

NOVA

IMS

Information
Management
School

MDSAA

Master's Degree Program in
Data Science and Advanced Analytics

Evaluating Recommender Systems Qualitatively

A survey and comparative analysis

Tiago Alexandre Vaz Faria

Dissertation

presented as a partial requirement for obtaining the Master's Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

**EVALUATING RECOMMENDER SYSTEMS QUALITATIVELY. A SURVEY
AND COMPARATIVE ANALYSIS**

by

Tiago Alexandre Vaz Faria

Dissertation presented as a partial requirement for obtaining the Master's degree in Advanced Analytics, with a Specialization in Business Analytics

Supervisor: Prof. Roberto Henriques

November 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Tiago Faria

[Lisboa, November/2022]

ABSTRACT

Recommender systems have improved users' online quality of life by helping them find interesting and valuable items within a large item set. Most recommender system validation research has focused on accuracy metrics, studying the differences between the predicted and actual user ratings. However, recent research has found accuracy to underperform when systems go live, mainly due to accuracy's inability to validate recommendation lists as a single entity, and shifted to evaluating recommender systems using "beyond-accuracy" metrics, like novelty and diversity.

In this dissertation, we summarize and organize the leading research regarding the definitions and objectives of the beyond-accuracy metrics. Such metrics include coverage, diversity, novelty, serendipity, unexpectedness, utility, and fairness. The behaviors and relationships of these metrics are analyzed using four different models, two concerning the items characteristics (item-based) and two regarding the user behaviors (user-based). Furthermore, a new metric is proposed that allows the comparison of different models considering their overall beyond-accuracy performance. Using this metric, a reranking approach is designed to improve the performance of a system, aiming to achieve better recommendations. The impact of the reranking technique on each metric and algorithm is studied, and the accuracy and non-accuracy performance of each system is compared. We realized that, although the reranking technique can increase most beyond-accuracy metrics, the accuracy of that system starts to worsen due to the negative correlation between these two dimensions. We also found that item-based models tend to achieve much lower values of coverage and diversity than user-based models.

KEYWORDS

Recommendation System; Validation Metrics; Beyond-Accuracy; Offline Evaluation; Comparative Study

INDEX

1. Introduction	1
2. Beyond Accuracy Metrics	3
2.1. Coverage	3
2.1.1. Prediction Coverage.....	3
2.1.2. Catalog Coverage	4
2.1.3. Interest Coverage	4
2.2. Diversity	5
2.3. Novelty.....	7
2.4. Serendipity.....	10
2.4.1. Unexpectedness	10
2.4.2. Utility	12
2.4.3. Formulating Serendipity	12
2.5. Fairness	13
3. Beyond accuracy quality	15
3.1. Quality measures – item quality	15
3.2. Quality measures – system quality	16
3.3. Reranking approach.....	16
4. Experimental setup.....	18
4.1. Datasets	18
4.2. Algorithms	19
4.3. Performance metrics	19
4.3.1. Coverage	20
4.3.2. Diversity	20
4.3.3. Fairness	20
4.3.4. Novelty.....	21
4.3.5. Unexpectedness	21
4.3.6. Hit rate.....	22
5. Results and Discussion	23
5.1. Comparison of metric results.....	24
5.2. Enhancing Quality	25
6. Conclusion	29
6.1. Limitations and Future Work	30
7. References	31

LIST OF FIGURES

Figure 1- Setps for the reranking technique. After retrieving the model outputs, they are reranked and the top-N items are recommended	17
Figure 2 - Comparison of metric results for each evaluated model	24

LIST OF TABLES

Table 1 – GoodReads dataset features.....	18
Table 2 – Average, median, maximum and total number of recommendations per model ...	23
Table 3 – Average and median rating values per model	24
Table 4 – Metric and quality results for each reranked model.	26
Table 5 - Average and median rating values per reranked model.....	27
Table 6 - Hit rate and beyond-accuracy quality results per evaluated model.....	28

LIST OF ABBREVIATIONS AND ACRONYMS

HDI	Human Development Index
RMSE	Root-mean-square error
MAE	Mean Absolute error
ILS	Intra-list similarity
PM	Primitive System
PMI	Point-wise mutual information
CB	Content-Based
KNN	K-nearest neighbors
KNN_UN	User-based K-nearest neighbors
KNN_IB	Item-based K-nearest neighbors
ALS	Alternating Least Squares

1. INTRODUCTION

Since the first computer's appearance, their ability to perform recommendations based on probabilities was recognized, with one of the first known recommendation systems being the computer librarian Grundy (Rich, 1979). Nowadays, recommender systems are widely adopted and interact with people daily, influencing and advising them. Identifying the best algorithms for a given system has proven to be challenging. Properly evaluating the results is an even more significant challenge due to the existing disagreement between researchers on the best metrics and attributes to use.

There are two main approaches to evaluating recommender systems: online validation, through A/B tests to compare different systems on online users, and offline validation, testing on the user's historical data under the assumption that the offline data is an appropriate proxy to the natural behavior of the users in the future. Online validation is the most reliable approach and can produce results closer to the system's actual performance without requiring many assumptions about the user's behaviors. Unfortunately, it is costly and time-consuming and requires a large user base. A good and reliable offline evaluation can mitigate this by indicating which models are more likely to perform well online (Gruson et al., 2019).

Evaluating recommender systems offline is inherently complex and can be considered as much art as science. Algorithms designed for datasets with more users than items may be entirely inappropriate in a domain with many more items than users (McLaughlin & Herlocker, 2004). In the past, extensive research has been done regarding accuracy, precision, and error-based metrics to validate recommender systems. These metrics are still the most used today by researchers and developers. From 2006 to 2009, Netflix promoted a competition focused on improving the accuracy of its recommendation systems using the root-mean-square error (RMSE) (Bennett & Lanning, 2007). Since then, RMSE and the mean absolute error (MAE) have been the go-to metrics to validate recommender systems. Although, Netflix admitted to not using those metrics anymore as the systems were not producing the desired results.

Recommender systems must be helpful to the users, providing them with valuable recommendations. An increasing number of researchers worry that predictive accuracy might not be enough to provide a valuable experience to the user. Accuracy metrics lack in considering essential features of a recommender system. Examples are whether an item is relevant, whether the system provides novel (Oh et al., 2011), diverse (Vargas & Castells, 2014), and serendipitous recommendations. This occurs due to accuracy only being able to analyze a single item and not a complete list of items. McNee et al. (2006) believe that a list of recommendations should be judged not as a collection of individual items but as a single entity, which accuracy cannot cope with.

To tackle this weakness, several "beyond-accuracy" metrics were created to analyze a recommender system qualitatively, believing that a system must be accurate, helpful, and interesting (Kaminskas & Bridge, 2017). A system might achieve high accuracy but only recommend easy-to-predict or popular items, which may not benefit the user (Herlocker et al., 2004).

In this research, we examine previous works addressing distinct concepts proposed for recommendation system validation, focusing on metrics other than predictive accuracy, also called “Beyond-Accuracy” Metrics. The concepts investigated are *coverage*, *diversity*, *novelty*, *serendipity*, *unexpectedness*, *utility*, and *fairness*. We discuss the various definitions and review each metric. Furthermore, we believe a metric should enable the compression of a system's beyond-accuracy performance into a single value, much like countries are compared using the Human Development Index (HDI)¹. This will enable a more straightforward and concise comparison between the performances of each system and allow the optimization of a system to achieve higher beyond-accuracy performance. Thus, our work objectives are:

- 1) To present a detailed survey and review of the most prominent beyond-accuracy metrics, conceptually analyzing their strengths and weaknesses.
- 2) To conduct a set of offline experiments providing insights from the metrics' behaviors in item-based and user-based algorithms, as well as the relationships between those metrics.
- 3) To propose a new metric to validate the beyond-accuracy performance of a system.
- 4) To propose an optimization framework that utilizes the created validation metric to reorganize the recommended items and increase the system's performance.
- 5) Finally, compare the beyond-accuracy performance of each model with their respective hit rate to further understand the relationship between accuracy and non-accuracy systems.

The remainder of this document is structured as follows: In section 2 we will review the beyond-accuracy metrics available in the recommender systems literature, referring to their concepts, strengths, and weaknesses. In Section 3 we formulate the metrics for evaluating the beyond-accuracy quality of a system and describe the reranking optimization process using those metrics. Section 4 describes the dataset, algorithms and performance metrics used in the experimental setup. On section 5 we present and discuss the results. And on section 6 a conclusion is made, exposing the limitations of this work as well as suggesting future work.

¹ <https://hdr.undp.org/data-center/human-development-index#/indicies/HDI>

2. BEYOND ACCURACY METRICS

2.1. COVERAGE

Coverage is a system-level metric that measures the proportion of items the system can recommend or users the system can perform recommendations. By definition, *accuracy* and *coverage* can collide, as enforcing a higher precision threshold on the recommendations might improve accuracy at the expense of *coverage*. Herlocker et al.(2004) believed in the importance of *coverage*, theorizing that a system with lower *coverage* will provide limited choices to the user, thereby being less valuable. Adomavicius and Kwon stand that high *coverage* can benefit not only the users, as an increased item catalog can lead to higher satisfaction (Adomavicius & Kwon, 2012) but also benefit the business owners. This would allow them to increase product sales, especially from long-tail items (Armstrong, 2008). High *coverage* can create in the user's mind a perception that the system was designed with care and detail. By suggesting more products, the users will perceive a higher sensation of quality in the system (Ge et al., 2010).

