DePaul University

Data Science Capstone

Yelp Restaurant Recommendation Systems: Content and Collaborative Filtering

Jenish Dobariya, Jorge Fernandez

ABSTRACT

Recommender systems are powerful tools designed to filter and present relevant information to users, tailored to their preferences and behaviors. These systems have become integral in various domains, including e-commerce, entertainment, and social media, enhancing user experience by simplifying decision-making processes. We explore the fundamental principles behind recommender systems, clarifying their importance and widespread applicability.

We delve into the two primary approaches to recommender systems: collaborative filtering and content-based filtering. Collaborative filtering relies on the collective intelligence of user interactions and preferences, recommending items based on the similarities between users or items. In contrast, content-based filtering focuses on the characteristics of items and user profiles, suggesting items with similar attributes to those previously liked by the user. By comparing these methodologies, we highlight their respective advantages, limitations, and scenarios where each approach excels. Through this comparative analysis, we aim to provide a comprehensive understanding of how recommender systems function and their critical role in personalizing user experiences across digital platforms. Our study reveals that the best performing collaborative filtering model was a User-Based approach with adjusted cosine similarity, achieving a Root Mean Square Error (RMSE) of 0.927. On the other hand, the best performing content-based model was the stacked regressor, which achieved an RMSE of 1.02.

June 14, 2024

Contents

1 Introduction

Recommender systems have revolutionized user experience across various digital platforms, from e-commerce to entertainment, by effectively managing the extensive array of choices available and enhancing user engagement. These systems are help users navigate the complexity of digital environments by presenting tailored options that align with individual preferences, thus improving decision-making and user satisfaction. As integral components of personalized marketing and customer interaction strategies, recommender systems boost business performance by increasing user retention and engagement.

Collaborative Filtering: Collaborative filtering, one of the primary methodologies in recommender systems, leverages the collective intelligence of user behaviors to suggest items. This method can be segmented into user-based and item-based approaches. User-based collaborative filtering recommends items by identifying users with similar behaviors or preferences and suggesting items liked by these similar users. The strength of this approach lies in its intuitiveness and the direct way it leverages user opinions. However, it often suffers from scalability issues and can struggle with data sparsity where user interactions are limited. Conversely, item-based collaborative filtering focuses on the similarities between items, recommending items like those the user has previously liked. This method tends to be more stable and scalable as the relationships between items change less frequently than those between users. However, it may not capture the complexity of user preferences when users' tastes diverge significantly from historical patterns.

Content-Based Filtering: The rationale behind choosing a content-based approach stems from several compelling factors. Firstly, content-based recommender systems excel in new item recommendations, an essential feature for platforms like Yelp, which constantly receive new reviews and business entries. Unlike collaborative filtering systems that rely heavily on user interaction data, content-based systems analyze item attributes, ensuring that even newly added items can be accurately recommended based on their content. Secondly, Yelp reviews are rich in textual information, offering a valuable source of data for understanding user preferences and item characteristics. By utilizing word embeddings, we can capture the semantic meaning of words and phrases within reviews, creating a nuanced representation of user opinions and item descriptions. This granular understanding allows the recommender system to match users with items that genuinely reflect their tastes and preferences. Moreover, incorporating sentiment analysis enhances the recommender system’s accuracy and relevance. Sentiment analysis helps in discerning the underlying emotions and opinions expressed in user reviews, adding an additional layer of personalization. By evaluating the sentiment polarity of reviews, the system can better gauge user satisfaction and tailor recommendations that align with positive experiences.

The objective of this paper is to explore and evaluate these two strategic frameworks within the context of restaurant recommendations on Yelp, a platform rich with user reviews and restaurant information. This dataset provides a robust foundation for applying advanced modeling techniques and extracting insights into user preferences and behavior, which are critical for assessing the performance of different recommender systems in a real-world context.

1.1 Literature Review

Recommender systems emerged as a significant area of interest in the mid-1990s, primarily in response to the burgeoning amounts of digital content and the corresponding need for efficient information filtering technologies [1]. These systems are designed to facilitate user interactions by personalizing the user experience through targeted suggestions based on historical data and user activity. The GroupLens project, developed by Resnick, Iacovou, Suh, Bergstrom, and Riedl in 1994, introduced collaborative filtering to a wider audience, showcasing its utility in filtering news and helping users navigate large information spaces efficiently [2].

