Comparison of machine learning-based book comments sentiment analysis for constructing recommendation system

Jiaqi Weng

Department of Computer Science, Rensselaer Polytechnic Institute, Troy NY, 12180, USA

wengj3@rpi.edu

Abstract. The exponential growth in the volume of books available, along with the proliferation of online platforms, has made it increasingly challenging for readers to find books tailored to their interests. This research paper aims to address this challenge by developing an effective book recommendation system based on user reviews and ratings, primarily drawn from Amazon’s dataset covering the period from May 1996 to July 2014. Using a K-Nearest Neighbors (KNN) algorithm and a Random Forest baseline model, the study focuses on comparative analyses in terms of Mean Squared Error (MSE) and computational costs. The KNN model outperformed the baseline model with a lower MSE of 0.15 compared to 0.38 and proved to be computationally less exacting. While the KNN model is currently the more tenable option for deployment, the paper posits that an ensemble approach may offer a more robust solution. Future work aims to include sentiment analysis, explore other recommendation algorithms, and make use of more advanced evaluation metrics. This study provides a foundation for the advancement of book recommendation systems, offering insights into their efficiency and effectiveness.

Keywords: Sentiment Analysis, K-Nearest Neighbors, Random Forest, Recommendation System.

1. Introduction

With the increasing number of books available in the market, both in digital and printed format, finding the perfect book for every reader has become a significant challenge. The advent of online platforms like Amazon has made it possible to collect large datasets comprising user reviews and book details, which can provide valuable insights into user preferences and significantly enhance recommendation systems [1,2].

The primary objective of this research is to develop a recommendation system based on sentiment analysis using the various factors influencing book ratings provided by Amazon. To achieve this objective, this research has leveraged two datasets that capture nearly all information from May 1996 to July 2014 [3]. The datasets include the Reviews File and the Books Details File.

The Reviews File contains reviews from 3 million users on 212,404 unique books. This file includes features such as the book's ID, title, price, and the user's ID, profile name, review's helpfulness rating, review score ranging from 0 to 5, the time the review was given, as well as the review's summary and full text.

The Books Details File contains detailed information about the 212,404 unique books mentioned in the Reviews File, which were fetched using the Google Books API. This file includes attributes such as the book's title, description, authors, cover image URL, Google Books preview link, publisher's name, publishing date, additional information link, categories, and average ratings count.

This paper presents the methods used for the analyses, specifically focusing on the KNN-based recommender system [4,5]. The results and evaluations of the KNN-based recommender system have been compared with those of the Random Forest baseline model. The KNN model outperformed the baseline model with a lower Mean Squared Error (MSE) of 0.15 compared to 0.38 and proved to be computationally more efficient.

2. Method

2.1. Dataset and Preprocessing

The first critical step in building the recommender system was data collection and preprocessing. We utilized a subset of the Amazon Books Reviews dataset, which comprises multiple features such as book ID, user ID, review score, book title, and the full text of the review. The primary attributes used for the recommendation system were book ID, user ID, and review score, where the other elements were not considered as strong factors to us.

The dataset was then filtered based on the following criteria:

User Involvement: To ensure the robustness of the recommendations, we considered only the users who had provided ratings for more than 200 books. Filtering by active users allows the system to leverage more comprehensive user behavior, thus enabling more accurate recommendations.

Book Popularity: Only books with a minimum of 50 reviews were included in the dataset. This step helps in focusing the recommender system on books that have been widely received, thereby increasing the generalizability of the model.

Text Normalization: The review text was stripped of commas and numerical digits to achieve a cleaner dataset, primarily to facilitate any text analytics that could be involved in future extensions of the system [6].

The filtered data was then saved to a new file for further analysis and model training. The rationale behind using these specific filtering thresholds was to balance the need for a sufficiently large dataset with the computational efficiency and model generalizability. A detailed statistical summary was generated post-filtering to ensure that the data distribution was conducive for model training. By adopting this approach, we aimed to tackle issues like data sparsity and cold-start problems in the recommender system, which often plague less rigorous data preprocessing steps. This meticulous preprocessing ensures a high-quality dataset that can support the building of an efficient and effective recommender system. In the next subsections, the methods for generating recommendations would be elaborated upon, taking into consideration the final filtered dataset.

Now the basic steps of data preprocess are finished. The subsequent steps of our research involve building the actual recommender models and assessing their performance. Various collaborative filtering models were explored using the Surprise library, a Python scikit building and analyzing recommender systems.

2.2. Random Forest

Before stepping into more advanced model training, it is crucial to establish a baseline for performance comparison. In this research, the Random Forest method is selected as the baseline model [7,8]. First of all, one-hot encoding was employed using Scikit-Learn's OneHotEncoder to transform categorical variables into numerical variables. The Random Forest Classifier was instantiated with 80 estimators. This number was chosen to balance between computational efficiency and model performance. Random Forests are advantageous due to their ability to manage high-dimensional data and tolerate missing values, making them an apt choice for our baseline model.