There are two approaches to measuring recommendation *coverage*, one focused on the users and the other on the items. The "*user coverage*" is the proportion of users that can receive recommendations from the system (Shani & Gunawardana, 2011). It might happen that the system is not confident enough in the predictions calculated for a particular user or that the user is recent in the system and does not yet have a purchase history. In those situations, it can be difficult for the recommendation system to suggest items for that user, possibly making the system ignore the user entirely and not recommend anything. The "*item coverage*" is the one we will focus on in this work because it is the most common formulation in the recommender system literature. It measures the proportion of items that are or can be recommended.

Shani and Gunawardana (2011) believe that recommendation systems should be evaluated considering a trade-off between accuracy and coverage, choosing systems with a higher *coverage* ratio. Herlocker et al. (2004) defines three types of *coverage* – *prediction coverage*, *catalog coverage*, and *interest coverage*.

2.1.1. Prediction Coverage

Prediction Coverage corresponds to the number of items the system can recommend (I_r) compared with the total number of items in the catalog (I). As Herlocker et al. (2004) suggests, prediction coverage answers the question: "What percentage of items can this recommender form predictions for?".

A simple way to measure *prediction coverage* is given by:

$$Prediction\ Coverage = \frac{|I_r|}{|I|} \quad (1)$$

There are various methods to obtain I_r , depending on the technology used to collect the data and build the system. The technique used to filter the item set is highly dependent on the goal and domain of the system. For example, if we consider a recommender system that relies on item ratings to make its predictions, the value of I_r will be the number of items with enough ratings for the algorithm to form predictions. All items that do not meet that rating threshold are not recommended.

2.1.2. Catalog Coverage

Contrary to the *prediction coverage*, the *catalog coverage* measures the percentage of items that effectively are recommended, answering the question: "What percentage of available items does this recommender ever recommend to users?" (Herlocker et al., 2004). This metric is not as popular as *prediction coverage*. However, it is advantageous for measuring *coverage* in a system that produces Top-N lists (Ge et al., 2010). It is essential to consider that *catalog coverage* represents a specific period in time, meaning that recommendations given in one recommendation session might produce a different *coverage* than in the previous or following sessions. This time sensitivity also helps distinguish *catalog coverage* from *prediction coverage*, the latter being less time-sensitive, depending on the technique applied to filter the items.

Catalog coverage can be calculated through the union of all items in the Top-N lists recommended at a given time, compared with the entirety of items in the item catalog. Kaminskis and Bridge (2017) proposed the following formula for measuring *catalog coverage*:

$$Catalog\ Coverage = \frac{|\cup_{u \in U} R_u|}{|I|} \quad (2)$$

Where U represents all users in the system, R_u all recommendations made to user u and I all items in the catalog.

In most cases, it is not adequate to increase the *coverage* if it would result in recommendations that are not interesting to the user. Herlocker et al. (2004) proposed calculating coverage by considering user interests.

2.1.3. Interest Coverage

The main difference between *interest coverage* and the *coverage* measures mentioned earlier is regarding the item catalog. *Interest coverage* does not consider the entirety of the item set, only the items a user is or might be interested in. In other words, this metric considers the item's usefulness for a user. However, usefulness is a concept with a comprehensive meaning and susceptible to various interpretations, although many researchers believe *accuracy* to be a good proxy for the usefulness of an item. Ge et al. (2010) believe that item *novelty* is also a good indicator of usefulness, assuming that a novel item has the potential to be attractive to a user unaware of the item's existence. *Utility* and

serendipity can also be indicative of an item's usefulness. All these metrics will be discussed forward in this work. Nonetheless, the best and most precise method to measure the usefulness of an item is by directly asking and studying the response of users in an online setting.

Ge et al. (2010) proposed an *interest coverage* formula, which they called *weighted catalog coverage*. This formula is an adaptation from equation (2), where the intersection between the set of recommendations and the collection of useful items (B) is considered:

$$\text{Interest Coverage} = \frac{|\bigcup_{u \in U} R_u \cap B|}{|B|} \quad (3)$$

Although this metric will, by definition, have a lower value than other coverage metrics, the loss in *coverage* is balanced by the worthiness of the items recommended. This metric can validate systems that better correspond to the user's needs since the system will not waste resources suggesting items the user is not interested in (Herlocker et al., 2004).

Common to all these *coverage* metrics is the aggregation of all items into a single value regardless of the number of recommendations an item gets. An item recommended to all users or an item recommended to a single user will have the same weight on coverage. This means that if a system is prone to popularity bias, measuring *coverage* will not be enough to notice this fault in the system. Coverage should be measured alongside other metrics to obtain the most trustworthy validation of the system.

2.2. DIVERSITY

The essence of a recommender system should be linked to a feeling of discovery, which, driven by *diversity*, helps users find items they would not have found by themselves. A system with high accuracy might not be able to generate this sense of discovery because it is more prone to suggest obvious items (Vargas & Castells, 2011).

The notion of *diversity* is highly consensual among researchers, and it is defined as the amplitude of variation within the characteristics of the items in a recommendation list. If a recommendation list contains all *Harry Potter* books, the diversity of that list is low due to the high similarity between those books. Hence Ricci et al. (2011) defined diversity as being the opposite of similarity. However, the similarity between items can be wildly subjective. For example, recommending a *Harry Potter* book and a *Lord of the Rings* book can be considered a diverse recommendation, even though both books have the same category (i.e., fantasy) and be considered similar by unfamiliar users.

Some recommendation systems are built using algorithms based on similarity, for instance, recommending items according to trends and item features, i.e., content or item-based algorithms. Consequently, Zhou et al. (2010) believe a popularity bias is created where only the most popular items are recommended, and the niche products, which can arguably be more interesting for the user, are overlooked. Furthermore, as Herlocker et al. (2004) identified, a paradox is created where the most

accurate systems are based on similarity. Still, the most valuable recommendations are of diverse and niche products that the users would hardly find alone. A close parallel can be analyzed in how often the most helpful advice does not come from close friends (similar items) but from people with whom we have limited connections ("weak ties") (diverse items) that open our horizons for possibilities outside our everyday experience (Granovetter, 1973).

The concept of *diversity* was first introduced in Information Retrieval literature, namely regarding the diversification of results in search engines and similarity in user queries (Carbonell & Goldstein, 1998). When a user searches for something, it usually uses small and not very specific queries, which can originate ambiguous results due to the different interpretations the system can make. If a user utilizes words with various meanings, the system will not know which topic the user is interested in. Thus, to help decrease ambiguity, *diversity* was introduced in search engines allowing the system to retrieve various documents that encompass the highest number of possible interpretations for that query, increasing the probability of satisfying the user's needs.

Ziegler et al. (2005) proposed a formula for measuring *diversity* in recommendation systems. They defined a *diversity* metric called *intra-list similarity (ILS)*, representing the aggregate sum of the similarity between items in a recommendation list. Their formula is presented as shown below, with R_u representing a user's recommendation list:

$$ILS(R_u) = \sum_{i \in R_u} \sum_{j \in R_u \setminus \{i\}} d(i, j) \quad (4)$$

An older version of this formula was proposed by Smyth and Mclave (2001), the difference being instead of considering the aggregated sum, they would consider the mean of the pairwise distances within a recommendation list:

$$Diversity(R_u) = \frac{\sum_{i \in R} \sum_{j \in \{i\}} d(i, j)}{|R_u|(|R_u| - 1)} \quad (5)$$

Where $d(i, j)$ represents the distance function between items i and j . The lower the value of this metric, the higher the similarity between the products.

The calculation of the distance function $d(i, j)$ differs from author to author and depends on the type of feature representing the item, resulting in numerous approaches to calculating *diversity*. If the feature is a descriptor of the item - e.g., the name, category, or description - the distance function can be computed using a taxonomy-based metric (Ziegler et al., 2005) or the complement of Jaccard Similarity (Vargas et al., 2011). Furthermore, if the items are being represented numerically, e.g., using ratings, Ribeiro et al. (2012) and Zhang et al. (2012) prefer the Cosine similarity as the distance function, whereas Kelly and Bridge (2006) favor the Hamming Distance and Vargas and Castells (2011) the Pearson Correlation.

Vargas and Castells (2011) criticized the current metrics for not considering the item's relevance or position in the recommendation list, believing items that appear first should be the most relevant ones. They created a new metric that considers both these aspects, assigning a discount depending on the position of each item in the list, with $disc(k)$ and $disc(k|l)$ representing the relative rank discount of an item at position l knowing position k has been taken:

$$Diversity(R|u) = \sum_{\substack{ik \in R \\ il \in R \\ l \neq k}} disc(k)disc(l|k)d(i_k, i_l), \forall i_k, \neq i_l \quad (6)$$

However, Vargas et al. (2014) believe metrics based on item-to-item distance like the ones mentioned above might not be able to create a sense of variety in the user perception, despite influencing the inherent diversity of a recommendation. To achieve this, the item genres should be used to calculate *diversity*, assuming if the recommendation span across various genres (e.g., action, comedy, drama), the user will perceive the recommendations as more diverse. A recommendation list should encompass the user's genres of interest and avoid the existence of genre redundancies. Namely, if a list contains three movies that span six different genres (e.g., Western, Space), but all are Comedy movies, the list will have some diversity but is redundant.

All metrics mentioned are measured on recommendation lists generated by the algorithms and are evaluated in a post-modeling stage. The advantage of this method is reproducibility on any system independently of the algorithm chosen to create the lists, explicitly controlling the level of diversification. However, a strand of research refers to creating recommendation algorithms that automatically incorporate *diversity* during their modeling process (Kaminskas & Bridge, 2017).

