Judging a Book by its Cover: Book Rating and Would-Read Predictions using Historical trends, Temporal trends and Latent Factor Models

Samira Sebt, Vikrant Jaltare

February 6, 2025

1 Introduction

1.1 Exploratory Analysis of data (Section 1)

The data used for this project consists of Amazon book reviews[1]. Because of system capability limitations and time constraints only 500k of the 51 million reviews were randomly chosen from this dataset. Some basic data analysis revealed the spread of the timestamps for the reviews dates back to as early as 1996 and as late as 2018. This data contains 326,981 unique users and 3,673 unique books. A look at the average book rating distribution reveals a heavy bias of book ratings towards 4.5 and 5 (out of 5) as shown in Figure 1. This observation will later be leveraged as a baseline for book rating predictions.

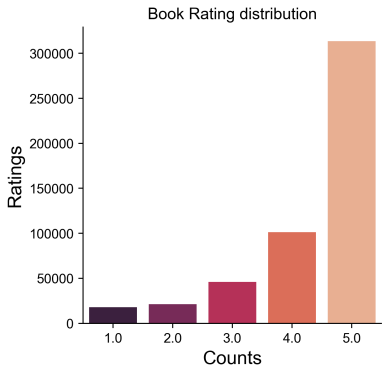

Figure 1: Number of ratings for every category from 1 to 5; 5 being the highest.

Another interesting trend to consider is temporal influences on average book ratings across years and months. Figure 2 shows the apparent trend that the book ratings are significantly influenced by temporal dynamics related to months and years, which therefore should also be considered when choosing model features. It should be noted that there are very few ratings for 1996 and 2018 as shown in Figure 3, so the average is over a very small number of ratings and not very informative. This fact is important to consider when designing features. Lastly, the temporal distribution of the number of reviews was plotted. This is done to ensure what results we are seeing from Figure 2 are actually informative and not skewed from lack of data availability.

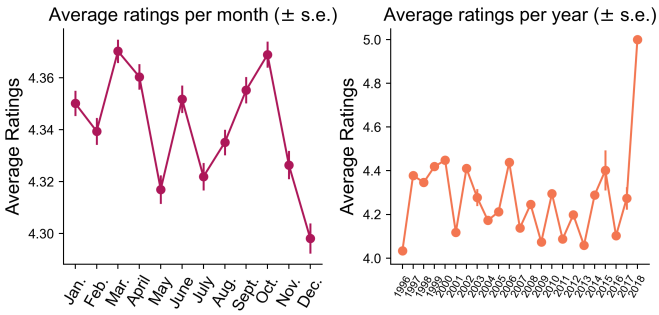

Figure 2: Average ratings per month and year. Errorbars are standard errors.

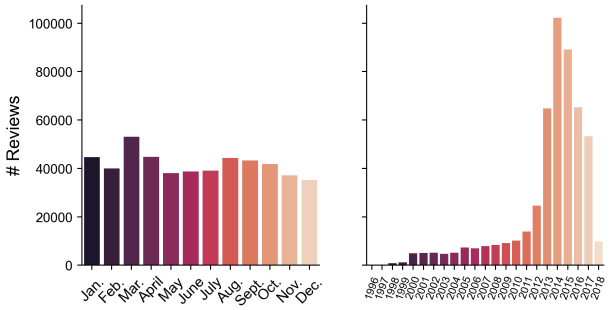

Figure 3: Number of reviews posted every year from 1996 - 2018. Note that there are very few samples from 1996 and 2018.

1.2 Identifying Predictive Task (Section2)

This work consists of two predictive tasks: 1. Rating predictions and what influences them in a population of readers. 2. Predicting whether a reader would read a certain book or not.

