How To Transfer? An Experimental Model for Combining Transfer Learning Techniques in Cross-Domain Recommender Systems

Master's Thesis

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

ICT in Business

Leiden University

by

Famke Nouwens

s3318508

Supervisor:
Dr. Marc Hilbert

Co-Supervisor:
Prof. Dr. Joost Visser

Company Supervisor
Koen Zandvliet

August 22, 2023

Abstract

Addressing data challenges, like data sparsity and the cold-start issue, remains pivotal for enhancing recommender systems. Cross-domain recommendation emerges as a solution to combat these data challenges. While contemporary single-target Cross-Domain Recommendation (CDR) methods concentrate on various aspects of transfer learning, there exists a research gap concerning the fusion of distinct techniques. This study centers on integrating content-based and rating pattern-based transfer strategies to gauge their potential impact on recommendation accuracy. Starting with the DARec model developed by Yuan et al. (2019), this framework is extended with content-based transfer based on text information. The resulting hybrid model is evaluated across two distinct domain sets, including one featured in the original DARec model. The experimental investigation unveils a small to adverse impact on recommendation accuracy, providing novel insights into the combination of techniques of cross-domain recommendation.

Contents

1 Introduction

In the era of abundant information and rapid digitisation, recommender systems (RS) have become indispensable tools that shape online experiences and influence users' decisions. These systems play a pivotal role in filtering vast amounts of content, products, and services, tailoring personalised recommendations to each individual user. From e-commerce platforms to entertainment streaming services, recommendation algorithms are ubiquitous in modern-day digital ecosystems. Notable examples of recommender systems include those used by online streaming platforms like Netflix[1] and Disney+[2], as well as online webshops such as Amazon[3] and Bol.com[4]. These services entice users to continue watching or purchasing by suggesting films, series, or products based on their previous interactions or current popularity.

1.1 Problem Statement

The development of accurate and effective recommender systems encounters challenges, with data scarcity emerging as a prominent concern. In many domains, the insufficiency of relevant user preferences and interactions hampers the training of robust recommender models. Recommender systems encounter two distinct data-related challenges: the *cold-start problem* and *data sparsity*. The cold-start problem emerges when novel users or items are introduced, rendering accurate predictions difficult for the recommender algorithm. Data sparsity, on the other hand, arises when only a small portion of the total items in a database have received user ratings (Isinkaye et al., 2015). To address this issue, the concept of cross-domain recommendation (CDR) has emerged as an intriguing solution. CDR employs transfer learning to share knowledge between domains, aiming to overcome data sparsity and the cold-start problem, ultimately enhancing recommendation performance across diverse contexts.

It is proven that transfer learning works and has many benefits, as will be further explained in Section 3. Within the domain of cross-domain recommendation, there are several approaches, including single-target CDR. In this research, a novel and advanced approach is taken by integrating two specific transfer learning methods: content-based recommendation and rating pattern-based recommendation. By merging content-based recommendation, which utilises additional item attributes and textual descriptions for personalisation (Berkovsky et al., 2007), with rating pattern-based recommendation, which identifies user behaviour patterns across domains (Li et al., 2009), the research endeavours to develop an intelligent, adaptable recommender system tailored to specific requirements

[1] www.netflix.com
[2] www.disneyplus.com
[3] www.amazon.com
[4] www.bol.com

even when little data is available. The objective is to explore the synergy between these approaches, aiming to improve the accuracy of cross-domain recommendations. Through empirical evaluations, the compatibility of these two techniques is investigated.

The research commenced by selecting a cutting-edge rating pattern-based CDR model as its foundation. The choice of the "*DARec*" model (Yuan et al., 2019) as the starting point is substantiated by its exceptional performance within the rating pattern-based CDR domain. Section 4 illustrates the DARec model architecture in detail.

DARec has consistently surpassed numerous other existing CDR models, rendering it a significant and fitting selection as benchmark for this study. Moreover, the research itself recommends investigating the combination of DARec with content-based transfer, which aligns with the direction the research aims to pursue. Previous research studies also support that incorporating associated metadata can significantly enhance recommendation accuracy (Gogna & Majumdar, 2015; Zhao et al., 2016). Thus by building upon the accomplishments of this well-performing model, the thesis proceeds to explore the potential augmentation of recommendation accuracy and effectiveness through the fusion of content-based and rating pattern-based transfers.

1.2 Research Question

Considering the recurrent data challenges faced by recommender systems, the thesis examines the potential impact of integrating content-based and rating pattern-based transfer learning methods on enhancing recommendation accuracy. The assumption is made that combining two transfer learning techniques will positively influence the recommendation accuracy, which forms the hypothesis that is tested. The ensuing research question is articulated as follows:

Research Question (RQ): *To what extent does the combination of content-based and rating pattern-based transfer learning strategies positively influence the recommendation accuracy of cross-domain recommendation systems?*

1.3 Thesis Outline

The structure of the thesis is organised as follows. In the initial section, the significance and relevance of the research to businesses are expounded. Next, an in-depth exploration of various types of recommender systems is presented. The literature review concludes by examining diverse techniques adopted in cross-domain recommendation. The subsequent section elabourates on the DARec model and the combination of DARec and content model employed within the study. Moreover, the research outlines the utilised dataset and explains the steps taken for data pre-processing. The ensuing section encompasses the executed experiments and provides visual representations of the outcomes. Finally, the thesis culminates with a comprehensive summarisation and discussion, followed by an exploration of potential future directions.

2 Business Case

In today's business landscape, companies, specifically e-commerce companies, face the formidable challenge of captivating customers effectively. The following section delves into the compelling rationale behind the need for businesses to utilise robust recommender systems. It emphasises the profound influence of personalised recommendations on revenue and customer engagement, illustrating their pivotal role in navigating the modern business landscape.

2.1 Business Significance

In the face of expanding volumes of available information, businesses encounter challenges in effectively reaching their customers. Once a customer engages with a business, there exists a narrow timeframe before their attention shifts elsewhere. An insightful study by Accenture Interactive expands on this; revealing that nearly 40% of customers opt to exit retail websites when confronted with an overwhelming plethora of choices (Accenture, 2016). This finding highlights the notion that an excess of options can lead to decision fatigue, discouraging further interaction. Therefore, it is imperative to seize and maintain the user's interest within the initial moments of their digital interaction (Agarwal, 2021). This accentuates the need for personalisation, conveying information in manners aligned to individual preferences.

In this regard, recommender systems play a vital role in providing that personalisation, effectively sustaining customer engagement with the business. Companies like Amazon and Spotify use recommendations to generate over 35% and 25% of their revenue, respectively (Cooper, 2018; MacKenzie et al., 2013), and in general over 26 % of revenue from e-commerce sites is driven by personalised product recommendations (Salesforce, 2017). This number has increased to 31% in 2023, according to a study conducted by Barilliance (Serrano, 2023).

Figure 1 illustrates the impact of personalised product recommendations on conversion rates, based on a study by Barilliance (Serrano, 2023). Conversion rate refers to the percentage of users who complete a desired action, such as making a purchase in e-commerce, out of the total number of users who engage with the platform. Evidently, the inclusion of personalised recommendations leads to a significant surge in conversion rate (rising from 1.02% to 3.96%), accentuating its significance.

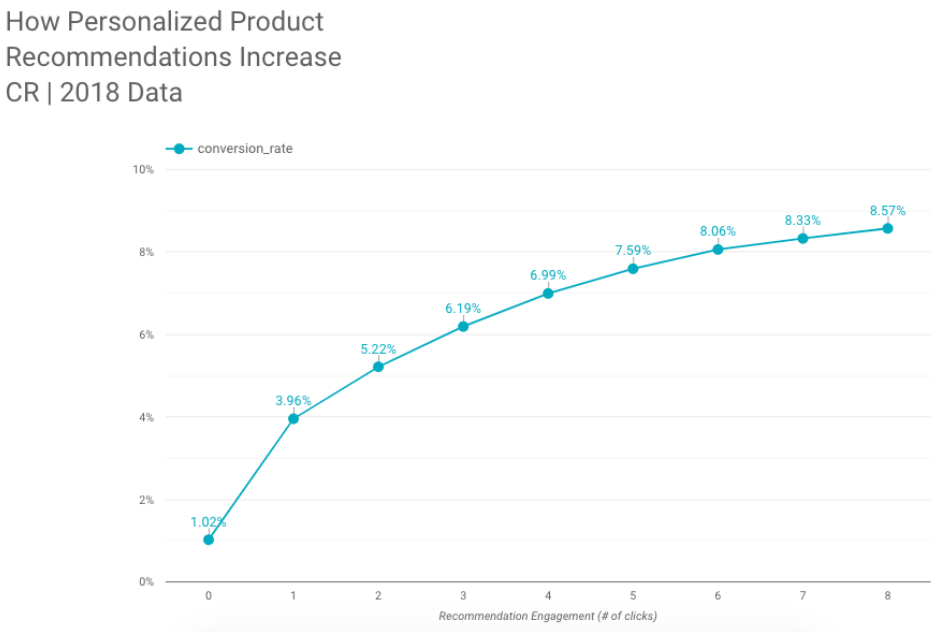

Figure 1: Impact of personalised recommendation on consumers' conversion rate (Serrano, 2023)

Similarly, a study conducted by SalesForce illustrated that personal recommendations amplify consumers' propensity to finalise a purchase by a factor of 4.5 (Salesforce, 2017). Figure 2 reveals the impact of personalised recommendation across devices.

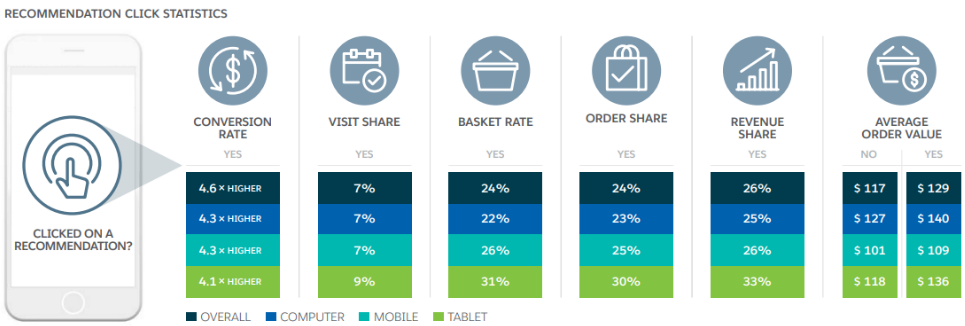

Figure 2: Impact of personalised recommendation on consumers' behaviour (Salesforce, 2017)

Thus, recommender systems play a crucial role in establishing a personalised link between consumers and businesses, encouraging shopping behaviour (Agarwal, 2021). Moreover, these systems simplify the purchasing process by narrowing down the search to products that are most likely relevant (Accenture, 2016).

These findings emphasise the crucial role of recommender systems in businesses and especially e-commerce, a concept many companies aim to adopt. However, not all businesses possess the necessary expertise to implement them effectively. As a result, in today's corporate landscape, such enterprises often turn to IT solution providers that specialize in serving clients without extensive IT knowledge. Within this environment, the challenge emerges when certain clients express a desire for recommender systems but struggle with a shortage of data needed to train robust algorithms. The reality is that conjuring additional data is beyond their reach, leaving them constrained by the confines of their modest dataset. The central focus of the thesis lies on a transfer learning solution: the integration of the client's exclusive data with either an existing dataset from the IT company or leveraging a pre-trained model from the IT company, or alternatively, tapping into publicly available repositories.

The utilisation of transfer learning aims to enhance the efficiency of recommender systems, which can bring forth numerous noteworthy advantages for companies, exerting a positive influence on their overall operational performance and revenue generation (X. Zhang & Wang, 2005). This is particularly relevant in the context of managing the overwhelming amount of information. Additionally, as mentioned before, a proficient recommender system that offers accurate and pertinent recommendations to customers or users tends to result in heightened customer satisfaction and increased revenues (Agarwal, 2021; Cooper, 2018; Jiang et al., 2010). Beyond this, there are additional merits, including the ability to counteract shopper reluctance to embrace new products and the potential to introduce customers to novel product categories (Dias et al., 2008).

Such a system aligns with a core business objective of recommender systems, which is to effectively guide users through decision-making processes by providing tailored and valuable suggestions.

2.2 Legal implications

The realm of data sharing for research or training purposes is often closely connected with complex issues pertaining to privacy and intellectual property rights. When datasets (or models trained on a specific dataset) are shared, particularly across different organisations or domains, concerns arise regarding the potential disclosure of sensitive information. Additionally, the ownership and rights associated with the shared data can lead to legal challenges, potentially affecting the control and usage of the data or model beyond its original scope. While these concerns are valid and significant, it is important to note that addressing the legal and intellectual property aspects of data sharing falls beyond the scope of this research. Instead, the focus here is on exploring innovative approaches to enhancing cross-domain recommendation systems through the integration of transfer learning techniques.

3 Literature Review

The literature review begins with presenting a comprehensive overview of existing recommendation techniques. Following this, attention turns towards the examination of contemporary state-of-the-art cross-domain recommendation methods. The exploration encompasses illustrating the distinct categories of transfer learning and rationalising the emphasis on the rating pattern-based approach within CDR. Subsequently, an in-depth analysis delves into the varied content-based cross-domain recommendation techniques, including their strengths and limitations. Concluding the literature review, the rationale for selecting the DARec model as the initial framework for this investigation is discussed.

3.1 Types of recommendation techniques

As mentioned in the introduction (Section 1), recommender systems try to forecast user preferences for items that have not yet been viewed and attempt to get the user to buy them (Bobadilla et al., 2013). Recommender systems can use *implicit* or *explicit* manners for finding the information on users to make recommendations (Gope and Jain, 2017; Koren et al., 2011). Implicit information is considered data that implicitly shows user preferences, such as buying history and demographics, whereas explicit information is data that was obtained for this specific purpose, such as ratings or questionnaire results. The emphasis of the thesis lies on recommender systems employing *explicit* information.