2.3. NOVELTY

A recommender system aims to show the user something new and exciting (Vargas & Castells, 2011). Thus, apparent recommendations or recommendations of trendy items, although they have high levels of accuracy, will be of little value to the user, who might already know about those items. Herlocker et al. (2004) found two problems with recommending obvious items. Firstly, suggesting popular items is counter-productive since the users that want the item would probably already have bought it, and the ones that do not like the product already decided not to buy it and will ignore the recommendation. Secondly, the business owners know their most popular items and how to promote them. A recommendation system does not have to suggest such items to a user. In a movie recommendation system, the user might consider it more valuable to receive recommendations for new or less-known films aligned with his/her preferences instead of an Oscar-nominated movie. However, the system cannot only rely on new and obscure recommendations. Allowing suggestions of items familiar to the user would increase their confidence in the system, as they believe the system can provide recommendations suited to their preferences. This creates a complex task of conciliating *familiarity* with *novelty* and *relevance*. Celma (2008) believed that if a user is in a laid-back state of mind, the system should suggest more comfortable and familiar items. However, if they are curious, the

recommendations should be novel and allow the user to explore new items. The challenge is the system's ability to recognize the user's state. Random or popular recommendations are beneficial when the system does not have enough information about a user.

Similarly to the other validation metrics, the *novelty* concept originated from information retrieval literature, with Baeza-Yates and Ribeiro-Neto (1999) defining *novelty* in a set of documents as the proportion of unknown documents to the user. Later, Zhang et al. (2019) considered *novelty* as the opposite of redundancy, with a document being redundant if containing information from previously seen documents. They treated *novelty* as a boolean metric, with a document either being novel or redundant.

In recommender systems literature, *novelty* primarily focuses on two aspects, the item is unknown to the user, and the item is different from what the user has seen before (Kaminskas & Bridge, 2017). According to Zhang (2013), a novel item has three characteristics: it needs to be unknown to the user, must be relevant, and must be dissimilar from all other items in the user profile. In 2015 Kapoor et al. (2015) extended the definition of *novelty* by also considering novel the items a user knows but has long forgotten or stopped interacting with. Recommending a song that a user has not listened to in a long time, but was once a highly reproduced song, will create a positive emotional response associated with that recommendation. However, the definition proposed by Kapoor et al. (2015) can only be applied to recommender systems of recurrent consumption, like a music or grocery shop recommender.

To better organize the different notions and definitions of *novelty*, Silveira et al. (2019) proposed three levels of *novelty*:

1. Life Level Novelty: when an item is an absolute novelty for the user, i.e., the user never knew about the item's existence.
2. System Level Novelty: the item is unknown to a user just considering the user profile and history of consumption. The user might know the item, yet there is no way for the system to confirm it due to the user having never interacted with it.
3. Recommendation List Level Novelty: this happens when a recommendation list is devoid of redundant items.

Celma (2008) proposed a Life Level Novelty formula, where *novelty* is calculated through the ratio of the unknown items in a Top-N list:

$$Novelty(u) = \frac{\sum_{i \in R} (1 - Knows(u, i))}{|R|} \quad (7)$$

Where R represents the Top-N list and $Knows(u, i)$ is a binary function returning 1 if the user u knows item i , or 0 otherwise. It is not a trivial task to calculate $Knows(u, i)$ in an offline validation context. This is because the system cannot easily understand whether or not the user knows an item. We could consider that if an item is not present on a user profile or the user did not review or rate it, it must mean the user is unaware of such an item (system-level novelty). Still, the user may be familiar with the item but did not interact with it in the system or did not feel like rating it. The only way to precisely understand whether a user is familiar with an item is by asking the user directly, for instance, by

creating a field on the item page asking, "Did you know about this product before?". However, interacting with the user through the system might create a problem of cognitive load and harm the user experience (Kaminskas & Bridge, 2017). A recommender system has a high user interaction where all features and gimmicks must be carefully considered. Sometimes elements that help the developers and business owners do not always provide a good experience to the user.

Therefore, the most popular and accepted method to measure *novelty* is through the popularity of an item, assuming that the more popular an item is, the more significant the probability of it being well-known. The method to measure popularity is highly dependent on the domain. In some instances, it might be the number of views and, in others, the number of times the item was bought. The number of ratings is the most used method when measuring popularity.

Lu et al. (2012) understood that regarding *novelty*, popular items have a higher rate of true positives than the items in the long-tail. If a user does not rate or consume a popular item, it probably means they are familiar with the product but not interested in it. Contrarily, if a user does not rate a long-tail item, there is a high probability of that item being genuinely unknown to the user.

However, some researchers are against the use of popularity. Celma (2008) believes that if a user is familiar with a rare item (long-tail), the probability of being knowledgeable about other rare items is increased, making popularity a poor method to know if a user is familiar with an item or not.

The user interacting with an item belonging to the long-tail is an infrequent event that can be measured using the *self-information* of the item, a concept introduced by Zhou et al.(2010) that gives more importance to less known items. An item's *self-information* (I) represents the chance of a random user observing an item (i). Self-information can be characterized by the following equation, where $p(i)$ represents the popularity of the item i :

$$I_i = \log_2(p(i)) \quad (8)$$

If we average the inverse of the self-information of all items in a recommendation list, we arrive at a formula of *novelty* (Kaminskas & Bridge, 2017):

$$Novelty(R) = \frac{1}{|R|} \sum_{i \in R} -\log_2(p(i)) \quad (9)$$

Closely associated with the concept of *novelty* is *serendipity*.

2.4. SERENDIPITY

Serendipity is concerned not only with an item's *novelty* but also with how surprising the item is. As Herlocker (2004) stated, all serendipitous recommendations are novel, yet not all novel recommendations are serendipitous. An example can be given using a music recommender to understand the difference better. If the system recommends a song from the user's favorite band that he has not yet listened to, that is a novel recommendation. Still, hardly a surprising one as, eventually, the user would come to know that song. However, if the system recommends a piece from an unknown band to the user and he likes the song, the recommendation is novel and serendipitous. Hereupon, laquinta (2008) realized that the lower the probability of an item being known by a user, the higher the chance of that item becoming a serendipitous recommendation.

It is important to note that a serendipitous recommendation must be of an item relevant to the user and considered a pleasant surprise. Otherwise, the system would recommend items irrelevant to the user just for the sake of them being surprising, hurting the user experience and confidence in the system. Ge et al. (2010) ultimately defined serendipity as pleasant surprises with two essential characteristics: the item must be unknown and unexpected by the user, and the item has to be attractive to the user.

Calculating *serendipity* is a challenging process, given the intricacy of its definition. In an offline context, it is complicated for a system to accurately measure the "level of surprise" or the *utility* an item has for a user. Considering that the core components of *serendipity* are surprise and utility, we must formulate *unexpectedness* and *utility before* formulating *serendipity*.

2.4.1. Unexpectedness

Unexpectedness is highly connected with surprise (Ge et al., 2010) and is defined as a divergence from expected recommendations. A user has certain expectations about what items are recommended to him, and all suggestions deviating from those expectations will be regarded as unexpected (Adamopoulos & Tuzhilin, 2014). Adamopoulos (2014) defined *unexpectedness* as the distance between an item and the set of expected items. However, if an item reaches a certain distance from the set, it might become irrelevant to the user due to its significant dissimilarity from the expected items. The set of expected items is commonly composed of the items in the user's profile and purchase history or frequently bought items. However, items with high similarity to those in the group of expected items can also be considered expected items. Adamopoulos presented the following formula, representing the distance between item i and the set of expected items E_u of user u :

$$Unexpectedness_{u,i} = d(i; E_u) \quad (10)$$

Ge et al. (2010) suggest creating two systems. The first system, which he called the "primitive system" (*PM*), is built to target the highest accuracy possible, producing the most expected recommendations. The second system is the system being validated. In this case, the unexpectedness will be the number

of items in the system that are missing in the primitive system. The *unexpectedness* of the system can be given by:

$$Unexpectedness(R) = R - PM \quad (11)$$

Adamopoulos (2014) transformed this formula into the ratio of items in a recommendation list (R_u) that don't belong to the set of expected items (E_u):

$$Unexpectedness(R_u) = \frac{R_u - E_u}{|R_u|} \quad (12)$$

However, a disadvantage exists in using a primitive system as the set of expected items. The *unexpectedness* might vary depending on the algorithm used to build the primitive system. Furthermore, due to the focus of the primitive system on the accuracy, most recommendations are of popular items, creating the assumption that unpopular items will be unexpected, which is not always true (Silveira et al., 2019).

Kaminskas and Bridge (2014) measure *unexpectedness* through the probability of an item being rated. To do so, they calculated the *point-wise mutual information* (PMI) of two items being rated by the same user:

$$PMI(i, j) = \frac{\log_2 \frac{p(i, j)}{p(i)p(j)}}{-\log_2 p(i, j)} \quad (13)$$

Where $p(i)$ and $p(j)$ represent the individual probabilities of items i and j being rated, and $p(i, j)$ the probability of both items being rated by the same user. This metric ranges from -1 to 1, with -1 meaning the items are never rated together and 1 total co-occurrence between the items. If a user has item j in their profile and receives a recommendation of item i , higher values of PMI between the two items result in a lower level of surprise because that would mean those items are commonly consumed together.

Kaminskas and Bridge (2014) then defined *unexpectedness* as the average PMI between the recommended items and the items in the user profile (P_u):

$$Unexpectedness(R_u) = \frac{\sum_{i \in R_u} \sum_{j \in P_u} PMI(i, j)}{|P_u|} \quad (14)$$

Serendipity and *unexpectedness* are often confused due to their similar definition. However, *serendipity* measures novel, useful and surprising items, whereas *unexpectedness* is only concerned with surprise.