Content-Based Filtering: Content-based filtering algorithms recommend items by analyzing the content associated with items that a user has previously liked, employing the attributes of the items themselves to make predictions. This approach is well-suited to scenarios where rich item metadata is available, such as in recommending articles or movies based on genres or tags. Early implementations often used simpler, heuristic methods, but the field has evolved to incorporate more sophisticated machine learning techniques. Pazzani and Billsus (1997) provided a foundational overview of the approach, highlighting its reliance on item-specific characteristics to drive recommendations [3]. More recent advancements have integrated methods like decision trees, neural networks, and support vector machines, enhancing the system's ability to handle complex and nuanced item features. [4]

Collaborative Filtering: Collaborative filtering techniques analyze patterns of user behavior to recommend items by identifying relationships between users or between items. Initially focused on user-based methods, the approach recommends items based on the preferences of similar users. This method, while intuitive, often struggles with issues such as scalability and data sparsity. Item-based collaborative filtering, popularized by Sarwar, Karypis, Konstan, and Riedl (2001), addresses some of these challenges by comparing items directly, thus providing more stable and scalable recommendations [5]. The evolution continued with the introduction of matrix factorization techniques, which have significantly improved the ability to model user preferences at scale. Koren, Bell, and Volinsky's work on matrix factorization has become a cornerstone in the field, particularly their application in the Netflix Prize competition, which underscored the effectiveness of these models in handling large, sparse datasets from real-world scenarios. [6]

2 Data

2.1 Data Collection

The dataset utilized in this study is a subset of Yelp's extensive businesses, reviews, and user data, made available through the Yelp Dataset Challenge. The dataset encompasses information from eight major metropolitan areas across the USA and Canada, provided in five JSON files. Each file corresponds to a single object type with one JSON-object per line.

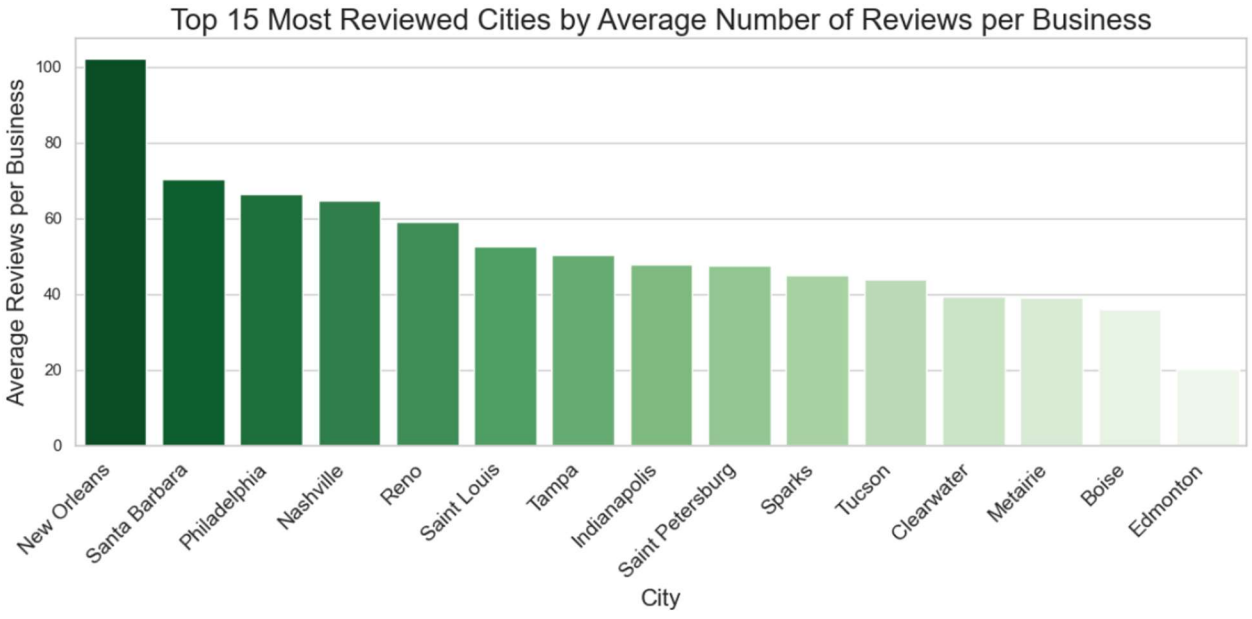
The files of interest for this analysis include:

- business.json: Contains data about businesses, including location, attributes, and categories.
- review.json: Comprises full review texts along with associated user and business IDs.

2.2 Exploratory Data Analysis

Initial exploratory data analysis revealed that the dataset is vast, encompassing over 6 million reviews across approximately 1,400 cities over 16 years. The sheer volume and diversity of the data posed significant challenges in terms of processing power and analytical focus.

Given this extensive scope, we decided to narrow our focus to enhance the specificity and relevance of the recommender system. New Orleans was selected as the primary city for this study, primarily due to having the highest number of reviews per business among the larger cities in the dataset, as seen below.