Recommender systems are algorithms that provide recommendations based on input data. The nature of the input data varies depending on the type of recommender system used. Within the scope of the thesis, recommender systems are categorised into two distinct groups: non-personalised and personalised. Non-personalised recommender systems offer generic recommendations without considering individual user preferences, while personalised recommender systems tailor recommendations to the specific interests and characteristics of individual users or items.

3.1.1 Non-personalised recommender systems

The most naive RS is a non-personalised recommender system. The non-personalised system offers recommendations solely based on information unrelated to the user. It delivers identical suggestions to all users, irrespective of demographics, preferences, or other individual details. Non-personalised recommender systems offer valuable advantages as they can be presented to users even with limited knowledge about their preferences. Examples of non-personalised recommenders are algorithms that simply take the N-most popular items to recommend to a user, like the homepages of grocery shop websites that display their top ten products (Ricci et al., 2010).

A non-personalised recommender system does not mean that it provides bad recommendations. Some argue that non-personalised recommendations should only be used until the system gathers enough information to provide more personalised suggestions. However, it is essential to recognise that humans are fundamentally social beings, and love to know (and potentially follow) what is currently most popular, if only to discern what might not align with their interests (Falk, 2019).

3.1.2 Personalised recommender systems

Personalised recommender systems are designed to provide different recommendations based on the user. The recommenders utilise information about the user to offer more tailored and ultimately more accurate suggestions (Resnick & Varian, 1997). Various approaches are employed to achieve personalisation, such as collaborative filtering, content-based filtering, and hybrid methods that combine both techniques. Figure 3 illustrates the theoretical working principles of each of these approaches, demonstrating how they cater to users' diverse needs and interests to enhance the overall recommendation experience. The following subsections elabourate on the approaches in detail.

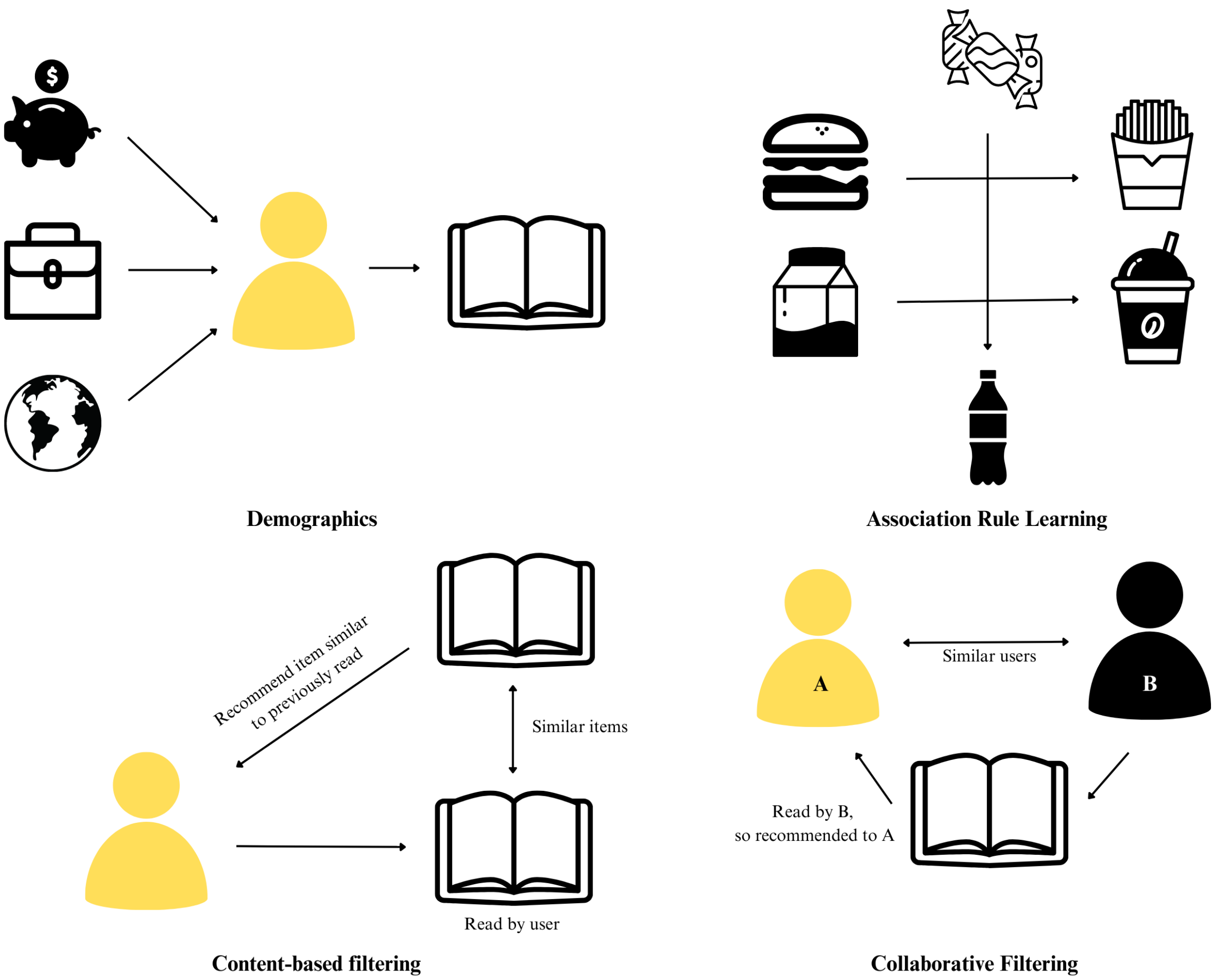

Figure 3: Types of personalised recommender systems

3.1.2.1 Demographics

A demographic-focused personalised recommender system disregards a user's preference or historical behaviour and instead relies solely on defining characteristics of the user to generate recommendations. It identifies the common traits shared by users who favour a particular item and leverages this information to make future suggestions (Pazzani, 1999). For instance, older women (characterised by age and gender) may receive recommendations for cleaning supplies, while students (distinguished by age and occupation) might be suggested school supplies.

A drawback of the demographic-focused recommender system is the challenge of acquiring accurate user demographics. Gathering such information can be cumbersome, requiring user consent and often relying on self-reported data. In many cases, users might be unwilling to share personal details, leading to incomplete or unreliable demographic profiles. Moreover, the recommendations generated solely based on demographic attributes may lack precision and personalisation. Without considering individual preferences or past behaviour, the system might suggest items that do not align with users' actual interests or needs. As a result, the accuracy and relevance of the recommendations may be compromised (Pazzani, 1999).

3.1.2.2 Association Rule Learning

Association Rule Learning (ARL) recommender techniques operate on association analysis; deriving recommendations from patterns or rules discovered in the data. ARL is a data mining technique that aims to discover relationships between variables. For instance, in a supermarket setting, this method can analyse customer purchase data to identify frequently co-purchased products. By leveraging association rules, the supermarket can gain valuable insights into customer behaviour, enabling them to enhance their marketing strategies and make more informed product recommendations (Kumbhare & Chobe, 2014). The significance of an association rule is typically measured using two key metrics:

1. **Support**: This metric reflects the frequency with which products A and B are purchased together. Rules with higher support values are considered more significant, indicating strong associations between the items (Fürnkranz & Kliegr, 2015).

2. **Confidence**: The confidence metric measures the conditional probability of buying product A when product B is purchased. It assesses the likelihood of one item being recommended when another item is already chosen by the user.

These measures play a vital role in identifying relevant and meaningful associations between products, facilitating the generation of item recommendations based on observed purchase patterns.

While association rule learning is a useful technique for discovering relationships between variables, it does have some drawbacks. One major limitation is that it only identifies associations between items that are frequently co-purchased but does not consider the underlying reasons or causal relationships. As a result, the identified associations may not always lead to meaningful or actionable insights. Additionally, association rule learning can be sensitive to noise in the data, leading to the discovery of spurious or irrelevant associations. Moreover, as the dataset grows larger, the number of potential rules increases exponentially, making the process computationally expensive and challenging to interpret. Finally, the method does not take into account user preferences or personalisation, providing non-personalised and static recommendations that may not accurately reflect individual tastes and needs (Kumbhare & Chobe, 2014).

3.1.2.3 Content-based Filtering

Content-based recommender systems utilise content information from items to suggest similar items to users. For instance, if a user expresses interest in a song of the genre "*jazz*," a likely recommendation for that user would be another song of the same genre. Content-based RS construct a user profile based on the characteristics of a user's rated items. The recommendation process involves comparing other item characteristics to the user's profile, which represents a structured representation of their preferences and interests (Lops et al., 2011). By utilising item attributes and user preferences, content-based recommender systems aim to provide personalised recommendations tailored to the unique preferences of individual users.

Content-based filtering in recommender systems offers several advantages. One key strength is its user independence, as it only requires information about the active user and not ratings from others. Similarly, content-based RS provides explainable recommendations by clearly stating the characteristics on which the suggestions are based, enhancing the interpretability of the system. Furthermore, content-based approaches are not affected by the cold-start problem for items, allowing them to recommend unrated items since they rely on item attributes rather than user behaviour (Lops et al., 2011). By leveraging item content, content-based recommendation systems can infer user preferences and offer relevant and targeted suggestions, making them valuable in scenarios with limited user feedback or explicit ratings (Pazzani & Billsus, 2007).

However, content-based systems have their limitations. One major flaw is the system's limit in content analysis; it can only analyse a limited amount of content, which may not be sufficient to accurately profile a user's interests. Additionally, content-based RS may encounter difficulty providing accurate recommendations to new users as it first needs ratings to determine their user profile (Lops et al., 2011). The effectiveness of content-based systems heavily relies on the accuracy and comprehensiveness of item descriptions, which directly impacts the quality of recommendations. Sparse or inadequate item descriptions may limit the performance of content-based approaches. Furthermore, content-based systems may suffer from the “over-specialisation” problem, frequently suggesting similar items with content that a user has already seen or liked before, making it challenging to suggest novel items outside of the user’s usual preferences (Pazzani & Billsus, 2007).

Overall, content-based recommendation systems offer a valuable approach to personalised recommendations, especially in situations where user feedback is scarce or difficult to obtain. They complement other recommendation techniques and play a crucial role in diverse recommendation scenarios, including domains with rich and informative item content.

3.1.2.4 Collaborative Filtering

Collaborative Filtering (CF) is a popular and widely used approach in recommender systems that leverages user and item similarities to provide personalised recommendations. The CF algorithm, also referred to as ”*people-to-people correlation*,” identifies users with similar preferences and items that have been highly rated together with the current item to make recommendations (Ricci et al., 2010). By effectively combining user-user and item-item correlations, CF aims to deliver accurate and relevant suggestions. As a result, collaborative filtering has become a well-liked and frequently employed strategy in the field of recommender systems .

Collaborative Filtering can be broadly categorised into two main approaches: *neighbour-based approaches* and *latent factor models*. Neighbour-based methods focus on establishing connections between items or users and generate recommendations based on similar items the user has already rated or items that are highly rated by similar users. These methods, such as nearest-neighbour techniques, are popular due to their simplicity and effectiveness.

On the other hand, latent factor models aim to transform both items and users into a shared latent factor space. One example of a latent factor model is matrix factorisation. In the latent space, items and users are represented based on variables that can be automatically derived from user input (Koren et al., 2011). These latent features are unobserved or indirectly defined features inferred through the model from the directly observed variables. These latent factors serve as valuable generalisations obtained by compressing the data. By incorporating latent factor models, collaborative filtering can effectively capture underlying patterns and relationships in the user-item interactions, leading to more accurate and comprehensive recommendations.

Neighbourhood-based models

Neighbourhood-based collaborative filtering models aim to mimic human interaction by identifying users who share similar preferences with the active user. Among these approaches, the most common implementation is the *user-user* approach, which estimates unknown ratings by leveraging confirmed ratings from similar users (Herlocker et al., 1999). Over time, the *item-item* approach gained popularity, estimating an item's rating using known ratings provided by the same user for comparable items. This approach offers advantages in terms of scalability, explainability, and accuracy (Bell & Koren, 2007; Koren et al., 2011).

In neighbourhood-based models, based on how similar the users are to one another, weights are allocated; the closer the resemblance, the more weight. Neighbourhood-based models can deliver personalised recommendations that are in line with the active user's tastes by taking into account the preferences of similar users (Bobadilla et al., 2013). Both *user-user* and *item-item* recommendation embody the same principle:

General Algorithm

Given an active user a and a target item j without a rating:

1. Identify a set of similar users (neighbours) or items (neighbour items) based on past interactions and historical ratings.

2. Calculate interpolation weights w for each neighbour based on their similarity to the active user a (for user-user) or their similarity to the target item j (for item-item). The similarity metric can be cosine similarity, Pearson correlation, or any other appropriate measure.

3. Predict the missing rating for item j by combining the ratings of the neighbours using the calculated interpolation weights w.

In user-user recommendation, the process involves locating akin users to the active user a and interpolating their ratings to estimate the absent rating for item j (Herlocker et al., 2017; Breese et al., 1998). Conversely, item-item recommendation entails identifying similar items to the target item j and interpolating their ratings to formulate the prediction (Sarwar et al., 2001; Linden et al., 2003).

By following this general algorithm, collaborative filtering models can effectively leverage the similarities between users and items to provide personalised recommendations. The specific implementation may vary depending on the chosen similarity metric and interpolation method, but the underlying idea remains the same for both user-user and item-item recommendation. In general, latent factor models achieve better results. However, neighbour-based approaches are often used due to their simplicity and intuitive reasoning, as they are quite logical and reasonably easy to use.

Some of the drawbacks of neighbour-based models are that they are typically relatively local in nature, focusing solely on a limited selection of similar ratings, in contrast to matrix factorisation. The models also have limited coverage, since users are only considered neighbours if they have common rated items (Desrosiers & Karypis, 2010).