2.4.2. Utility

Shani and Gunawardana (2011) defined *utility* as the value a particular recommendation provides to the system, user, or business owner. The more valuable a recommendation, the more useful it is. For a business owner, the value of a recommendation can be directly linked with the monetary gain that the recommendation provides. However, for the user, the most common way to measure *utility* is using the ratings of an item, assuming that items with higher ratings have more usefulness to the user than items with low ratings.

Adamopoulos et al. (2014) defined a utility function in terms of the perceived quality and unexpectedness of an item:

$$Utility_{u,i} = q_u * r_{u,i} - \lambda_u * |\delta_{u,i} - \delta_u^*| \quad (15)$$

Where they assume that a user values the quality of an item by a constant q_u and the quality of the item for the user is $r_{u,i}$. In their work, they measure $r_{u,i}$ by using the predicted ratings given by their model. λ_u represents the assumed tolerance a user has for redundant recommendations. There is also a presumption of the ideal level of unexpectedness, given by δ_u^* , which is compared with the actual unexpectedness $\delta_{u,i}$ of the item. One way to calculate the ideal level of unexpectedness for a user is by averaging the distance of the rated items to the ones on the set of expected items.

Furthermore, an additional method to measure utility can be performed by tracking the item after the user consumes it. If a product is returned after purchase or a movie is not watched until the end, those might indicate a lack of utility to the user.

2.4.3. Formulating Serendipity

The difference between unexpectedness and serendipity is the necessity for serendipity to consider utility. Much like *unexpectedness*, a standard method to measure *serendipity* is by building a primitive recommender under the premise that the primitive system will recommend items easy to predict, and the serendipitous system will recommend items hard to predict or unanticipated items.

Get et al.(2010), and Adamopoulos et al.(2014) measure serendipity by a similar formula, only having terminology discrepancies (Ge et al. use PM_u and Adamopoulos et al. E_u), with both procedures being an adaption from equation (12):

$$Serendipity(R_u) = \sum_u \frac{|(R_u - PM_u) \cap Utility_u|}{|R_u|} \quad (16)$$

$Utility_u$ can be obtained using the utility function in equation (15) or, as Ge et al. (2010) proposed, using a binary process returning 0 if not useful. This concept deviates from the *serendipity* notion introduced by Herlocker (Herlocker et al., 2004), where he believes a serendipitous item has to be novel.

Zhang et al.(2012) measured serendipity through a clustering method. They built a music recommendation system where users were clustered according to their taste in musical artists. This enabled them to calculate the distances between an artist vector and the artists in a user profile. If an artist is outside a user's cluster of liked artists, that might mean that artist is a serendipitous recommendation. However, this concept is more related to *unexpectedness* rather than *serendipity*.

2.5. FAIRNESS

Our society is replete with algorithmic systems that help people and organizations make decisions and form opinions. However, several systems encapsulate various biases, some coming from the developers and some that the system creates according to given data (Chouldechova & Roth, 2020). Due to those biases, some minority groups can be affected, namely women (vs. men) and seniors (vs. youth), receiving weak or biased recommendations that make those minorities lose their trust in the system. Because fairness does not increase the monetary value or help create useful suggestions, sometimes even having the opposite effect of usefulness, it is not given as much attention in the recommender systems literature as other non-accuracy metrics (Bobadilla et al., 2020). A study conducted by Ferraro et al. (2021) concluded that gender fairness is one of the female artists' most significant concerns in a music recommender system since most systems do not give them the same exposure as male artists. The study analyzed 300.000 users and their music recommendations and concluded that songs from female artists only represented 25% of the songs recommended to users. The system made recommendations based on previously listened songs, amplifying the existing biases and reducing diversity.

Pitoura et al. (2021) define fairness as the absence of discrimination in a system, i.e., a system devoid of bias that does not favor any group based on its inherent characteristics (Mehrabi et al., 2019). Unfairness is usually influenced by sensitive attributes, which generally include age, race, religion, gender, sexual orientation, etc. These sensitive attributes, also called protected attributes, describe the users and can be allocated to certain representative minorities. Hiding these attributes from the model might help prevent unfairness. However, if the other features in the model are somewhat

correlated with the protected attributes, the model will discover that connection and possibly maintain the bias (Yao & Huang, 2017), keeping being unfair.

A well-known way of reducing unfairness is through *demographic parity* (Zemel & Swersky, 2013), where classes and attributes have equal proportions in the model. However, in recommendation systems, *demographic parity* is not always desired. For a recommendation system to be effective, it needs to study user preferences and behaviors. Behaviors that are most likely influenced by the gender, race, or age of the user (Chausson, 2010), forcing parity in these attributes would probably hurt the quality and usefulness of the recommendations.

Yao et al. (2017) conceptualized three categories of unfairness:

- *Value unfairness* – Occurs when one class of users consistently receives predictions contrary to their tastes just because they belong to that class. For example, male users receive recommendations for action movies even if they are not interested in them, while female users do not receive recommendations for action movies even if they are interested in such genres.
- *Absolute unfairness* – Unlike value fairness, it does not consider the error's direction, just the error's distance. If recommendations for females and males have errors in the same proportion, there is no absolute unfairness. However, if female ratings are off by 1 point and males by 2 points, absolute unfairness is prominent.
- *Underestimation/overestimation unfairness* happens when a type of user is missing recommendations (underestimation) or, conversely, when a user is overwhelmed with suggestions (overestimation). Varying amounts of either can cost one type of user more than the other. If a user needs to spend considerable time analyzing each recommendation, being overwhelmed with recommendations could hurt the experience.

Pitoura et al. (2021) suggest using fairness-aware programming to control fairness. The programmers and developers of a system should, a priori, set their fairness goals and expectations, and the system should alert them in case of violation of those expectations. Additionally, they propose three methods to reach fairness. The first method happens in a pre-processing phase, transforming the data and ensuring no biases leak into the model. Secondly, creating new models or modifying existing ones so they output unbiased recommendations, called in-processing methods, where fairness-aware programming is included. Lastly, there are post-processing methods that aim to rearrange the final outputs of a model. However, this method interferes with the model's accuracy, considering that the algorithm was built to achieve the best results, and we would be changing those outputs.

Unfairness is a highly mutable concept with different levels of importance according to the system's domain. When building a system, a developer should weigh the importance of each attribute and the impact that unfairness could have on the user experience. Fairness is imperative in a system that recommends candidates for a university program, as the system should not discriminate on sensitive attributes like gender and race. However, in a book recommendation system forcing gender fairness might produce negative results and hurt the user's confidence in the system.

3. BEYOND ACCURACY QUALITY

All metrics in section 2 are considered beyond-accuracy metrics in the recommendation system literature because, as the name implies, they evaluate a system without considering the algorithmic precision or error.

We propose using a summary measure that encapsulates the beyond-accuracy metrics and allows the validation of a system using a single value representative of its beyond-accuracy performance. This metric can be denominated as *beyond-accuracy quality* and is formulated in equation (17) in the section below.

In the literature, there are several methods to optimize a recommender system for accuracy to recommend the most accurate items possible. By having a *beyond-accuracy quality* measure, we can rerank the recommendations of a system to retrieve the items that have the most significant influence on this quality metric. However, to optimize and rerank the recommendations of a system, the *beyond-accuracy quality* measure must be adapted into two variations due to the reranking technique only working at an item level. The first variation will measure the *beyond-accuracy quality* of a single item, and the other will validate the overall *beyond-accuracy quality* of a system. An item does not have the same properties as a system, making specific metrics impossible to be calculated at an item level.

In the forward sections, we will formulate the two variations of the *beyond-accuracy quality*, referred to as *quality* for simplicity. We will also provide a framework to optimize a system for *quality* by using a reranking approach (section 3.3).

3.1. QUALITY MEASURES – ITEM QUALITY

We propose the quality of an item i to be represented by the geometric mean of the beyond-accuracy elements of that item:

$$item_quality(i) = \sqrt[4]{i_diversity(i) * i_novelty(i) * i_unexpectedness(i) * i_rating(i)} \quad (17)$$

The *diversity*, *novelty*, and *unexpectedness* are calculated for each item in a recommendation set, as well as the rating of that item. The $i_rating(i)$ is the predicted rating calculated by a model. Specific algorithms cannot predict ratings. In such cases, the $i_rating(i)$ is the average rating given by the users for that item. The assumption is that items with higher ratings will be considered to have more quality.

Metrics like *coverage* and *fairness* are system-level metrics, which means they can only operate at a level where all recommendations are already made. One singular item does not have a coverage value; only the aggregation of all items can allow coverage measurement. Therefore, both those metrics are excluded in the computation of a single item's *quality*.

Although each element can be weighted differently in different domains when measuring *quality*, all should be considered, and none should be zero. According to this definition, an item will have no *quality* when one element is zero. If the item has a *diversity* value of 0, meaning it is highly similar to other items in the user profile, it cannot be considered an item of quality.

A final consideration about *item_quality* is that each user's quality is unique, and the same item might have different quality values depending on the user. To measure a system's overall quality, we propose the calculation of *system_quality*.

3.2. QUALITY MEASURES – SYSTEM QUALITY

We propose the following metric to evaluate the *quality* of a system:

$$system_quality(R) = \sqrt[5]{s_diversity(R) * s_novelty(R) * s_unexpectedness(R) * s_fairness(R) * s_coverage(R)} \quad (18)$$

System_quality is the metric that most closely follows the concept of the *beyond-accuracy quality* of a system. The result of this metric will enable the ranking and comparison of the algorithms according to their performance in beyond-accuracy metrics, as higher values of *diversity*, *novelty*, *unexpectedness*, *fairness*, and *coverage* will correspond to a system with higher *quality*.