This model will later be compared with other advanced models in the next section. By establishing this baseline, we aim to gauge the relative effectiveness of advanced recommendation algorithms and understand the incremental benefits they offer over simpler models. The ultimate goal is to identify the most efficient and accurate recommendation system strategy, balancing factors like model complexity, computational cost, and accuracy.

2.3. K-Nearest Neighbors (KNN)

The first advanced technique deployed in this study for the recommendation system is collaborative filtering via KNN. This method was selected as it often produces excellent results with minimal configuration, and it complements the other models well [9,10].

The KNN algorithm was specifically configured using Pearson's baseline similarity measure with a focus on item-based collaborative filtering (user_based: False). The choice of Pearson's baseline over other similarity measures like cosine or plain Pearson is due to its intrinsic adjustment for baseline ratings of users and items, thus accounting for the varying average rating behavior among different users and items. This is especially pertinent in recommendation systems where different users have different rating scales.

3. Result

The baseline estimates were calculated using a stochastic gradient descent algorithm (SGD) with Alternating Least Squares (ALS) regularization. The learning rate for the SGD algorithm was set at 0.003. The choice of using ALS regularization with SGD is strategic, as it helps prevent overfitting while optimizing the ratings matrix. KNNWithMeans, a variant of basic KNN, was used with k=10 and min_k=7. KNNWithMeans computes the mean of the ratings and adjusts the ratings based on the mean, making it sensitive to the average ratings by users.

The algorithm was trained on the 80% training set, and its performance was tested using the 20% test set that had been reserved. The algorithm's accuracy was assessed using the Mean Squared Error (MSE), which is particularly useful for evaluating models that predict ratings. The losses are demonstrated in Figure 1. The model yielded an MSE of 0.1498, which provides a reasonable level of accuracy for a recommendation system.

The KNN model will be evaluated in conjunction with the baseline model Random Forest, to understand how these different approaches might be combined to achieve optimal performance. The goal is to assess if an ensemble of these models or a standalone model could better serve the recommendation purpose. Therefore, the individual and collective performance metrics of each model, such as accuracy and MSE, will be closely monitored. This multi-pronged approach to recommendation systems provides a holistic view and allows for more nuanced recommendations, taking into account not only individual preferences but also broader patterns in user-item interactions.

To quantify the model's predictive power, accuracy was used as the primary evaluation metric. While accuracy is a rudimentary measure, it provides an initial understanding of how well the model performs on unseen data. Further evaluations with more sophisticated metrics such as precision, recall, and F1-score are considered for future work. Comparison with Advanced Models This baseline Random Forest Classifier will serve as a point of comparison against more sophisticated models developed later in the study.

As a result, our KNN model was used to do actual recommendation tasks. We used a loop to go through each book that this user had not reviewed yet, calculating their predicted sentiment score, and recommend the top 15 candidates, as displayed in Table 1.

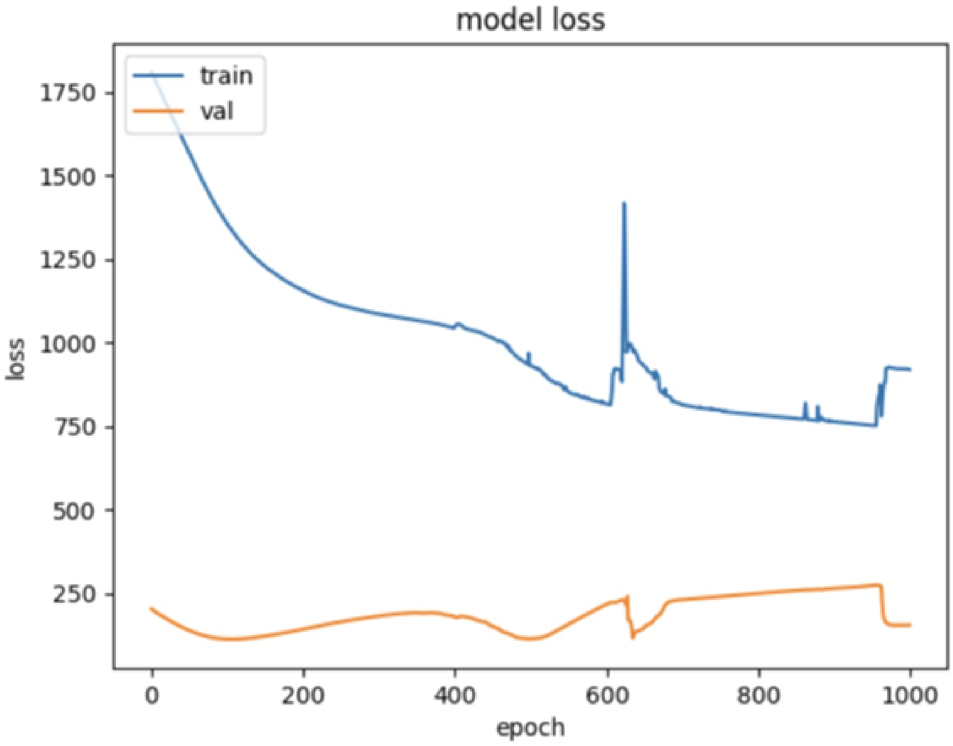

Figure 1. Loss curves of training and validating (Figure Credits: Original).