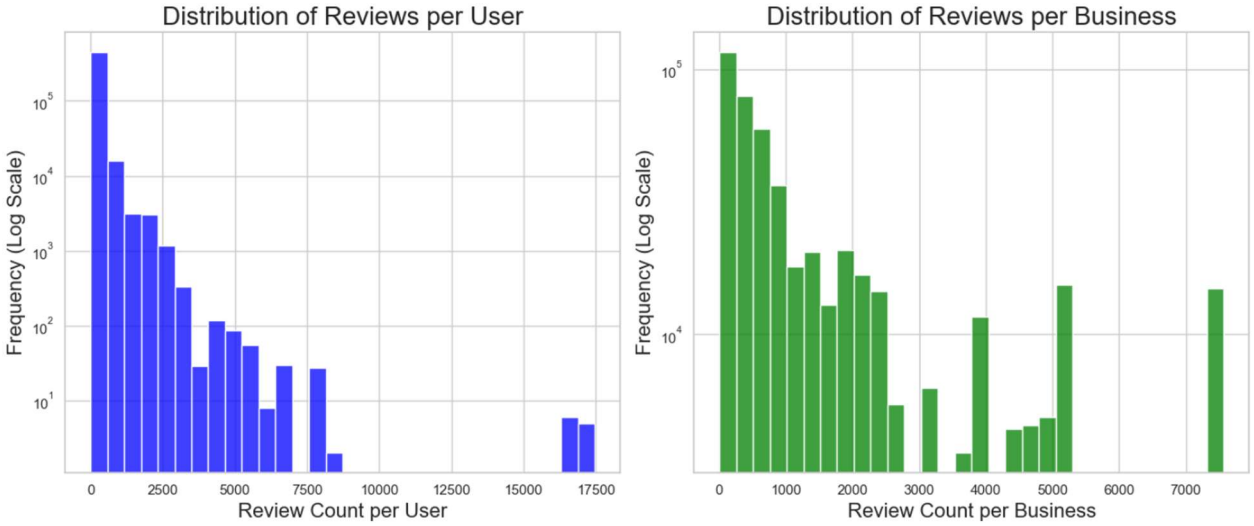
This high density of reviews per business provides a rich dataset that is more suited for developing an accurate and reliable recommender system. More reviews per business means

more comprehensive user feedback and opinions, which significantly enhances the system's capacity to learn and predict user preferences with higher precision. Localized data may also show more coherent patterns in consumer behavior and business attributes, which can improve the accuracy and usefulness of the recommendations.

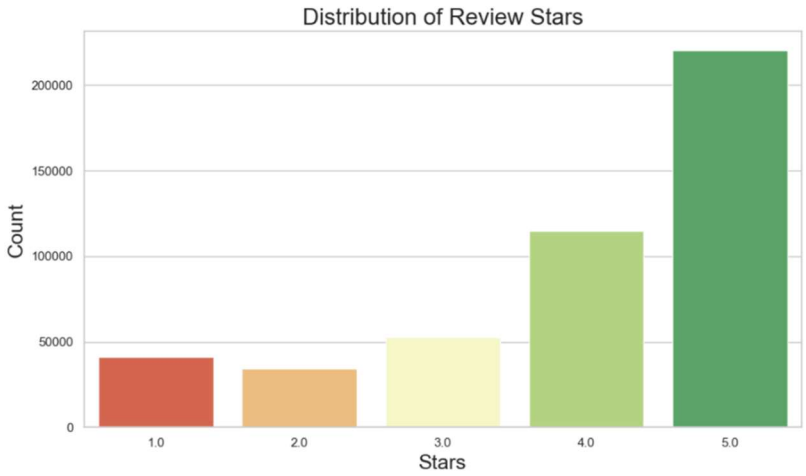
After narrowing our focus to New Orleans due to its high density of reviews per business, we conducted an exploratory analysis of the remaining 466,007 restaurant reviews to better understand the dynamics of user feedback and business performance in this locale.

The following histograms for both reviews per user and reviews per restaurant in New Orleans reveal a distribution that is heavily skewed towards smaller values. Most users and businesses have a relatively low number of reviews, while a small fraction have significantly more

These patterns suggest that the recommender system needs to handle a wide range of user engagement levels.



Further analysis of the distribution of star ratings given by users to New Orleans restaurants is shown below. This distribution is very positively skewed, with higher frequencies of 4-star and 5-star reviews. However, there is a noticeable prevalence of 1-star reviews over 2-star ratings, which could indicate user tendencies to leave feedback on extreme experiences.