The formulas of each metric included in these equations are illustrated in section 4.3.

3.3. RERANKING APPROACH

We propose a methodology that aims to generate recommendations with the highest *beyond-accuracy quality* by reranking the outputs of a given algorithm. This approach allows the usage of any recommendation algorithm, as this method only takes place in a post-modeling stage and only alters the final outputs of a model.

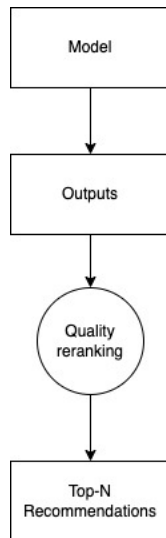


Figure 1- Steps for the reranking technique. After retrieving the model outputs, they are reranked and the top-N items are recommended

As standard practice, a model is optimized to achieve the highest accuracy possible. Therefore, the outputs of a recommendation algorithm would be the Top-N most accurate items, even if they are not the most adequate regarding the beyond-accuracy metrics. Those models select the top-N items with the highest predicted rating for a particular user. However, an item with a high predicted rating does not necessarily mean it is a good recommendation for that user. Another factor that should be considered when ranking an item is the *beyond-accuracy quality*.

We propose using each item's *quality* (equation 17) criterion for rearranging the model's outputs. A model suggests a certain number of candidate items to be recommended to a user, and the *quality* of those items is calculated. Depending on the values of *quality*, each item is rearranged, and the Top-N best items are retrieved. Using an illustrative example of a recommendation system aiming to recommend ten books to each user. If the algorithm is built to output the top 20 most accurate books, we could rearrange them according to their *quality* and retrieve the ten books with the highest value. Those ten books would be the ones recommended to the user. This method enables the recommendation of the best ten books, among the twenty most accurate, with the highest *quality*.

Kaminskas and Bridge (2017) also optimized the recommendations of a system by using a reranking technique. Their approach is reranking an item by greedily maximizing a metric using a function proposed by Smyth and McClave (Smyth & McClave, 2001). However, their approach only allows the maximization of one beyond-accuracy metric, e.g., optimizing the system for *diversity*. Our approach aims to optimize and validate a system across several beyond-accuracy metrics, allowing for an assessment of the overall *quality* of that system. Kaminskas and Bridge's (Kaminskas & Bridge, 2017) technique should be applied when a system developer intends to improve the recommendations according to one specific metric.

4. EXPERIMENTAL SETUP

4.1. DATASETS

The Goodbooks-10K² dataset, retrieved from Amazon's Goodreads platform, was utilized to study the relationships between the metrics and test the reranking approach.

The dataset contains approximately ten thousand books and fifty-four thousand users that gave close to six million ratings, measured from 1 to 5. All users made at least two ratings, with the median number of ratings per user being eight. There is also information regarding each user's books on their "to_read" list.

Each book is represented by eleven features, including genre, number of pages, author, and book description. An additional feature representative of the author's gender was created to enable the measurement of *fairness*, resulting in 2678 male and 1882 female authors.

Furthermore, another feature was added, compressing the number of pages into four groups:

- Group A - Less than 252 pages
- Group B - Between 252 and 336
- Group C - Between 336 and 422
- Group D – More than 422

Table 1 below lists the features available in each dataset table.

Books	Ratings	To read
book_id	user_id	user_id
title	book_id	book_id
author	rating	
author_gender		
pages		
pages_group		
description		
genres		
release_year		
average_rating		
ratings_count		

Table 1 – GoodReads dataset features

² <https://github.com/zygmuntz/goodbooks-10k>

4.2. ALGORITHMS

Four algorithms commonly found in the recommendation systems literature were used to perform this study. A content-based algorithm (CB) implemented using a vectorizer and cosine similarity (Salton & Buckley, 1987) to measure the distance between the vectors. Each book is represented by a vector containing that book's characteristics (name, description, author, etc.), and the CB algorithm evaluates the similarity between the vectors of each book. Two *k-nearest-neighbor*(KNN) algorithms – a user-based (KNN_UB) and an item-based (KNN_IB) (Desrosiers & Karypis, 2011). And an iterative algorithm - *alternating least squares* (ALS) (Y. Zhou et al., 2008). A grid-search technique was used on the ALS algorithm, achieving an RMSE of 0.811. No further accuracy optimization was performed on any algorithm. The purpose of this work is not to achieve the highest possible accuracy in recommendations but rather to study the behavior of the metrics and compare the system's *quality*.

The KNN_IB and CB are two item-based algorithms, so they could only make suggestions specific to each book, resulting in each book instead of each user having a set of ten recommendations. To solve this issue, we looked at all books a user has read and retrieved the top 10 recommendations for each of those books given by the algorithms. Then, we count the number of times a book appears among all those sets, and the ten books with the highest frequencies are the ten books recommended to the user.

Furthermore, all algorithms were reranked according to the methodology described in section 3.3, forward identified as *CB_quality*, *KNN_IB_quality*, *KNN_UB_quality*, and *ALS_quality*.

4.3. PERFORMANCE METRICS

In section 2, several metrics were introduced alongside their advantages and use cases. In this section, we will specify the metrics used in our experiment.

Because *utility* and *interest* inherently require the developer to make significant assumptions about the user, we opted not to use metrics that need those elements. In an online validation setting, or in cases where a user specifies the usefulness of a particular item, metrics using *utility* or *interest* should be analyzed, as the assumptions made are minimal. However, in an offline validation setting, the evaluation of a system should be impartial and not allow many assumptions that can influence the results. Therefore, metrics like *serendipity* are not considered going forward.

As mentioned above, the reranking approach will need to calculate an *item's_quality* (equation (17)), which requires slight variations on the beyond-accuracy metrics for them to make measurements at an item level. Therefore, when relevant, each metric will have two versions, one to measure at an item level and the other at a system level.

On all metrics R_u represents the set of recommendations for user u , and R the total amount of recommendations in the system.

4.3.1. Coverage

In the chosen dataset, all books have many reviews, and all users read and rated several books, making it unnecessary to calculate *prediction coverage*. We opted to compute the *catalog coverage* (equation (2)), as it is also the most appropriate when validating top-N lists.

4.3.2. Diversity

To measure the *diversity* value of an item, the average distance of that item with all other items in a user's recommendation list is calculated:

$$item_Diversity(i) = \frac{\sum_{j \in R_u \setminus \{i\}} d(i, j)}{R_u} \quad (20)$$

The *diversity* metric proposed by Smyth and Mclave (2001) was computed to measure *diversity* across the system:

$$system_Diversity(R) = \frac{\sum_{u \in U} ILS(R_u)}{R} \quad (21)$$

To calculate the similarity between books, each book was represented by a vector containing the following characteristics: author, average rating, the author's gender, page group, and the top 5 genres associated with the book. The Jaccard Similarity was chosen as the function to measure the item's distance.

4.3.3. Fairness

As previously mentioned in Section 2.5, fairness is a concept with a definition highly dependent on the domain and system objective. In this research, we evaluate fairness by comparing the number of recommended books written by a female author with the ones written by a male author. We can compute fairness with the equation below:

$$Fairness(R) = 1 - \left| \frac{\sum_{i \in R} male_i - female_i}{R} \right| \quad (22)$$

Where R represents the total amount of recommendations, $male_i$ defines the list of recommended male-written books and $female_i$ the list of recommended female-written books.

The output of this equation ranges between 0 and 1, with 1 representing complete fairness. In this case, it would mean that the number of recommended books written by male authors is the same as

the number of recommended books written by female authors. Complete unfairness is achieved if no male-written or female-written book is ever recommended.

4.3.4. Novelty

To measure an item's *novelty*, the inverse of that item's self-information is calculated:

$$i_novelty(i) = -\log_2(p(i)) \quad (23)$$

Regarding each user value of *novelty*, two different metrics were computed.

The first metric, inspired by equation (7), calculates the ratio of unknown items in the recommendation list. An assumption is made, and a book is considered unknown if it is not present in the user profile or in the set of "to read" books of the user:

$$Novelty(u)_{binary} = \frac{\sum_{i \in R_u} (1 - Knows(u, i))}{|R_u|} \quad (24)$$

The second metric is based on equation (9) and considers each book's self-information. Where $p(i)$ is obtained from the number of ratings a book has. A book with a higher number of ratings is considered to be more popular.

$$Novelty(u)_{self-info} = \frac{1}{|R_u|} \sum_{i \in R_u} -\log_2(p(i)) \quad (25)$$

In both cases, the overall system *novelty* is given by the average *novelty* of all user recommendations, as presented below:

$$s_novelty_{binary/self-info}(R) = \frac{\sum_{u \in U} Novelty(u)_{binary/self-info}}{|R|} \quad (26)$$

4.3.5. Unexpectedness

The unexpectedness of an item is given by the maximum PMI value that item achieves in a user's recommendation set:

$$i_unexpectedness(i) = \max(PMI(i, j)), j \in R_u \setminus \{i\} \quad (27)$$

To measure the system's overall unexpectedness, we first had to calculate the $PMI(R_u)$ of each user, as shown in equation (13). For each user, the $PMI(u)$ is estimated between all recommended items and all items in that user profile. Because the higher the $PMI(u)$ the lower the surprise, the maximum value of $PMI(u)$ in each user is stored and is representative of that user. Therefore, we are assessing the lowest possible value of surprise each system is achieving.