Latent Factor models

Latent factor models serve as a generalisation of content-based filtering, aiming to overcome its limitations by inferring latent factors responsible for user-item preferences (Hofmann, 2004; Hofmann & Puzicha, 1999; Koren et al., 2009). In contrast to content-based filtering, where specific factors must be explicitly defined, latent factor models do not require such prior knowledge. Instead, the model infers latent representations, represented by u_i for users and v_j for items, capturing underlying patterns in the data (Mongia et al., 2020). The model predicts higher ratings when the latent factors of users and items align, enabling flexible and versatile recommendations based on these inferred representations.

The rating (x) of item i for an active user a can be modeled by calculating the inner product between the latent factors of the user (u_a) and the item (v_i):

$$x_{a,j} = u_a v_j \tag{1}$$

While latent factor models, such as matrix factorisation, excel in rating prediction accuracy, they are not without their limitations. One significant challenge is the mentioned cold-start problem, where these models struggle to make accurate recommendations for new users or items with sparse data. Additionally, the lack of transparency can be a drawback, as users might not understand the reasoning behind the recommendations. Furthermore, for less popular or niche users or items, latent factor models might not offer the most optimal recommendations due to limited data availability, potentially leading to suboptimal user experiences (Cheng et al., 2018).

3.2 Cross-Domain Recommendation

As discussed in the preceding section, personalised and non-personalised recommender systems each have their strengths and limitations. Often, the limitations are due to some form of data sparsity. Cross-domain recommendation (CDR) emerges as a strategic approach to address the challenges of data sparsity and cold-start problems in recommender systems. Cross domain recommender systems embody the concept of *transfer learning*:

taking information from a source domain and applying it to have better knowledge in a target domain (Khan et al., 2017). By leveraging data from multiple domains, CDR aims to mitigate data sparsity, address the cold-start problem, and ultimately improve recommendation performance in a broad array of domains.

Transfer learning encompasses both positive and negative transfer scenarios. *Positive transfer* occurs when knowledge or experience in one domain enhances performance or learning in a related domain. For example, possessing the skill of riding a bicycle can significantly facilitate the process of riding a scooter, in comparison to someone who has never encountered either mode of transportation. Similarly, someone who knows how to play the violin might pick up the piano faster than someone who has never touched an instrument before (Zhuang et al., 2021). However, this generalisation of experience can also have downsides. *Negative transfer* occurs when having experience or knowledge in one domain negatively influences performance or learning in another domain due to conflicting information or inappropriate mappings between the domains. An intuitive example for this is languages: acquiring proficiency in Spanish could pose more challenges for someone already familiar with English and French, compared to an individual with only a background in English (Zhuang et al., 2021).

Within the scope of the thesis, attention is directed towards positive transfer, with an emphasis on enhancing cross-domain recommender systems through knowledge transfer. Through the utilisation of positive transfer, the objective is to surmount challenges linked to data sparsity and cold-start issues, thereby enabling the generation of meaningful recommendations even in domains marked by constrained user interaction data.

3.2.1 Building Blocks for a CDR System

In the design of cross-domain recommender systems, three crucial aspects must be addressed: domain difference, user-item overlap, and recommendation task (Khan et al., 2017). These factors play a pivotal role in shaping the system's architecture and approach, and thus require careful consideration during the system's development.

1. **Domain difference**
   The first building block examines whether the various domains are distinguishable based on their respective systems, data types, or different times, or if items can be distinguished based on their attributes, types, medium, or system (Cantador et al., 2015; Li, 2011). The focus is on understanding the distinctions between domains in order to choose the design of the recommendation system.

2. **User-Item overlap**
   The second building block investigates the overlap between users and items in the

different domains. Four cases of overlap are considered (Cremonesi et al., 2011):

- *User-Item overlap (U-I)*: there are common users and common items in both domains
- *User-No Item overlap (U-NI)*: there are common users but no common items in both domains
- *No User-Item overlap (NU-I)*: there are no common users but common items in both domains.
- *No User-No Item overlap (NU-NI)*: there are no common users or items in both domains

This analysis helps in understanding the available data and user-item relationships between the domains. The four scenarious are depicted in Figure 4.

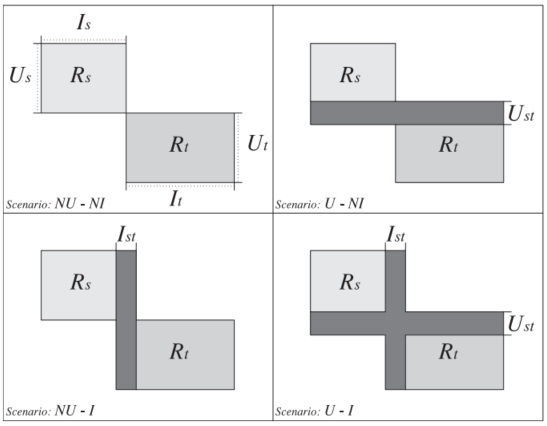

Figure 4: Four scenarios of user-item overlap (Cremonesi et al., 2011)

3. **Recommendation Task**

The third building block categorises the type of recommendation that occurs based on where the recommended items or users come from. Within CDR, there are four scenarios for recommendation (Zhu et al., 2021):

- *Single-Target CDR*: This approach involves transferring knowledge from a source domain, where there is sufficient data, to a target domain, where data is limited or missing. The goal is to leverage the information learned from one domain to improve recommendations in another domain. For example, if a recommender system has ample data on user preferences for movies, it can transfer this knowledge to enhance recommendations for books, even with limited book-related data.
- *Dual-Target CDR*: This approach leverages information from two domains to improve recommendation accuracy in both domains simultaneously. It is a

variation of Single-Target CDR, but instead of having a fixed source and target domain, Dual-Target CDR uses both domains depending on which domain has richer information.

- *Multi-Target CDR*: In this scenario, information from multiple domains is used to improve recommendation accuracy across all domains simultaneously. The objective is to recommend items from multiple target domains to a single user based on their diverse interests or historical interactions. For instance, a Multi-Target CDR system might suggest movies, books, and music to a user who has demonstrated interests in all three domains.

- *Multi-Domain CDR*: This approach leverages information from multiple domains to improve the recommendation accuracy of a set of items from multiple domains to a single group of users. The goal is to provide personalised recommendations to users who have preferences spanning across various domains. For instance, a Multi-Domain CDR system could be used in an e-commerce platform to recommend a set of products from different categories (e.g., electronics, clothing, and home decor) to a specific group of customers (e.g., frequent shoppers or premium members).

Within the context of the current investigation, particular focus is directed towards the *single-target* cross-domain scenario. In this scenario, recommendations are made either from the source domain to users in the target domain or in the reverse direction (Cremonesi et al., 2011; Zhu et al., 2021). In single-target CDR, there are three variations of transfer learning: content-based, embedding-based, and rating pattern-based transfer. The following Subsections explain these variations in detail.

3.2.2 Content-based transfer

Content-based recommendation is an intuitive approach that enhances the recommendation process by utilising additional information about items or users. This technique leverages knowledge of the product or user to generate more accurate suggestions and is based on the content-based filtering (Section 3.1.2.3). This section delves into the various types of content that can be utilised for transfer in content-based recommendation.

A straightforward example of content-based transfer considers the domains of movies and books, where similarities between the two domains can lead to relevant recommendations. Having substantial knowledge about a movie and a book allows for the generation of consistent suggestions. If a user enjoys fantasy films with child protagonists, they are likely to appreciate fantasy books with similar characteristics. However, without information about the genre or storyline of the book and movie, generating accurate recommendations becomes challenging. This exemplifies how additional information, such as user/item attributes, can enhance the accuracy of content-based recommendations (Berkovsky et al., 2007).

Other examples of content-based recommendation include utilising social tags, which are user-generated metadata about a product that can be shared across different domains (Fernández-Tobıas et al., 2019; Wang & Lv, 2020). Semantic properties derived from textual reviews help measure similarity between reviews, enabling the identification of similar items (Q. Zhang et al., 2019). Votes reflecting the helpfulness of reviews also play a role, with higher ratings carrying more weight (Shapira et al., 2013). Additionally, textual information can be analysed using techniques like topic modelling to model user and item interests, creating a shared topic space (Tan et al., 2014). Browsing and watching history can serve as implicit feedback, indicating user interest through actions like clicking on or watching specific content, even in the absence of explicit ratings (Kanagawa et al., 2019). These content-based techniques contribute to generating more accurate and relevant recommendations for users in various domains.

3.2.2.1 User/Item attributes

In content-based recommendation, user and item attributes are utilised to understand users' preferences and item characteristics (Berkovsky et al., 2007). User attributes include information about the users themselves, such as demographics and past behaviour. On the other hand, item attributes represent the features of the items being recommended, like descriptions and metadata. An important distinction for user/item attributes is that they provide explicit information that can often be directly observed and is based on the metadata of the users/items (Leung et al., 2007; Zhu et al., 2021).

By extracting meaningful features from these attributes, user profiles are created to capture the users' preferences, and item representations are established to capture the important characteristics of the items. Similarity between the user profile and each item can be calculated using measures like cosine similarity or distance metrics. Based on the similarity scores, items that are most similar to the user profile are selected as potential recommendations. These recommendations can be further refined using techniques like ranking, filtering, or combining with other recommendation approaches.

An example of an algorithm that uses user/item attributes is by Melville, Mooney, and Nagarajan (2002). They propose a content-based predictor that creates a pseudo user-ratings vector containing the user's actual ratings and content-based predictions for the unrated items. These pseudo-ratings vectors replace the original (sparse) ratings matrix.

In summary, user/item attributes are utilised to create user and/or item profiles, calculate similarity, and generate personalised recommendations based on the user's preferences and the characteristics of the items (Jannach et al., 2010; Ricci et al., 2010).

3.2.2.2 Social Tags

Social tags are user-generated labels or keywords assigned to items by users in a social tagging system. These tags represent user perceptions, opinions, or associations related to the items. Social tags can provide additional information about the items, capturing user-generated knowledge and subjective aspects of the items that may not be explicitly captured by predefined attributes (Fernández-Tobıas et al., 2019). For example, in the MovieLens database, the film "*Remember the Titans*" is associated with social tags such as "football," "race issues," "feel-good movie," and "based on a true story" (Harper & Konstan, 2015).

The difference with user/item attributes is that user/item attributes represent predefined characteristics and preferences associated with users and items, while social tags represent user-generated labels or keywords that capture subjective aspects of items. While user/item attributes are based on predetermined categories and metadata, social tags provide insights into how users personally perceive and categorise the items, making them valuable for understanding subjective preferences and associations.

3.2.2.3 Semantic Attributes

Semantic attributes focus on intrinsic item characteristics and their relationships, which can be extracted from both structured and unstructured textual information. Instead of focusing solely on predefined characteristics or user-generated labels (like user/item attributes or social tags), semantic attributes aim to uncover the underlying meaning or context of items. Semantic attributes help establish connections and similarities between items from diverse domains, even if they lack explicit overlap in traditional attributes or tags. As an illustration, contemplate a situation involving movie and book domains. Conventional attributes within these domains could encompass elements such as genre, director, author, or release date. In addition to these, social tags emerge as user-generated labels, encompassing descriptors like "feel-good," or "tear-jerking". Semantic attributes, however, go beyond these predefined labels and uncover deeper relationships between items. These attributes could include elements like "emotional intensity," "atmosphere," or "character development."

To extract semantic attributes, advanced techniques like text clustering or topic modeling are applied to analyse textual data from multiple items. These techniques identify common themes, concepts, or topics present in the text, enabling the system to group similar items together based on their semantic content. Unlike social tags, which capture users' individual perceptions and associations with items, semantic attributes represent the intrinsic characteristics of items.

An example algorithm using semantic attributes is presented by Kumar et al. (2014), who utilised item description data to identify the top 25% important words and establish semantic relations using WordNet ontology (Miller, 1995). Fraihat and Shambour (2015) extract semantic attributes by using text clustering on the item descriptions.

Although similar to user/item attributes, semantic attributes differ in that they represent the underlying meaning or context of items. They provide a more abstract and meaningful representation of users and items, capturing their essence in a generalised and interpretable manner. Moreover, semantic attributes are shared among items/users, while user/item attributes often pertain to specific individual users/items, making them distinct in their scope and application.

3.2.2.4 Text Information

Text information in CDR focuses on leveraging textual content associated with items, which includes product descriptions, reviews, summaries, and other text-based data that provides valuable information about the items. By analysing the text data using natural language processing (NLP) techniques, the CDR system can extract meaningful features, sentiments, and themes, gaining a deeper understanding of the items' characteristics.

Scholars have investigated the utilisation of textual data within CDR systems to enhance the precision of recommendations. For example, Tang et al. (2012) calculate the similarity of authors based on the papers they publish, using text data to establish connections between authors and their works. Similarly, Tan et al. (2014) utilise text data, such as movie summaries, meta data, and user-generated tags in the MovieLens dataset, as well as summary text and comments in the Amazon dataset, to enhance the understanding of users' preferences and item characteristics, leading to more effective and personalised recommendations. Sahebi and Walker (2014) use a convoluted neural network (CNN) to process text information (e.g. course introductions) and predict latent factors, which are then utilised to match students with new learning resources.

The difference between semantic attributes and text information lies in their approach. Semantic attributes are determined using automated techniques like text clustering or topic modeling, which span several users/items, without delving into specific textual content. In contrast, text information focuses on the actual individual textual content, providing a more detailed understanding of item content, such as analysing movie reviews to comprehend user opinions. Semantic attributes, on the other hand, offer a broader perspective for categorisation and grouping, like genre labels of movies.

3.2.2.5 Votes/Thumbs-up

The concept of votes or thumbs-up refers to the user feedback provided to items in the form of positive ratings or endorsements. In many recommendation systems, users have the option to express their satisfaction or preference for an item by giving it a positive rating, such as a thumbs-up or a like.