- **Positive Bias**: The prevalence of high ratings may impact the predictive accuracy of the system if it overestimates the likelihood of positive experiences.
- **Critical Feedback**: The relatively high occurrence of 1-star ratings suggests that users are also keen to report negative experiences.

Understanding these trends helps in refining the algorithms used in the recommender system to better predict user preferences and to balance the impact of extreme reviews.

A word cloud of the most frequent words used in the reviews is shown below. It shows a visual representation of common themes and concerns expressed by users. This visualization can help to identify factors that influence user satisfaction. This word cloud however mostly reinforces the positive bias in reviews, as seen from the prevalence of positive phrases in the visualization, and lack of negative ones.

3 Content Based Filtering

3.1 Methodology

3.1.1 Data Preprocessing

Data preprocessing involved several stages to ensure the dataset was clean and suitable for analysis. This included:

1. Text Cleaning: Utilizing the Natural Language Toolkit (NLTK), we removed stop words, special characters, and extraneous whitespace from the review texts.
2. Tokenization: Splitting review texts into individual words or tokens.
3. Stemming and Lemmatization: Reducing words to their base or root form to standardize variations.

Additionally, we handled missing values by removing reviews with incomplete information and businesses without any associated reviews.

3.1.2 Feature Extraction

Feature extraction was conducted to convert textual data into numerical representations that could be utilized by machine learning algorithms. We employed word embeddings to capture the semantic meaning of words within the reviews. Specifically, we used GloVe embeddings with a vector length of 50 to transform the text into dense vector representations.

For each review, the word embeddings of individual words were averaged to create a single vector representation of the review. This approach enabled us to capture the overall sentiment and content of each review effectively.

3.1.3 Sentiment Analysis

To enhance the personalization of recommendations, we incorporated sentiment analysis to evaluate the sentiment polarity of user reviews. We utilized the VADER sentiment analysis library, a popular tool for analyzing social media text, to classify reviews and generate sentiment scores. These scores were integrated into the feature vectors, providing an additional dimension of user preference information.

3.1.4 Content-Based Filtering

We implemented several regression models to predict the relevance of items based on user reviews. The models used included:

Random Forest Regressor

- Decision Tree Regressor
- Linear Regressor
- K-Nearest Neighbors Regressor
- Support Vector Regressor

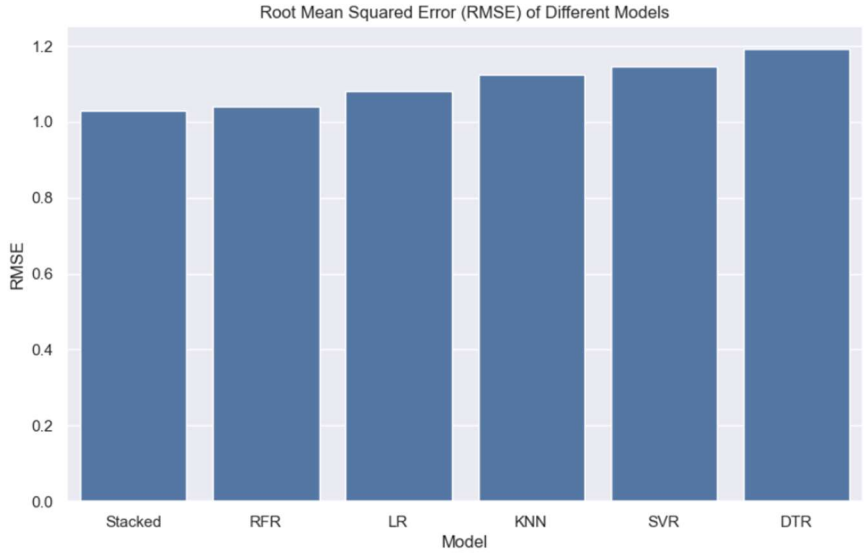
Hyperparameter tuning was performed to optimize the models. For cases with fewer than 10 hyperparameters, we employed brute force GridSearchCV. For scenarios with more than 10

hyperparameters, we used randomized grid search to efficiently explore the parameter space and identify the best parameters.

3.1.5 Model Stacking

To further enhance prediction accuracy, we created a stacked model using the best performing models from the individual regressors. In this stacked model, the Random Forest Regressor served as the final decider, leveraging its robustness and ability to handle complex interactions within the data.

3.2 Model Evaluation



The performance of the different regression models used in our recommender system was evaluated using the Root Mean Squared Error (RMSE) metric, as illustrated in the provided bar graph. The RMSE values for the models are as follows: Stacked (1.02), Random Forest Regressor (RFR) (1.04), Linear Regressor (LR) (1.08), K-Nearest Neighbors Regressor (KNN) (1.05), Support Vector Regressor (SVR) (1.06), and Decision Tree Regressor (DTR) (1.20). These results highlight the predictive accuracy of each model, with lower RMSE values indicating better performance.