The PMI value of the system is the mean value of the $PMI(u)$ in all users, and unexpectedness will be the inverse of that:

$$s_unexpectedness(R) = 1 - \frac{\sum_{u \in U} PMI(R_u)}{|R|} \quad (28)$$

4.3.6. Hit rate

The *hit rate*, a measure proposed by Deshpande et al. (2004) and highly similar to the concept of *recall*, is considered an accuracy measure that is a good representative of the *accuracy quality* of a system.

For this research, we will consider a "hit" when a book recommended to a user is also present in that user's "to read" set of books. The *hit rate* will be the number of users with hits compared to the total number of users.

With U as the total amount of users, the hit rate is computed as:

$$Hit\ Rate(R) = \frac{Total\ number\ of\ user\ with\ hits}{U} \quad (29)$$

5. RESULTS AND DISCUSSION

We performed two primary investigations. Firstly, we compared the results of the beyond-accuracy metrics displayed in section 4.3 across the different algorithms described in section 4.2. Then the results from the reranking approach were analyzed to understand if reranking the items would increase the *quality* of the recommendations. Furthermore, a comparison of the overall *quality* of the original and reranked algorithms is performed alongside the *hit rate* values of each algorithm. We will use the hit rate to understand the behavior between *beyond-accuracy quality* and accuracy quality.

All models generated ten recommendations, resulting in 534 130 recommendations per model.

Table 2 is representative of the recommendations distribution on each model and can be interpreted as follows (CB as an example):

- In total, 2259 unique books were recommended
- Each book was recommended to 236 users on average
- 50% of the books were recommended at least 23 times
- The most frequently recommended book was suggested to 12980 users

	Nº of recommended books	Average nº of recommendations	Median nº of recommendations	Maximum recommendation frequency
CB	2259	236	23	12980
KNN_IB	3620	147	12	26027
KNN_UB	8956	59	14	5798
ALS	8020	66	9	13639

Table 2 – Average, median, maximum and total number of recommendations per model

The table below (table 3) represents the rating distributions of the recommended books of each model. It enables the assessment of the popularity bias in a model. It reads as follows (CB as an example):

- On average, a recommended book was rated 130035 times
- 50% of the books were rated at least 40523 times

	Average nº of ratings	Median nº of ratings
CB	130035	40523
KNN_IB	98660	34716
KNN_UB	58585	23157
ALS	60788	22500

Table 3 – Average and median rating values per model

The item-based algorithms (CB and KNN_IB) are the ones that have the lowest number of recommended books, with KNN_IB having the book with the highest recommendation frequency (table 2). As seen in table 3, item-based algorithms are also the ones recommending books with the highest ratings, in other words, more popular books. Joining both these phenomena, recommending fewer books and highly rated books, allows us to understand that item-based algorithms are more prone to popularity bias within this domain. Zhou et al.(2010) also achieved similar conclusions where content and item-based models create higher popularity bias.

5.1. COMPARISON OF METRIC RESULTS

Figure 2 shows the metrics results obtained for each model.

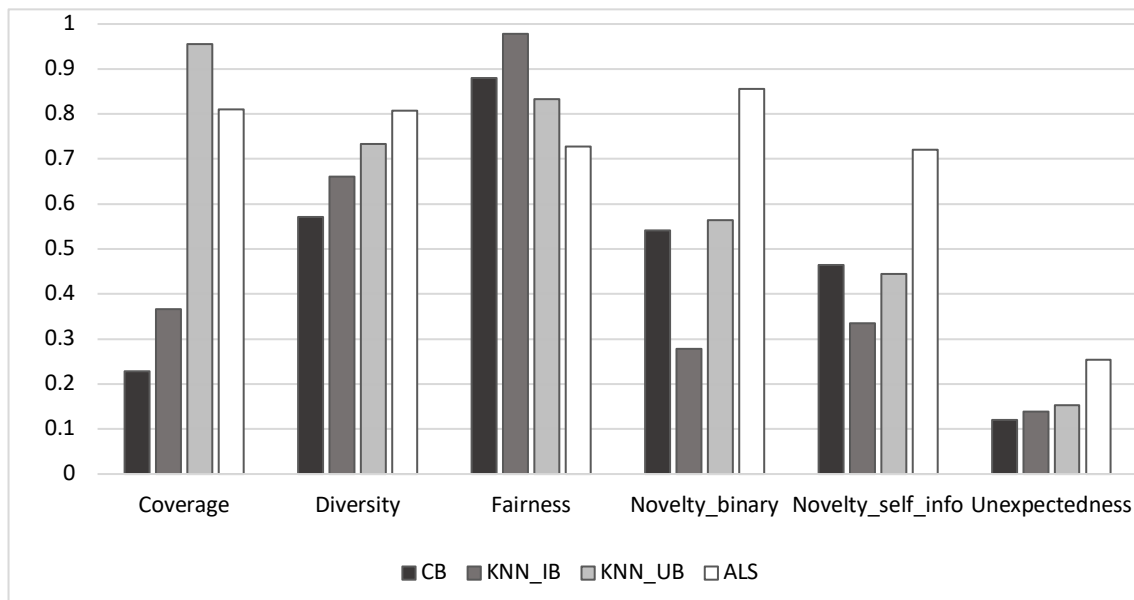


Figure 2 - Comparison of metric results for each evaluated model

Because diversity is measuring similarity between the books in a set of recommendations by looking at features like the title and genres, it is understandable that item-based algorithms are also underperforming in this metric. In the Harry Potter example, it is not a very diverse set of recommendations if a user is receiving six recommendations of Harry Potter books. These results concord with the ones obtained by Kaminskas et al. (Kaminskas & Bridge, 2017), where their item-based algorithms also achieved lower *diversity*. It is also observable that *diversity* has a high correlation with *unexpectedness*, and the more diverse an item, the higher the probability of that item being unexpected by the user.

Regarding fairness, contrary to the previous metrics, the user-based algorithms, KNN_UB and ALS, underperform the item-based algorithms. It is essential to consider that the database has a higher number of male authors (59%) than female authors, achieving a base database fairness of 0.82. This does not necessarily mean that the user-based algorithms are intrinsically more biased towards one gender. These results are due to one of the features given to the item-based algorithms being the *author_gender* which makes those algorithms aware of the author's gender. If a user has female-written books on his profile, the probability of the recommended books containing female-written books is higher. The opposite happens in user-based algorithms. Because a bias already exists in the data, a user-based algorithm will only extrapolate that gap because it represents a user's behavior. This can be confirmed by observing that the item-based algorithms surpassed the database fairness value of 0.82 and reduced the bias. In contrast, the user-based algorithms failed to achieve the inherent database fairness, amplifying the bias effect.

The ALS considerably outperformed the other models, providing the most novel and surprising recommendations. In both *novelty* metrics, the CB algorithm outperforms KNN_IB, almost equaling the KNN_UB, which may seem counterintuitive but can be explained by the intrinsic characteristics of the CB algorithm. CB has a high popularity bias and lower coverage, meaning a few popular books are being recommended with high frequency, with most books belonging to fictional and fantasy genres. Some readers might not usually read fictional books and tend to be more niche, so popular or fictional books can be scarce in their user profile and become novel when recommended. CB *unexpectedness* is lower, meaning the recommended items have a high general probability of being consumed, which, allied to the *novelty* value, further proves the system is suggesting popular books to readers that might not fit into those genres of books. Because KNN_IN is aware of what type of user reads a specific kind of book, it can produce more relatable recommendations at the expense of *novelty*. Our *novelty* results follow the work by Bellogín et al. (Bellogín et al., 2013), where the user-based KNN also generated more novel recommendations than the item-based KNN.

5.2. ENHANCING QUALITY

All algorithms suggested twenty items to run the re-raking approach, and the best ten were selected after the reranking. The four algorithms were subject to the *beyond-accuracy quality* reranking approach and achieved the results displayed in table 4 below. The percentages inside the brackets represent the variation compared to before the reranking.

	CB_quality	KNN_IB_quality	KNN_UB_quality	ALS_quality
Coverage	0.287 (26%)	0.391 (7%)	0.944 (4%)	0.693 (-32%)
Diversity	0.645 (13%)	0.730 (10%)	0.781 (7%)	0.857 (7%)
Fairness	0.803 (-9%)	0.971 (-1%)	0.833 (0%)	0.81 (8%)
Novelty_binary	0.767 (42%)	0.414 (48%)	0.752 (33%)	0.971 (41%)
Novelty_self_info	0.562 (21%)	0.348 (4%)	0.524 (18%)	0.8 (24%)
Unexpectedness	0.198 (64%)	0.199 (43%)	0.237 (55%)	0.355 (73%)
Quality	0.439 (21%)	0.454 (12%)	0.598 (15%)	0.671 (14%)

Table 4 – Metric and quality results for each reranked model.

The percentages inside the brackets represent the variation compared to the value of that model before being optimized by the reranking.

All reranked algorithms achieved a better *quality* value than before the reranking, with improvements across almost all metrics. The reranking approach significantly impacted *unexpectedness* and *novelty* across all models, producing more surprising recommendations of novel items. This means that the improvement in *beyond-accuracy quality* is mainly caused by the recommendations becoming more serendipitous, as per the definition given by Herlocker et al. (Herlocker et al., 2004).

Fairness is the metric with the lowest variation after reranking because the reranking approach does not account for fairness in its formula (equation (22)). However, it is interesting to note that the item-based algorithms decreased their fairness, and the user-based either maintained or improved it. The increase in *unexpectedness* and *novelty* can explain this. An item-based algorithm like CB tries to match the most similar books, considering the author's gender. If the system increases the *unexpectedness*, it recommends books that are not usually read together. In other words, it suggests more dissimilar books (as the *diversity* also increased), and the author's gender is no longer a factor.