The bigger data pool was shuffled and randomly sampled to avoid any biases in the dataset. This sample was split into training, validation, and test sets with an 80-10-10 split. Interpretable timestamps were then obtained by using the `dateutil` library in Python for all sample reviews. For all tasks, all models were trained using the training data, hyper-parameters were tuned using the validation set and the model validity was assessed using the test set. Different metrics for the loss/performance were chosen appropriately for each task. Based on the exploratory analysis, it seems the data is heavily driven by temporal trends and book rating history/distribution. These will be the metrics that will be exploited when constructing useful features for the predictive tasks.

For rating predictions, various models were used and compared to the baseline. The performance of these models were evaluated using the mean-squared-error (MSE). From Figure 1 it became clear that a simple baseline for the rating predictions task can be obtained by simply predicting the average rating of books. This is a simple user-free model that performs surprisingly well. This further confirms that history of book ratings are significant in predicting ratings and should be appropriately leveraged.

In addition to this feature, from Figures 3 and 2 we can also conclude that leveraging the months could be a worthwhile endeavor as the number of reviews per month, Figure 3 left, seem quite uniform and yet the average ratings per month, Figure 2 left, are quite variable. This tells us that months of the year seem to have influence on how people rate the books they read. Furthermore, as we saw, this variability is not due to large variations in the number of reviews per month which make it a good feature to use. Pairing these results with the image on the right of 3, it is apparent that there should be a special focus on roughly the most recent 6 years which have the most relevant data (although 2018 is excluded from this for limited number of reviews). These years have disproportionately more reviews than other years and thus are thought to be the main driving factor behind the average historic ratings. Thus, rating history, temporal data with an emphasis on the months and ratings from most recent years were largely the focus of the features for this predictive task.

For the next predictive would-read task, again user-item and item-item interactions were chosen as well as book popularity. For this task, the performance metric used was accuracy. The reasoning for this

is again the aforementioned trends in the data. Popularity was used as a baseline following the potential trend that people tend to read popular books, people tend to watch popular movies, and people tend to play popular games. To improve this baseline, we also considered that people may read similar books to their previous books. Therefore item-item similarities and user-item interactions were the main focus of this predictive task.

2 Methods/ Model Selection (Section 3)

2.1 Predicting Ratings Task

To establish the baseline for the models, we first used a model that would always predict the global average of the ratings. Surprisingly, this model performed quite well due to the fact that most of the ratings were clustered around the 4-5 mark. We also used a similar model that depended on the user and the book such that the model would always predict their respective averages from the training dataset. With these three baselines we could establish that models generally predicting high ratings would perform better. In the baselines case, we didn't account for the temporal trends in the data.

We first tried to use a latent factor model using `Surprise` library's built in `SVD` and `SVD++` functions. The idea behind this approach was that finding low dimensional representations of users and books might be able to pick out notions like the likelihood that user rates the books high/low or the chance that the book in question in generally rated high or low based on user-item preferences. Thus, in essence we did not hand-pick any features for the matrix factorization method of latent factor model explicitly, rather the model inferred all the important characteristics. Spelled out in equations the model was

$$\hat{r}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i \tag{1}$$
$$= \text{SVD}(R) \tag{2}$$

Where $R \in \mathbb{R}^{u \times i}$ is the sparse user-book interaction matrix.

The library minimizes the squared errors with L2 penalty to find the parameters

$$\text{argmin}_{\hat{r}_{u,i}} \sum_{u,i} (\hat{r}_{u,i} - r_{u,i})^2 + \lambda(\alpha^2 + \beta_u^2 + \beta_i^2 + \|\gamma_u\|^2 + \|\gamma_i\|^2) \tag{3}$$

For our subsequent approach we performed ablation studies to see which factors influenced the rankings most – was it the book popularity, the temporal trends like the month or the year or the history of users purchases. To test this we trained linear regressors considering popularity, month, year and past history features. For popularity, we used binary encoding $X_{pop} \in 0, 1$ such that if a queried book b_i ranked in the list of top k books then the feature associated with it would be 1, i.e.

$$\theta_{b_i} = 1 \quad \text{if rank} \quad b_i \leq k, \quad \text{else} \quad 0 \tag{4}$$

