AMAZON REVIEW DATA ANALYSIS

Songqiao Li
University of California, San Diego
sol018@ucsd.edu

Wen-Hsuan Hung
University of California, San Diego
w2hung@ucsd.edu

Hongyuan Zhang
University of California, San Diego
hoz010@ucsd.edu

ABSTRACT

We cleaned, analysed, and made predictions for our dataset, “Amazon Games & Toys reviews”. Insights concerning user behaviors are generated from our visualizations. To make comparison between baseline models, we chose three machine learning models that all out performed the baseline models in terms of their accuracy, F-score, or MSE.

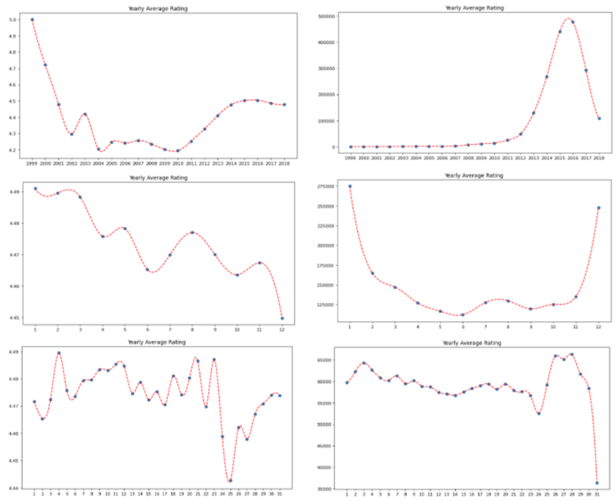
Beside, sentiment analysis is done with word cloud techniques. Clear discrepancies are observed between positive and negative reviews.Our visualizations include trend lines, bar graphs, and word clouds, for which we maximized the simplicity to make them easier to understand and more representative.

1. THE DATASET

The dataset we have is the Toys and Games Amazon review data by Ni et.al, UCSD[1] from 1999 to 2018. Our team chose the 5-core dataset to meet our analysis goal because the dataset is filtered to contain only users who purchased 5 items and above. We believe that with the data length of 1828971 and 11 variables, the dataset is sufficient for our analysis.

Some basic dataset statistics include the mean of the rating to be 4.47. Additionally, the mean of review length and name is 226.99 and 9.86 respectively. The number of unique users is 208180 and 286726 for unique items.

1.1 Ratings over time



Rating average & amount for year, month, day

We plotted the rating average and review amount by year, month, and day. Trends are observed from left three plots (average ratings) that people gave lower ratings as year increases before 2004, and gave higher ratings after 2010. We noticed that people tend to give lower ratings in the later months of the year, the same applies to days of the months. In terms of the review amounts, it reaches its peak in 2016 and then gradually goes down. Also, people are more willing to write reviews during Christmas months.

1.2 Influential factors on rating

Through linear regression, our team found the overall rating is positively correlated with verified user, length of reviewerName, and time. However, the overall rating is negatively correlated with the number of votes and the length of reviewText.

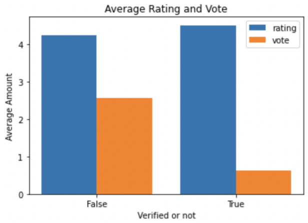
[1] https://nijianmo.github.io/amazon/index.html#subsets

$$
\begin{aligned}
overallRating = 4.12 &- 1.11e^{-04} * vote \\
&+ 1.64e^{-01} * verified \\
&+ 5.99e^{-03} * reviewName \\
&- 2.44e^{-04} * reviewText \\
&+ 1.42e^{-10} * time
\end{aligned}
$$

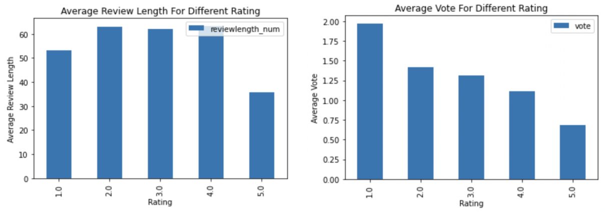
Figure X. Linear regression equation

Comparing theta in the regression model, we noticed verified as the most influential factor followed by vote and review text. Therefore, our team decided to take a deeper look into the vote, verified and to perform a sentiment analysis on review text.