Among the models, the stacked model exhibited the best performance with the lowest RMSE of 1.02. This demonstrates the effectiveness of combining multiple models to capture the complex patterns in the data, leveraging the strengths of individual models while mitigating their weaknesses. The stacked model outperformed all individual models, including the Random Forest Regressor, which had the next best RMSE of 1.04. This indicates that the ensemble approach not only improves prediction accuracy but also provides a more robust and reliable recommendation system. The superior performance of the stacked model validates its use as the final decider in our content-based recommender system, ensuring high-quality, personalized recommendations for users.

4 Collaborative Filtering

4.1 Methodology

4.1.1 Data Preprocessing

Data Preparation: Prior to analysis, duplicate reviews for the same user-business pair were averaged to ensure each interaction was uniquely represented, preserving the integrity of the dataset for collaborative filtering approaches. Most content data was also excluded, as collaborative filtering only requires ratings and user-business interactions.

Sparse Matrix Construction: Collaborative filtering techniques require a matrix format where rows represent users and columns represent items (restaurants in this case), with matrix values denoting ratings. Given the vast number of users and restaurants, most of the matrix elements are zero, so a sparse matrix format was used to efficiently store and manipulate the data. The transformation of the data into a sparse ratings matrix was achieved by converting user IDs and business IDs to categorical codes, which are then used as indices in the sparse matrix. The ratings themselves fill the matrix such that each element (i, j) in the matrix corresponds to the rating given by user i to restaurant j. The sparse matrix is implemented using SciPy's csr_matrix, which provides efficient row slicing.

Train-Validation-Test Split: To evaluate the performance of our recommender system, it was necessary to split the data into training, validation, and test sets. This split was conducted to ensure that each user's data appeared in only one of the sets, preserving the integrity of the evaluation and avoiding data leakage. The splitting process involved randomly shuffling the indices of rated items for each user to prevent bias in the allocation to subsets. Allocating two ratings per user to the test and validation sets, with the remaining ratings used for training. This process used a lil_matrix format for efficient assignment operations during the split, before converting the matrices back to a csr_matrix for optimized storage and processing. Post-split, integrity checks confirmed that the training, validation, and test sets were indeed disjoint, with no overlapping user-item interactions between them.

4.1.2 Similarity Calculations

Collaborative filtering relies on the calculation of similarity between users or items. The primary similarity metric in this analysis for both user- and item-based methods was the adjusted cosine similarity. Pearson Correlation and Jaccard Similarity were evaluated as well.

Adjusted Cosine Similarity: This method adjusts the ratings for user or item bias, focusing the similarity calculation on the variations of ratings from their respective means, thereby potentially enhancing the performance of by focusing on patterns of rating deviation rather than absolute ratings. While this approach is generally reserved for item-item similarity calculations, it was used in both user- and item-based methods here.

The general formula for adjusted cosine similarity between items is:

$$\text{similarity}(i,j) = \frac{\sum_{u \in U}(r_{ui} - \overline{r}_i)(r_{uj} - \overline{r}_j)}{\sqrt{\sum_{u \in U}(r_{ui} - \overline{r}_i)^2}\sqrt{\sum_{u \in U}(r_{uj} - \overline{r}_j)^2}}$$

Where r_{ui} is the rating of user u on item i, and $\overline{r}_i$ is the average rating of item i. U is the set of all users who have rated both items i and j.

When adapted to a user-user context, this becomes:

$$\text{similarity}(u,v) = \frac{\sum_{i \in I}(r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in I}(r_{ui} - \overline{r}_u)^2}\sqrt{\sum_{i \in I}(r_{vi} - \overline{r}_v)^2}}$$

Where r_{ui} is the rating of user u on item i, and $\overline{r}_u$ is the average rating of user u. I is the set of all items that both users u and v have rated.

4.1.3 Predictive Model Building

In our collaborative filtering approach, the model prediction process involves calculating user and item biases, and then leveraging these biases along with similarity metrics to predict ratings. This process enhances the accuracy of predictions by adjusting for inherent biases in user behavior and item characteristics before applying neighborhood-based filtering techniques.