Table 5 below represents the changes in popularity bias. As the *quality* increases, the average and the median number of ratings per recommended book decreases, meaning the systems become less popularity biased. The justification for this behavior is the increased *diversity*, *novelty*, and *unexpectedness*, as, after reranking, systems are more suited to suggest long-tail items. These long-tail items, by definition, have a lower number of ratings and can be perceived as novel or unexpected by most users. ALS_quality is the model with the highest *quality* and has the highest values across most metrics, especially *novelty* and *unexpectedness*. It is also the model recommending the less popular books, which allows the assumption that it is also the one suggesting the higher number of long-tail items.

	Average nº of ratings	Median nº of ratings
CB_quality	117267 (-10%)	39925 (-1%)
ALS_quality	47763 (-21%)	19555 (-13%)
KNN_UB_quality	56754 (-3%)	22327 (-4%)
KNN_IB_quality	97422 (-1%)	34390 (-1%)

Table 5 - Average and median rating values per reranked model.

The percentages inside the brackets represent the variation compared to the value of that model before being optimized by the reranking.

The ALS and ALS_quality outperform the other models in most metrics and can be defined as the models with the highest *quality*. By looking at the results of table 6, both these models are the ones with the lowest *hit rate*. They could only "hit" 5% and 2%, respectively, of the books present in each user's "to_read" list. As perceived in most cases, a higher *quality* represents a lower *hit rate*. In other words, the *quality* and *accuracy* of a system are negatively correlated. This means that the ALS models are making recommendations of novel and unexpected books that might be of no interest to the users.

It is also noticeable that the CB model was the only model that, after re-ranking, improved both the hit rate and the beyond-accuracy quality. By increasing *diversity* and *coverage*, the reranking approach removes some of the item similarity present in the CB model allowing for a vaster offer, indirectly increasing accuracy.

The KNN_UB_quality is the most balanced model between both dimensions. It is the third-best model in beyond-accuracy *quality* and, coincidentally, the third-best in *hit rate* values. In this case, the reranking approach improved the system's quality without heavily impacting its *accuracy*.

As shown in previous works by Cremonesi et al. (Cremonesi et al., 2010), Jannach et al. (Jannach et al., 2013), and Kaminskis et al. (Kaminskas & Bridge, 2017), the KNN algorithms are able to achieve higher values of *hit rate* (in their works considered as *recall*) when validating the top-N recommendation list, which is also observable in our research.

	Hit rate	Quality
ALS_quality	0.020	0.671
ALS	0.050	0.613
CB	0.118	0.365
KNN_IB_quality	0.132	0.454
CB_quality	0.132	0.439
KNN_UB_quality	0.140	0.598
KNN_UB	0.153	0.518
KNN_IB	0.159	0.405

Table 6 - Hit rate and beyond-accuracy quality results per evaluated model.

Depending on the system's objective and the developer's goals, a decision must be made regarding which dimension is more important. In some cases, the developers might want to focus on providing the most diverse, fair, and novel recommendations. In this case, they should choose systems with a higher *quality* even if the *accuracy* will be lower. On the contrary, by choosing higher-accuracy systems, the developers and business owners must accept that they may compromise the system's *quality*.

Chen and Pu (2010) performed an online study about the user's perceived usefulness of a system. They concluded that the users found more helpful systems focused on *novelty* and *accuracy* rather than *diversity*. Contrarily, Ekstrand et al. (2014) discovered that *diversity* provided more usefulness than *novelty*. The difference between these two studies is in the domains used. One author worked on an Amazon product recommender, and the other on a movie recommender. This further demonstrates that each metric's relevance depends on the domain chosen.

6. CONCLUSION

As Herlocker et al. (2004) mentioned, making an effective and meaningful evaluation of a recommender system is challenging. This chapter will summarize our key research findings and expose their limitations and suggestions for future work.

In this research, a survey of the state-of-the-art beyond-accuracy metrics was performed, namely on *coverage*, *diversity*, *novelty*, *serendipity*, *utility*, and *fairness*. For each, we exposed the relevant definitions and where their strengths lie. Each of these metrics measures a different aspect of a recommendation system that is not linked with *accuracy*. *Coverage* covers the number of items that are or can be recommended and the *diversity of* how different those items are from each other. *Novelty* and *serendipity*, although often confused, measure how novel (*novelty*) and how surprising, novel, and valuable (*serendipity*) a recommendation is. *Utility*, as the name implies, tries to understand how beneficial a recommendation is for a user. However, significant assumptions must be made to predict an item's utility without directly asking the user through an online survey or by adding a new feature in the system questioning users. Finally, *fairness* is an ever more relevant metric, measuring how biased a system is regarding a specific type of user or item. In certain domains, it is paramount that the recommendations are fair, especially if it is a domain where the gender or ethnicity of a user is an input feature. Furthermore, we proposed using a summary metric, called *beyond-accuracy quality*, that would enable the comparison of items and systems regarding their *quality*. This metric is then used on a suggested reranking approach to increase the beyond-accuracy performance of a system.

An offline experimentation was performed to study the relationships between the beyond-accuracy metrics and to implement *the beyond-accuracy quality* reranking approach mentioned above. A comparison was made between all models, before and after the reranking, with the respective *hit rates* to understand the behaviors and potential impacts on *accuracy*. Four algorithms were chosen for the experimentation. One content-based algorithm (CB), two memory-based algorithms (one item-based and one user-based) (KNN_IB and KNN_UB), and a matrix factorization algorithm (ALS). The reasoning behind using these algorithms is that they are very commonly used in the recommender system literature and represent different machine learning methods in recommendation systems modeling. The research was conducted in the domain of book sales, using a subset of Amazon's Goodreads dataset. As the dataset is not commonly found in the recommender system literature, we hope this research could provide a helpful reference regarding the beyond-accuracy performance in this domain. Each algorithm was modeled to retrieve the top 10 best items for each user as per that algorithm criterion, which for most algorithms, is the predicted rating for the item.

We realized that, within the book domain, item-based algorithms tend to recommend fewer books that are amongst the most popular in terms of rating numbers. They were also the ones achieving higher levels of *fairness*, with *fairness* being regarded as the recommended proportion of male and female written books. The ALS algorithm was the one suggesting the most novel and unexpected items. Furthermore, the four algorithms were optimized for their *beyond-accuracy quality*. A slight change in the modeling was performed for the models to output twenty items instead of ten. All those twenty items were validated according to the *item_quality* metrics, and the best ten items were chosen. The reranking approach generally increased the beyond-accuracy metric values across all algorithms, especially their *novelty* and *unexpectedness*. All four achieved a higher *beyond-accuracy quality* and a

lower popularity bias than before the reranking. We believe this reranking approach, alongside the proposed *beyond-accuracy quality*, will help researchers and developers easily compare the *quality* of different algorithmic options and optimize their systems. By studying the beyond-accuracy performance of the models alongside their *hit rate* values, we realized that as the *quality* of the recommendations increases, the *hit rate* decreases. This helps demonstrating the negative correlation between accuracy and non-accuracy, as referenced in the literature.

We believe the goal should not be creating a system with the highest value of neither dimension but a balanced system with minimal trade-offs between the dimensions. A system with very low accuracy will be no better than recommending random items, but a system with high accuracy will most likely only recommend obvious items and have no value to the users. Therefore, equilibrium is critical. The increase of beyond-accuracy metrics must be strategically done not to confuse the users and create distrust in the system. This research found that the KNN_UB model provides the most balanced recommendations within both dimensions.

6.1. LIMITATIONS AND FUTURE WORK

Our research is limited to only being deployed in an offline validation setting. This is due to the existing constraints on performing an online validation, namely the need for a dynamic database fed by a live recommendation system interacting with real users. However, for future work, we suggest testing the reranking approach and validation using the beyond-accuracy quality on a live system. It could be interesting to understand if the recommendations provided by this approach translate well in an online scenario and can provide users with a good experience with valuable recommendations.

We only focused on four commonly used algorithms to perform our study, meaning we are bound to those algorithms' limitations. We incentivize testing on different types of algorithms, especially on deep learning models, and comparing the results with the ones present in this work. Furthermore, it could be interesting to test the beyond-accuracy quality reranking approach on other algorithmic models and observe if it can also improve the quality of recommendations. It is also worth studying if these results and the behaviors of the beyond-accuracy quality reranking approach translate well to other domains and systems. As stated before, the importance and behavior of a beyond-accuracy metric can be highly dependent on the system's domain. The reranking approach might need some adaptations to work appropriately in different domains. A book recommender and a music recommender behave differently, and the users have different needs. In the first, novelty and unexpectedness might be desired, while in the music recommender, users might prefer more familiar recommendations [Kapoor et al. 2015]. Furthermore, we believe more research should be done regarding the user perceptions of accuracy and non-accuracy metrics, specifically understanding which significantly impacts user satisfaction and provides a better experience using the system.

A final limitation is regarding the temporal aspect of recommendations. In this research, we did not consider time as a dimension for analysis. However, user tastes and desires change over time, and a user might not like an author that was his favorite years ago. Research could be done regarding beyond-accuracy metrics that use time as a factor and try to understand the metric's influence as time advances. An item is not novel forever, so when validating a system, it is essential to note its ability to reshape recommendations as time progresses.