1.3 "Rating" and "Verified"



Average rating and vote



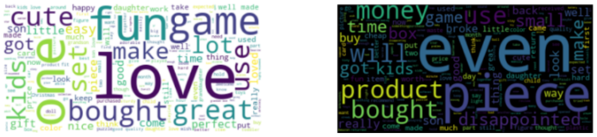
Average review length and vote

The average rating and vote for verified and not verified charts shows that the average rating for verified users is a bit higher than not verified users, and the average votes not verified users get is higher than verified users. The average review length for different rating charts shows that overall when users give lower ratings, they write longer reviews, and people who give 5 star reviews usually write shorter reviews.

The average vote for different rating charts shows that 1 star rating has the highest average votes, and 5 star rating has the lowest votes. These bar charts make us rethink what are the features that affect rating predictions, thus in one part of our later predictive models, we try different approaches in the hope to predict what rating a user would give.

1.4 Sentiment Analysis on review text

We classified the ratings as "Positive" (ratings of 5) and "Negative" (ratings from 1 to 4) and generated word clouds focusing on the latest two years, 2017 & 2018.



Word Cloud 2017 Positive & negative

Word Cloud 2018 Positive & Negative

It is shown that positive and negative word clouds represent distinct attitudes for which the positive word cloud includes words like "love", "great", and "fun", while the negative word cloud includes words like "even", "broken", or "time".

2. PREDICTIVE TASKS

Based on the exploratory analysis, we have two objectives, the prediction of rating and product purchase based on review information. We have one baseline for each of the objectives. To better achieve the prediction, we manipulated the data and identified ways to evaluate our model.

2.1 Baselines

The baseline model for rating prediction is: predicting the average rating for that product. Let's say for product A, user 1 gives it 5 star rating, user 2 gives it 3 star rating, then when we predict rating for a new user buying product A, we expect he/she to give the product (5 + 3) / 2 = 4 star rating. The accuracy for the baseline model is 0.54, and f1 score is 0.57.

We used Logistic modeling as our baseline model for “would users buy this item” prediction. We parsed user Id and item Id into each letter/number and transformed them into letters using the absolute value of ord function minus 96 (all non-capitalized). Labels consist 0 and 1 for which represent negative and positive cases (did the user buy it or not).

2.2 Methodology

For the data cleaning, we remove the comma in the vote column, and convert its type from string to integer. Next, we converted the NaN values to 0 in vote data. To validate our prediction, we split the dataset into 70% training set and 30% testing set, and also generate a negative set the same length as the original dataset and add it to the original data.

Specifically, the negative set generation is based on two variables, usersPerItem and itemsPerUser. The usersPerItem includes all interacted items per user and the itemsPerUser includes all users per item. To avoid repetition, we randomly selected ungrouped users and items and compared them with the existing groups of user and item. From the negative set, we are able to thoroughly testify our model performance and avoid overfitting issues for prediction tasks.

In collaborative filtering models, we include the average rating given by a user, and the average rating received for each product in the model. In the random forest classifier, we included the number of votes, binary outcome of whether the user is verified, the length of review text, the length of review name, and the unix time of the review.

In the Catboost training process, userId and itemId are parsed and converted into lower-cased letters respectively. We used them as categorical features to predict the binary outcome of whether the user bought this item or not.

The model evaluation metrics we use vary from MSE, accuracy, and f1 score. We used the MSE score to assess the collaborative filtering model because it is based on user and item average. Whereas, we also include the f1 score for the filtering model to compare its performance with random forest model and baseline. Here, we chose the f1 score because it does not require a balanced dataset.

Additionally, we choose to use accuracy for the catboost model. While accuracy measures the correctly classified positive and negative observation, it requires a balanced dataset. Since we added a negative set, the assumption of accuracy is satisfied.

3. MODELS

3.1 Rating Prediction - Collaborative Filtering

In this prediction, rating is composed of two parts - the average rating given by each user, and the average rating received for each product. The sum of the weight for the two features is equal to 1, and the formula of the model is:

*weight*average rating given by a user +*
*(1-weight)*average rating received for a product*

For example, if the average rating given by each user is 0.6, then the weight of the average rating received by the product is 1 - 0.6 = 0.4. We also need to take new products into consideration, therefore when there is a new product, we predict the rating as the average rating for all the items.