Bias Calculation

- The initial step in our prediction workflow involves calculating the biases associated with each user and item
    - User Bias is computed as the average of the deviations of a user's ratings from the global mean.
    - Item Bias is calculated based on how an item's ratings deviate from the global mean

Rating Prediction Using k-Nearest Neighbors

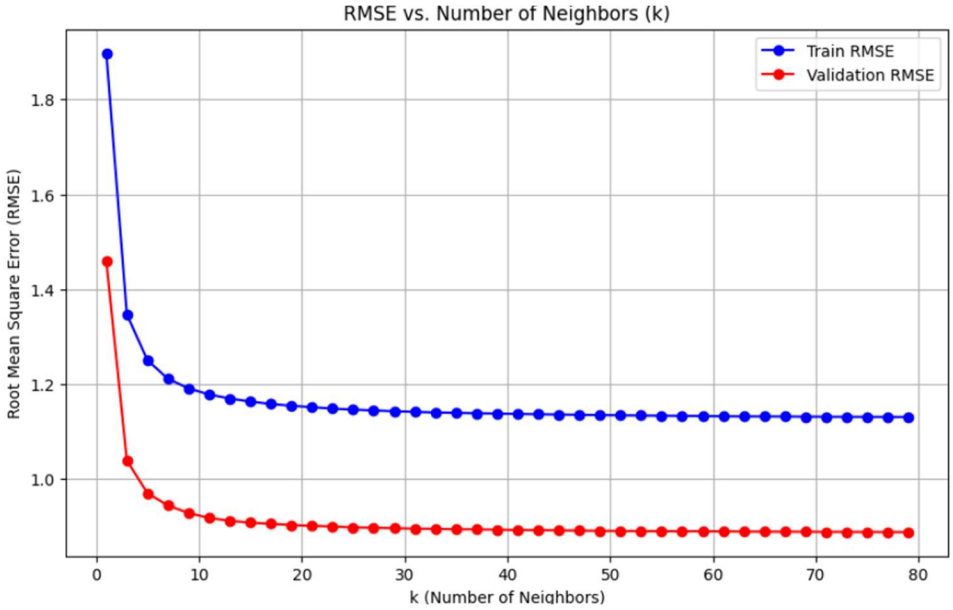
The core of our prediction methodology is the use of the k-nearest neighbors algorithm to predict ratings based on user-user or item-item similarities.

- **Inputs**: This process takes as inputs the ratings matrix, a precomputed similarity matrix (either user-user or item-item) and specifies whether to perform user-based or item-based predictions.
- **Bias Adjustment**: It begins by calculating the global average rating and initial biases calculated earlier. It then iteratively refines these biases to account for systemic biases in user and item rating behaviors.
- **Similarity-Based Prediction**: Depending on the type (user or item), the function:
    - **User-Based**: For each user, it identifies the top-k similar users based on the similarity matrix. It then predicts ratings as a weighted average of these top-k users' ratings, adjusted for user biases.
    - **Item-Based**: Alternatively, for each item, it calculates the predicted ratings as a weighted average of ratings from the top-k similar items, adjusting for item biases.

4.1.4 Parameter Optimization

The goal of parameter optimization in our model is to find the optimal number of nearest neighbors (k) that minimizes the prediction error, measured by the Root Mean Squared Error (RMSE). This process helps to enhance the recommendation quality by tuning the model to capture the most relevant user or item interactions.

To determine the optimal k-value for both user-based and item-based collaborative filtering approaches across the various similarity metrics, we implemented a systematic testing procedure to test the different k-values' impact on the prediction accuracy on both the training and validation sets. An example plot from this evaluation can be seen below:

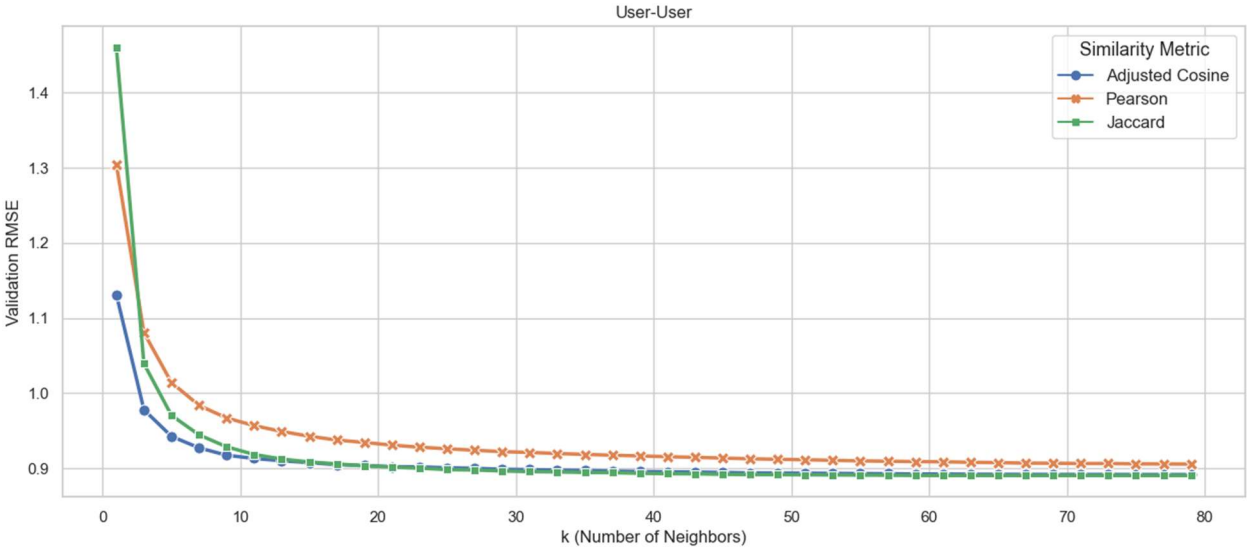


Observations:

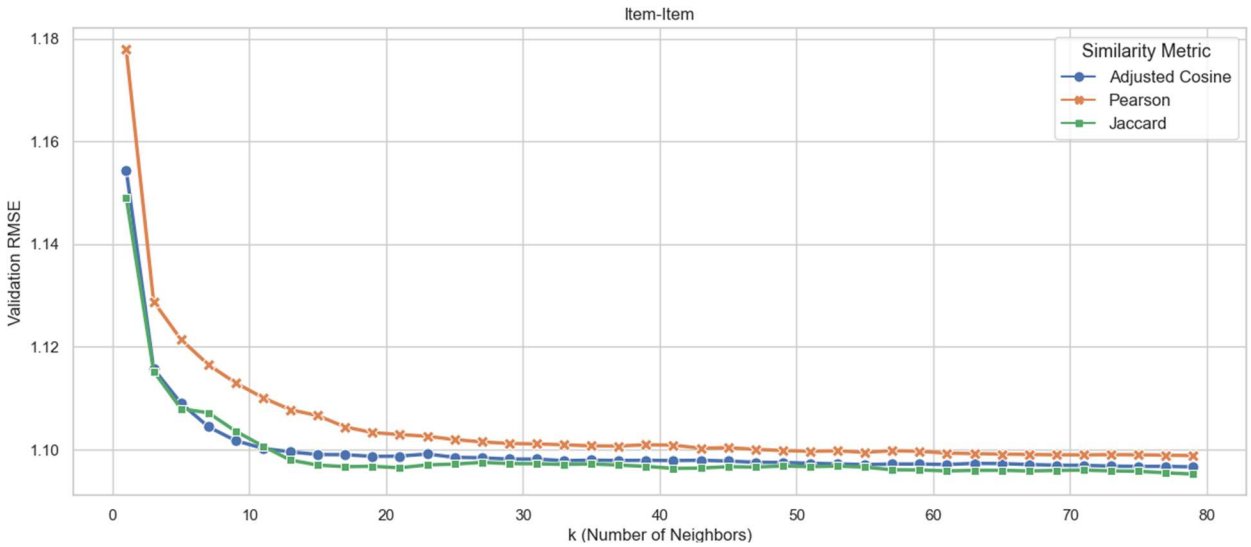
- **Early Performance Gains**: There is a significant drop in RMSE when increasing the number of neighbors from 1 to 5. This indicates that incorporating more neighbors initially leads to a substantial improvement in prediction accuracy.
- **Diminishing Returns**: Beyond k=10, the improvement in RMSE becomes marginal

4.2 Model Evaluation

Similarity Metric Performance:

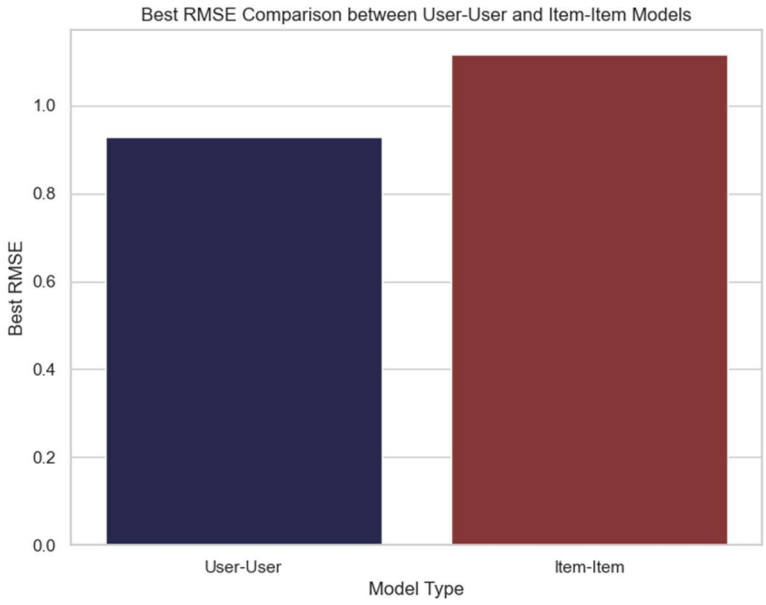


User-Based: Adjusted cosine similarity was the most effective metric in minimizing RMSE, as seen from the above plot. This metric's focus on item-specific user rating deviations seems to make it adept at capturing the nuances of user preferences.



