Neural Collaborative Filtering:

A Recommender System for Matching on Exchange Platforms

DECEMBER 19, 2024

AZRA PARK

[redacted]

Abstract

This paper develops and evaluates a Neural Collaborative Filtering (NCF) recommender system designed to facilitate user exchanges on Ratebeer.com, a niche platform for beer enthusiasts. Using a dataset of 211,082 transactions, 118,453 wish list entries, and 62,879 give-away list entries collected between 2003-2016, I compare the performance of NCF against traditional matrix factorization approaches in generating accurate item recommendations. The study employs Bayesian Personalized Ranking (BPR) loss function to optimize for ranking performance, aligning with Ratebeer.com's objective of maximizing successful trades and user engagement. While my NCF model achieves a strong Area Under the Curve (AUC) score of 0.961, I find that incorporating additional embeddings for bidirectionality, social bias, and temporal dynamics yields only modest improvements, reaching a maximum AUC of 0.963. This contrasts notably with matrix factorization models, where these same embeddings produce substantial performance gains. My findings suggest that NCF's neural architecture may inherently capture certain user-item relationships that matrix factorization requires explicit embeddings to model. This work contributes to the growing literature on recommender systems for exchange platforms by offering a detailed comparative analysis of matrix factorization and neural collaborative filtering methods, building on prior research that debates their relative accuracy and applicability in different contexts. My findings provide insights into how these methods perform in facilitating user engagement and successful exchanges, benefiting the platform by increasing traffic, sustaining community participation, and enhancing revenue streams from advertisements and premium memberships.

1 Introduction

Recommender systems play a crucial role in modern digital platforms, analyzing user behaviour patterns to enable the prediction of user preferences and the delivery of personalized recommendations. While such systems are often studied in commercial settings where platforms aim to maximize profits through advertising or sponsored content, there are platforms with more niche objectives. One such example is Ratebeer.com, a community-driven platform for beer enthusiasts. Ratebeer.com generates revenue primarily through banner advertisements (e.g., Google Ads) and premium memberships, which offer an ad-free experience. To sustain its model, the platform focuses on increasing user traffic and engagement. Consequently, Ratebeer.com prioritizes optimizing the accuracy and relevance of its recommendations to facilitate successful transactions and exchanges through its forum, thereby encouraging continued user participation and growth.

Traditional approaches to matching users on exchange platforms, such as wishlist-based and give-away list matching, suffer from significant limitations. These methods rely exclusively on explicit user preferences, which are often incomplete and fail to account for items users might be unaware of but would find appealing if recommended. Moreover, matching pairs of eligible trading partners-where one user's wishlist aligns with another's give-away list-is inherently scarce, leading to few or even zero recommendations per user. This highlights the need for recommender systems capable of inferring implicit user preferences from transactional histories and serendipitously suggesting relevant items beyond explicitly declared interests.

To address these limitations, this paper extends the foundational work of Rappaz, Vladarean, McAuley, and Catasta (2017), who developed a linear matrix factorization-based recommender system tailored for exchange platforms. Their model incorporated key components such as bidirectional preference matching, social bias to capture repeated trades, and temporal dynamics to account for seasonality and user activity patterns. While effective, matrix factorization assumes a linear structure of user-item interactions, which may oversimplify the nuanced preferences and trading dynamics present on platforms like Ratebeer.com.

In this study, I adopt Neural Collaborative Filtering (NCF) as introduced by He et al. (2017) to overcome the linearity limitations of matrix factorization. NCF replaces the dot product with a deep neural architecture, enabling the model to capture complex, non-linear, and higher-order relationships between users and items. By leveraging the Multi-Layer Perceptron (MLP) framework, I aim to enhance the model's ability to identify hidden user preferences and improve recommendation accuracy. To optimize the NCF model for the implicit feedback setting inherent to exchange platforms, I employ the Bayesian Personalized Ranking (BPR) loss function (Rendle, Freudenthaler, Gantner, and Schmidt-Thieme (2012)), which directly optimizes for ranking performance rather than prediction error.

My evaluation criteria measures the accuracy of recommendations by using a user-specific Area Under the Curve (AUC), which evaluates ranking performance on a per-user basis. This formulation ensures that the model effectively prioritizes relevant items for each individual user, aligning with the personalized nature of exchange platforms. By comparing the performance of NCF against the traditional matrix factorization baseline, I investigate whether introducing non-linearities and higher-order interactions leads to superior recommendation outcomes in the context of an exchange platform.

This paper contributes to the growing literature at the intersection of recommender systems and matching theory by offering a comparative analysis of matrix factorization and neural collaborative filtering methods. By examining their relative effectiveness in enhancing recommendation accuracy, I aim to support platforms like Ratebeer.com in achieving their core objective: maximizing successful trades and fostering an active, engaged user community. My findings highlight how the choice of recommendation method impacts user engagement and exchange success, ultimately driving platform sustainability through increased traffic, advertising revenue, and premium memberships. This work highlights the nuanced trade-offs between classical and deep learning-based methods, offering insights into their role in maximizing accuracy, which supports community-oriented objectives and, in turn, serves the goal of profit maximization.

2 Literature Review

This work builds upon the foundational approach of Rappaz et al. (2017), who developed a comprehensive model for exchange platform recommendations using matrix factorization. Their model enhanced basic matrix factorization with several important components to capture the unique dynamics of bartering platforms, including bidirectional preference matching between users, social bias to account for repeated trading patterns, and temporal dynamics to reflect changing activity levels and seasonality. Through experiments on multiple real-world exchange platform datasets, they demonstrated that each additional component provided cumulative improvements to recommendation accuracy as measured by AUC.

Early work on recommender systems was not just focused on technical implementations, but also on their economic implications. Resnick and Varian ((1997), (1999)) provided foundational analyses of recommender systems from an information economics perspective, examining critical issues that remain relevant today. They highlighted key economic challenges, including privacy concerns around user data collection, incentive structures to encourage user participation in generating recommendations, and the monopolistic tendencies of these systems due to economies of scale. Larger user bases enable better preference matching, leading to network effects that predict market consolidation around a few dominant players, as observed in modern digital platforms. Furthermore, they addressed the tension between monetization strategies, such as advertising, and maintaining recommendation credibility-a challenge that remains central to recommender system design.

The economic value of recommender systems has since been empirically validated. Pathak, Garfinkel, Gopal, Venkatesan, and Yin (2010) demonstrated that recommender systems positively impact sales through mechanisms such as reducing consumer search costs, enabling cross-selling opportunities, and increasing customer loyalty by raising switching costs with increasingly accurate personalized recommendations. Similarly, Li, Grahl, and Hinz (2022) conducted a field experiment showing that personalized recommendations significantly increase both consumers' propensity to buy and average basket value.

Comprehensive surveys of the field provide essential context for understanding the evolution and methodologies of recommender systems. Lü et al. (2012) presents an extensive review, covering applications, algorithmic approaches, and evaluation methodologies, establishing fundamental frameworks for assessing recommendation performance. With the emergence of deep learning techniques, Zhang, Yao, Sun, and Tay (2019) provides an overview of neural approaches to recommendation, examining architectures, technical frameworks, and limitations while addressing key evaluation criteria and implementation challenges.

Building upon traditional collaborative filtering methods, He et al. (2017) introduced neural collaborative filtering (NCF) as a novel approach. Their work replaced the standard matrix factorization's inner product with a neural architecture, enabling the learning of arbitrary user-item interaction functions and capturing complex non-linear relationships. The Multi-Layer Perceptron (MLP) framework, in particular, allows for the learning of higher-order interactions through its deep architecture-a feature I adopt in this work. However, recent studies have questioned the purported superiority of neural methods. Ferrari Dacrema, Boglio, Cremonesi, and Jannach (2021) raised concerns about reproducibility in recommender systems research, emphasizing the importance of standardized baseline comparisons. Rendle, Krichene, Zhang, and Anderson (2020) demonstrated that properly tuned matrix factorization methods could outperform NCF on several datasets, though they acknowledged that relative performance may depend on the specific application domain. Similarly, Anelli, Bellogín, Di Noia, and Pomo (2021) expanded these investigations, examining NCF performance across multiple evaluation dimensions beyond accuracy, providing a more nuanced understanding of the contexts in which neural approaches may be advantageous.

The emerging intersection of recommender systems and matching theory provides a valuable lens for understanding user-item interactions on digital platforms. Ren et al. (2021) offer a comprehensive taxonomy of matching algorithms, distinguishing between explicit and implicit matching approaches. In explicit matching-traditionally studied in economics and mathematics-preference lists are known, and the objective is to achieve stable or optimal matches. However, the rise of large-scale digital platforms has shifted focus toward implicit matching, where preferences must be inferred or predicted. Recommender systems represent a key application of implicit matching, specifically in the domain of user-item matching.

In optimizing the recommendation model, I employ the Bayesian Personalized Ranking (BPR) loss function proposed by Rendle et al. (2012). BPR optimizes for pairwise ranking, ensuring that observed interactions (positive feedback) are ranked higher than unobserved interactions. This loss function is particularly effective in implicit feedback settings, where only positive interactions (e.g., clicks or transactions) are observed. By directly optimizing for ranking instead of prediction, BPR improves the model's ability to generate accurate personalized item rankings.

By building upon prior contributions, including those of Rappaz et al. (2017), He et al. (2017), and Rendle et al. (2012), I aim to advance the understanding of preference prediction and matching dynamics in exchange platforms, by leveraging deep learning architectures to capture nuanced user interactions.

3 Data

The dataset used in this study is sourced from Ratebeer.com, an online platform for beer enthusiasts, as originally collected by Rappaz et al. (2017). Ratebeer hosts a vibrant global community of users who engage in discussions, rate beers, and participate in local craft beers exchanges. The data was obtained through web crawling by the authors of the original study, extracting transactional details from users' textual submissions. These submissions provided information about completed exchanges, including the parties involved and the items exchanged.

The dataset includes 118,453 wish list entries, representing items users expressed interest in acquiring, and 62,879 give-away list entries, reflecting items users already own and are willing to exchange. Additionally, the data contains 211,082 total transactions, encompassing both unidirectional and bidirectional exchanges between users. While the data has been transformed into a dense matrix, the underlying user-item interaction data is highly sparse, as it is common on platforms for users to interact with only a small subset of the total items available. This sparsity underscores the inherent challenges that come with recommendation tasks, necessitating advanced techniques like Neural Collaborative Filtering (NCF) to infer latent preferences and improve recommendation accuracy. The data spans the period from October 1, 2003, to February 1, 2016, offering a comprehensive view of user activity and global craft beer trading over more than a decade.

4 Methodology

4.1 Neural Collaborative Filtering

Collaborative filtering generates recommendations by analyzing historical interactions between users and items. These interactions can be categorized into two types of feedback:

- **Explicit feedback**, such as user-provided ratings, reviews, or likes, which directly reflect preferences.
- **Implicit feedback**, such as clicks, browsing history, purchases, or item views, which indirectly indicate user interest.

Matrix factorization, as used by Rappaz et al. (2017), is a classical collaborative filtering technique that represents these interactions as dot products between user and item latent feature vectors. While effective, this approach assumes a linear relationship between users and items, which may oversimplify the complex nature of user preferences and interactions.

Neural Collaborative Filtering (NCF) addresses this limitation by replacing the dot product with neural networks, enabling the model to learn non-linear and higher-order relationships between users and items. Specifically, I employ the multi-layer perceptron (MLP) framework, where the NCF model begins by embedding users, items, and optional additional features (e.g., social connections, temporal dynamics) into dense, high-dimensional vectors.

$$\mathbf{h}^{(0)} = [\mathbf{e}_u, \mathbf{e}_i, \dots].$$

These embeddings are concatenated and then passed through hidden layers of the MLP to learn complex interactions with non-linear activation functions, in this case, ReLU:

$$\mathbf{h}^{(k)} = \text{ReLU}\left(\mathbf{W}^{(k)}\mathbf{h}^{(k-1)} + \mathbf{b}^{(k)}\right) \tag{1}$$

The final layer produces a predicted preference score $\hat{y}$ that predicts the likelihood of interaction between a user and an item, calculated as:

$$\hat{y} = \mathbf{w}^\top \mathbf{h}^{(L)} + b \tag{2}$$

Where:

- $\text{ReLU}(x) = \max(0, x)$ is the Rectified Linear Unit activation function.
- $\mathbf{h}^{(0)}$ is the initial concatenated vector of user, item, and optional embeddings.
- $\mathbf{W}^{(k)}$ and $\mathbf{b}^{(k)}$ are the weight matrix and bias vector for the k-th layer.

By leveraging deeper neural networks, NCF models capture complex, non-linear interactions between users and items. This richer representation allows the model to learn intricate user preferences, item attributes, and their interplay, resulting in improved recommendation accuracy over traditional linear models like matrix factorization.

4.2 Loss Function: Bayesian Personalized Ranking (BPR)

Bayesian Personalized Ranking (BPR) is a pairwise ranking loss function designed for implicit feedback scenarios, such as recommendations, introduced by Rendle et al. (2012). It acts as the objective function we seek to minimize. It aims to optimize the relative ranking of items for a given user by encouraging the model to assign a higher predicted score to items the user has interacted with (positive items) than to those they have not (negative items). The BPR loss is defined as:

$$\mathcal{L}_{BPR} = -\frac{1}{N}\sum_{j=1}^{N} \ln \sigma\left(\hat{y}_{u_j,i_m} - \hat{y}_{u_j,i_n}\right)$$

where:

- N: Total number of samples in the batch.
- U: Set of users.
- I: Set of items.
- u_j: A specific user $u_j \in U$.
- i_m: A positive item $i_m \in I$ that user u_j has interacted with or expressed preference toward.
- i_n: A negative item $i_n \in I$ that user u_j has not interacted with.
- $\hat{y}_{u_j,i_m}$: Predicted preference score of user u_j for item i_m.
- $\hat{y}_{u_j,i_n}$: Predicted preference score of user u_j for item i_n.
- $\sigma(x) = \frac{1}{1+e^{-x}}$: Sigmoid function.

The BPR loss encourages the model to maximize the margin between the predicted scores of positive items and negative items. Specifically:

- If $\hat{y}_{u_j,i_m} > \hat{y}_{u_j,i_n}$, the loss decreases, indicating that the model successfully ranks the positive item i_m higher than the negative item i_n.
- The sigmoid function σ smooths the loss and ensures differentiability for optimization.

BPR is particularly effective for learning personalized rankings in recommendation systems with implicit feedback data. Unlike explicit ratings (e.g., 1-5 star ratings), implicit feedback consists of observed user behaviours such as clicks, purchases, or views, where the absence of interaction is treated as a negative signal. BPR formulates the problem as a pairwise comparison between positive and negative items for each user:

- **Positive Interaction**: An item i_m that the user has interacted with is assumed to be preferred over items they have not interacted with.
- **Negative Interaction**: Items i_n without interaction are sampled as "negative" examples for learning.

By minimizing the BPR loss, the model learns to prioritize user preferences based on relative comparisons rather than absolute ratings. This is what makes BPR highly suitable for large-scale recommendation tasks with sparse, implicit feedback data.

Gradient of the BPR Loss Function

The gradient of the BPR loss with respect to the model parameters θ is given as:

$$\nabla_\theta \mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial \theta} \ln \sigma \left(\hat{y}_{u_i,i_m} - \hat{y}_{u_i,i_n} \right)$$

Here, the gradient measures the model's error in predicting the relative ranking of positive and negative items for a user. This gradient is then propagated through the model's parameters to update them during training.

By optimizing the BPR loss, the model improves its ability to produce personalized rankings that align with users' observed preferences in implicit feedback scenarios.

4.3 Parameter Updates: Adam Optimizer

To optimize the model's parameter updates, I employ the Adam optimizer, a widely used algorithm in deep learning that combines the advantages of momentum-based and adaptive learning rate methods. Adam stands for *Adaptive Moment Estimation* and maintains two moment estimates for each parameter during training: the first moment (mean of gradients) and the second moment (uncentered variance of gradients).

The parameter updates at time step t are defined as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta \mathcal{L}, \tag{3}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_\theta \mathcal{L})^2, \tag{4}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{5}$$

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}. \tag{6}$$

Where:

- $\nabla_\theta \mathcal{L}$: Gradient of the loss function $\mathcal{L}$ with respect to the model parameters θ.
- m_t: First moment estimate (exponentially decaying average of past gradients).

- v_t: Second moment estimate (exponentially decaying average of squared gradients).
- $\hat{m}_t$, $\hat{v}_t$: Bias-corrected first and second moment estimates, compensating for initialization biases.
- β_1, β_2: Exponential decay rates for the first and second moment estimates.
- η: Learning rate that controls the step size of updates.
- ϵ: A small constant to ensure numerical stability and prevent division by zero.

The parameter θ represents the model weights, which include:

- User embeddings e_u, item embeddings e_i, and any optional embeddings, which are dense vector representations.
- Weight matrices $\mathbf{W}^{(k)}$ and bias vectors $\mathbf{b}^{(k)}$ for each neural network layer.
- Final prediction layer weights $\mathbf{w}$ and bias b.

4.4 Evaluation Criteria: Area Under the Curve (AUC)

To evaluate the performance of the recommender system, I adopt the Area Under the Curve (AUC) metric, which measures the model's ability to rank items with positive user feedback higher than items with no observed feedback (negative samples). This methodology aligns with the approach outlined in Rappaz et al. (2017).

Evaluation Set E

The evaluation set E consists of triplets (u_j, i_m, i_n), where:

- $u_j \in U$: A user from the user set U,
- $i_m \in I$: A positively observed item (e.g., an item from the user's history H_j^r),
- $i_n \in I$: A negative item randomly sampled from the unobserved items, ensuring $i_n \notin W_j$ (user's wish list) and $i_n \notin H_j^r$ (user's history).

AUC Definition

The AUC is formally defined as:

$$\text{AUC} = \frac{1}{|E|} \sum_{(u_j, i_m, i_n) \in E} \mathbb{1}\left(\hat{y}_{u_j, i_m} > \hat{y}_{u_j, i_n}\right),$$

where:

- $\hat{y}_{u_j,i_m}$: Predicted preference score of user u_j for the positive item i_m,
- $\hat{y}_{u_j,i_n}$: Predicted preference score of user u_j for the negative item i_n,
- $\mathbb{1}(\cdot)$: Heaviside step function, which is defined as:

$$\mathbb{1}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0.5 & \text{if } x = 0, \\ 0 & \text{if } x < 0. \end{cases}$$

The AUC measures how well the model distinguishes between positive and negative interactions. Specifically:

- If $\hat{y}_{u_j,i_m} > \hat{y}_{u_j,i_n}$, the model correctly ranks the positive item i_m above the negative item i_n.
- The closer the AUC is to 1, the better the model's ranking performance.

Sampling Strategy for Positive and Negative Items

- **Positive items** i_m: These are sampled from the user's history of interactions H_j^r, which includes items the user has expressed preference toward.
- **Negative items** i_n: These are randomly sampled from the set of items the user has not interacted with, ensuring they are excluded from both the user's history H_j^r and wish list W_j.

Average AUC Across Users

The final evaluation metric is the average AUC across all users in the evaluation set. This metric is particularly suitable for ranking tasks in recommendation systems because it assesses how effectively the model ranks relevant (positive) items above irrelevant (negative) ones for each user. Therefore, I cannot visualize a single ROC curve representative of the entire dataset as the AUC is considered on an individual basis (i.e., the TPR and FPR are calculated separately for each user).

To ensure robustness of the AUC scores, I run each embedding configuration 5 times and average the AUC scores across runs. This procedure follows the methodology used in the paper I am replicating, where multiple runs help mitigate the impact of randomness and stochasticity in training and evaluation.

By following this evaluation methodology, I ensure consistency with the prior work of Rappaz et al. (2017), and provide a robust measure for ranking quality in implicit feedback recommendation systems.

4.5 Hyperparameters

After performing extensive hyperparameter tuning, the best combination of parameters for the model was found to include an embedding latent dimension of 8, a batch size of 512, 100 epochs, a single hidden layer with 2 neurons, and a learning rate of 0.001. The embedding latent dimension determines the size of the learned representation for users and items, with a smaller value like 8 balancing model complexity and generalization. The batch size defines the number of examples processed per training iteration, where a value of 512 allows for stable gradient updates while optimizing memory usage. Training for 100 epochs provides sufficient cycles through the dataset to converge without risking overfitting. The hidden layer, consisting of a single layer with 2 neurons, captures simple non-linear relationships, preventing excessive complexity while still enabling meaningful patterns to be learned. Lastly, the learning rate of 0.001 ensures gradual, stable updates to the model's parameters, balancing convergence speed and stability.

5 Results

Using the results from Rappaz et al. (2017) as a baseline, Table 1 shows a comparison between my Neural Collaborative Filtering (NCF) model and the matrix factorization (MF) baseline with additional embeddings for bidirectionality (B), social bias (S), and temporal dynamics (T).

My NCF model performs exceptionally well compared to the base matrix factorization model (MF), achieving a significant AUC of 0.961. When additional features such as bidirectionality (B), social bias (S), and temporal dynamics (T) are incorporated into the NCF framework—effectively adding these terms to the embeddings—their impact remains modest. However, consistent improvements were observed with the social bias and temporal embeddings across all runs, suggesting that these terms do contribute positively to the model's performance, albeit not to a significant degree.

One unexpected discovery in my experiments is the slight drop in performance when bidirectionality is added to the NCF model. This could be explained by the following:

- **Implicit Capturing of Bidirectionality**: The embeddings learned by the neural network may already infer and encode bidirectional relationships implicitly, reducing the need for an explicit bidirectional term.

- **Symmetrical Representations**: During training, the neural network might naturally learn symmetrical representations, particularly when optimizing pairwise rankings, making the addition of a bidirectional term redundant.

Table 1: AUC Performance Comparison

	Base	Base+B	Base+B+S	Base+B+T	Base+B+S+T
Baseline (MF)	0.824	0.892	**0.962**	**0.969**	**0.983**
NCF	**0.961**	**0.959**	**0.962**	0.963	0.962

Note: Upon attempting to replicate the findings of Rappaz et al. (2017), I was only able to achieve a maximum AUC of approximately 0.93, despite their published results demonstrating higher scores. However, I did not have access to all of the exact hyperparameters used in their experiments.

6 Conclusion

The results of this study highlight a nuanced dynamic between the base models of Matrix Factorization (MF) and Neural Collaborative Filtering (NCF) in the context of recommender systems for exchange platforms. While the additional embeddings significantly improved the performance of the MF model, their impact on the NCF model was negligible. This suggests that NCF's architecture may inherently capture certain relationships that MF relies on explicit embeddings to model. Future work could expand upon the NCF architectureâsuch as incorporating attention mechanisms or adopting a NeuMF frameworkâto better leverage additional embeddings and enhance its performance further.

These findings contribute to the ongoing discussion about the comparative strengths of MF and NCF, suggesting that the choice between the two should consider the specific objectives and characteristics of the application. For Ratebeer.com, whose primary goals are to drive user engagement and facilitate meaningful exchanges between users, the developed recommender system aligns closely with these objectives by maximizing recommendation accuracy. This work demonstrates how advanced recommendation techniques can be tailored to meet the specific needs of niche platforms, contributing valuable insights to the broader discourse on the comparative strengths of MF and NCF in recommender system applications.

A potential avenue for future research would be to incorporate user ratings of beers into the analysis. This addition could provide valuable explicit feedback, complementing the implicit feedback already used in the model, and may offer further insights into user preferences. Exploring how the inclusion of such data impacts the performance of both MF and NCF models could deepen our understanding of their applicability and effectiveness in niche platforms.

References

Anelli, V. W., Bellogín, A., Di Noia, T., & Pomo, C. (2021). Reenvisioning the comparison between neural collaborative filtering and matrix factorization. In *Proceedings of the 15th acm conference on recommender systems* (pp. 521–529).

Ferrari Dacrema, M., Boglio, S., Cremonesi, P., & Jannach, D. (2021). A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, *39*(2), 1–49.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

Li, X., Grahl, J., & Hinz, O. (2022). How do recommender systems lead to consumer purchases? a causal mediation analysis of a field experiment. *Information Systems Research*, *33*(2), 620–637.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics reports*, *519*(1), 1–49.

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R., & Yin, F. (2010). Empirical analysis of the impact of recommender systems on sales. *Journal of Management Information Systems*, *27*(2), 159–188.

Rappaz, J., Vladarean, M.-L., McAuley, J., & Catasta, M. (2017). Bartering books to beers: A recommender system for exchange platforms. In *Proceedings of the tenth acm international conference on web search and data mining* (pp. 505–514).

Ren, J., Xia, F., Chen, X., Liu, J., Hou, M., Shehzad, A., ... Kong, X. (2021). Matching algorithms: Fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *5*(3), 332–350.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th acm conference on recommender systems* (pp. 240–248).

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56–58.

Varian, H. R. (1999). *Markets for information goods* (Vol. 99). Citeseer.

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.