The reason that we take the average rating given by each user into account in the model is that we believe users who give higher rating tend to give higher rating at all times, even though they aren’t too satisfied with the product, and users who give lower rating tend to be less satisfied with the product they bought.

Since there are only 1, 2, 3, 4, 5 star ratings, and the number that the model generates contains float, thus we round to the nearest whole number to increase model performance.

The advantage of the model is that collaborative filtering is a method of making predictions about the interests of a user by collecting preferences or taste information from many users, therefore it is easy to implement, and has few limitations. The constraint is there must be historical data. However, the disadvantage of the model is that since it's based on a group of people, and people differ from each other, thus the prediction might not be too accurate.

To optimize the weight between the rating given by each user, and the rating received for each product, we define a function to find the weight which generates the lowest MSE for prediction, then build the prediction model based on this weight.

The accuracy for this model is 0.57, f1 score is 0.59, and the MSE is 1.032. Comparing the f1 score with the baseline model (0.57), we can tell that the collaborative filtering has better performance.

3.2 Rating Prediction - Random Forest

One advantage of a random forest classifier is that it solves the prediction task efficiently with implicit feature selection and balance in the bias and variance tradeoff. However, there exist certain disadvantages. While the bootstrap sampling algorithm keeps the classifier less influenced by outliers, it also leads to less accuracy for rare traits.

The core of random forest is to random subset features to split nodes and to random sample form training data to build decision trees. In each decision tree within the random classifier we work with minimizing the classification error, denoted by the Gini index[2].

$$I(t) = \Sigma_{s=1,\dots,n} [1 - \overline{Y}_s^2 + (1 - \overline{Y}_s^2)]$$

Here, $\overline{Y}_s$ stand for the mean value of the classified results under a certain node. Though the minimizing of Gini index might lead to potential overfitting problem, the random forest classifier avoids the problem from the random samples [1].

We first predict the rating with review information in the default setting of the random forest classifier. To optimize the model, we graphed the model performance to visualize the effects of changes in number of estimators and max depth in the classifier. Both the two models beat the baseline and the optimized model attains higher f1 score, an increase from 0.79 to 0.83.

3.3 Catboost

Catboost is an algorithm for gradient boosting on decision trees. It is developed byYandex researchers and engineers, and is used for search, recommendation systems...[3]

During training, a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to precious trees. The number of trees is controlled by the stating parameters.

The Catboost algorithm is a high performance and greedy novel gradient boosting implementation. Thus, when implemented appropriately, Catboost either leads to ties in competitions with standard benchmarks.

Catboost produces good results without extensive hyper-parameter tuning. For data with categorical features the accuracy would be better compared to other algorithms.

In our "would the user buy this item?" predictions, over 400,000 negative data points are generated based on one other thing that a certain user did not purchase. Total of 600,000 are trained with Catboost as 0.2 test-train split from the 800,000 dataset which has equal size of bought and didn't bought (classified as 1 & 0). Our accuracy from test prediction and test dataset reached 0.9146 in terms of only 13732 false predictions out of 160844 test dataset.

[2] https://www.stat.cmu.edu/~larry/=sml/forests.pdf

[3] https://catboost.ai/

3.4 Model comparisons

The models we have answers two different questions, how to predict ratings and how to predict the decision of the user to buy an item from review information. Then, we compare and evaluate the model based on their prediction task.

The performance (f1 score) of random forest is better than collaborative filtering on rating prediction. In the random forest model, we labeled all 5 star ratings as 1, and other ratings as 0, it might be easier for the machine to make the correct guess between 1 and 0.

However, in collaborative filtering models, we ask the machine to predict a real number, thus it's reasonable that the performance isn't as good as the random forest model.

Catboost outperformed logistic regression by having accuracy of 0.9146, which is approximately 80% higher than the accuracy of logistic's.

4. LITERATURE

Our team came up with the project idea while we are talking about our plans for Christmas. Besides travel, we found we are all deciding gifts for our family, especially for teenagers and children. Given our interest, we found the Amazon dataset about the Toys and Games category developed by UCSD researchers to explain and justify recommendations to users [2]. The dataset contains various review information and allows us to analyze customer psychology by predicting whether they will purchase a product and what rating they will give to the product.

