A recommender system based on item properties

Maria João Lavoura
Mestrado em Ciência de Dados
Departamento de Ciência de Computadores
2023

Orientador
João Vinagre, Professor auxiliar convidado, Faculdade de Ciências

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

MSc

Acknowledgements

This thesis was made possible with the help of many people, to whom I am deeply grateful. To Pedro Silva, for all the insights and support. To João Vinagre, for guiding me through this idea of modelling items properties. To Nirbhaya Shaji, for being always available to give feedback. To my grandma, for all the times she told me ”You can do it!”.

UNIVERSIDADE DO PORTO

Abstract

Faculdade de Ciências da Universidade do Porto

Departamento de Ciência de Computadores

MSc. Data Science

A recommender system based on item properties

by Maria João Lavoura

After the user views three long-sleeved flowy dresses, he purchases one recommended long-sleeved flowy dress. Recommending a purchase is the ultimate goal of recommender systems. Therefore, is there a discernible pattern in the side information of the few clothing items viewed by an active user in sessions where they make a purchase, which could be relevant to the implicit user preferences, given the explicit preference revealed in the purchase action? The aim of this dissertation is to recommend items based on users' implicit preferences about item properties, with a focus on the cold start problem in the fashion domain. Hence, the study proposes a novel approach to handling side information of items as user-item interaction data in recommender systems. Items-based sessions were transformed into sessions comprised of properties of items. The final recommendation of the proposed model is a ranking of items based on the initial recommendation of item attributes. In addition, this dissertation proposes a hybrid model that integrates the item-properties approach with a traditional one. The main advantages of the proposed framework are the independence of the algorithm used and the domain to which it is applied. Additionally, it can ease the addition of new items to the catalogue as long as they share the properties with the existing items. Results reveal that the traditional modelling approach outperformed the item properties-based model.

UNIVERSIDADE DO PORTO

Resumo

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

MSc. Data Science

A recommender system based on item properties

by Maria João Lavoura

Depois de ver três vestidos fluidos de manga comprida, o utilizador compra um vestido fluido de manga comprida recomendado. Recomendar uma compra é o objetivo final dos sistemas de recomendação. Assim, será que existe um padrão discernível na informação lateral dos poucos artigos de vestuário vistos por um utilizador ativo em sessões em que este efectua uma compra, que possa ser relevante para as preferências implícitas do utilizador, dada a preferência explícita revelada na ação de compra? O objetivo desta dissertação é recomendar artigos com base nas preferências implícitas dos utilizadores sobre as propriedades dos artigos, com enfoque no problema do cold start no domínio da moda. Assim, o estudo propõe uma nova abordagem para tratar a informação lateral dos artigos como dados de interação utilizador-artigo em sistemas de recomendação. As sessões baseadas em itens foram transformadas em sessões compostas por propriedades de itens. A recomendação final do modelo proposto é uma classificação dos itens com base na recomendação inicial dos atributos dos itens. Para além disso, esta dissertação propõe um modelo híbrido que integra a abordagem das propriedades dos itens com uma abordagem tradicional. As principais vantagens do modelo proposto são a independência do algoritmo utilizado e do domínio em que é aplicado. Além disso, pode facilitar a adição de novos itens ao catálogo, desde que estes partilhem as propriedades com os itens existentes. Os resultados revelam que a abordagem de modelação tradicional superou o modelo baseado nas propriedades dos itens.

Contents

List of Figures

List of Tables

Glossary

RS	Recommender Systems
CS	Cold Start
CF	Collaborative Filtering
MRR	Mean Reciprocal Rank
HR	Hit Ratio
KNN	K Nearest Neighbours
RPV	Recommended Property Value

Chapter 1

Introduction

Faced with the need to navigate enormous amounts of irrelevant information to access products or services suited to an individual's preferences, consumers benefit from suggested personalised recommendations. This is the goal of Recommender Systems (RS).

When entering a fashion e-commerce store, the user might be casually shopping (window shopper) or specifically shopping for a particular garment, such as a *black oversized sweatshirt*. A window shopper may decide to make a purchase if the print, fabric or size of the clothes item is to their taste. Hence, in the fashion sector, the attributes of an item highly influence the user's purchase decision. Therefore, modelling such content-level attributes would help capture particular user preferences and user shopping intentions, ultimately leading to improved recommendations. Additionally, in situations where minimal information about a user is known, information about item-level attributes could help mitigate the cold-start problem in RS.

In a session where the active user made a purchase after viewing four clothing items, was the user browsing or intentionally seeking the purchased item? Short sessions may contain several irrelevant interactions, but is there a discernible pattern in the side information of items that could be pertinent in "purchasing sessions"?

This dissertation proposes a novel approach to handle item content data as user-item interaction data in a cold start setting, *i.e.*, a RS based on item properties. The aim is to recommend items based on user implicit preferences on item properties. Instead of considering the user's history of viewed items, the algorithm analyses the user's history of viewed attributes of items. Furthermore, it proposes a hybrid model that combines the item recommendations of a traditional approach with those of the item-properties proposed modelling approach. Therefore, it enhances conventional recommendations with the nuances of item properties information.

The goal of this research is to address the following research questions:

- **Q1:** How does the proposed hybrid method perform compared to a classical method?
  - **Q1.1:** Does reordering the baseline model recommendation space, based on the recommendations of the proposed properties model, improve performance?
  - **Q1.2:** Does the proposed item-properties model introduce new items to the recommendation space of the proposed hybrid model?
- **Q2:** How does the proposed method perform with different approaches to calculate the item-translating scores of the properties model?
- **Q3:** How does the proposed hybrid model distribute the weight between the state-of-the-art model and the proposed properties model?

The main contributions of this dissertation are the following:

- creation of a novel content-based modelling approach to Recommender Systems.
- implementation of a hybrid model that combines the traditional modelling approach with the proposed one.

The thesis is organized as follows. Chapter 2 gives a brief background on Recommender Systems, Cold Start and fashion e-commerce, and presents state-of-the-art works on the topics mentioned. Chapter 3 describes and analyses the data. Chapter 4 explains the proposed item-properties modelling approach and hybrid model. Chapter 5 presents and discusses the results obtained. Finally, chapter 6 presents the conclusion of the work with remarks on the main contributions and limitations, and future directions to continue this research study.

Chapter 2

Background

There is a massive amount of clothes to buy, music to listen to, movies to watch, news to read, etc. New products (which are generically referred to as items) are released every week. In the present age of the digital economy, individuals are required to sift through copious amounts of superfluous data before accessing the limited items relevant to their needs and tastes. When overwhelmed by choices, the user becomes more likely to leave the site unsatisfied, without making any purchases and with a decreased sense of loyalty. This is the problem of *information overload* [1]. Recommender systems (RS) solve the problem by suggesting personalised items to the user.

This chapter is organised as follows. Section 2.1 introduces the topic of recommender systems and the cold start problem. Section 2.2 contextualises the application of recommender systems in the fashion e-commerce domain.

2.1 Recommender Systems

The personalised list of recommended items is based on the user's expressed preference for items, either explicit preference in the form of ratings or implicit preference inferred from their purchase history and patterns. The RS predicts the user's ratings on items that have not yet been experienced from their history and then produces a ranked list of new relevant items based on the predicted rate.

Approaches on item recommendation use algorithms to make predictions and fall into two broad categories: non-personalised and personalised [2]. In the absence of more precise information regarding user preferences, a popular item that is highly valued by many users is more likely to appeal to a generic user than a randomly selected one. Hence, a simple and non-personalised recommendation algorithm example is one that suggests only the most popular items [2]. In contrast, when there is significant information regarding user preferences, personalised algorithms yield higher-valued recommendations.

2.1.1 Types of Recommender Systems

There are three different types of data that traditional personalised algorithms receive as input: user-item interactions (ratings and buying behaviour), item attribute information and domain information. Following the aggregation referred in Aggarwal [3], RS that use:

- user-item interactions of all users, are called **Collaborative Filtering** RS.
- user-item interactions of a specific user and item attribute information, are called **Content-based** RS.
- explicit user requirements, item attribute information and external domain awareness are called **Knowledge-based** RS.

The fundamental concept of **Collaborative Filtering** (CF) RS is that unspecified ratings can be inputted because observed ratings are often highly correlated between various users and items [3]. There are two widely utilized CF approaches, commonly known as neighbourhood-based and model-based methods.

1. Neighbourhood-based CF methods base their predictions on either user-centred or item-centred neighbours.
   - User-based CF considers that neighbours are users with similar rating patterns. Predicting a target user's rate for an unseen item makes use of the ratings of that item by the target user's neighbours.
   - Item-based CF considers that neighbours are items with similar ratings by multiple users. The prediction of a target user's rating, for an item they have not yet seen, relies on the ratings that same user has given to similar items.
2. Model-based CF methods differ from Neighbourhood-based CF methods in that they do not use stored ratings directly for prediction. Instead, they use these ratings to develop a predictive model that captures the significant features of both users and items in the form of model parameters. These parameters are learned from training data and subsequently used to predict new ratings. [2]

The fundamental concept of **Content-based RS** is to model user interests using the attributes of items, *i.e.*, content data, they previously rated or accessed. The aim is to identify the common features of highly ranked items by a user and suggest new items with similar characteristics.[2]

Knowledge-based RS incorporate the user's needs through the use of constraints and the recommendation processes are carried out on the basis of similarities between explicit user needs and item descriptions [3]. For instance, in the fashion domain, the visual compatibility of garments may be personalized to the style of each user. Therefore, it is common to encode the visual compatibility of garments in the definition of the outfit composition scoring function, in order to incorporate domain knowledge [4].