We determined the optimal value of k using validation accuracy of the regressor and found it to be $k = 1000$. We had 3673 unique books in the dataset so $k \approx 0.3 \times |\mathcal{D}|$. We used one hot encoding for months and years. For the book history, we created a list of past t ratings that were given to these books. The regressors had L2 penalty thus we solved:

$$\vec{y} = \mathbf{X}\vec{\theta} \tag{5}$$
$$\text{argmin}_\theta \sum_i (\vec{y}_{i,pred} - \vec{y}_i)^2 + \lambda\|\theta\|^2 \tag{6}$$

To avoid overfitting we used the regression methods with L2 regularizer and cross-validation.

2.2 Would-Read Task

For this task as mentioned previously, we used popularity as the baseline for our models and obtained accuracy on the test set using the aforementioned optimized k value for popularity. Next, to leverage the influence of item-item interactions and user-item interactions we first constructed a Jaccard similarity model and a Bayesian Personal Ranking (BPR-based) model. The Jaccard simularity is simply

$$J(U_i, U_j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \tag{7}$$

Where U_i and U_j are the sets of books that users i and j reviewed respectively. Similarly the BPR framework is given as

$$p(i \text{ preferred over } j) = \sigma(\beta_i + \gamma_u \cdot \gamma_i - \gamma_i \cdot \gamma_j) \tag{8}$$

The code maximizes the likelihood

$$\max \ln \sigma(\beta_i + \gamma_u \cdot \gamma_i - \gamma_i \cdot \gamma_j) \tag{9}$$

where $\sigma(x)$ is the sigmoid function. In order to use these models, we constructed an augmented training, validation, and test sets to include"negative" interactions. Since the dataset is only comprised of books that users have already read, we needed to make sure we are providing the model with books the user has not read in order to properly train the models. The negative samples were chosen from the training set to avoid "data-leaking" and inflated accuracy results.

For Jaccard similarity, we compared each user-item interaction with the history of that user's items (books). This means for predicting whether or not a user would read a given book from the augmented validation set (which now consists of both positive and negative interactions), we computed the Jaccard similarity between the query book and the history of books the user has read from the training set (positive interactions only). This Jaccard similarity was calculated based on user overlaps for each item in question. We then took the maximum similarity score for each user-item interaction and found a threshold (by sweeping over an array of appropriate values) for would-read predictions on the augmented validation set. The accuracy, defined as correct predictions/total predictions, was then calculated on the augmented test set. As suspected, the Jaccard similarity performed better than the baseline popularity model.

Lastly, we decided to incorporate user-item interactions by using a BPR-based model. This is traditionally a "ranking" model which we decided to exploit to predict whether a user would read a particular book. The BPR model was adapted from McAuley *Personalized Machine Learning* [2] trained using `TensorFlow` and essentially is designed to assign a higher score on positive user-item interactions than negative user-item interactions during training. During training, for each positive user-item interaction a random negative user-item interaction was selected from the training set. Once this model was trained, in order to utilize this for would-read predictions, the predictive score for each user-item interaction was fed into a logistic regressor and trained on the augmented training set. This is the same augmented training set (same positive interactions but different random negative interactions from the training set) used for the BPR training, again to avoid any data leakage. Once the training was done, the accuracy was computed on the augmented validation set which was comprised of data the model has not seen in either of the steps. The accuracy obtained from this method was quite high but not significantly higher than the Jaccard method. This is not uncalled for since we already observed that the data is heavily driven by the fact that users typically tend to read books that are similar to their history of readings. BPR will further exploit this fact by capturing low dimensional latent factor features for user-item interactions as well as taking into account individual user and item attributes (in this case ratings, even though we are still predicting would-read). In this way it grants an improvement but does not provide a gross advantage over Jaccard for this dataset.