Item-Based: Jaccard similarity was the top performer. This metric is suitable in capturing item interactions where binary presence rather than rating scale plays a larger role.

Optimal k-Value Selection: While the RMSE continued to decrease marginally with increasing k-values beyond the scope of the plots, a k-value of 20 was selected as the optimal balance. This choice was made to prevent the model from becoming overly complex while still capturing sufficient neighborhood information to make accurate predictions.



Comparative Performance:

A comparative plot of RMSE for the best configurations in both the user-based and item-based methods is seen above.

- **User-Based Method**: Achieved an RMSE of **0.927** on the test set using adjusted cosine similarity with k=20.
- **Item-Based Method**: Also recorded an RMSE of **1.11** on the test set using Jaccard Similarity and the same k-value.

5 Results

Here we consolidate the findings from both the content-based and collaborative filtering evaluations of our recommender system.

Content-Based Model Performance

The content-based models were evaluated based on their RMSE, with a focus on the stacked regression model due to its superior performance. Here are the RMSE scores for each model:

- **Stacked Model**: 1.02
- **Random Forest Regressor (RFR)**: 1.04
- **Linear Regressor (LR)**: 1.08
- **K-Nearest Neighbors Regressor (KNN)**: 1.05
- **Support Vector Regressor (SVR)**: 1.06
- **Decision Tree Regressor (DTR)**: 1.20

The stacked model, which integrates multiple regression techniques, demonstrated the lowest RMSE. This model effectively leverages the strengths of the underlying models while mitigating their individual weaknesses.

Collaborative Filtering Based Model Performance

In the collaborative filtering domain, both user-based and item-based methods were assessed:

- **Best User-Based Method**: RMSE of 0.927
- **Best Item-Based Method**: RMSE of 1.11

The user-based method outperformed the item-based approach.

Comparative Analysis and Final Selection

User-Based Collaborative Filtering yielded the lowest RMSE of 0.927, surpassing the stacked model's RMSE of 1.02.

For this dataset and set of features, collaborative filtering, specifically the user-based method using adjusted cosine similarity, is more effective in predicting user ratings than the best content-based approach. This superior performance could be attributed to the collaborative filtering method's ability to leverage collective user behavior patterns effectively, which are particularly strong indicators of individual preferences in this dataset.

6 Future Work

Development of a Hybrid Recommender System

A hybrid recommender system that integrates the content-based and collaborative filtering approaches could leverage the detailed item-specific information from the content-based models with the user behavior insights from the collaborative models. This integration can offset the weaknesses inherent in each model type.

Implementation Strategies:

1. **Feature Combination:**
    - One simple approach could be to combine the features or predictions from both models, using techniques such as ensemble learning where predictions from multiple models are weighted and aggregated.
2. **Model Stacking:**
    - Another approach could involve using the output of one model as an input feature for another. For example, the predictions from a collaborative filter could serve as features within a content-based model, or vice versa, to refine the final recommendation.
3. **Switching Systems:**
    - Depending on the context or user segment, the system could dynamically switch between collaborative and content-based recommendations. For instance, for new users with sparse interaction data, content-based recommendations could be prioritized.

The continuous improvement of our existing models and the exploration of hybrid recommender systems represent a promising avenue for future research.

References

[1] Z. Dong, Z. Wang, J. Xu, R. Tang and J. Wen, "A brief history of recommender systems," *arXiv,* 2022.

[2] P. Resnick, N. Iacovou, B. Suh, P. Bergstrom and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proc. 1994 ACM Conf. on Computer Supported Cooperative Work*, 1994.

[3] M. J. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," in *Machine Learning*, 1997, pp. 313-331.

[4] F. Ricci, L. Rokach and B. Shapira, "Recommender systems: Techniques, applications, and challenges," in *Recommender Systems Handbook*, 2021, pp. 1-35.

[5] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," in *Proc. 10th Int. Conf. on World Wide Web*, 2001, pp. 285-295.

[6] R. B. Y. Koren and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer,* vol. 42, no. 8, pp. 30-37, 2009.