Votes and thumbs-up are a form of indirect user-generated data that can be utilised in content-based recommendation approaches. For instance, in a music recommendation system, users can express their approval by giving thumbs-up to songs they like, enabling the system to suggest similar songs or artists based on shared attributes or genres. Additionally, voting can augment direct user input. An illustrative example involves a model that integrates explicit ratings with user preferences obtained from their Facebook profile. This model considers a user's direct feedback and incorporates indirect feedback based on the user's activities on Facebook, such as liking pages related to specific films or series (Shapira et al., 2013).

This approach allows the model to provide more accurate and personalised movie recommendations that align with the user's tastes and interests. For instance, if a user explicitly rates several action movies highly and has also liked multiple action movie pages on Facebook, the model can deduce the user's strong preference for action-packed content. Consequently, it can offer movie recommendations that cater specifically to the user's affinity for action films.

3.2.2.6 Browsing/Watch History

Browsing/watch history is gathered to gain insights into user preferences and interests by analysing the web pages or items they have interacted with. It serves as a form of *soft rating*, as it reflects the user's level of interest or disinterest without a direct numerical rating. This valuable information is then utilised to construct user profiles, which capture the user's preferences and behaviour across various domains.

By analysing the browsing/watch history, the recommender system can discern patterns and trends in the user's online activities, enabling it to make more informed and personalised recommendations. For example, if a user frequently visits web pages related to travel destinations and adventure activities, the system can infer their interest in traveling and adventure-oriented experiences. This knowledge contributes to the creation of a comprehensive user profile that enhances the accuracy and relevance of future recommendations.

Algorithms using browsing/watch history can also replace ratings entirely, as demonstrated by Kanagawa et al. (2019). Others use the search history to build user features, which are then used to create user profiles (Elkahky et al., 2015).

3.2.3 Embedding-based transfer

Embedding-based cross-domain recommender systems concentrate on transferring embeddings obtained from collaborative filtering methods. The process begins by training a collaborative filtering model on the source domain to generate the embeddings. These embeddings, which represent the underlying user and item preferences, are then shared with the target domain using machine learning algorithms. By leveraging these shared embeddings, the recommender system can effectively transfer knowledge from the source domain to the target domain, enhancing the accuracy of personalised recommendations in the target domain (Zhu et al., 2021).

There are various approaches that have been utilised in embedding-based transfer. These include multi-task learning, where the model simultaneously learns from multiple tasks to improve generalisation; transfer learning, which leverages knowledge from the source domain to enhance performance in the target domain; clustering, where similar users or items are grouped together to improve recommendation accuracy; deep neural networks, which can capture complex patterns and representations in the data; relational learning, which focuses on learning relationships between users and items to make recommendations; and semi-supervised learning, which uses both labeled and unlabeled data to improve the learning process. These techniques play a crucial role in enhancing the capabilities of embedding-based cross-domain recommender systems and facilitating knowledge transfer between different domains (Zhu et al., 2021).

Despite the potential of embedding-based methods to transfer user and item preferences across domains, the direction of the thesis research does not extend to this domain. Multiple factors influenced this choice, including data availability and alignment, existing research saturation, and the intricacies involved in advanced embedding-based transfer techniques and its integration with rating pattern-based transfer.

3.2.4 Rating pattern-based transfer

Rating pattern-based transfer is the CDR approach that emphasises the transfer of general patterns that emerge across domains, rather than focusing on specific users or items (Zhu et al., 2021). It aims to capture recurring patterns in ratings data and leverage this shared knowledge to improve recommendations in the target domain. For instance, consider a pattern where users who rated films A and B with five stars tend to rate film C with only one star. This pattern can then be employed to predict potential items for a user who has highly rated film A. By identifying and utilising such rating patterns, the rating pattern-based transfer enables more effective knowledge transfer between domains, contributing to enhanced cross-domain recommendation performance (Yuan et al., 2019).

The fundamental concept of rating patterns remains constant across different domains and data in rating pattern-based transfer. However, the distinguishing factor lies in the algorithmic approaches employed. These approaches can vary from focusing on domain-specific rating patterns, to solely considering cluster-level rating patterns, or even combining both aspects (Gao et al., 2013; Li et al., 2009). In addition, some strategies acknowledge that user-item interactions may vary across domains, leading to the exploration of methods that leverage all three aspects to effectively exploit user preferences (Hu et al., 2013; Loni et al., 2014). These diverse ideas contribute to the comprehensive exploration of transfer learning and rating patterns to enhance recommendation accuracy in cross-domain scenarios.

Recent advancements in transfer learning have shown a growing interest in integrating deep learning techniques with cross-domain recommendation. As mentioned in the Introduction (Section 1), a noteworthy example is the work presented by Yuan et al. in "*DARec*," which combines rating pattern-based transfer with deep learning and achieves SOTA performance compared to other models on the same dataset (Yuan et al., 2019). By leveraging domain adaptation alongside cross-domain recommendation, DARec effectively addresses scenarios with user-no item (U-NI) overlap, enhancing its applicability in diverse recommendation settings. Section 4 presents the comprehensive architecture of the DARec model.

4 Model

The present section focuses on the implemented models in the thesis. As introduced previously, the research begins with the DARec model as foundation. Following this, the integration of content-based features into the framework is implemented to explore the outcomes of combining two distinct transfer learning techniques within CDR. The section first elaborates on the DARec model and subsequently provides details on the incorporation of content-based features. It is important to note the presence of two variations of the DARec model: U-Rec and I-Rec. U-Rec adopts a user-centric approach, aiming to forecast rating vectors for individual users. Conversely, I-Rec takes the inverse route by predicting rating vectors for specific items.

The emphasis of the research is directed towards implementing the I-Rec (Item-Based) model, rather than the U-Rec (User-Based) model. The decision is driven by the findings from the DARec paper (Yuan et al., 2019), where the I-Rec model outperformed its user-based counterpart in terms of recommendation accuracy and cross-domain adaptation capabilities (demonstrated by Figure 9). As a result, the implementation of the I-Rec model as the initial step in our investigation is prioritised. Emphasising this variant allows the utilisation of its strengths in capturing item-based features and facilitating the recommendation process adaptation across diverse domains. It is important to note that the same data pre-processing steps are used for both variants of the model, which are explained further in Section 5.

4.1 Research Methodology

The purpose of the research aims to provide insight in the impact of combining transfer learning techniques in cross-domain recommendation. To test the hypothesis stated in the Section 1.2, the research focuses on building a recommender system that evaluates the recommendation accuracy of the model using only rating pattern-based data and a system using a combination of rating pattern-based data and content. The two distinct models are evaluated with the aim of enhancing recommendation accuracy.

The methodology to achieve this objective is built upon the utilisation of two models that share the same initial pipeline and input data. The key distinction arises from the inclusion of content features within one of the models. At each phase of the pipeline, experiments are conducted to attain optimal outcomes before progressing to the subsequent stages. A detailed description of the fundamental pipeline is presented in Subsection 4.2. In the research, it is ensured that the input data has explicit feedback (user ratings) and source and target domain have a shared set of users, but no overlapping items (i.e.

scenario U-NI from Section 3.2). The following notations are used:

$$\text{Set of users: } \mathcal{U} = \{1, 2, ..., U\} \tag{2}$$

$$\text{Set of items in the source domain: } \mathcal{I}_{\mathcal{S}} = \{1, 2, ..., I_S\} \tag{3}$$

$$\text{Set of items in the target domain: } \mathcal{I}_{\mathcal{T}} = \{1, 2, ..., I_T\} \tag{4}$$

All variations of the model are scored with the Root Mean Squared Error (RMSE) metric. RMSE is widely adopted as a key metric in evaluating the performance of recommender systems due to its ability to capture the extent of prediction errors. RMSE provides a comprehensive insight into how well the model's predictions align with the actual observed ratings. By considering the squared differences between predicted and actual ratings, RMSE not only measures the magnitude of errors but also penalizes larger deviations more severely. This sensitivity to larger errors makes RMSE an effective measure for assessing the overall accuracy of recommendation predictions, offering a clear and interpretable benchmark to compare different models and techniques. Equation 5 outlines the formula employed for computing the RMSE score, with M and N representing the quantities of items and users within the test set, respectively. $\hat{y}_{i,u}$ represents the predicted rating for item i and user u, and $y_{i,u}$ represents the actual observed rating for item i and user u.

$$\text{RMSE} = \sqrt{\frac{1}{(M \times N)} \times \Sigma_{i=1}^{M} \Sigma_{u=1}^{N} (\hat{y}_{i,u} - y_{i,u})^2} \tag{5}$$

4.2 DARec

The DARec ("Deep Domain Adaptation for Cross-Domain Recommendation via Transferring Rating Patterns") paper proposes a novel approach to address the challenges of cross-domain recommendation (CDR) by leveraging domain adaptation techniques (Yuan et al., 2019). The key idea behind DARec is to learn the underlying rating patterns from source domains and then adapt this knowledge to the target domain. The model incorporates a deep learning architecture that effectively captures the intrinsic relationships between users and items in different domains. Figure 5 shows the architecture of the whole DARec pipeline.

4.2.1 AutoEncoder

The first step of the DARec model, inspired by AutoRec (Sedhain et al., 2015), employs an AutoEncoder to address the issue of unknown ratings in the dataset. The primary

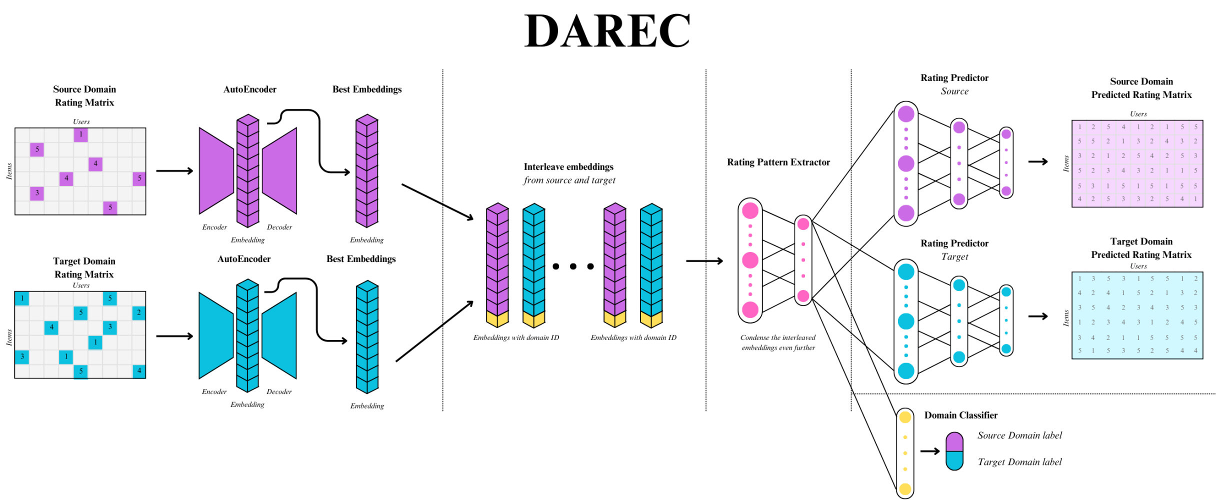

Figure 5: DARec architecture

objective is to encode and decode the rating matrix in a manner that reduces its sparsity effectively. To achieve this, two separate AutoEncoders are utilised within the architecture: one trained on the source domain and another on the target domain. Each AutoEncoder takes the rating vector for every item (y_i) as input and maps it into a low-dimensional latent space (the mapping layer is referred to as the *encoder*, and the latent space are the *embeddings*). Subsequently, a reconstruction layer (known as the *decoder*) generates values for the missing ratings, yielding rating vectors with predicted values. The predicted vectors are defined as follows:

$$\hat{y} = \mathbf{W_2} \cdot g(\mathbf{W_1} \cdot y + b_1) + b_2) \tag{6}$$

where W1, W2, b1, and b2 are the relevant weights and biases of the AutoEncoder, and g($\cdot$) is an activation function.

$$\begin{aligned} \text{loss}_{\text{AutoEncoder}} = &\sum_{u=1}^{U} \|\hat{y}_u - y_u\|_{\mathcal{O}}^2 \\ &+ \lambda \left(\|\mathbf{W_1}\|_F^2 + \|\mathbf{W_2}\|_F^2 + \|\mathbf{b_1}\|_2^2 + \|\mathbf{b_2}\|_2^2\right) \end{aligned} \tag{7}$$

Equation 7 shows the loss function for the AutoEncoder, which is calculated by calculating the masked loss between the true and predicted ratings and adding the regularisation term. The masked loss is explained in Section 4.2.4. The regularisation term consists of the matrix Frobenius norms of W_1 and W_2, and the vector l^2-norm of b_1 and b_2, and λ is the regularisation strength.

From the trained AutoEncoder, the *encoder*-layer is used to map the rating vectors of both source and target to the low-dimensional latent space. These *embeddings* form the input for the rest of the model. The algorithm for training the AutoEncoder is shown in Algorithm 1.

Algorithm 1: Training the AutoEncoder

input: rating matrix
output: embeddings, predicted rating matrix
Training AutoEncoder

for rating matrix **in** domains **do**
    Forward pass to get embeddings and predictions
    Compute loss using predictions and true values
    Update model parameters using backpropagation
end for
Save trained AutoEncoder

4.2.2 Interleaving Embeddings

Before using the embeddings as input for the subsequent part of the architecture, the source and target domains are combined to create a new dataset. To extract shared rating patterns effectively from the latent space, the sequence of latent factors for the target domain is interleaved with those from the source domain. This interleaving process includes transferring the embeddings from both domains to create a new dataset. Each embedding is associated with an item ID and a domain ID to indicate to which item and domain (source or target) it belongs. The interleaving algorithm is further detailed in Algorithm 2. This approach allows the model to capture and leverage shared patterns between the two domains, enhancing the performance of cross-domain recommendation.