It is worth noting that for this predictive task 500,000 data points were used due to the limitations of our devices and time. The run-time of data sizes over 100k took a substantial amount of time. Therefore, we opted to use a smaller subset of the data. To avoid over-fitting the common practice is to implement an early-stop on the MSE or accuracy of the validation set and stop training once these values start increasing over multiple iterations. However, in this case it was challenging to implement anything like this since the BPR model outputs scores which don't necessarily correlate to anything"meaningful" by themselves. These scores were then fed into the logisitic regressor and so any tuning would have to be done on this part of the model using ridge or lasso penalty terms. However, this did not seem to make a significant impact as the model did not seem to be over-fitting. This was assessed by comparing the accuracy of the model on the training data, $\approx 90\%$, and on the test data, 88%

3 Results

3.1 Ratings Prediction Results (Section 4)

As discussed earlier, the baseline predictions used for this task were average ratings for each user, average ratings for each book, and the global average ratings across all books. Looking at the values in Table 1 we can see that the dataset is heavily influenced by the book ratings, while not as dependant on the

individual user ratings. Implementing a handful of other models with various features helped us further confirm the most important features of this dataset. Again looking at Table 1 we can see that temporal features are not as influential as we once previously thought from the exploratory analysis. While they do offer a reasonable MSE value (see ridge regression with month, year, month and year), it is still not as strong of a feature as the baseline using only average book ratings.

Naturally then it makes sense that the MSE acquired from linear regression using only the past ratings for each book would offer better results. It can be seen that the results are not significantly dependant on how far back the rating histories go. Although the most recent rating offers a better MSE than the last 6 ratings, and the last 10 ratings as features offers an even better MSE. However, based on the baseline, we know if we were to include the entire history of each book rating (book average rating MSE), the MSE would increase as the older ratings become less relevant.

The best performing model turned out to be the SVD model which leverages both user and book averages as well as low dimensional user-item attributes. This shows that there does seem to be some user-item attributes that are influential in the user rating of books. This could be the author, the genre, etc. These results show book ratings on average are quite consistent across different users. A reason for this may be because people with similar tastes tend to read similar books and that people are generally habitual in the type of books they prefer to read.

	Model	MSE
Baseline		
	Global Average rating across all books	1.093527
	User Average rating	11.69986
	Book Average rating	**1.012898**
Benchmark		
	SVD	**0.952935**
	SVD++	0.962546
	Ridge regression with only popularity	1.09121
	Ridge regression with month and year	1.079749
	Ridge regression with month ONLY	1.093205
	Ridge regression with year ONLY	1.080130
	Ridge regression with popularity and month	1.091675
	Linear Regression with only last book	0.989441
	Ridge regression with last three books	1.000133
	Linear Regression with last six books + month	0.998597
	Linear Regression with last 10 entries	0.983619

Table 1: Rating prediction results. The baseline models always predict either global or user or book average rating. The other trained models compare latent factor models with ablation studies

3.2 Would-Read Results

For this task, as mentioned in the above sections, the baseline used was popularity. For this task the top 1/3rd of books was deemed as an appropriate threshold for book popularity. The accuracy from this was higher than expected at $\approx 73\%$. Further confirming our hypothesis that people tend to read popular books. Looking at Table 2 we can see that Jaccard similarity and BPR both work quite well with accuracy well above the baseline. This is in line with our previous observation from the rating prediction tasks, showing that people tend to have specific taste in reading books. Thus, it makes sense that Jaccard similarity would yield such high accuracy results on this dataset. Lastly, the BPR model ensembled with a logistic regressor provided the best accuracy among the others. This is further proof that readers generally read books in accordance with specific preferences. The latent factor BPR model manages to somewhat discern this user-item relationship/preference and use it to make more well-informed decisions about whether a user would read a certain type of book or not. Although this is not explicitly done

through this method. As mentioned before, the BPR tensor flow model returns a score, and that score is subsequently fed into a logistic regressor and trained to find some correlation between the scores and would-read prediction. Over-fitting did not seem to be an issue here since the accuracy on the training set was $\approx 90\$\%$, and thus doesn't seem to showcase gross discrepancy between the training set accuracy and the test set.