In terms of customer review behavior, around the world, November and December mark the busiest shopping time of the year and a make-or-break sales period for retailers and brands. What may be more of a surprise, though, is how closely the spike in review usage during the holiday season parallels holiday sales.

According to the Census Bureau of the U.S. Department of Commerce, U.S. e-commerce sales for the fourth quarter of 2013 totaled an estimated $83.7 billion – a 39 percent increase from the average of e-commerce sales during Q1-Q3 2013 [4]. In parallel, review page views for November and December exceeded the monthly average for the rest of the year by 38 percent and 48 percent, respectively[4].

Similar research we found includes *Predicting User Ratings Using Status Models on Amazon.com* from B. Wang, G. Wang, Z. Li [3]. The result shows that the model relies more on analyzing network structure and using status theory has better performance than the model focusing on review based information. This indicates that applying network and status concepts in the model is useful for improving prediction accuracy for Amazon data. However, users' ratings of a product might produce bias for rating prediction.

Furthermore, we found a state-of-art model within a recent publication on amazon science platform. In the article, Danushka Bollegala discussed the impact of counterfactual phrases in sentiment analysis [5]. As mentioned in the article, counterfactual phrases in reviews might lead to frustrating experiences for customers. To deal with the issue, the author suggests baseline models such as the cross-lingual language model, and anticipates further filtration by other types of linguistic constructions.

In future study, beyond the existing sentiment analysis on review text and our models that contain review text length, we can also include analysis on different types of linguistics constructions. The improvement in review text analysis would facilitate the product retrieval systems in the Amazon Store.

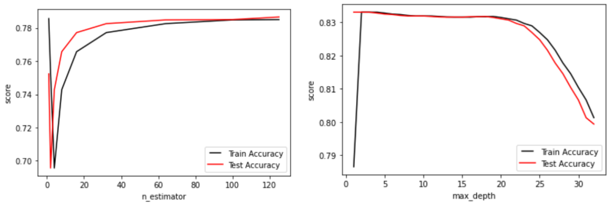
5. RESULTS AND CONCLUSIONS

The result of the collaborative filtering model shows an f1 score of 0.59, higher than the rating prediction baseline model. The weight we choose for average rating per user is 0.61, and the weight for average

[4] https://www.quirks.com/articles/how-consumer-feedback-can-change-what-you-know-about-seasonal-buying-patterns

rating per item is 0.39. Compared with random forest classifiers, the model performance of collaborative filtering doesn't have significant performance. One of the reasons might be that average ratings of a product would produce bias for rating prediction, since people differ from each other. Thus, we have to consider more complex models when working on rating prediction models .

The result of the random forest classifier shows an f1 score of 0.83, higher than the rating prediction baseline. The parameters we choose are 100 estimators and 22 maximum depth. The final model is achieved through an optimization process.



Random forest result interpretation

There exists a certain threshold that using a large value of the parameters would instead deteriorate the model performance. We found the random forest classifier using review information such as vote, verified, review text, review length and time successfully predicted a binary rating outcome.

On the other hand, in terms of buyers' behavior predictions, Thanks to Catboost's high performance and greedy novel gradient boosting implementation, the model trained by Catboost performed incredibly well, for which it has accuracy of 0.9146 among 160844 test data points.

Thus, when dealing with datasets which contain large numbers of categorical variables, and you have limited time for tuning parameters, algorithms like Catboost are highly recommended.

6. REFERENCE

[1] Wasserman, L. (2019). Random Forests. https://www.stat.cmu.edu/~larry/=sml/forests.pdf

[2] Ni, J., Li, J., & McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *Empirical Methods in Natural Language Processing (EMNLP)*.

[3] Wang, B., Wang, G., Li, Z. (2019). *Predicting User Ratings Using Status Models on Amazon.com*

[4] Hayes, A., (2014). *How Customer Feedbacks can change what you know about seasonal buying patterns*

[5] Bollegala, D., (2021). Amazon releases dataset to help detect counterfactual phrases. https://www.amazon.science/blog/amazon-releases-dataset-to-help-detect-counterfactual-phrases