Algorithm 2: Interleaving Embeddings

Input: Embeddings source, Embeddings target
Output: Interleaved Embeddings
Creating train-test split

for $domain$ **in** $[source, target]$ **do**
    Get embeddings from trained AutoEncoder
    Create train and test split on embeddings
end for
for $embedding$ **in** $train/test$ **do**
    Interleaved $\leftarrow$ embeddings source + id + $domain_{id}$
    Interleaved $\leftarrow$ embeddings target + id + $domain_{id}$
end for

4.2.3 Modified DANN

The second part of the DARec model has its focus on taking in the embeddings and trying to predict the original source and target rating vectors from it. This part of the model is inspired by the Domain Adversarial Neural Networks (DANN) (Ganin & Lempitsky, 2015). The DANN architecture proposed by Ganin et al. consists of a domain classifier, a deep label predictor, and a deep feature extractor. A gradient reversal layer, which aligns the feature distributions between several domains, connects the domain classifier to the feature extractor during training. In order to make the features as indistinguishable from one another as possible for the domain classifier, this alignment seeks to make them domain-invariant. Effective domain adaptation results from the model's training to minimise both label prediction loss for source instances and domain classification loss for all samples.

The modified DANN model, henceforth referred to as the DARec-part of the model, undergoes two modifications: the utilisation of two label predictors (one for the source and one for the target) instead of one, and the omission of the gradient reversal layer due to the exclusive focus on I-DARec within this thesis. The altered DARec model comprises three distinct elements:

1. Rating Pattern Extractor
The first component is the Rating Pattern Extractor (RPE). This is a fully-connected neural network that is applied to the task of feature extraction. It takes in the interleaved embeddings and extracts domain-invariant features.

2. Rating Predictor
The Rating Predictor (RP) is a component that is domain-specific and therefore applied twice: once for source (RPS) and once for target (RPT). It consists of a deep feed-forward neural network that takes in the domain-invariant features from the Rating Pattern Extractor and reconstructs the original source or target rating matrix. Its hidden layers follow a pyramid shape that varies based on the output dimensions from the RPE.

Its dimensions are calculated based based on the number of rows in the input matrix ($\mathcal{I}$) and the # of features ($\mathcal{F}$) in the RPE:

1 Layer. In: # of features RPE ($\mathcal{F}$). **Out**: $x = (\mathcal{F} + \mathcal{I})/2$

2. Layer. In: x. **Out**: $y = (x + \mathcal{I})/2$

3. Layer. In: y. **Out**: $\mathcal{I}$

Based on thesizeof the input matrix, the layers either follow a pyramid or an inverted pyramid shape.

3. Domain Classifier
The last component of the DARec model is the Domain Classifier (DC), a component solely implemented to force the model to learn domain-invariant features. It is a neural network consisting of two layers that tries to predict the domain label of the input. The domain label $c \in \{0, 1\}$ denotes 0 as belonging to the source domain, and 1 as belonging to the target domain.

As previously mentioned, the model does not require a gradient reversal layer prior to the domain classifier when extracting the patterns. Since there is no overlap between items, the RPE attempts to *reduce* the domain classification loss to more effectively adjust the model to the varied item attributes that are present across domains. As a result, the following is an expression for the total loss function of the DANN-part of the DARec model:

$$\begin{aligned} L(\Theta_f, \Theta_r, \Theta_c) &= loss_{pred}(\Theta_f, \Theta_r) + \mu \cdot loss_{DC}(\Theta_c) + \lambda \cdot R \\ &= \sum_{i=1}^{I_{S,T}} \|\hat{y}_{i,S} - y_{i,S}\|_{\mathcal{O}}^2 + \beta \cdot \|\hat{y}_{i,T} - y_{i,T}\|_{\mathcal{O}}^2 \\ &+ \mu \cdot \sum_{i=1}^{I_S,T} \hat{c}_i \log{(c_i)} + (1 - \hat{c}_i) \cdot (1 - c_u) \\ &+ \lambda \cdot R \end{aligned} \tag{8}$$

where Θ_f is the parameter set of the rating pattern extractor, Θ_r the parameter set for the rating predictor, and Θ_c is parameter set for the domain classifier. The loss of the domain classifier is calculated by employing binary cross-entropy loss, where $\hat{c_u}$ is the predicted domain, and c_u is the true domain label.

Algorithm 3 shows the pipeline for this process, after the embeddings are generated by the trained AutoEncoder.

Algorithm 3: DARec model

Input: Interleaved Embeddings

```
for embedding in interleaved embeddings do
    feature ← AutoEncoder(embedding)
    prediction_source ← RPS(feature)
    prediction_target ← RPT(feature)
    prediction_domain ← DomainClassifier(feature)
    Update DARec model based on loss function
end for
best_RPE ← trained-RPE
best_RPS ← trained-RPS
best_RPT ← trained-RPT
```

4.2.4 Masked Loss

In the calculation of the loss for the AutoEncoder (as indicated in Equation 7) and the predictors in Equation 8, the employed approach involves utilising a concept known as "masked loss." This approach serves to address the issue arising from ratings that are unknown or missing within the rating matrix.

The masked loss function focuses only on the observed ratings when training the AutoEncoder or DARec, i.e., the ratings that are known in the training data. It disregards the missing entries in the rating matrix by masking them. This approach is essential, as including the missing ratings, which are represented by zeroes, in the loss calculation would introduce biases and inaccuracies in the model's learning process.

By using a masked loss function, the model can concentrate on optimising the predictions for the observed ratings only, which enables it to learn the underlying patterns and relationships more effectively. This helps in mitigating the impact of data sparsity and improves the recommendation accuracy, especially when dealing with large and sparse rating matrices.

4.3 Hybrid model

The objective of the DARec model is to identify domain-invariant features for items within the datasets, as elaborated in the preceding section. To enhance recommendation performance further, the study explores the integration of content-based recommendation in conjunction with the rating pattern-based recommendation. By enhancing the DARec model with content features, the aim is to leverage supplementary item information, which in turn enhances feature learning and deepens the comprehension of item attributes. It is expected that the addition of content-based recommendation provides extra input, aiding in accelerating the learning process. However, it is anticipated that the ultimate outcomes are similar, since the content-based features are presented as supplementary information.

One method for experimenting with the combination of two transfer learning strategies is to build two distinct models (one based on rating patterns and the other on content), and combine the two outcomes. Due to the notion that using numerous models is preferable to using only one, it is expected that this will probably result in improved recommendations. However, such an approach does not effectively address the core objective of this research, which is to explore the *integration* of transfer learning techniques. To truly assess the impact of combining content-based and rating pattern-based transfers, a more integrated approach is necessary to investigate the impact of the combination. Section 4.3.2 addresses the proposed model for this research.

4.3.1 Obtaining Content Features

The first step of the hybrid model is the creation of the content features. Section 4.2 already explains the entire architecture of the DARec pipeline, which remains the same for this model. Section 3.2.2 explains the types of content features that are used typically in cross-domain recommendation. For this research, leveraging text information by analysing the item descriptions and item titles is the approach chosen. Section 5.1.2 explains how the content is pre-processed in the data before it is used to generate content features. The content features itself are based on the top N most-popular terms in the TF-IDF algorithm.

4.3.1.1 TF-IDF algorithm

The TF-IDF algorithm (Term Frequency-Inverse Document Frequency) (Robertson, 2004) is used to gauge a term's significance within a group of documents. It vectorises text by multiplying a word's frequency in a document by the word's inverse frequency throughout the entire dataset to determine how relevant a term is to a particular document within a broader corpus. Because of this, TF-IDF considers a word's "uniqueness" rather than merely its frequency. The output of TF-IDF is a sparse vector with zero values for words that do not appear in any particular document and one value for each word that does. The TF-IDF algorithm consists of two main components:

Term Frequency (TF)
It calculates the frequency of a term (word) within a document. It indicates how often a term appears in a document relative to the total number of terms in that document. The intuition is that more frequent terms may have higher importance within the document.

$$TF(t, d) = \frac{\text{Number of occurrences of term } t \text{ in document } d}{\text{Total number of terms in document} d} \tag{9}$$

Inverse Document Frequency (IDF)
It measures the rarity of a term across the entire corpus. It penalises terms that appear frequently in many documents as they are considered less informative. Rare terms that appear in only a few documents are considered more valuable in distinguishing those documents.

$$\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents in the corpus}}{\text{Number of documents containing term } t}\right) \tag{10}$$

Finally, the TF-IDF score for a term in a document is obtained by multiplying its TF and IDF values:

$$\text{TF-IDF}(t, d, D) = TF(t, d) \times IDF(t, D) \tag{11}$$

To extract content features, the TF-IDF score is computed for every word in the content descriptions. The top N content words are then determined based on their highest cumulative TF-IDF scores. These selected content words serve as essential representations of the most relevant and distinguishing features found in the item descriptions.

Subsequently, the identified top N content words are repurposed into content features for each item. In this process, a value of 0 is assigned if a content word is absent in the item description, and a value of 1 is assigned if it is present. This procedure generates content features that effectively indicate the presence or absence of specific content words in each item description. For instance, an example content feature would look as follows:

Source	**Black**	**Wood**	**Red**
Item 1	1	1	0
Item 2	0	1	1

Table 1: An example content feature

For both the source and the target domains these features are saved and stored for later use.

4.3.2 Integration content and DARec

Figure 6 shows the model depicting the integration of content-based features within the DARec pipeline. As can be seen in the Figure, the original DARec pipeline remains unchanged. The change occurs at the end of the pipeline, where instead of returning the predicted rating matrices, the steps of the DARec model are repeated, with the addition of content features.

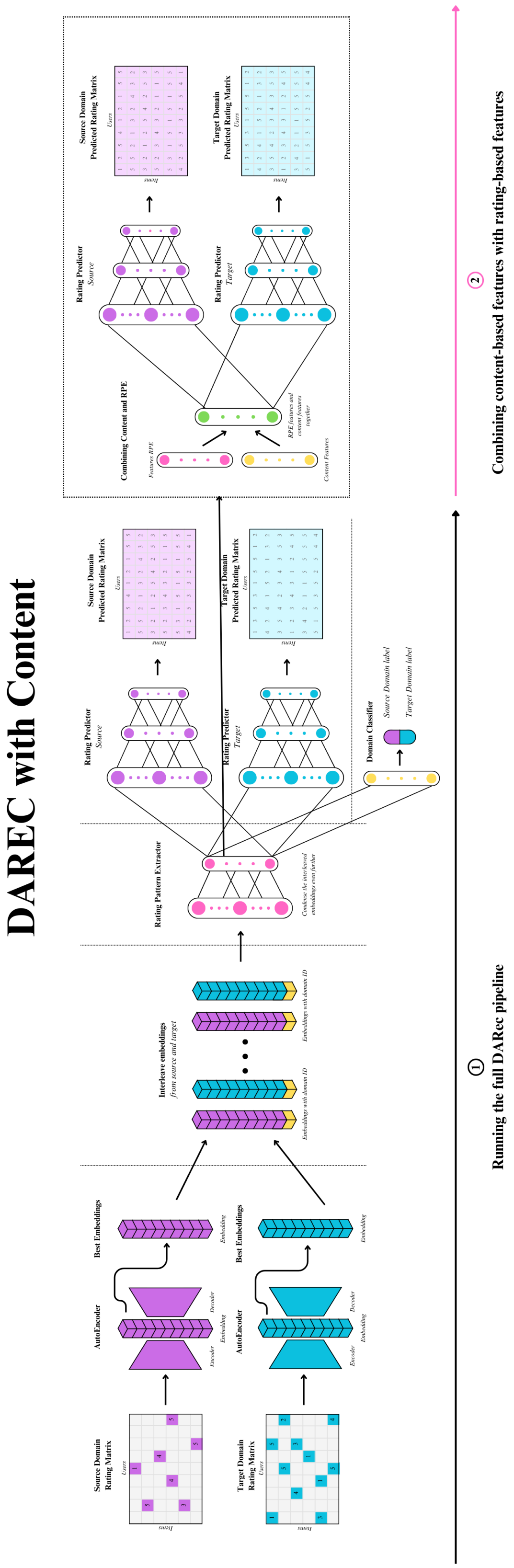

Figure 6: Hybrid model architecture (DARec with content)

The integration process begins with the utilisation of the content features in conjunction with the pre-trained DARec model. From the pre-trained DARec, the pre-trained AutoEncoder, rating pattern extractor, and rating predictors are saved. The domain classifier is left out of the process, as the content features are domain-specific.

The pipeline remains unchanged until the rating pattern extractor (RPE) stage, where the same steps as before are followed to obtain embeddings from the rating matrices. The dimensions of the rating pattern extractor remain consistent with those in the original DARec model. Using the RPE, the features are extracted from the generated embeddings.

The actual integration occurs just before passing the new features into the rating predictors. In this stage, the content features are added to the output of the rating pattern extractor. This strategy is based on the idea that the AutoEncoder within DARec strives to achieve a comparable objective (identifying distinct features for optimal dataset representation), but it must start learning from the beginning. By furnishing it with well-defined features through content features, the intention is to support its learning endeavour. Placement of the content features after the rating pattern extractor is guided by sizing considerations; since the output is already reduced at this point, the incorporation process becomes more seamless.

To accommodate the new input dimensions resulting from the addition of content features, a layer is inserted before the original pre-trained rating predictors. Once the content features are combined with the output features from the rating pattern extractor, the process of training the rating predictors remains the same. The same loss function, without the domain prediction loss, is employed to compare the final results to those of the model without content features. This approach facilitates the assessment of the influence of content integration on recommendation accuracy.

The modified algorithm, after the main DARec model is trained, is shown in Algorithm 4.

Algorithm 4: DARec model with content

```
Input: Interleaved Embeddings, content
for embedding in interleaved embeddings do
    feature ← AutoEncoder(embedding)
    feature_content ← feature + content
    prediction_source ← RPS(feature_content)
    prediction_target ← RPT(feature_content)
    Update DARec model based on loss function
end for
```

5 Data

The primary dataset utilised for experimentation is sourced from Amazon reviews, originally curated by J. McAuley (McAuley et al., 2015). The dataset was initially developed for one-class collaborative filtering research and encompasses product reviews and associated information from 24 distinct Amazon product categories. Moreover, each category is accompanied by a corresponding dataset containing metadata for all products within that particular category. As highlighted in Section 2.2, the issue of intellectual property rights surrounding data becomes a notable concern when employing transfer learning across datasets or models sourced from distinct companies. To circumvent this potential challenge, a deliberate choice was made to utilise a publicly available dataset. By opting for such a dataset, the discussion regarding intellectual property becomes a non-issue, providing a clear and ethical pathway for the research investigation.

The dataset has been previously employed in the DARec paper (Yuan et al., 2019) and, consequently, serves as a benchmark for the present study. However, it is important to note that since the publication of the DARec paper, the dataset has undergone significant expansion, incorporating numerous additional reviews. Consequently, while the results obtained in this thesis may exhibit slight variations due to the dataset extension, the original DARec results continue to serve as valuable guidelines for comparison and evaluation. Due to the different domains as well as its accompanying content metadata, the dataset is ideal for this research and better suited than other datasets often used in recommendation (i.e. MovieLens (F. Maxwell Harper, 2005)).

Due to hardware and availability constraints, not every domain set was used in the experiments. The original DARec paper employed four domain sets, aiming for domains as dissimilar as possible: Office Products & Movies and TV, Sports and Outdoors & CDs and Vinyl, Android Apps & Video Games, and Toys and Games & Automotive. Among the domain sets used in the original DARec paper, Sports and Outdoors, Toys and Games, Movies and TV, and Automotive are notably large datasets. Additionally, one domain (Android Apps) was not available anymore in the updated dataset. Therefore, to also accommodate hardware limitations, the research focuses on two domain pairs: Office Products & Movies and TV, and All Beauty & Appliances. These domain pairs strike a balance between datasetsizeand the number of mutual users, aligning with the research objectives. The selection was made considering the availability of data and computational feasibility while maintaining the essence of the original approach (dissimilar domains).

Table 2 shows the summary statistics for the four chosen domains.

	Dataset		# of items		# of users	# of ratings		Avg rating per user		Avg # of ratings per item	
	Source	*Target*	*Source*	*Target*	*Shared*	*Source*	*Target*	*Source*	*Target*	*Source*	*Target*
1	All Beauty	Appliances	58	68	11	69	71	4.4928	4.1549	1.1897	1.0441
2	Office Products	Movies and TV	11143	29667	3355	47302	117699	4.4941	4.3109	19.1550	39.7250

Table 2: Summary statistics domains

The sparsity of each domain is calculated by using the following Equation:

$$\textbf{Sparsity} = \frac{\text{\# of ratings}}{(\text{\# of items} \times \text{\# of users})} \times 100\% \tag{12}$$

In the context of the Beauty and Appliances domains, the sparsity rates are 89.18% and 90.51%, respectively. For Office Products and Movies and TV, notably higher sparsity levels of 99.87% and 99.88% are observed.

5.1 Data pre-processing

5.1.1 Generating rating matrix

For each dataset, Figure 7 shows the data pre-processing steps taken to obtain the rating matrices. The data underwent a thorough cleansing process, involving the removal of irrelevant columns and duplicate entries. To handle instances where users submitted multiple reviews for the same product, only the latest review (and rating) was retained. Additionally, to ensure sufficient data for effective learning, users with a limited number of reviews (e.g., 5 for Beauty & Appliances and 10 for Office & Movies) were excluded. Furthermore, in the case of Office and Movies, items with fewer than 5 ratings were also filtered out to create a computationally manageable dataset. The selection of the threshold value of 5 aligns with the criteria outlined in the DARec paper. Section 6.1 provides detailed insight into the rationale behind the choice of a threshold value of 10 for users for the Office & Movies domain.

Having extracted the desired rows, the next step involved identifying the shared users between the source and target domains. Subsequently, the rating matrix was constructed based on these shared users, with users serving as rows in the matrix. To align the data in the format suitable for I-DARec, the matrix was then transposed. This process ensured a well-prepared dataset, conducive to effective experimentation and analysis.

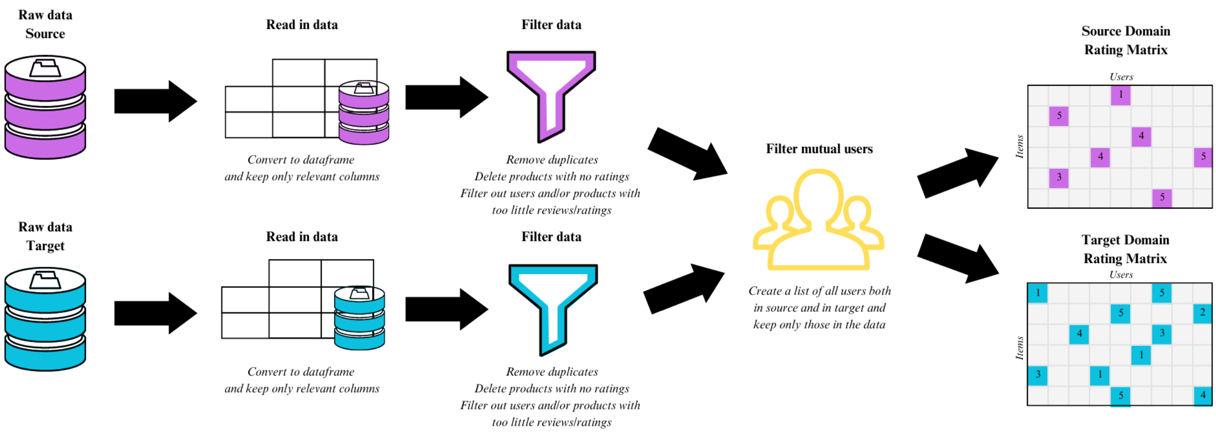

Figure 7: Data pre-processing steps per domain set

5.1.2 Generating content

Each domain has extensive metadata on all its products. This metadata is used to generate the content features for the integration of content features and rating pattern-based DARec. Before the content can be used, it first needs to be pre-processed as well. The items have the following types of information available:

- **Reviews**:
  - Reviewer ID: the ID of the reviewer
  - Reviewer name: the name of the reviewer
  - Review text: the text of the review
  - Review summary: a summary of the review, where it is unknown who creates the summary
  - Review time: at what time the review was written
  - Number of "helpful review"-votes: how many other users found this review helpful
- **Items**:
  - Title: the name of the item
  - Description: the description of the item
  - Features: a bulleted list of features of the product
  - Related products: products that are similar (either also bought, also viewed, or bought together)
  - Brand: the name of the brand

- Price: the price in US dollars
- Categories: which categories the item belongs to (can be the domain itself, or other categories)

It is essential to acknowledge that not every item in the dataset contains all the mentioned data fields. Some items may lack certain information, such as descriptions, features, or related products.

The choice of utilising item descriptions for analysis was made based on the data availability, time constraints, and the quality of the textual content. In cases where item descriptions were absent, the item titles were used to complement the missing information.

Figure 8 illustrates the sequence of pre-processing actions conducted to extract relevant content from the metadata. The specific actions taken at each stage are described below. It is important to note that these steps were executed without extensive optimisation. While further refinement could potentially yield more insightful content features, this aspect was not prioritised due to the constraints of time within the scope of this research. Section 4.3.1 provides a detailed account of the content feature extraction algorithm. To maintain conciseness, duplicating the comprehensive explanation here is avoided.

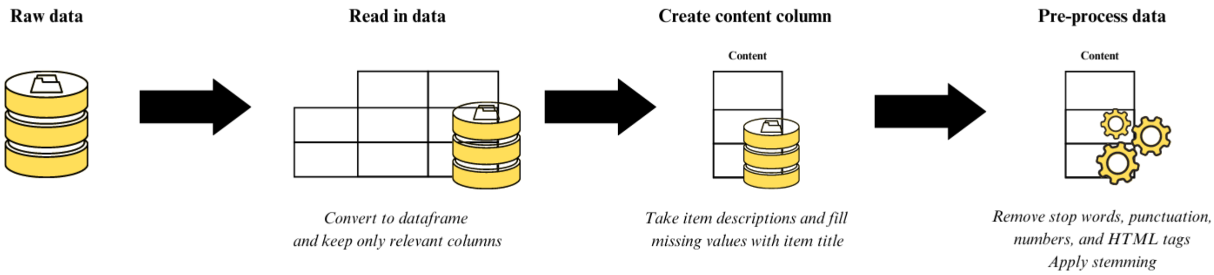

Figure 8: Data pre-processing steps for content

5.1.2.1 Remove stop words, punctuation, and numbers

Stop words are common words that do not carry significant meaning in the text. Examples of stop words include "a," "the," "is," and "and." They are words that occur often without incurring any meaning. Punctuation is removed from the text because it provides no meaningful context. Additionally, numbers are also deleted from the contents. The reasoning behind this step is that numbers do usually not provide a distinguishing feature of an item; in the dataset numbers are often product codes, prices, or measurements. These are often not generalisable and only make sense when made relative to other products. Preferable attributes include terms like "large", "cheap", or "travel-sized" as opposed to terms such as "1 (L)", "2.99", or "50 (ml)". Furthermore, HTML tags were also removed from the text.

Stop words, punctuation, and numbers are removed for multiple reasons:

- **Noise reduction and focus on important terms:** Eliminating elements like stop words, punctuation, and numbers, causes the overall text to be cleaner and less cluttered and allows the analysis to focus more on the content words that carry significant meaning. This can help make the analysis and interpretation of the text more straightforward and efficient, and can improve the accuracy and relevance of the results obtained from NLP models.

- **Memory and storage efficiency and improved computational efficiency:** The stop words, punctuation, and numbers can consume memory and storage resources, especially in large text corpora. Removing them can lead to more efficient processing and storage of textual data. Since the stop words, punctuation, and numbers occur frequently, their presence in the text can increase the computational burden during various natural language processing tasks. Removing them can speed up processing and analysis.

5.1.2.2 Applying stemming

Stemming is a technique used in natural language processing to reduce words to their root or base form, known as the *stem*. It involves removing common word endings or suffixes to simplify words and group them together based on their common root form. The purpose of stemming is to normalise words so that variations of the same word can be treated as the same word. For example, stemming can convert words like "running," "runs," and "ran" to their common stem "run," allowing algorithms to recognise them as similar forms of the same word.

Stemming algorithms work by applying a set of predefined rules or heuristics to remove suffixes and convert words to their stem form. These rules are based on linguistic knowledge and patterns. The most commonly used stemming algorithm is the Porter stemming algorithm; a technique used to remove common morphological and inflectional endings from English words (Porter, 1980).

It is important to note that stemming is a simple and rule-based process, and it does not always produce accurate results. Stemming can result in the stem not being an actual word or it can create stems that are not necessarily related in meaning. Therefore the performance of the algorithm with and without stemming was tested. The results are recorded in Table 11 and 12.

5.2 Train/test split

In order to create a train-test split for the data, unique identifiers are generated for the rating matrix. The test percentage is used to determine the size of the test set, which is a subset of these unique identifiers. Random selection without replacement ensures that the test set does not overlap with the training set. After obtaining the test and train IDs, the rating matrix is split into separate matrices for training and testing. The train/test split is generated on the source and target domain separately for the AutoEncoder, and on the interleaved embeddings (consisting of both source and target embeddings) for the DARec and hybrid model.

6 Experiments

The upcoming section meticulously outlines the experiments and the resulting performance of the different models, as detailed in Section 4. It commences with comprehensive explanations of the experiments and data settings. Subsequently, some of the models' performances are compared against the benchmark set by Yuan et al. (2019). The section concludes by presenting the outcomes related to the hybrid model.

6.1 Parameter Settings

This subsection delves into an exploration of parameter settings for three distinct models, namely the AutoEncoder, the DARec model and the hybrid model. Understanding the appropriate configuration of these hyperparameters is important in optimising the models' performance for cross-domain recommendation tasks. Systematic experimentation involves the assessment of diverse hyperparameters, including learning rates, weight decay, number of hidden layers, and regularisation rates, among others. For all experiments, the *batch size* is set to 512 and the train/test split is set to 10%. The results presented are derived from the models' performance on the test set.

6.1.1 AutoEncoder

The first step in both the regular DARec model and the hybrid model is the AutoEncoder. It is crucial for the AutoEncoder to perform well, as its embeddings influence the performance of the other models afterwards. There are four hyperparameters to be considered in the AutoEncoder: the number of features, the regularisation rate (λ), the learning rate (lr), and weight decay (wd). The sets of values that are tested for each of the hyperparameters are displayed in Table 3. An essential consideration in the AutoEncoder implementation is to maintain an identical number of features for both the source and the target domains. This ensures that the resulting embeddings possess uniform dimensions, facilitating their seamless interleaving during the subsequent stages of the model. As mentioned in Section5.1, there is also the additional test on what criteria to use for the Office Products & Movies and TV domain set. This is done to simulate the number of shared users this domain set has in the original DARec paper (5,154).

Domain Set	**Embedding Size**	λ	**lr**	**wd**$*$	**Criteria**
1	{5, 50, 100}	{1e-6 - 0.01}	{1e-4 - 0.25}	{1e-6 - 0.1}$*$	5
2	{200, 400, 600, 800, 1000, 1200}	{1e-6 - 0.01}	{1e-5 - 0.01}	{1e-6 - 0.1}$*$	{9,10,15}

Table 3: Hyperparameter settings AutoEncoder

*Weight decay is initially taken into account in the first few experiments, but it is eventually set to 0. This decision is made because the regularisation term (λ) in the optimiser already serves as a means of controlling overfitting, similar to weight decay. Simultaneously hypertuning both parameters could result in conflicts, hence weight decay is fixed at 0 to avoid any potential interference.

The optimal-performing AutoEncoder (best RMSE) on both source and target domain is saved and retained for the subsequent phase of the DARec model.

6.1.2 DARec

For the DARec model, the pre-trained AutoEncoder is used to obtain the embeddings ($\mathcal{E}$). After the sequence of embeddings from source is interleaved with the embeddings from target, the interleaved input is fed into the DARec model. The sets of values that are tested for each of the hyperparameters are displayed in Table 4. The hyperparameters considered are: the number of features in the RatingPatternExtractor, the strength of the domain classifier loss in the overall loss function (μ), the regularisation term (λ), the importance of the target domain (β), and the learning rate (lr). Equation 8 details the exact loss function used.

Domain set	**# of features RPE**	μ	λ	β	**lr**
1	{50-200}	{1e-4 -10000}	{1e-4 -10000}	{0.1-1}	{1e-4 - 0.1}
2	{50-500}	{1e-4 -10000}	{1e-4 -10000}	{0.1-1}	{1e-4 - 0.1}

Table 4: Hyperparameter settings DARec

The individual models of the top-performing DARec model - the rating pattern extractor, the rating predictor source, and rating predictor target - are saved and retained for the integrated DARec with content model. The domain classifier is not used any further.

6.1.3 Hybrid model (DARec with content)

By utilising the pre-trained DARec model and adapting it to incorporate the added content features, the same parameters as DARec are hypertuned within this model. Furthermore, experimentation encompasses varying the quantity of incorporated content features. The selection of the number of content features is influenced by computational constraints to avoid potential memory issues. Table 5 shows the hyperparameters tuned for this integrated model. The hyperparameters for the hybrid model are again the regularisation term (λ), the significance of the target domain (β) and the learning rate (lr)

For both distinct domains, the point-biserial correlation coefficient between the content-based features and the features extracted through the Rating Pattern Extractor was computed to determine the uniqueness of the content-based features.

Domain set	# content features	λ	β	lr
1	{10, 25, 50}	{1e-4 - 10000}	{0.1-1}	{1e-4 - 0.1}
2	{50, 100, 200, 400}	{1e-4 - 10000}	{0.1-1}	{1e-4 - 0.1}

Table 5: Hyperparameter settings hybrid model

6.2 Results

An overview of the findings derived from the conducted experiments is presented. The initial subsection outlines the benchmark results from the original DARec model. Afterwards, the focus lies on the outcomes of the AutoEncoder experiments. This is followed by a subsection on the results from the DARec experiments. The last subsection details the results from the experiments with the content-integrated DARec model. Every experiment is conducted with an implementation of early stopping to counteract potential overfitting. The following section (Section 7) analyses the achieved results.

6.2.1 Benchmark DARec

Figure 9 shows the results obtained by the original DARec model. For each embedding size value, the combined RMSE results of four models are displayed: U-AutoRec, U-DARec, I-AutoRec, and I-DARec. U-AutoRec showcases the outcomes of the AutoEncoder with a user-centric approach, while U-DARec represents the collective results of the DARec model using the specified embedding size. Similarly, I-AutoRec and I-DARec exhibit identical outcomes but from the item-centric perspective.

Within this diagram, the emphasis is based on the red and green lines, pertaining to the I-rec version of the model. These particular lines display the performance of the original I-rec model on the domainset of Office Products & Movies and TV and can therefore be used for comparison. It should be noted that for embedding size 800, the best RMSE for the AutoEncoder is 0.985, and for DARec the obtained RMSE is 0.974. The overall best achieved RMSE for I-DARec is 0.9582, for embedding size 400.

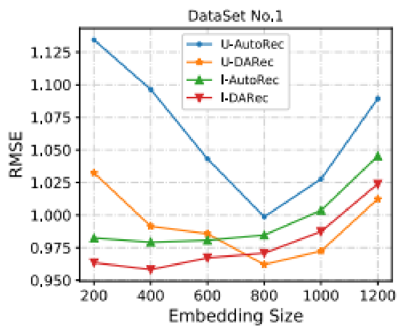

Figure 9: Benchmark results DARec (Yuan et al., 2019)

6.2.2 AutoEncoder

The presentation of experiment outcomes begins with the analysis of the AutoEncoder results. The results are categorised by the two domain sets: Beauty & Appliances and Office Products & Movies and TV (referred to as "Office & Movies" hereafter).

	Criteria	**Shared Users**	**Domain**	**RMSE**
1	15	1098	Source	0.8444
			Target	0.9059
2	10	3355	Source	0.8828
			Target	0.8853
3	9	4412	Source	1.1863
			Target	1.2982

Table 6: Results experiments criteria Office & Movies

Starting with the experiments involving the criteria number for the Office Products & Movies and TV domain set, Table 6 outlines the outcomes of the best AutoEncoder model, acquired from the evaluation of three distinct criteria-values, accompanied by their corresponding quantities of mutual users and RMSE-scores. As mentioned in Section 5.1, the criteria-value stands for the minimum number of reviews a user must have to be considered useful in the dataset. Following an exhaustive assessment of the experiments and thoughtful consideration of computational efficiency, the decision to employ *criteria* = 10 for all ensuing experiments was made.

The learning curves of the best AutoEncoders for the source and target domains on the first domain set are depicted in Figures 10 and 11, respectively.

Figures 12 and 13 present the learning curves for the second domain set: Office & Movies. Moreover, Figure 14 illustrates the ideal embedding size for both domains. In the right figure, the benchmark from the original DARec model is highlighted.

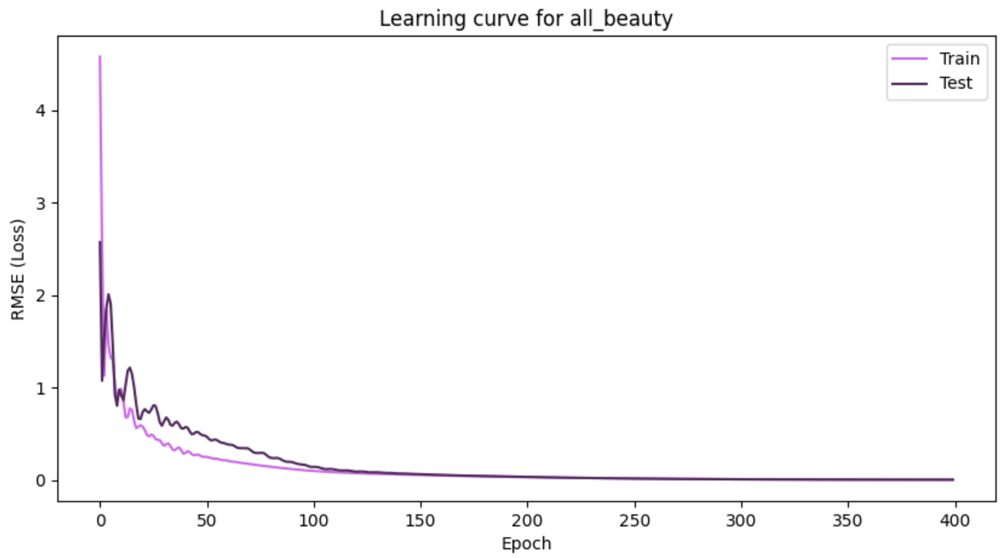

Figure 10: Learning curve Autorec All Beauty

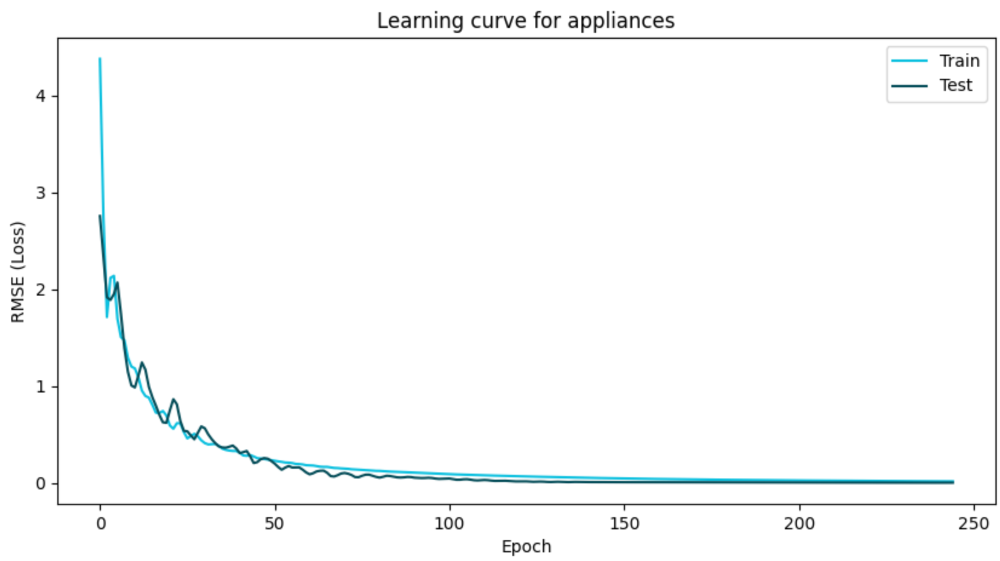

Figure 11: Learning curve Autorec Appliances

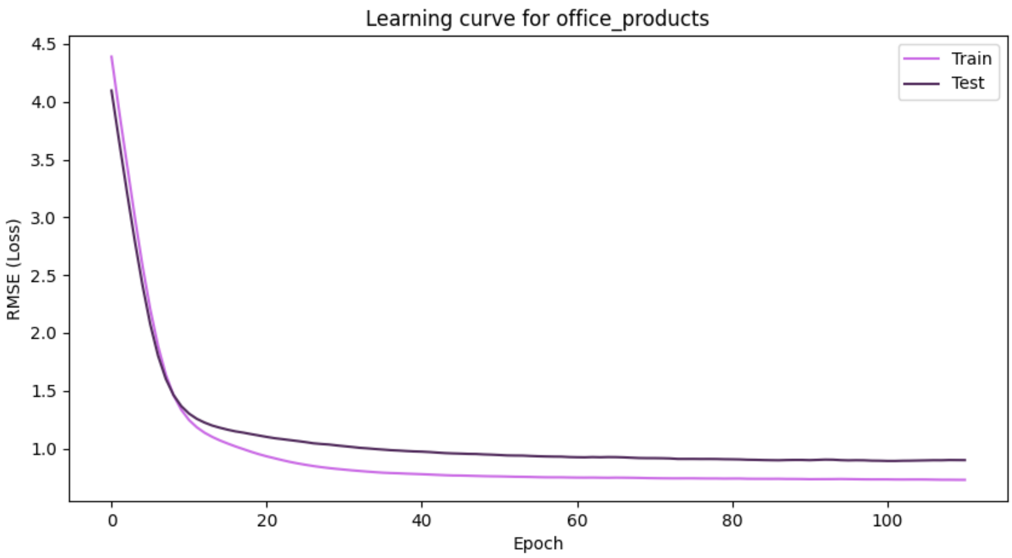

Figure 12: Learning curve Autorec Office Products

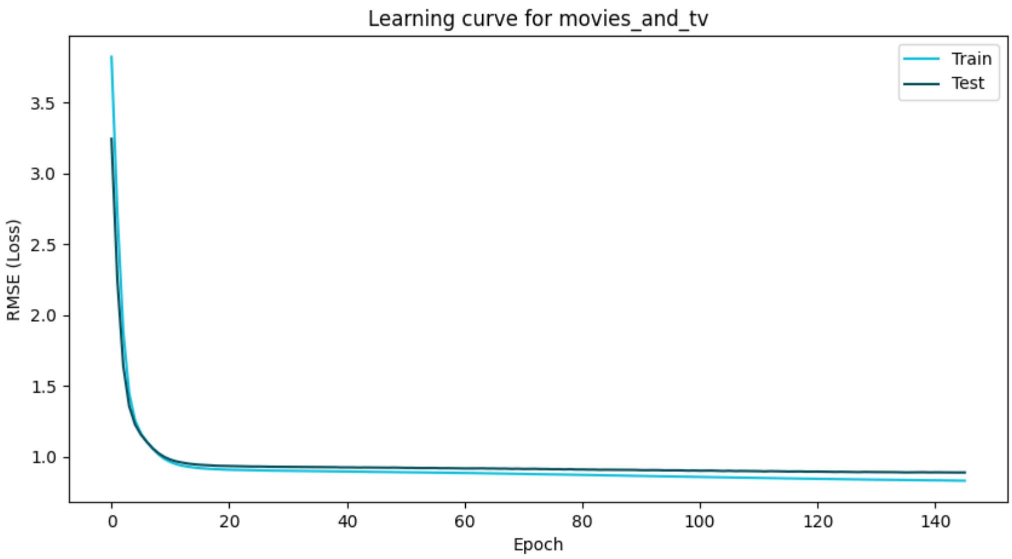

Figure 13: Learning curve Autorec Movies and TV

For comprehensive experiment results, Table 7 shows the performance of the AutoEncoder for Beauty & Appliances and Table 8 shows the results for the Office & Movies domainset. An interpretation of the results is provided in Section 7.

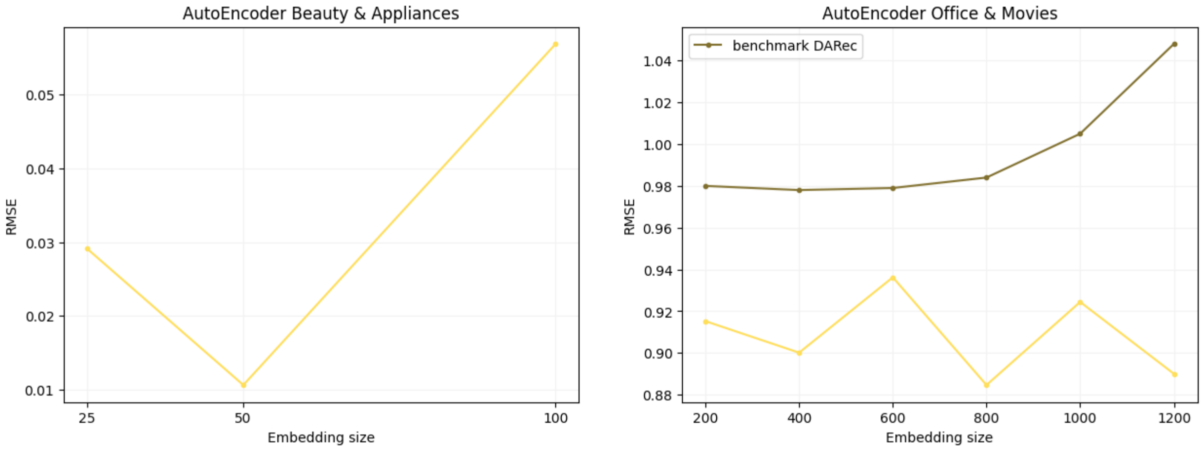

Figure 14: Impact of embedding size on RMSE

Beauty & Appliances

All the results presented are based on the optimal models obtained by running the hyperparameter optimisation program for 150 trials.