	Model	Accuracy
Baseline		
	Popularity based prediction	0.736002
Benchmark		
	Jaccard similarity based prediction	0.857234
	Bayesian Personalized Ranking and logistic regression	**0.881065**

Table 2: Would read prediction results

4 Literature Review and Conclusion (Section 5)

In previous studies the Amazon Book review dataset has been extensively studied for rating prediction tasks. A recent paper from Ping Lin et al. [3] explored multiclass classifiers trained using the review text as features. They combined various text mining approaches like bag-of-words, word2vec, TF-IDF and even transformer BERT to get embeddings of the text. This paper, however, did not consider the co-dependence of user-book interactions and our models were able to surpass the accuracy from Ping Lin et al. Another recent paper from Torbati et al. [4] the problem of recommendation for users who have sparse but informative reviews was adressed. They used the same dataset as us and took a transformer-based approach. This was perhaps a case of 'latent factor models' where large language models (BERT, T5 and ChatGPT) were used to capture the user review key points. Although direct comparison with this model is difficult, our model is relatively simple and perhaps more interpretable than this study. However, our model is also biased towards high-rating books and users who review a lot of books. There have been also studies showing how different modalities of user interactions can be leveraged for better recommendation, including implicit feedback [5] such as mouse clicks. We would like to potentially extend our model to include some aspects of implicit feedback to improve book recommendations. In a similar study conducted on the LitRec dataset, Vaz et al. [6] used ensemble of collaborative filtering and content-based filtering to generate if a user will read a given book. In all, the state-of-the-art algorithms heavily rely on some transformer-based approach to extract the latent representations of user-book interactions. However, our method and the ones described in some of the literature pieces here demonstrate a more interpretable and computationally-efficient approach to book recommendation.

In conclusion, this study aimed at developing models for rating prediction and book recommendation for users from the Amazon Book review dataset[1]. We applied latent factor models to predict user ratings for a given book which seemed to outperform some of the similar models in the literature. We also did ablation studies to determine the features with the best predictive power for ratings and found that although temporal features like the month in which the book was reviewed had influence on the rating prediction, it was not as strong of a factor as previously thought. The strongest features turned out to be latent user-item attributes and history of the query book's ratings. In addition to this, we also built BPR and Jaccard similarity based models for book recommendation (i.e. would-read recommendation). We found that given the nuances of the dataset where popular books tend to be read more, and the fact that generally users tend to give high ratings to books -perhaps because they tend to choose books that they assume they would like in the first place- the chosen similarity metrics were an acceptable measure for good would-read predictions. In all, the results provide an interpretable approach to recommendation that leverages the features of the dataset to provide useful recommendations of books.

Acknowledgements

We would like to thank Prof. McAuley and TAs of the CSE 258R class for their invaluable inputs and help throughout the quarter.

References

1. Ni, J., Li, J. & McAuley, J. *Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (eds Inui, K., Jiang, J., Ng, V. & Wan, X.) (Association for Computational Linguistics, Hong Kong, China, Nov. 2019), 188–197. https://aclanthology.org/D19-1018.
2. McAuley, J. *Personalized machine learning* (Cambridge University Press, 2022).
3. Lin, H.-P., Chauhan, S., Chauhan, Y., Chauhan, N. & Woo, J. Amazon Books Rating prediction & Recommendation Model. *arXiv preprint arXiv:2310.03200* (2023).
4. Torbati, G. H., Tigunova, A., Yates, A. & Weikum, G. Recommendations by Concise User Profiles from Review Text. *arXiv preprint arXiv:2311.01314* (2023).
5. Wan, M. & McAuley, J. *Item recommendation on monotonic behavior chains* in *Proceedings of the 12th ACM conference on recommender systems* (2018), 86–94.
6. Vaz, P. C., Martins de Matos, D. & Martins, B. *Stylometric relevance-feedback towards a hybrid book recommendation algorithm* in *Proceedings of the fifth ACM workshop on Research advances in large digital book repositories and complementary media* (2012), 13–16.