Table 1. Example result of top 15 candidate books

Redommendation results
'A striving after wind'
'House of the Sleeping Beauties'
'A Case of Conscience'
'The Scarlet Letter A Romance'
'The Rebel'
'Sea Wolf'
'A11 quiet on the western front'
'Shakespeare's Macbeth; (Macmillan's English classics) '
'Consilience (University Press Audiobooks)'
'Beethoven (Oxford paperbacks)'
'Counterpoint'
'Taiji Chin Na: The Seizing Art of Taijiquan (Chinese Internal Martial Arts)'
'The New Americans: How the Melting Pot Can Work Again'
'Small-Circle Jujitsu'
'SPACE, TIME AND ARCHITECTURE: THE GROWTH OF A NEW TRADITION.'

4. Discussion

One of the primary objectives of this study was to develop an effective book recommendation system by comparing various algorithms. In this regard, Random Forest and k-Nearest Neighbors (KNN) models were evaluated.

4.1. Performance Metrics

Both models were initially evaluated using Mean Squared Error (MSE). To quantify and compare the model's predictive power, accuracy was used as the primary evaluation metric. More specifically, we used Mean Squared Error (MSE) in this case, which is particularly useful for evaluating models that predict ratings. Our baseline model using Random Forest yields an MSE of 0.38 while the KNN model yielded an MSE of 0.15, which provides a reasonable level of accuracy for a recommendation system.

4.2. Computational Costs

While the Random Forest model provided a reasonable baseline MSE, it was computationally more intensive compared to the KNN model, taking almost double the time to train on the same dataset. This difference in computational cost could be crucial when deploying the model in a real-world application where timely recommendations are needed.

4.3. Future Work

The current dataset is sourced exclusively from Amazon and spans from May 1996 to July 2014. This time-frame and source could introduce inherent biases or limitations in generalizing the model to other platforms or more current user behaviors. Future work could include collecting more data of both books and reviews, integrating sentiment analysis to better understand the qualitative aspects of user reviews. Moreover, other recommendation algorithms like Matrix Factorization or Neural Collaborative Filtering could be tested to see if they offer any performance advantages. While MSE and accuracy have been useful for initial evaluations, more nuanced metrics like AUC-ROC and Cumulative Gain could offer deeper insights into model performance.

5. Conclusion

This study set out with the primary objective of developing an efficient and effective book recommendation system. Both Random Forest and KNN algorithms were tested against this objective. While both models showed promise, KNN offered distinct advantages in terms of computational cost and applicability to recommending different types of books to users.The KNN model demonstrated lower MSE and was computationally less intensive, making it the current choice for deployment. However, an ensemble approach incorporating both models could potentially offer a more robust and versatile system.By highlighting both the successes and limitations of our approach, it is expected to provide a foundation for future research in the field of recommendation systems, particularly those focused on book recommendations.

References

[1] Islek, I., & Oguducu, S. G. (2022). A hierarchical recommendation system for E-commerce using online user reviews. Electronic Commerce Research and Applications, 52, 101131.

[2] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 52(1), 1-38.

[3] Amazon Books Reviews, URL: https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews. Last Accessed: 2023/09/18

[4] Subramaniyaswamy, V., & Logesh, R. (2017). Adaptive KNN based recommender system through mining of user preferences. Wireless Personal Communications, 97, 2229-2247.

[5] Bahrani, P., Minaei-Bidgoli, B., Parvin, H., Mirzarezaee, M., & Keshavarz, A. (2023). A new improved KNN-based recommender system. The Journal of Supercomputing, 1-35.

[6] Anandarajan, M., Hill, C., Nolan, T., Anandarajan, M., Hill, C., & Nolan, T. (2019). Text preprocessing. Practical text analytics: Maximizing the value of text data, 45-59.

[7] Rigatti, S. J. (2017). Random forest. Journal of Insurance Medicine, 47(1), 31-39.

[8] Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25, 197-227.

[9] Kramer, O., & Kramer, O. (2013). K-nearest neighbors. Dimensionality reduction with unsupervised nearest neighbors, 13-23.

[10] Jiang, L., Cai, Z., Wang, D., & Jiang, S. (2007). Survey of improving k-nearest-neighbor for classification. In Fourth international conference on fuzzy systems and knowledge discovery, 1, 679-683.