Hybrid RS typically combine content-based and collaborative filtering methods. There are various approaches to creating a hybrid system. One approach involves consolidating individual predictions to produce a stronger overall prediction. Alternatively, each model could recommend to the best-performing user. Another option is to include content information in the model. Several studies indicate that hybrid approaches perform better than pure RS, particularly in a cold start setting [2].

2.1.2 Cold Start problem

A recommender system with little to no information on which to support the predictions leads the system to yield unreliable recommendations. This is the problem of the *Cold Start* (CS).

Abdullah et al. [5] has categorised the CS problem in terms of the entity (user or item) and data sparsity (unavailable or insufficient information). In terms of entity, the RS does not have prior information on brand-new users or brand-new items, which has a negative impact on its performance. In terms of data sparsity, when there is no available information, the problem is a pure cold start. On the other hand, when the information is sparse and insufficient to provide recommendations, the problem is of a warm start.

Of the RS types mentioned above, in the presence of cold start, Content-based and Knowledge-based RS are more robust than Collaborative Filtering RS. [3]

2.2 A fashion case study

The application domain of a RS is a major factor in the choice of the algorithmic approach to follow. Ricci et al. [2] has grouped RS applications in entertainment (movies, songs), content (newspapers, webpages), e-commerce (fashion, books), services (travel plans), and social (person profiles in social media).

Fashion is the largest business-to-consumer e-commerce market of 2023, with a global size estimated at US$768.7 billion and an expected growth rate of 9.4% per year [6]. Even the slightest increase in e-commerce conversion rates (*i.e.*, the ratio of user sessions resulting in a monetary transaction compared to the total number of sessions) can result in significant revenue growth for businesses. Despite varying conversion rates across different countries, industries and websites, the majority of sessions involve window shoppers (users with limited purchasing intent) [7]. Therefore, converting window shoppers into purchasing customers is of utmost importance for fashion brands.

In the fashion domain, the attributes of a garment, *i.e.*, the content-level attributes of fashion items, highly influence the user's purchase decision. For instance, after viewing a *mini yellow dress with a v-neckline*, the window shopper clicks on a *yellow mini skirt*, followed by a *white v-neckline top*. By the end of the session, the user purchases a *maxi yellow dress with a v-neckline*. This example shows that when a user picks an item, the interest might be in the colour, style, fit, or any other design attributes. Therefore, modelling such content-level attributes would help capture more specific user preferences, improve the recommendations given to a window shopper and increase the likelihood of a sale happening.

This study proposes a new approach to handle item content data as user-item interaction data in a cold start setting. In other words, the *item property* replaces the *item* as the object of interest by the anonymous user. Consequently, the *item* is also replaced by the *item property* in the recommendation model. Therefore, this dissertation proposes user-item property interaction data as input data of a recommender system in a cold start setting. Moreover, this dissertation proposes a hybrid model that combines the item recommendations, of a traditional approach, with the item-properties recommendations, of the proposed modelling approach. To the authors' best knowledge, this is a brand new approach to modelling content data within the fashion domain.

Fashion RS commonly falls within the following recommendations tasks: single-item, fashion similarity, fashion item recognition, garment size, complementary items, outfit and user intent. The majority of public fashion data sets available are image-based or based on user-item ratings and reviews [8]. The ability to undertake this study was made possible by the release of the Dressipi dataset in 2022. The Dressipi dataset comprises anonymous click-stream session-based apparel data, with item properties, and a purchase by the end of each session. Consequently, it was possible to shape the data into user-item properties interactions. The data is further explored in chapter 3.

2.3 State of the Art

What influences the shopping behaviour of the user? What is the influence of the item's features? How can those influences be included in a Recommender System? These were the major broader questions that guided the study of the state of the art for this dissertation.

The authors of Ding et al. [9] explored the intent of users when they enter an online store in order to have a tailored way of reducing the recommendation space. The instant user intent was defined as the relationship between seen adjacent items: *match*, *substitute* and *others*. They proposed a content-aware translation-based fashion recommender, which effectively improved the accuracy and interpretability of the proposed framework by modelling the instant user intent and the intent-specific item transitions.

The main paper reference on this study, Deldjoo et al. [4], gathered the main prominent attributes used for item representation: colour, brand, texture, textual features, etc. This survey revealed that most publicly available fashion data sets are based on images or user rated and reviewed items. The following works, cited in the survey, investigate the fusion of user-item interaction and item side information.

Tangseng and Okatani [10] propose a method for explaining outfits based on its item features. They achieve this by quantifying how influential each feature is given how much it affects the outfit score. They define the outfit in the image by its human interpretable item properties, including texture, shape and colour. Given a bad outfit, the authors aim to find the reason why it is so.

The authors Al-Rawi and Beel [11] aim to extract, from image data, the dominant colours of clothing and accessories with a resort to clustering and deep learning.

Goncalves et al. [12] explored the luxury fashion sector, in which the audience follows closely each designer's work for the respective brand. Hence, the need for RS to understand what a brand means. They proposed a content-based hybrid to extract the fashion's expert knowledge regarding brands, in the form of textual data.

Yan et al. [13] defined user-item relationships as *buy_after_viewing*, *also_viewed*, *buy_together* and *also_bought*. Using nodes to represent items and users, and links to represent relationships, they implemented a graph called a knowledge graph to represent these relationships. The knowledge graph was an effective solution for mitigating the cold start problem.

Bracher et al. [14] makes use of tags (textual data) and images of curated article information and is able to circumvent the cold start problem by resorting to the proposed concept of Fashion DNA - a mapping of fashion articles to vectors, in an abstract space, using deep neural networks.

Nguyen et al. [15] infers the user's ranking on items based on an implicit preference score. The score is inferred from the user's actions (*clicks*,*wants* and *purchases*) on other similar items.

The following works participated in the Dressipi challenge and took interesting and in-depth approaches to handling the items' side information.

Lu et al. [16] took an extensive feature engineering approach aiming to describe the item and session contexts. Instead of using the original item content features, they summarized it regarding two axes: item-content attributes (resorting to learnable embeddings) and item-item similarities (by computing the co-occurrences of the items). Additionally, they created separate features based on the count of purchases and views for each item.

Schifferer et al. [17] created features rooted in the duration and quantity of viewed items during a session and the time elapsed between two consecutive views. They employed an ensemble of models, including the session K Nearest Neighbours algorithm. They also performed some items' feature selection based on their frequency and stability over time.

Chapter 3

Data

This chapter describes the data used in this work. Section 3.1 presents in detail the original data set. Section 3.2 explains the division of the original data into *Items data set* and *Properties data set*. Finally, section 3.3 presents the exploratory data analysis performed on the original data.

3.1 Data set

The data comprises 1.1 million online fashion retail sessions resulting in a purchase. It exclusively features clothing and footwear. Additionally, all items are labelled with content data, *e.g.*, the item *dress* is labelled with *colour: red*. Furthermore, the data set was sampled over a period of 18 months and anonymised [18]. Figure 3.1 presents a representation of items seen in a session and the purchased item.

FIGURE 3.1: Example of session in the data set. Image retrieved from Landia et al. [18].

The Dressipi data set contains four types of features:

- **sessions**: the list of items that were viewed before completing the purchase. A session is anonymous and can last up to a day.
- **purchase**: the item bought after the session ends.
- **item properties**: descriptive labels of items composed by categories and categories values. There are categories such as *color*, *fit*, *occasion*, etc; and values such as *blue*, *body skimming* and *casual*, respectively. Different types of clothing share different categories IDs for the same category. For instance, the category ID for the *colour* property is different for dresses and trousers. Figure 3.2 presents an illustration of descriptive labels of a dress.
- **timestamp**: timestamp of the seen item.

FIGURE 3.2: Example of content data for a dress. Image retrieved from Landia et al. [18].

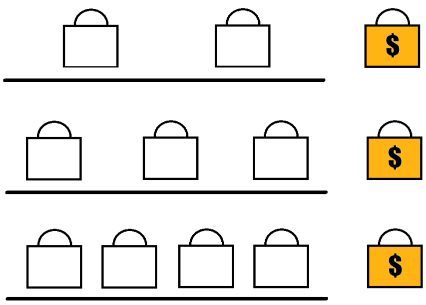
The data set was part of the RecSys Challenge 2022 [19], organized by Dressipi - a leading provider of personalized e-commerce solutions for the retail sector.

3.2 Data cleaning

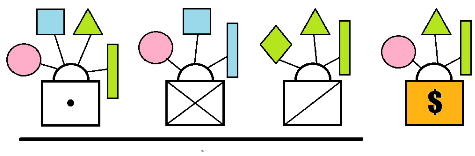
In the original data set, one session is equal to a day. However, the items can either have all been seen within the first 10 minutes or spread through the day or within the last 10 minutes. Likewise, the purchase can happen one minute or 23h after the end of the session . Moreover, in the same session, one item can show up more than once. Hence, for instance, one session with 20 items seen, can be reduced to a total of five different items.

In regard to this research work objective, which is to recommend items based on their properties in a cold start setting, the repetition of items added no relevant information. On top of that, the scope of this work does not address sequential data dependencies. Therefore, all repeated items were removed and the *timestamp* feature was also removed from the data set. The sessions considered had two, three or four different items.

The cleaned data set is composed of 380045 sessions of which 48% (183261) are two-item sessions, 31% (116288) are three-item sessions and 21% (880496) are four-item sessions. There are 23691 items, 73 properties and 904 property values. Figure 3.3 presents a generic representation of the cleaned data set sessions and content data.



(A) The representation of a two-item session, a three-item session and a four-item session. The bag symbol represents an item. The items above the back line are part of the same session *i.e.*, the seen items of a session. The golden bag with the $ character represents the purchased item of the session.

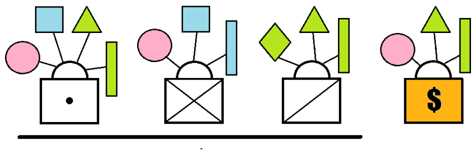


(B) One session and purchase of the cleaned data set with items (bag), item's properties (shapes) and values (shape colour).

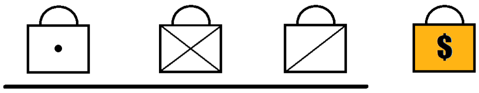
FIGURE 3.3: The representation of the cleaned data set sessions and content data. All values in the data set are encoded. Hence, the symbolic illustration.

The proposed modelling approach in the context of Recommender Systems (RS) is based on the side information of items. Accordingly, instead of having the clothing items as items in an RS, this modelling approach takes the clothing item properties as items. In order to treat the content data as interaction data and recommend properties-values for a given session, the item-based sessions were transformed into property-values-based sessions. Figure 3.4 presents a schematic representation of the transformation of the data set. One session in the original data is comprised of both items and item property values.

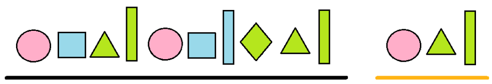
The same session in the *Items data set*is represented by the items alone. In the *Properties data set*, the session is comprised of all property values that were present in the seen items.



(A) One session and purchase of the cleaned data set with items, item's properties and values.



(B) One session and purchase of the *Items data set*.



(C) One session and purchase of the *Properties data set*.

FIGURE 3.4: The symbolic representation of one session of each data set - *Items data set* and *Properties data set* - obtained from the cleaned data set.

3.3 Exploratory data analysis

This section starts by giving a description of the simple statistics of the data. Section 3.3.2, section 3.3.3, section 3.3.4 and section 3.3.5 explain, respectively, the concepts of occurrence, insistence, co-occurrence and insistence of co-occurrence applied to the particular case of this data set. Finally, section 3.3.6 presents the analysis of the item properties according to the introduced concepts.

3.3.1 Simple statistics

The property-values-based sessions data set (*Properties data set*) exponentially increased in size compared to the item-based sessions data set (*Items data set*). The new data set required computation power to perform data analysis that was not available. Therefore, the following exploratory data analysis was performed on the data set's first 10% sessions (a total of 38000 sessions).

The sampled data set has a minimum of two items per session and a maximum of four items per session. The average is 2.73 session items and the median is three items. There are 10762 different purchased items. An item was purchased on average 3.5 times, with a median of two times and a maximum of 347 times.

Figure 3.5 presents the distribution of the number of properties per session. A session has at least two properties and a maximum of 49 properties. On average, there are 25 properties in a session. 13% of the sessions have 18 properties and another 13% have 25 properties. Sessions with less than 13 properties or more than 44 properties are fewer than 0.1% of the sessions. The distribution is quite similar to the one with the whole data that can be found in Landia et al. [18]. Hence, it validates the 10% sessions are representative of the whole data set and the analysis performed is significant.

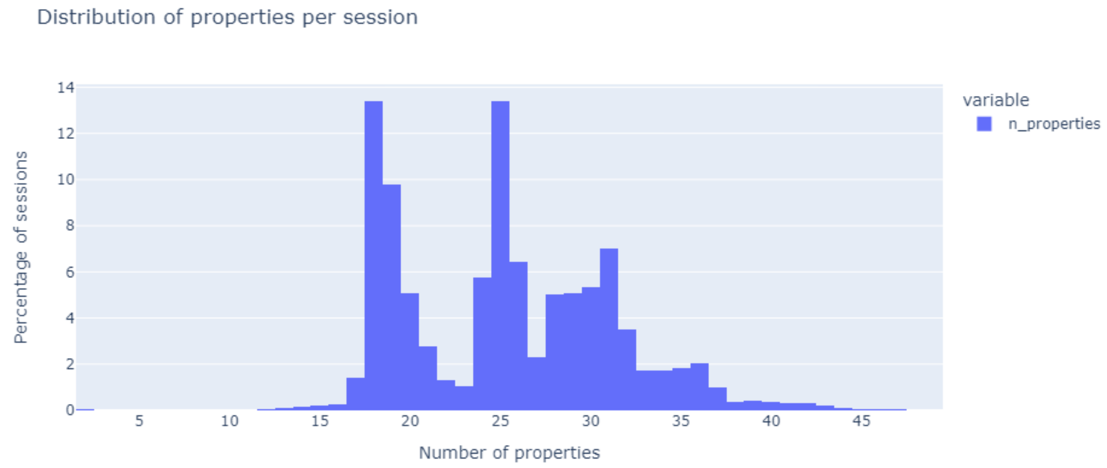


FIGURE 3.5: Distribution, in percentage of sessions, of the number of properties per session.

Figure 3.6 presents the distribution, in percentages, of the values of the properties present in the purchased item that also exist in the session items. 10.5% of the sessions have 40% of the purchase property values present in the session. Sessions with less than 20% of purchase property values present in the session or more than 77% are fewer than 4% of the sessions. Both the instances with no purchase property values previously seen in the session and the instances where all property values seen in the session were present in the purchase happen in 0.005% of the instances. Therefore, there is an overlap of information between sessions and purchases that allows for the proposed methodology to be studied.

In order to verify the content data reliability to perform this study, in other words, to understand if the properties and properties values had a consistent behaviour intra-sessions and inter-sessions, the analysis of its occurrence, insistence, co-occurrence and insistence of the co-occurrence was performed.

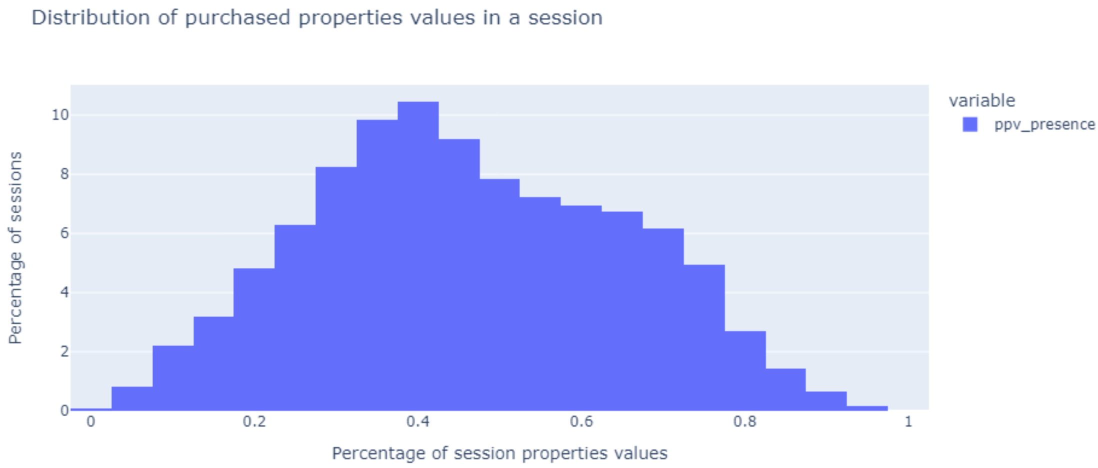


FIGURE 3.6: Distribution, in percentages, of the values of the properties present in the purchased item that also exist in the session items.

3.3.2 Occurrence

As a simple and intuitive approach, a property that appears more frequently within a session is more likely to be something the user is interested in. Hence, it is more relevant. The occurrence of one property is given by the percentage of items, in a session, where it is present. This percentage quantifies the relevancy of a given property in one session. Figure 3.7 presents a schematic representation of the occurrence of a property.

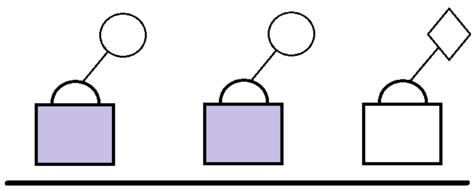


FIGURE 3.7: In this sample session, the circle property appeared twice, while the diamond property only once. Intuitively, the former is more relevant than the latter. In this session, the circle property occurrence is 2/3 and the diamond property occurrence is 1/3.

A property occurrence is calculated in the following way:

```
for each session
  for each property in the session
      property_occurrence = number_of_items_with_the_property /
                                        session_total_number_of_items
```

3.3.3 Insistence

On one hand, the occurrence reflects the property's relevancy in a session. On the other hand, the insistence reflects the persistence of a property's value. A property value that appears consistently within a session is more likely to be something the user is interested in. Hence, it is more relevant. A property insistence is given by the percentage of items with the property value that has the highest presence, in the session. Figure 3.8 presents a schematic representation of the insistence of a property.

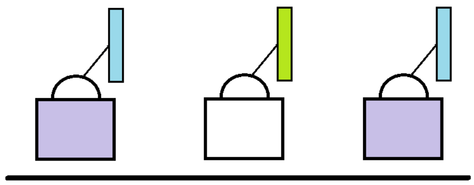


FIGURE 3.8: In this sample session, the rectangle property appeared in all three seen items, but with different values: blue and green. The blue value appeared more times than the green. Intuitively, the former is more relevant than the latter. The rectangle property insistence, in this session, is 2/3. This means the rectangle property is persistent in 66% of the items of the session.

A property insistence is calculated in the following way:

```
for each session
  for each property in the session
      property_insistence = highest_count_of_a_property_value /
                                          frequency_of_the_property
```

3.3.4 Co-occurrence

Two properties that appear together more frequently in many sessions are more likely to keep co-occurring. Hence, a popular co-occurring pair of properties that is present in the session, if relevant in that session, will also be relevant in the session's purchase. The co-occurrence of two properties is given by the percentage of items, in a session, where both are present simultaneously. Figure 3.9 presents a schematic representation of the occurrence of a property.

The co-occurrence of two properties is calculated in the following way:

```
for each session
  for each pair of properties in each item
    properties_cooccurrence = number_of_items_with_both_properties /
                                          session_total_number_of_items
```

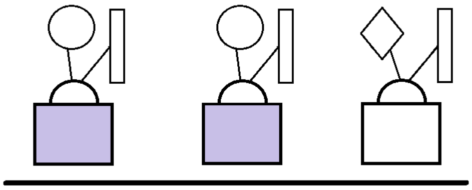
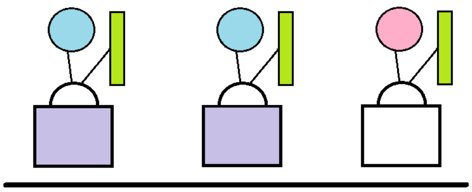


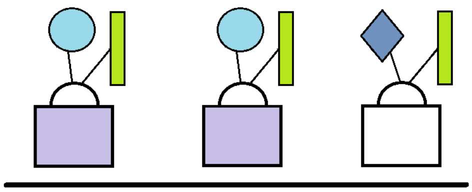
FIGURE 3.9: The circle and rectangle properties happened together twice out of the three seen items, and the pair diamond and rectangle happened once. In this sample session, the circle-rectangle co-occurrence is 2/3 and the diamond-rectangle co-occurrence is 1/3. In this example, the circle and diamond properties never co-occurred. So, it is not counted.

3.3.5 Insistence of co-occurrences

In the same line of thought of the previous sections, the co-occurrence reflects the pair of properties' relevancy in a session, while the insistence reflects the persistence of the value held by the pair of properties. A popular co-occurring pair of properties that is present in the session, is most likely to be liked by the user. On top of that, if its value appears consistently within a session, then it corroborates the user interest. The insistence of a co-occurring pair of properties is given by the percentage of items with the pair of values that has the highest presence, in the session, where both are present simultaneously. Figure 3.10 presents a schematic representation of the insistence of a co-occurring pair of properties.



(A) In this example, the circle and rectangle properties are present simultaneously in all three seen items. The blue and green pair happened twice, while pink and green only happened once. Therefore, in this session, the insistence of the pair circle and rectangle is 2/3.



(B) In this example, the circle and rectangle properties are present simultaneously in two of the three seen items. The blue and green pair happened in all of the items the pair of properties were present simultaneously. Therefore, in this session, the insistence of the circle-rectangle pair is 2/2 and the insistence of the diamond-rectangle is 1/1.

FIGURE 3.10: The schematic representation of the insistence of a co-occurring pair of properties.

The insistence of a co-occurring pair of properties is calculated in the following way:

```
for each session
```

```
for each pair of properties in each item
  cooccurrent_properties_insistence = highest_count_of_a_property_value_pair /
                                      frequency_of_the_properties_pair
```

3.3.6 Analysis of the properties

The previous sections introduced the metrics of occurrence, insistence, co-occurrence and insistence of co-occurrence applied to the theme of this dissertation. This section performs the analysis of the item properties, according to the aforementioned metrics. The values obtained from each property for each metric were session-specific. The average of each metric was calculated and plotted in terms of the percentage of items in a session where they were present in the data set.

Figure 3.11 presents the average insistence versus the average occurrence of each property. The occurrence values are all above 40%. Only four properties have values below 50% and there is a small group whose values are above 90%. Having no values below 40% indicates that there is no property that is present on average in one item of a three-item or a four-item session. Properties above 90% are present on average in all session items. All insistence values are generally high, above 60%. On one hand, this indicates that all properties that occurred in half or fewer session items did not keep the same value. On the other hand, this indicates that properties that were present in the majority of the session items did keep the same value. The majority of the properties have high occurrence and high insistence. In other words, this group of properties are present in almost all session items and they keep a persistent value within a session. For instance, the property 'fc_5' has an average occurrence of 0.749 and an average insistence of 0.945. This indicates this property, on the average of the sessions it has occurred, was present in 75% of the seen items of each session. Moreover, the value was consistently the same in 95% of the items where it was present. Hence, in a four-item session, this property would have kept a persistent value in all three items where it occurred.

Half the pair of properties have not co-occurred in any of the sessions. Figure 3.12 presents the average insistence co-occurrence versus the average co-occurrence of each pair of properties that did. The co-occurrence values are all above 35%, with the exception of three pairs. The majority of the pairs lay between 40% and 83%. The remaining pairs are all above 83%. The pairs below 35% were present simultaneously on average in one item of a three-item or a four-item session. The pairs above 90% were present simultaneously

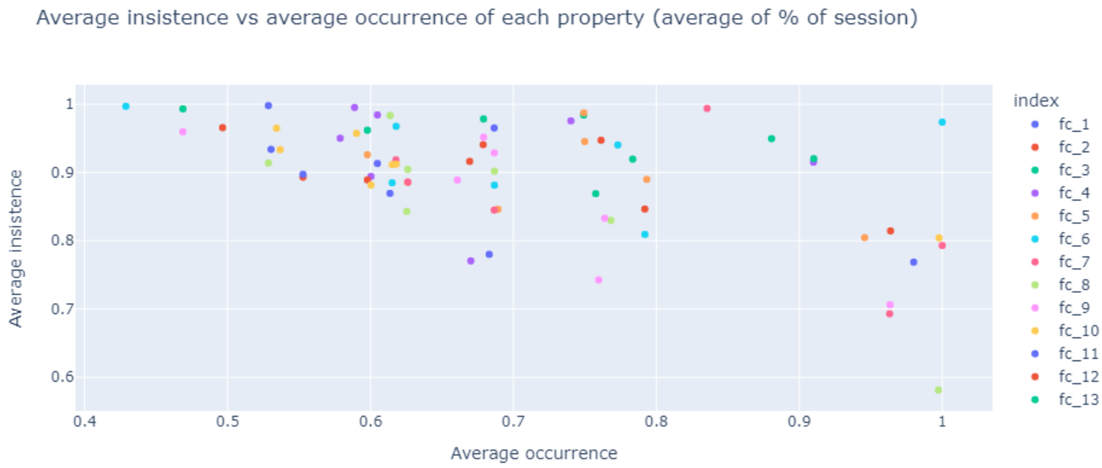


FIGURE 3.11: Average insistence vs average occurrence of each pair of properties.

on average in all session items. The majority of pairs co-occurred in half the items of a two-item or a four-item session; or in all but one item of a three-item or a four-item session. All insistence values are above 99%. Hence, when a pair co-occurs, their value very rarely differs from item to item in a session. For instance, the pair of properties 'fc_47' and 'fc_56' has an average co-occurrence of 1 and an average insistence of 0.99. Hence, this pair, on the average of the sessions it has co-occurred, was present in all items of the sessions and kept their pair of values also persistent.

FIGURE 3.12: Average insistence of co-occurrences vs average co-occurrence of each property.

Taking into account all that was previously mentioned, the properties have a behaviour that seems consistent, valid and that reflects the user's intention.

Chapter 4

Methodology

The proposed framework of this dissertation comprises a Hybrid Model that is the result of the combination of two other models: the Items Model and the Properties Model. The Items Model follows the classical modelling approach to Recommender Systems (RS). The Properties Model represents the proposed modelling approach of this dissertation. Figure 4.1 presents a visual representation of the Hybrid Model. The proposed framework is independent of the algorithm used to obtain the recommendations. Hence, it was evaluated with one baseline algorithm.

This chapter is organised as follows. Section 4.1 explains in detail the choice taken for the baseline algorithm. Section 4.2 and section 4.3 describe, respectively, the Items Model and Properties Model specific algorithm parameters. Section 4.4 explains how the hybridization is performed. Lastly, section 4.5 summarises the whole methodology.

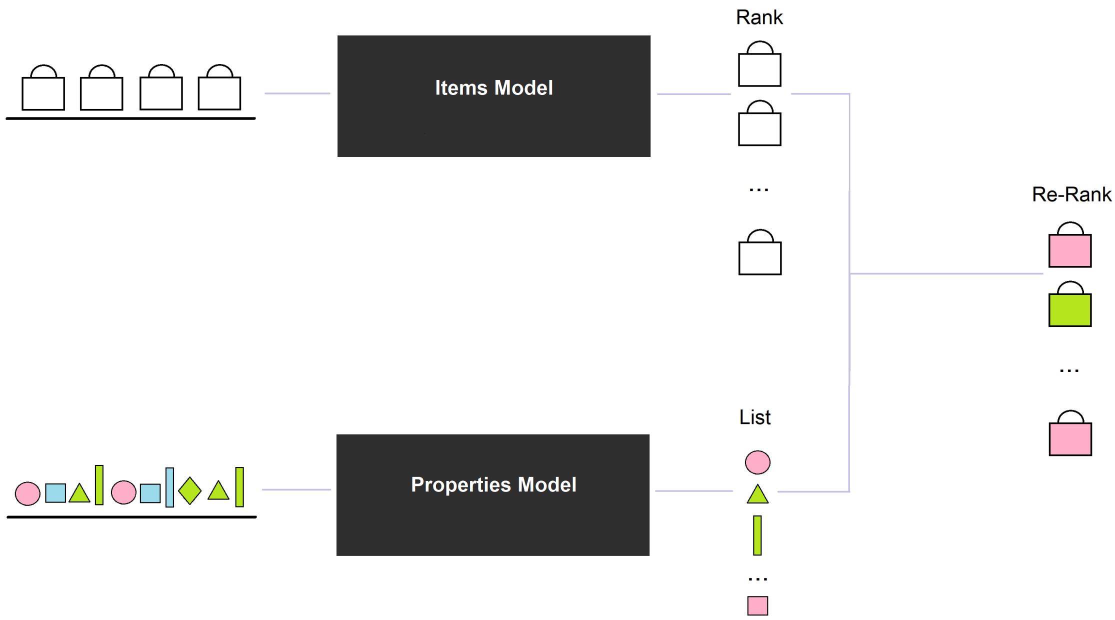


FIGURE 4.1: Symbolic representation of the Hybrid model.

4.1 Baseline algorithm

Given the focus of this dissertation is the study of the properties modelling approach, the baseline algorithm used in the framework was required to be fully known - advantages and issues alike - so its influence would be trackable. Additionally, to prevent unnoticed errors and facilitate result comparison, the baseline algorithm was standardized across all models.

This section is organised as follows. Firstly, section 4.1.1 frames the choice taken on the type of Recommender System used in the framework. Secondly, section 4.1.2 contextualises the choice of the algorithm. Lastly, section 4.1.3 explains the approach taken on the evaluation of the model.

4.1.1 Item-based Collaborative Filtering

The key elements of RS research are Collaborative Filtering (CF) methods, content-based methods and knowledge-based methods [3]. Out of the three, CF is the most well-developed, extensively studied and revised framework [5]. CF can be model-based or neighbourhood-based, as explained in section 2.1.1. A neighbourhood-based CF approach was followed for its simplicity of implementation and the easily explainable recommendations [3].

The neighbourhood-based CF can be user-based or item-based. As mentioned in Ricci et al. [2], the following five criteria are referred to opt between them:

- **Accuracy**: When the number of users greatly exceeds that of the items, implementing item-based methods can yield more accurate recommendations.
- **Efficiency**: When the number of items is inferior to that of the users, implementing item-based methods will use less memory.
- **Stability**: If the list of either users or items is constantly changing, it will heavily delay the computation of the similarity rate between the elements of the list. Therefore, if the catalogue of items does not change, the item-based method is more advantageous.
- **Justifiability**: Items do not have the subjective nature of users. Hence, the recommendations of an item-based method can be easily used as justification compared to the user-based method.
- **Serendipity**: Given the objectivity of items, their similarity is more stable than that among users. Therefore, utilizing a user-based approach is more likely to provide serendipitous recommendations.

The data set used has 16 times more sessions than items and 420 times more sessions than property values. Hence, the item-based approach was adopted, fulfilling the accuracy, efficiency and justifiability criteria.

4.1.2 K Nearest Neighbours

This dissertation used the implemented K Nearest Neighbours (KNN) Collaborative Filtering Recommender on the Elliot framework *. The algorithm generates recommendations based on a few similar items measured by the cosine similarity between vectors, in which each vector corresponds to an item and the vector's M dimensions correspond to sessions (anonymous users) who have purchased that item [21]. The cosine similarity is presented in eq. (4.1) where A_i and B_i are item vectors.

The K Nearest Neighbours (KNN) model is widely used compared to other methods, especially among other neighbourhood-based CF approaches, in both RS and cold start

*The models were run on the Elliot framework - a recommendation framework that centralizes several relevant RS models able to conduct the whole pipeline [20].

settings [5] [2] [22]. Despite the metric simplicity, this is due to the model's efficiency, robustness and interpretability.

$$cosine\_similarity(A, B) = \frac{\sum_{i=1}^{n} \mathcal{A}_i \mathcal{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathcal{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathcal{B}_i)^2}} \tag{4.1}$$

- **The value of k in KNN.** Finding the value of k of KNN is a difficult task.[2] The best approach to find the optimal value of k is found with cross-validation over the train set [23]. However, given the size of the data, alongside the high computation time required and time constraints, k was set to the square root of the number of points in the training data set - an often-used rule of thumb [24].

  Hence, in this study, the KNN Items Model's value of k equals the square root of the number of items in the catalogue. The KNN Properties Model's value of k equals the square root of the number of unique property values.

- **Anonymous sessions interpretation.** In a classical RS setting, the data contains users, items and user-item-specific rates. The sessions of the original data set were all anonymous. To address this issue, each session was interpreted as a different user from the point of view of the KNN algorithm.

4.1.3 Evaluation

The number of relevant recommendations in RS goes up to 20 [2] [1]. An exaggerated number of recommendations can overwhelm the end user, which can increase the likelihood of user dissatisfaction and further decrease the sales at the site [3] [1]. In order to yield a high-quality recommendation list, metrics in which the top-ranked items are given more importance - top-heavy ranking measures - like the Mean Reciprocal Rank (MRR) are used [3]. MRR and Hit Rate were used to measure the performance of the recommender.

MRR indicates at which rank the first relevant item occurs on average [25]. The metric is calculated as presented in eq. (4.2), where N is the number of recommendations.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i} \tag{4.2}$$

The Hit Rate measures the ratio of relevant recommended items to all relevant items. The metric is defined in eq. (4.3), where $|recommended\_relevant|$ is the number of relevant recommended items and $|all\_relevant|$ is the number of all relevant items. In this study, the number of relevant items was only one, the purchase.

$$HitRate = \frac{|recommended\_relevant|}{|all\_relevant|} \tag{4.3}$$

In summary and for all previously mentioned, the baseline algorithm of both models of the proposed framework was the item KNN, with cosine similarity measure. All models output 20 items ranked by the predicted score, which were evaluated with MRR and HR metrics.

4.2 Items Model

The Items Model follows the classical modelling approach to RS. It models the interaction data, *i.e.*, the user's seen items in a session, and predicts the next item score in the session.

The input of this model is the *Items data set* which is composed of three columns: *session_id*, *item_id* and *score*. Figure 4.2 presents session with id 18 of the *Items data set*.

```
Session id = 18
    session_id  item_id  score
0           18  i_18316    1.0
1           18  i_23914    0.0
2           18  i_28086    0.0
3           18  i_11815    0.0
4           18   i_4026    1.0
5           18   i_2507    1.0

The purchased item was :
    session_id  item_id  score
0           18  i_24911    1.0
```

FIGURE 4.2: Seen items and purchase of session with id 18 of the *Items data set*. The session data has binary scores and negative sampling.

The original data was comprised of implicit item ratings *i.e.*, observed or not observed items. Given the nature of the data, the scores were defined as binary:

- score 1 - the item was present in the session.
- score 0 - the item was not present in the session.

However, without further adjustments, all items in each session only would have one value, which is not useful. Therefore, random negative sampling was performed in each session in order to balance the scores. Items that were not seen in the session, were randomly sampled until the same number of positive and negative rates were the same, and inputted in the session.

For the reasons aforementioned in section 4.1.3, the model outputs 20 items ranked by the predicted score.

4.3 Properties Model

The Properties Model follows the proposed modelling approach to RS. It models the content data - item property values - as interaction data. Drawing the parallel between the two approaches, it models the user's seen items' values of properties in a session and predicts the scores of the next batch of properties in the session.

The input of this model is the *Properties data set* which is composed of three columns: *session_id*, *property_value_id* and *score*. Figure 4.3 presents session with id 18 of the *Properties data set* fed to the model. The score is explained next in this section.

Session id = 18

	session_id	property_value_id	score
0	18	5689	0.333333
1	18	1461	1.000000
2	18	55267	0.666667
3	18	55267	0.666667
4	18	50901	0.333333
..	...	...	...
67	18	3793	1.000000
68	18	3793	1.000000
69	18	3793	1.000000
70	18	7275	0.666667
71	18	7275	0.666667

[72 rows x 3 columns]

(A) Session property values

The purchased properties values were :

	session_id	property_value_id	score
0	18	30847	1.0
1	18	4618	1.0
2	18	48326	1.0
3	18	50328	1.0
4	18	32902	1.0
5	18	45559	1.0
6	18	1461	1.0
7	18	56365	1.0
8	18	47690	1.0
9	18	7837	1.0
10	18	17378	1.0
11	18	30482	1.0
12	18	58299	1.0
13	18	69709	1.0
14	18	63861	1.0
15	18	59180	1.0
16	18	55267	1.0
17	18	3793	1.0
18	18	26268	1.0
19	18	6831	1.0
20	18	5605	1.0
21	18	19765	1.0
22	18	46825	1.0
23	18	61706	1.0
24	18	7275	1.0

(B) Purchased property values

FIGURE 4.3: Seen items and purchase of session with id 18 of the *Properties data set*.

4.3.1 Property value score

As mentioned previously, the original data was comprised of implicit item ratings. A weighted importance approach was taken to calculate the scores of each property value in the session. This approach incorporates the relevancy that a property value has in the session. The score is calculated in the following way:

```
property_value_score =
                number_of_times_the_property_value_occurred_in_session /
                     number_of_items_in_session
```

In particular, the score ranges between zero and one, $property\_value\_score \in [0, 1]$.

4.3.2 Model outputs

This model has two outputs. The first output of the model is a list of property values. The length of this list was set equal to the maximum number of property values a catalogue item presented, that is 33. After that, the list is converted to a rank of 20 items - the final output.

There are two issues to take into consideration when performing this conversion: the relevancy of the property values in the list and in the item. In this regard, two methods of calculating an item score based on the recommended property values (RPV) were tested: the simple conversion method and the weighted conversion method. Section 4.3.2.1 and section 4.3.2.2 describes thoroughly these methods, respectively.

4.3.2.1 Simple Converted Score

The Simple Converted Score of item α, $\mathcal{SCS}_\alpha$, is given by eq. (4.4), where the numerator is the sum of the RPV's scores present in item α and the denominator is the sum of all RPV scores. In the numerator, $\mathcal{S}_i$ is the i^{th} RPV score in the P_α list of RPV present in item α.

$$\mathcal{SCS}_\alpha = \frac{\sum_{i \in P_\alpha} \mathcal{S}_i}{\sum_j^{n=33} \mathcal{S}_j} \tag{4.4}$$

The method has the following properties:

1. The score ranges between zero and one, $\mathcal{SCS}_\alpha \in [0, 1]$.
2. The score of an item with zero RPV equals zero.
3. The score of an item with all RPV equals one.

4. The score of an item with only one property value, where that property value is recommended, equals the score of an item with two or more property values, where only one of them is recommended. In summary, the score is independent of the number of property values an item holds.

In summary, the converted item score equals the sum of all scores of the RPV present in the item divided by the sum of all 33 recommended scores.

4.3.2.2 Weighted Converted Score

The Weighted Converted Score of item α, $\mathcal{WCS}_\alpha$, is given by eq. (4.5), where the numerator is the sum of the RPV weighted scores present in item α and the denominator, m, is the number of properties in the item α. All is multiplied by r - the number of property values recommended. In the numerator, $\mathcal{WS}_i$ is the i^{th} RPV score in the P_α list of RPV present in item α.

The Weighted Score of the i^{th} RPV, $\mathcal{WS}$, is given by eq. (4.6), where $\mathcal{S}_i$ is the i^{th} RPV score in the recommended list and the denominator is the sum of all RPV scores.

$$\mathcal{WCS}_\alpha = \frac{\sum_{i \in P_\alpha} \mathcal{WS}_i}{m} r \tag{4.5}$$

$$\mathcal{WS}_i = \frac{\mathcal{S}_i}{\sum\limits_{j=1}^{n=33} \mathcal{S}_j} \tag{4.6}$$

The method has the following properties:

1. The score ranges between zero and one, $\mathcal{WCS}_\alpha \in [0, 1]$.
2. The score of an item with zero RPV equals zero.
3. The score of an item with all RPV equals one.
4. the score of item A, with r number of recommended properties and t_A total number of properties, is greater than item B, with r number of recommended properties and a t_B, higher than t_A total number of properties. In summary, the score benefits an item with a bigger ratio of RPV and the total number of property values in an item.

In summary, the converted item weighted score equals the sum of the weighted scores divided by the number of properties in the item multiplied by the number of properties recommended. The weighted score is calculated by the score of the RPV divided by the sum of all recommended scores.

4.3.3 Recommendation space

Having a list of RPV and a method to rank items based on those there was a choice of which selection of items to apply the conversion method to. The selection of items could be the full catalogue of items or a smaller selection of those. Both selections of items were implemented, with the smaller selection being the Items Model's recommended items.

Using the Items Model's recommended items in the Properties Model immediately impacts the Hybrid Model. The final effect is that of a re-ranking method. The main advantage is the shorter list of items to calculate the conversion score. Hence, requiring significantly less computation power and time. The main drawback is that it eliminates the possibility of new items being introduced in the recommendation rank based on the Properties Model discovery. In opposition, using all the items in the catalogue allows the property model to reach its full individual potential by removing the bias of the Items Model. The main advantage is the higher likelihood of new items being recommended. The main drawback is the huge computation power required.

4.4 Hybrid Model

The proposed Hybrid Model is a combination of the Items Model and the Properties Model. It receives as input the rank of 20 items with the respective scores of both models and outputs a rank of 20 items of its own. The hybridisation follows a linear function given by eq. (4.7). $\mathcal{H}_\alpha$ is the Hybrid Model score of item α, $\mathcal{I}_\alpha$ and $\mathcal{P}_\alpha$ are the item's score of the Items Model and Properties Model, respectively. The constant ρ determines each model weight on the Hybrid Model score.

$$\mathcal{H}_\alpha = (1-\rho)\mathcal{I}_\alpha + \rho\mathcal{P}_\alpha \tag{4.7}$$

In particular,

- when $\rho = 0$, the Hybrid Model assumes the Items Model prediction.
- when $\rho = 1$, it assumes the Properties Model prediction.

- In cases where the list of items recommended by either model is different, the scores missing are inputted as 0.

4.5 Summary

In summary, the baseline algorithm of both the Items Model and the Properties Model was the item KNN, with cosine similarity measure. All models output 20 items ranked by the predicted score, which were evaluated with MRR and HR metrics.

The Items Model handles the user-item interaction data and predicts the next item score in the session. The items scores are binary and negative sampling was performed to balance them.

The Properties Model handles the user-property-values interaction data. The scores of the property values were calculated with a session-weighted importance approach. The model has two outputs. The first is a list of property values. The second and final output is a rank of items based on the first output. The score of an item based on the RPV can be calculated in two ways: the simple conversion method and the weighted conversion method. The model's recommendation space was tested in two sizes: the full catalogue of items and a smaller selection of those. In this study, the smaller selection was the Items Model's recommended items.

The Hybrid Model ranking score is given by a weighted linear combination of the Items Model score and the Properties Model score.

Chapter 5

Results

The framework was tested with a short selection of items to recommend versus the complete catalogue, in combination with the simple conversion method versus the weighted conversion method. The Hybrid model parameter ρ was tested with 0.75, 0.50, 0.25, 0.10, 0.05, 0.01 and 0.001.

The study of different approaches to calculate the item-translating scores of the Properties Model revealed the weighted conversion method to be the best one. However, in the Hybrid models, as the influence of the Items Model increases (*i.e.*, as the ρ decreases), the simple conversion method outperforms the weighted conversion method.

The study of the size of the recommendation space revealed that a smaller selection performed better. The Items Model influence was a major one in the Hybrid Model's performance. Moreover, the reordering of the recommendation space, based on the recommendations of the proposed Properties Model, did not yield a better-performing model than the traditional model.

This chapter is organized as follows. Section 5.1 goes through the main limitations of this dissertation. Section 5.2 throughoutly presents the interpretation of the results. Lastly, section 5.3 discuss the results obtained.

5.1 Limitations

The focus of this study was the test of a brand new approach to the Cold Start (CS) in the Recommender System (RS) setting. Therefore, the results of this study were not expected to be outstanding by themselves. The focus was on the comparison between the implemented models.

Moreover, due to memory and computation power restraints, it was not possible to run the experiments with the whole data set at once. Therefore, the data set was split into four files. Each file is composed of an equal number of sessions - each has a total of 95011 sessions.

5.2 Interpretation of the results

Table 5.1 and fig. 5.1 present the average Mean Reciprocal Rank (MRR) with a cutoff of 20, over the four files, of the Items, Properties and Hybrids models. The Items Model is independent of the variables studied. Hence, its value is constant across all columns. The model's average MRR of 8.81×10^{-2} means the model correctly predicts the item purchased, on average, in the 11$^{\text{th}}$ place.

Considering the shorter selection of the items to recommend (in this study, it is the Items Model recommended list of items) the Properties Model performed best with the weighted conversion method. The model's average MRR of 4.67×10^{-2} means the model correctly predicts the item purchased, on average, in the 21$^{\text{st}}$ place. It was an increase of 68% performance from the simple conversion method. The worst Hybrid Model was the one with $\rho = 0.75$ and the simple conversion method. Its average MRR 3.38×10^{-2} is an improvement of 22% from the Properties Model with the same conversion method. The best Hybrid Model was the one with $\rho = 0.001$. It achieved an average MRR of 8.79×10^{-2} with both conversion methods. This is an improvement of 217% and 88% from the Properties Model with the simple and weighted conversion methods, respectively. However, a decrease in performance of 0.23% from the Items Model.

Following in turn the analysis of the full catalogue recommendation space, the Properties Model performed best with the weighted conversion method. It was an increase in performance of 4000% from the simple conversion method. The model's average MRR of 1.74×10^{-3} means the model correctly predicts the item purchased, on average, in the 574$^{\text{th}}$ place. Nonetheless, it performed poorly while using the full catalogue recommendation space in comparison with the shorter recommendation space. Both the worst and best Hybrid Models used the simple conversion method. The worst hybrid was $\rho = 0.75$ with average MRR 5.03×10^{-5}, while the best one was $\rho = 0.001$ with average MRR of 7.52×10^{-2}. The results of the former model reflect an improvement in performance of 18% from the Properties Model with the same conversion method. The results of the latter model mean the model correctly predicts the item purchased, on average, in the 13$^{\text{th}}$

place. This is a huge improvement in performance from the worse-performing Properties Model. On top of that, the results reflect a decrease in performance of 14% from the best-performing Hybrid Model in the shorter recommendation space and of 15% from the Items Model.

TABLE 5.1: Average MRR @20, over the four data files, of each model.

Model	Items Model rec. items: Simple conversion	Items Model rec. items: Weighted conversion	Full catalogue: Simple conversion	Full catalogue: Weighted conversion
Items	8.81×10^{-2} $\pm 8.65 \times 10^{-4}$	8.81×10^{-2} $\pm 8.65 \times 10^{-4}$	8.81×10^{-2} $\pm 8.62 \times 10^{-4}$	8.81×10^{-2} $\pm 8.62 \times 10^{-4}$
Properties	2.77×10^{-2} $\pm 3.11 \times 10^{-4}$	4.67×10^{-2} $\pm 2.55 \times 10^{-4}$	4.24×10^{-5} $\pm 6.60 \times 10^{-6}$	1.74×10^{-3} $\pm 4.57 \times 10^{-5}$
Hybrid ($\rho = 0.75$)	3.38×10^{-2} $\pm 5.56 \times 10^{-4}$	4.99×10^{-2} $\pm 2.54 \times 10^{-4}$	5.03×10^{-5} $\pm 7.63 \times 10^{-6}$	4.09×10^{-3} $\pm 1.73 \times 10^{-4}$
Hybrid ($\rho = 0.50$)	5.13×10^{-2} $\pm 6.38 \times 10^{-4}$	5.36×10^{-2} $\pm 2.96 \times 10^{-4}$	5.19×10^{-3} $\pm 2.33 \times 10^{-4}$	4.16×10^{-3} $\pm 1.94 \times 10^{-4}$
Hybrid ($\rho = 0.25$)	7.58×10^{-2} $\pm 7.74 \times 10^{-4}$	6.25×10^{-2} $\pm 4.48 \times 10^{-4}$	5.95×10^{-2} $\pm 7.37 \times 10^{-4}$	5.89×10^{-3} $\pm 2.77 \times 10^{-4}$
Hybrid ($\rho = 0.10$)	8.48×10^{-2} $\pm 9.65 \times 10^{-4}$	7.50×10^{-2} $\pm 6.16 \times 10^{-4}$	7.51×10^{-2} $\pm 8.67 \times 10^{-4}$	3.84×10^{-2} $\pm 5.15 \times 10^{-4}$
Hybrid ($\rho = 0.05$)	8.66×10^{-2} $\pm 9.36 \times 10^{-4}$	8.20×10^{-2} $\pm 6.94 \times 10^{-4}$	7.52×10^{-2} $\pm 8.62 \times 10^{-4}$	6.54×10^{-2} $\pm 7.62 \times 10^{-4}$
Hybrid ($\rho = 0.01$)	8.77×10^{-2} $\pm 8.79 \times 10^{-4}$	8.72×10^{-2} $\pm 8.40 \times 10^{-4}$	7.52×10^{-2} $\pm 8.61 \times 10^{-4}$	6.88×10^{-2} $\pm 8.66 \times 10^{-4}$
Hybrid ($\rho = 0.001$)	8.79×10^{-2} $\pm 8.89 \times 10^{-4}$	8.79×10^{-2} $\pm 8.98 \times 10^{-4}$	7.52×10^{-2} $\pm 8.60 \times 10^{-4}$	6.88×10^{-2} $\pm 8.64 \times 10^{-4}$

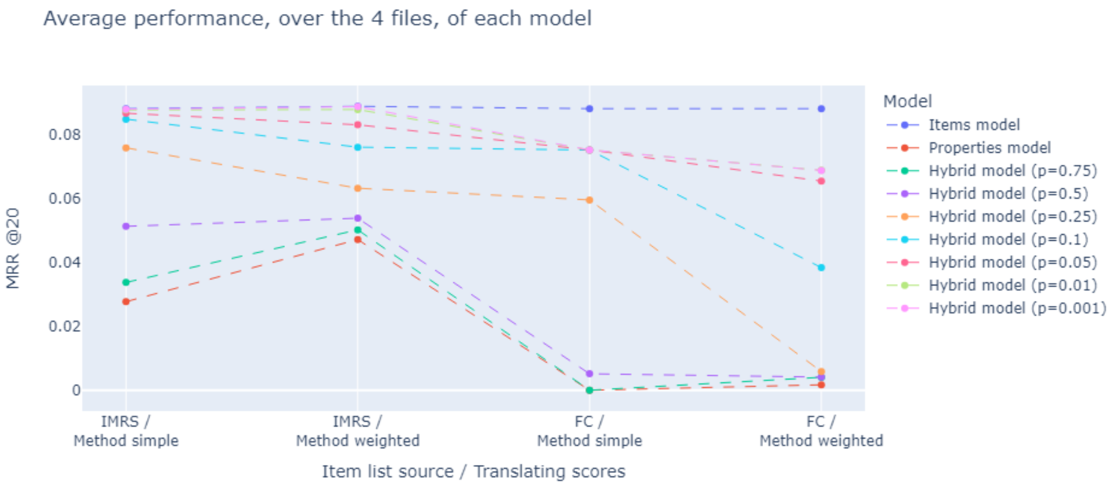


FIGURE 5.1: Average MRR @20, over the four data files, of each model. The study of the selection recommendation space is between the Item Model Recommendation Space (IMRS) and the Full Catalog (FC). The study of different approaches to calculate the item-translating scores of the Properties Model was between the Simple Conversion method and the Weighted Conversion method.

Table 5.2 presents the average Hit Ratio (HR), of the studied models with a cutoff of 20, over the four files. In the shorter recommendation space, given the recommended items did not change between the models, consequently neither did the hit ratio in the respective columns. Hence, all of these models had a hit ratio of 2.57×10^{-1}. Considering the full catalogue of items recommendation space, the best Properties Model - the one using the weighted conversion score - had a hit ratio of 8.49×10^{-3}, and the best Hybrid Model - the one with $\rho = 0.001$ and using the simple conversion score - had a hit ratio of 2.16×10^{-1}. Therefore, the models with a high influence of the Items Model have a bigger hit ratio.

TABLE 5.2: Average hit ratio @20, over the 4 files, of each model.

	Items model rec. space		**Full catalogue**	
Model	**Simple conversion**	**Weighted conversion**	**Simple conversion**	**Weighted conversion**
Items	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$
Properties	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.53×10^{-4} $\pm 5.21 \times 10^{-5}$	8.49×10^{-3} $\pm 2.90 \times 10^{-4}$
Hybrid ($\rho = 0.75$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.63×10^{-4} $\pm 4.99 \times 10^{-5}$	8.40×10^{-3} $\pm 3.05 \times 10^{-4}$
Hybrid ($\rho = 0.50$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	8.02×10^{-3} $\pm 2.87 \times 10^{-4}$	8.40×10^{-3} $\pm 3.03 \times 10^{-4}$
Hybrid ($\rho = 0.25$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	1.22×10^{-1} $\pm 1.63 \times 10^{-3}$	1.05×10^{-2} $\pm 4.66 \times 10^{-4}$
Hybrid ($\rho = 0.10$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.15×10^{-1} $\pm 1.22 \times 10^{-3}$	6.67×10^{-2} $\pm 1.14 \times 10^{-3}$
Hybrid ($\rho = 0.05$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.16×10^{-1} $\pm 1.23 \times 10^{-3}$	1.72×10^{-1} $\pm 1.64 \times 10^{-3}$
Hybrid ($\rho = 0.01$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.16×10^{-1} $\pm 1.22 \times 10^{-3}$	1.96×10^{-1} $\pm 1.26 \times 10^{-3}$
Hybrid ($\rho = 0.001$)	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.57×10^{-1} $\pm 1.13 \times 10^{-3}$	2.16×10^{-1} $\pm 1.21 \times 10^{-3}$	1.96×10^{-1} $\pm 1.21 \times 10^{-3}$

The Properties Model has two outputs: the property-values list and the items rank. The analysis made above was regarding the rank of the items. The following analysis is regarding the unordered list of property values. Hence, it is not comparable with the previously presented results. Nonetheless, it provides insight into the proposed modelling approach.

The average precision and average recall metrics were calculated over the four data files. Taking into account that an item can have between two and 33 property values, the number of the purchase item's property values is unpredictable. Hence, both values were considered as cutoff. Considering the maximum number of property values as the cutoff, the average precision was of $8.80 \times 10^{-2} \pm 4.46 \times 10^{-4}$ and the average recall was of $1.33 \times 10^{-1} \pm 6.32 \times 10^{-4}$. In turn, considering the minimum number of property values as the cutoff, the average precision was of $2.27 \times 10^{-1} \pm 1.70 \times 10^{-3}$ and the average recall was of $2.12 \times 10^{-2} \pm 1.46 \times 10^{-4}$. Therefore, the model is able to learn some of the data dependencies.

In summary, the Items Model was the best-performing model among all and the Properties Model performed the worst. The Hybrid Model performed best with a small ρ, *i.e.*, with a high influence from the Items Model.

5.3 Discussion

The significant difference in the performance between the Items and Properties models in the two settings, alongside the change in the hit ratio, reveals that both models favoured different items to feed the Hybrid Model recommendation space. However, the Hybrid Model, which follows a weighted linear function, never outperformed the traditional model.

Therefore, the presented results were unexpected. Other works that have used the Dressipi data [16] [17] noted a positive correlation between the purchases and the item properties. Hence, the use of the item properties should add value to the RS. The results indicate that the Properties Model had a negative influence on the Hybrid Model. Consequently, and in conclusion, the proposed modelling approach to explore content information does not add value to the proposed framework, without further adjustments.

Chapter 6

Conclusion

This work studies the modelling of item property content data using short anonymous sessions in the cold start fashion recommender systems setting. Additionally, it analyses the impact of the recommendations based on user-item property interaction data in a hybrid model. The study found that the proposed Properties Model performed best with the weighted conversion method. However, the performance was significantly lower than the traditional Items Model. On top of that, the Properties Model had a negative influence on the Hybrid Model. The Hybrid Model performed best when the influence of the Items Model was higher than the Properties Model - reflected in better results when the ρ value was small and the recommendation space was restricted to that of the Items Model recommendations.

The main contributions of this dissertation are the following:

- creation of a novel content-based modelling approach to Recommender Systems.
- implementation of a hybrid model that combines the traditional modelling approach with the proposed one.

One of the major challenges of this work was the size of the data alongside the computation power limitations, with the run time being in the order of weeks. Another major challenge of the study was how to take into consideration the relevancy of the property values in the recommendation list and in the item. In addition, the proposed framework was built on research-backed-up choices. Nevertheless, these would benefit from a more in-depth study, which suggests some future work.

The main advantages of the proposed framework are the independence of the algorithm used and the domain to which it is applied. Additionally, it can ease the addition of new items to the catalogue as long as they share the properties with the existing items.

As for future work, the authors suggest the property values score to be binary and to perform negative sampling. The implemented weighted importance approach (mentioned in section 4.3) is highly sensitive to how frequent the property value is in the session. A binary score can make the task of score prediction easier, while the negative sampling can increase the information given to the model - highly beneficial in a cold start setting.

The authors observed that the Properties Model might be boosting items with a larger percentage of (presumably relevant) properties that did not appear in the session. However, the purchased item has a significant number of overlapping properties with items seen by the user. Therefore, the recommendation might be highly serendipitous and different from the purchase, which could explain its negative impact. The authors suggest a different hybridization approach: an alternating recommendation list. Hence, the Hybrid Model's recommendations would be comprised of the recommendations of the Items Model alternated with the recommendations of the Properties Model.

Bibliography

[1] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus, "Understanding choice overload in recommender systems," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 63–70. [Cited on pages 3 and 22.]

[2] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Springer New York, NY, 2015. [Cited on pages 3, 4, 5, 6, 21, and 22.]

[3] C. C. Aggarwal, *Recommender Systems - The Textbook*. Springer Cham, 2016. [Cited on pages 4, 5, 20, and 22.]

[4] Y. Deldjoo, F. Nazary, A. Ramisa, J. Mcauley, G. Pellegrini, A. Bellogin, and T. Di Noia, "A review of modern fashion recommender systems," *arXiv preprint arXiv:2202.02757*, 2022. [Cited on pages 5 and 7.]

[5] N. A. Abdullah, R. A. Rasheed, M. H. N. M. Nasir, and M. M. Rahman, "Eliciting auxiliary information for cold start user recommendation: A survey," *Applied Sciences*, vol. 11, no. 20, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/20/9608 [Cited on pages 5, 20, and 22.]

[6] Statista. Fashion ecommerce report. [Online]. Available: https://www.statista.com/study/38340/ecommerce-report-fashion/ [Cited on page 6.]

[7] L. Bigon, G. Cassani, C. Greco, L. Lacasa, M. Pavoni, A. Polonioli, and J. Tagliabue, "Prediction is very hard, especially about conversion. predicting user purchases from clickstream data in fashion e-commerce," *arXiv preprint arXiv:1907.00400*, 2019. [Cited on page 6.]

[8] (2022) 16th acm conference on recommender systems. [Online]. Available: https://fashionxrecsys.github.io/fashionxrecsys-2022/ [Cited on page 7.]

[9] Y. Ding, Y. Ma, W. K. Wong, and T.-S. Chua, "Modeling instant user intent and content-level transition for sequential fashion recommendation," *IEEE Transactions on Multimedia*, vol. 24, pp. 2687–2700, 2021. [Cited on page 7.]

[10] P. Tangseng and T. Okatani, "Toward explainable fashion recommendation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2153–2162. [Cited on page 7.]

[11] M. Al-Rawi and J. Beel, "Probabilistic color modelling of clothing items," in *Recommender Systems in Fashion and Retail*. Springer, 2021, pp. 21–40. [Cited on page 7.]

[12] D. Goncalves, L. Liu, J. Sá, T. Otto, A. Magalhães, and P. Brochado, "The importance of brand affinity in luxury fashion recommendations," in *Recommender Systems in Fashion and Retail*. Springer, 2021, pp. 3–19. [Cited on page 8.]

[13] C. Yan, Y. Chen, and L. Zhou, "Differentiated fashion recommendation using knowledge graph and data augmentation," *Ieee Access*, vol. 7, pp. 102 239–102 248, 2019. [Cited on page 8.]

[14] C. Bracher, S. Heinz, and R. Vollgraf, "Fashion dna: merging content and sales data for recommendation and article mapping," *arXiv preprint arXiv:1609.02489*, 2016. [Cited on page 8.]

[15] H. T. Nguyen, T. Almenningen, M. Havig, H. Schistad, A. Kofod-Petersen, H. Langseth, and H. Ramampiaro, "Learning to rank for personalised fashion recommender systems via implicit feedback," in *Mining Intelligence and Knowledge Exploration: Second International Conference, MIKE 2014, Cork, Ireland, December 10-12, 2014. Proceedings*. Springer, 2014, pp. 51–61. [Cited on page 8.]

[16] Y. Lu, Z. Gao, Z. Cheng, J. Sun, B. Brown, G. Yu, A. Wong, F. Pérez, and M. Volkovs, "Session-based recommendation with transformers," in *Proceedings of the Recommender Systems Challenge 2022*, 2022, pp. 29–33. [Cited on pages 8 and 33.]

[17] B. Schifferer, J. Liu, S. Rabhi, G. Titericz, C. Deotte, G. De Souza P. Moreira, R. Ak, and K. Onodera, "A diverse models ensemble for fashion session-based recommendation," in *Proceedings of the Recommender Systems Challenge 2022*, 2022, pp. 10–17. [Cited on pages 8 and 33.]

[18] N. Landia, R. Mcalister, D. North, S. Kalloori, A. Srivastava, and B. Ferwerda, "Recsys challenge 2022 dataset: Dressipi 1m fashion sessions," in *Proceedings of the Recommender Systems Challenge 2022*, ser. RecSysChallenge '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1–3. [Online]. Available: https://doi.org/10.1145/3556702.3556779 [Cited on pages ix, 9, 10, and 13.]

[19] (2022) Recsys challenge 2022. [Online]. Available: http://www.recsyschallenge.com/2022/ [Cited on page 10.]

[20] V. W. Anelli, A. Bellogín, A. Ferrara, D. Malitesta, F. A. Merra, C. Pomo, F. M. Donini, and T. D. Noia, "Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation," in *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai, Eds. ACM, 2021, pp. 2405–2414. [Online]. Available: https://doi.org/10.1145/3404835.3463245 [Cited on page 21.]

[21] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003. [Cited on page 21.]

[22] H. Guo, R. Tang, Y. Ye, F. Liu, and Y. Zhang, "A novel knn approach for session-based recommendation," in *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part II 23*. Springer, 2019, pp. 381–393. [Cited on page 22.]

[23] M. Azadkia, "Optimal choice of k for k-nearest neighbor regression," *arXiv preprint arXiv:1909.05495*, 2019. [Cited on page 22.]

[24] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, "Solving the problem of the k parameter in the knn classifier using an ensemble learning approach," *arXiv preprint arXiv:1409.0919*, 2014. [Cited on page 22.]

[25] N. Craswell, *Mean Reciprocal Rank*. Boston, MA: Springer US, 2009, pp. 1703–1703. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_488 [Cited on page 22.]