7. REFERENCES

- Adamopoulos, P., & Tuzhilin, A. (2014). On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Transactions on Intelligent Systems and Technology*, 5(4), 1–32. <https://doi.org/10.1145/2559952>
- Adomavicius, G., & Kwon, Y. (2012). *Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques*. 896–911.
- Armstrong, R. (2008). The Long Tail: Why the Future of Business Is Selling Less of More. *Canadian Journal of Communication*, 33(1), 127–128. <https://doi.org/10.22230/cjc.2008v33n1a1946>
- Baeza-yates, R., & Ribeiro-Neto, B. (1999). *Moderne Information Retrieval*. ACM Press, July.
- Belloñín, A., Cantador, I., Díez, F., Castells, P., & Chavarriaga, E. (2013). An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Transactions on Intelligent Systems and Technology*, 4(1). <https://doi.org/10.1145/2414425.2414439>
- Bennett, J., & Lanning, S. (2007). The Netflix prize. *Communications of the ACM*. <https://doi.org/10.1145/1562764.1562769>
- Bobadilla, J., Lara-Cabrera, R., González-Prieto, Á., & Ortega, F. (2020). *DeepFair: Deep Learning for Improving Fairness in Recommender Systems*. <https://doi.org/10.9781/ijimai.2020.11.001>
- Carbonell, J., & Goldstein, J. (1998). Use of MMR, diversity-based reranking for reordering documents and producing summaries. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 335–336. <https://doi.org/10.1145/290941.291025>
- Celma, Ò. (2008). Music recommendation and discovery in the long tail. *Media*, 32(3), 1–252. http://www.recolecta.net/buscador/single_page.jsp?id=oai:UPF.es:TDX-0612109-190038
- Chausson, O. (2010). *Assessing The Impact Of Gender And Personality On Film Preferences*.
- Chen, L., & Pu, P. (2010). A User-Centric Evaluation Framework of Recommender Systems. *Proceeding of UCERSTI Workshop of RecSys'10, January 2011*, 14–21.
- Chouldechova, A., & Roth, A. (2020). A snapshot of the frontiers of fairness in machine learning. In *Communications of the ACM* (Vol. 63, Issue 5, pp. 82–89). Association for Computing Machinery. <https://doi.org/10.1145/3376898>
- Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems, September*, 39–46. <https://doi.org/10.1145/1864708.1864721>
- Deshpande, M., Karypis, G., Karypis, G., & Deshpande, M. (2004). Item-Based Top-N Recommendation Algorithms. In *ACM Transactions on Information Systems* (Vol. 22, Issue 1).
- Desrosiers, C., & Karypis, G. (2011). A comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook* (Issue January). <https://doi.org/10.1007/978-0-387-85820-3>
- Ekstrand, M. D., Harper, F. M., Willemsen, M. C., & Konstan, J. A. (2014). User perception of differences in recommender algorithms. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 161–168. <https://doi.org/10.1145/2645710.2645737>

- Ferraro, A., Serra, X., & Bauer, C. (2021). Break the Loop: Gender Imbalance in Music Recommenders. *CHIIR 2021 - Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, 249–254. <https://doi.org/10.1145/3406522.3446033>
- Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010). Beyond accuracy: Evaluating recommender systems by coverage and serendipity. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems, January*, 257–260. <https://doi.org/10.1145/1864708.1864761>
- Granovetter, M. S. (1973). The strength of weak ties. *American Journal of Sociology*, 78(6), 1360–1380.
- Gruson, A., Chandar, P., Charbuillet, C., McInerney, J., Hansen, S., Tardieu, D., & Carterette, B. (2019). Offline evaluation to make decisions about playlist recommendation algorithms. *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 420–428. <https://doi.org/10.1145/3289600.3291027>
- Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (2004). Evaluating Collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53. https://doi.org/10.1007/978-3-540-72079-9_9
- Iaquinta, L., De Gemmis, M., Lops, P., Semeraro, G., Filannino, M., & Molino, P. (2008). Introducing serendipity in a content-based recommender system. *Proceedings - 8th International Conference on Hybrid Intelligent Systems, HIS 2008, October*, 168–173. <https://doi.org/10.1109/HIS.2008.25>
- Jannach, D., Lerche, L., Gedikli, F., & Bonnin, G. (2013). What recommenders recommend - An analysis of accuracy, popularity, and sales diversity effects. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7899 LNCS, 25–37. https://doi.org/10.1007/978-3-642-38844-6_3
- Kaminskas, M., & Bridge, D. (2014). Measuring Surprise in Recommender Systems. *RecSys REDD 2014: International Workshop on Recommender Systems Evaluation: Dimensions and Design*, 69, 2–7. <http://www.springerlink.com/index/N3JQ77686228781N.pdf%5Cnhttp://ir.ii.uam.es/redd2014/program/paper03.pdf>
- Kaminskas, M., & Bridge, D. (2017). Diversity, Serendipity, Novelty, and Coverage. *ACM Transactions on Interactive Intelligent Systems*, 7(1), 1–42. <https://doi.org/10.1145/2926720>
- Kapoor, K., Kumar, V., Terveen, L., Konstan, J. A., & Schrater, P. (2015). “i like to explore sometimes”: Adapting to dynamic user novelty preferences. *RecSys 2015 - Proceedings of the 9th ACM Conference on Recommender Systems*, 19–26. <https://doi.org/10.1145/2792838.2800172>
- Kelly, J. P., & Bridge, D. (2006). Enhancing the diversity of conversational collaborative recommendations: A comparison. *Artificial Intelligence Review*, 25(1–2), 79–95. <https://doi.org/10.1007/s10462-007-9023-8>
- Lu, Q., Chen, T., Zhang, W., Yang, D., & Yu, Y. (2012). Serendipitous personalized ranking for top-N recommendation. *Proceedings - 2012 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2012*, 258–265. <https://doi.org/10.1109/WI-IAT.2012.135>
- McLaughlin, M. R., & Herlocker, J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *Proceedings of Sheffield SIGIR - Twenty-Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 329–336. <https://doi.org/10.1145/1008992.1009050>
- McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. *Conference on Human Factors in Computing Systems -*

- Proceedings*, August, 1097–1101. <https://doi.org/10.1145/1125451.1125659>
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). *A Survey on Bias and Fairness in Machine Learning*. <http://arxiv.org/abs/1908.09635>
- Oh, J., Park, S., Yu, H., Song, M., & Park, S. T. (2011). Novel recommendation based on Personal Popularity Tendency. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 507–516. <https://doi.org/10.1109/ICDM.2011.110>
- Pitoura, E., Stefanidis, K., & Koutrika, G. (2021). Fairness in rankings and recommendations: an overview. *VLDB Journal*. <https://doi.org/10.1007/s00778-021-00697-y>
- Ribeiro, M. T., Lacerda, A., Veloso, A., & Ziviani, N. (2012). Pareto-efficient hybridization for multi-objective recommender systems. *RecSys'12 - Proceedings of the 6th ACM Conference on Recommender Systems*, 19–26. <https://doi.org/10.1145/2365952.2365962>
- Ricci, F., Rokach, L., & Shapira, B. (2011). Recommender Systems Handbook. In *Recommender Systems Handbook* (Issue October). <https://doi.org/10.1007/978-0-387-85820-3>
- Rich, E. (1979). User modeling via stereotypes. *Cognitive Science*, 3(4), 329–354. [https://doi.org/10.1016/S0364-0213\(79\)80012-9](https://doi.org/10.1016/S0364-0213(79)80012-9)
- Salton, G., & Buckley, C. (1987). Term-weighting approaches in automatic text retrieval. In *Information Processing & Management* (pp. 513–523).
- Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 257–297). Springer US. https://doi.org/10.1007/978-0-387-85820-3_8
- Silveira, T., Zhang, M., Lin, X., Liu, Y., & Ma, S. (2019). How good your recommender system is? A survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, 10(5), 813–831. <https://doi.org/10.1007/s13042-017-0762-9>
- Smyth, B., & McClave, P. (2001). Similarity vs. Diversity. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2080(Section 2), 347–361. https://doi.org/10.1007/3-540-44593-5_25
- Vargas, S., Baltrunas, L., Karatzoglou, A., & Castells, P. (2014). Coverage, redundancy and size-awareness in genre diversity for recommender systems. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 209–216. <https://doi.org/10.1145/2645710.2645743>
- Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems*, 109–116. <https://doi.org/10.1145/2043932.2043955>
- Vargas, S., & Castells, P. (2014). Improving sales diversity by recommending users to items. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 145–152. <https://doi.org/10.1145/2645710.2645744>
- Vargas, S., Castells, P., & Vallet, D. (2011). Intent-oriented diversity in recommender systems. *SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1211–1212. <https://doi.org/10.1145/2009916.2010124>
- Yao, S., & Huang, B. (2017). *Beyond Parity: Fairness Objectives for Collaborative Filtering*.

<http://arxiv.org/abs/1705.08804>

- Zemel, R., & Swersky, K. (2013). *Learning Fair Representations*. 28.
- Zhang, L. (2013). The definition of novelty in recommendation system. *Journal of Engineering Science and Technology Review*, 6(3), 141–145. <https://doi.org/10.25103/jestr.063.25>
- Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., & Jambor, T. (2012). Auralist: Introducing serendipity into music recommendation. *WSDM 2012 - Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 13–22. <https://doi.org/10.1145/2124295.2124300>
- Zhou, T., Kuscsik, Z., Liu, J. G., Medo, M., Wakeling, J. R., & Zhang, Y. C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(10), 4511–4515. <https://doi.org/10.1073/pnas.1000488107>
- Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5034 LNCS, 337–348. https://doi.org/10.1007/978-3-540-68880-8_32
- Ziegler, C.-N., McNeely, S. M., Konstan, J. A., & Lausen, G. (2005). *Improving recommendation lists through topic diversification*. January 2005, 22. <https://doi.org/10.1145/1060745.1060754>

NOVA

IMS

Information
Management
School

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa