtime. They give higher rating score to food product as time passed. There are no obvious pattern and strong relations for other features in reviews. Therefore, using linear regression and ridge regression will not predict the rating score so well. The Latent-factor model is the one which is more fitted with the dataset than other models. The purpose of this report is showing how to approach a model to a dataset and using it to predict the given tasks. We tried different models and compare the result at the end to see which models is the best among them. Although we picked the model which is better than others, there always exists some errors. There are many factors could affect the results which we can only assume, but we tried to reduce the errors as much as we could such as reduce overfitting. In addition, we realized that analyzing the data before doing the predictive task is very important because it helps us to choose the efficient features applying in the models. We found that after applying the models in the dataset and looking at the results, the reflection actually is very closed to what we expect when analyzing the data at the beginning. In conclusion, for the fine food products, the most important factors affect to the score are each user's behaviors and each items itself. The best model we have found is latent factor model whose advantages fit this task pretty well.

8 REFERENCES

1. "Linear Regression." *Linear Regression*. Yale University's Website. Web. 29 Nov. 2015. <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm>.
2. "1.1. Generalized Linear Models¶." *1.1. Generalized Linear Models — Scikit-learn 0.17 Documentation*. Web. 29 Nov. 2015. <http://scikit-learn.org/stable/modules/linear_model.html>.
3. McAuley, Julian, and Jure Leskovec. "From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews." Stanford University. Web. 1 Dec. 2015. <http://i.stanford.edu/~julian/pdfs/www13.pdf>.
4. Xu, Yun, Xinhui Wu, and Qinxia Wang. "Sentiment Analysis of Yelp‘s Ratings Based on Text Reviews." Stanford University. Web. 1 Dec. 2015. <http://cs229.stanford.edu/proj2014/Yun Xu, Xinhui Wu, Qinxia Wang, Sentiment Analysis of Yelp's Ratings Based on Text Reviews.pdf>.
5. McAuley, J., & Leskovec, J. (2013). Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In Proceedings of the 7th ACM Conference on Recommender Systems (pp. 165-172). New York, NY, USA: ACM. doi:10.1145/2507157.2507163.
6. G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In WebDB, 2009.
7. E. Frank and M. Hall. Additive regression applied to a large-scale collaborative filtering problem. In AI 2008: Advances in Artificial Intelligence, pages 435-446. Springer, 2008.
8. Kim, S. M., Pantel, P., Chklovski, T., & Pennachiotti, M. (2006, July). Automatically assessing review helpfulness. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (pp. 423-430). Association for Computational Linguistics.
9. Ghose, A., & Ipeirotis, P. G. (2011). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. Knowledge and Data Engineering, IEEE Transactions on. 23(10), 1498-1512.