Embedding Size	Combined RMSE	Domain	RMSE	λ	Lr	Wd
25	0.0291	Beauty	0.0415	5.18e-6	0.1597	0
		Appliances	0.0106	4.22e-6	0.2129	0
50	**0.0106**	Beauty	0.0083	1.14e-6	0.2473	0
		Appliances	0.0053	1.26e-6	0.2476	0
100	0.0569	Beauty	0.079	3.48e-5	0.1091	0
		Appliances	0.0269	1.01e-5	0.1431	0

Table 7: Results hyperparameter experiments AutoEncoder Beauty & Appliances

Office & Movies

All the results presented are based on the optimal models obtained by running the hyperparameter optimisation program for 50 trials.

Embedding Size	Combined RMSE	Domain	RMSE	λ	Lr	Wd
200	0.9152	Office	0.8276	1.02e-6	0.0006	0
		Movies	0.9105	1.03e-6	0.0004	0
400	0.9001	Office	0.9230	1.01e-6	0.0004	0
		Movies	0.8913	1.02e-6	0.0001	0
600	0.9362	Office	0.9352	2.04e-6	0.0009	0
		Movies	0.9366	1.01e-6	0.0002	0
800	**0.8847**	Office	0.8828	1.04e-6	0.0001	0
		Movies	0.8853	1.02e-6	0.0001	0
1000	0.9244	Office	0.8919	1.76e-6	0.0006	0
		Movies	0.9364	1.05e-6	0.0000	0
1200	0.8900	Office	0.8843	1.03e-6	0.0001	0
		Movies	0.8926	1.04e-6	0.0001	0

Table 8: Results hyperparameter experiments AutoEncoder Office & Movies

Based on the findings presented, the optimal-performing models are chosen for further progression. These selected models demonstrate superior performance and efficacy, making them ideal candidates for further experiments with DARec and DARec with content.

6.2.3 DARec

The outcomes presented in this section are based on utilising the most optimal performing AutoEncoder, as presented in the previous subsection.

Figure 15 shows the impact of size of the rating pattern extractor on the model's recommendation accuracy on the test set. The full parameter-settings for each experiment are presented in Table 9 for Beauty & Appliances and Table 10 for Office & Movies. Section 7 provides an interpretation of the results.

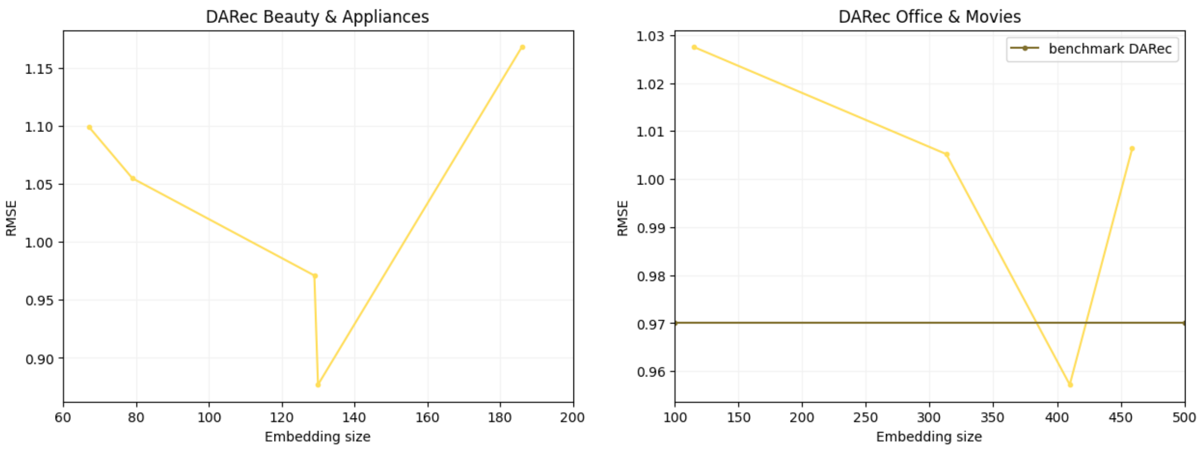

Figure 15: Results experiments nr of features RPE in DARec

Beauty & Appliances

All the results presented are based on the optimal models obtained by running the hyperparameter optimisation program for 250 trials.

Combined RMSE	Domain	RMSE	μ	λ	β	Lr	# factors RPE
1.0549	Source Target	0.7350 1.2811	191.3308	0.1415	1	0.0242	76
0.9815	Source Target	0.7044 1.1809	0.4515	0.9083	1	0.0867	129
0.8769	Source Target	**0.6715** **1.0309**	0.0092	0.0002	0.6604	0.0867	**130**
1.1680	Source Target	0.5945 1.5156	209.5544	1.6607	0.0067	0.0995	186

Table 9: Results hyperparameter experiments DARec Beauty & Appliances

Office & Movies

All the results presented are based on the optimal models obtained by running the hyperparameter optimisation program for 25 trials.

Combined RMSE	Domain	RMSE	μ	λ	β	Lr	# factors RPE
1.0275	Source Target	0.9220 1.1458	0.6782	0.0004	0.7886	0.0179	115
1.0549	Source Target	0.7350 1.2811	3.9322	1017.2699	0.2540	0.03892	313
0.9572	Source Target	**0.8503** **1.0660**	172.8745	0.9771	0.1923	0.005	**410**
1.0064	Source Target	0.8971 1.1284	0.0002	51.5055	0.1252	0.0423	459

Table 10: Results hyperparameter experiments DARec Office & Movies

6.2.4 Hybrid model

The results depicted in the tables within this section stem from employing the most optimal-performing DARec model, identified as Experiment 0 in the tables. Figure 16 illustrates the combined RMSE scores of the hybrid model plotted against different numbers of content features for both domain sets. Detailed values, along with the separate source and target RMSE, are provided in the corresponding paragraphs for each domain set.

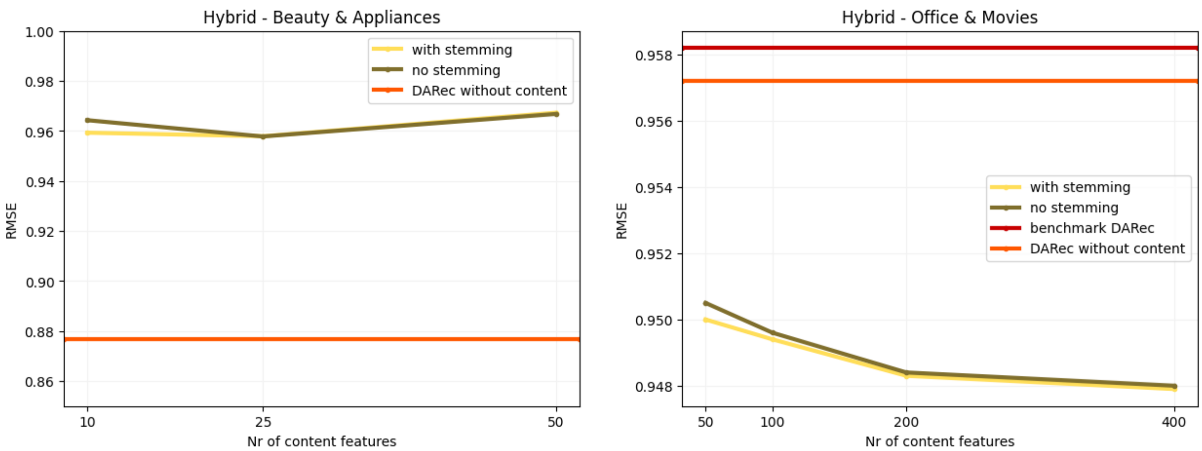

Figure 16: Results experiments Hybrid model

Beauty & Appliances

Table 11 shows the highest-achieved RMSE score on the test set for the domain set Beauty & Appliances. The '# of features' denotes the number of content features used in the experiment. All the results presented are based on the optimal models obtained by running the hyperparameter optimisation program for 250 trials.

	# of Features	Stemming	Combined RMSE	Domain	RMSE	λ	β	lr
0	0	-	0.87693	Source Target	0.67146 1.03087	-	-	-
1a	10	True	0.9593	Source Target	0.7543 1.1156	3.1411	0.7144	0.0019
1b	10	False	0.9643	Source Target	0.7645 1.1531	2.3178	0.5431	0.0019
2a	25	True	0.9579	Source Target	0.7869 1.0922	0.0002	0.7787	0.0003
2b	25	False	0.9578	Source Target	0.78632 1.0925	0.0003	0.6858	0.0003
3a	50	True	0.9673	Source Target	0.7960 1.1020	0.0004	0.2378	0.0005
3b	50	False	0.9668	Source Target	0.7951 1.1017	0.0015	0.9994	0.0005

Table 11: Results experiments hybrid model Beauty & Appliances with hyperparameters

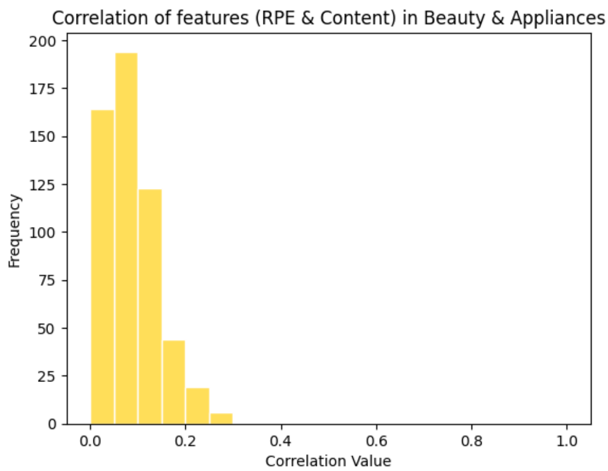

Figure 17: Histogram of correlation-values between RPE-features and content-features for Beauty & Appliances

Office & Movies

Table 12 shows the highest-achieved RMSE scores on the test set for the Office & Movies domain set. All the results presented are based on the optimal models obtained by running the hyperparameter optimisation program for an average of 10 trials.

	# of Features	Stemming	Combined RMSE	Domain	RMSE	λ	β	lr
0	0	-	0.9572	Source Target	0.8503 1.0660	-	-	-
1a	50	True	**0.9500**	Source Target	0.8461 1.0560	11.2807	0.3754	0.0348
1b	50	False	**0.9498**	Source Target	0.8457 1.0561	0.0004	0.4782	0.0003
2a	100	True	**0.9494**	Source Target	0.8427 1.0581	0.0193	0.9996	0.0002
2b	100	False	**0.9496**	Source Target	0.8424 1.0586	0.0236	0.8117	0.0001
3a	200	True	**0.9483**	Source Target	0.8420 1.0565	0.0107	0.6839	0.0001
3b	200	False	**0.9484**	Source Target	0.8430 1.0558	0.0011	0.1912	0.0002
4a	400	True	**0.9479**	Source Target	0.8413 1.0563	0.0070	0.6545	0.0001
4b	400	False	**0.9480**	Source Target	0.8415 1.0565	0.0007	0.1176	0.0003

Table 12: Results experiments hybrid model Office & Movies with hyperparameters

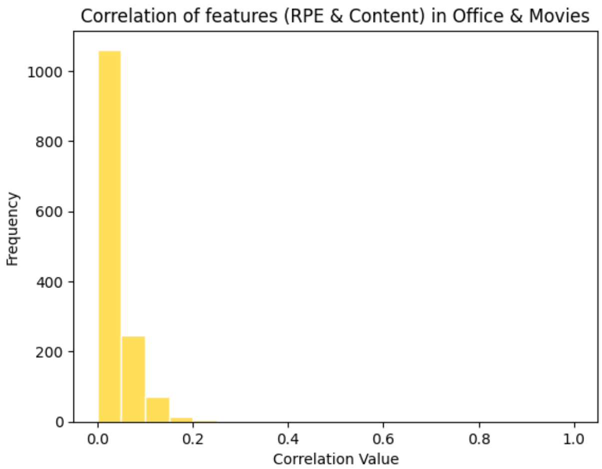

Figure 18: Histogram of correlation-values between RPE-features and content-features for Office & Movies

7 Discussion

The discussion regarding the experimental outcomes revolves around key observations and insights derived from the conducted research. It begins with interpreting the results presented in the previous section (Section 6.2), followed by a discussion of result limitations. The relevance of the outcomes to the business case is then explained, and potential directions for future research are highlighted to conclude this section.

7.1 Interpretation of Results

The interpretation of the outcomes presented in Section 6.2 is divided into three distinct subsections, each dedicated to a specific model: AutoEncoder, DARec and the Hybrid model.

7.1.1 AutoEncoder

Figure 14 presents the influence of the embedding size of the AutoEncoder on its predictive accuracy. Notably, the combined RMSE of the Beauty & Appliances domain set is markedly superior to that of the Office & Movies AutoEncoder. This divergence can be attributed to the substantial dissimilarity in size between the two domain sets, affording the AutoEncoder the opportunity to achieve near-perfect prediction of the rating matrix in the smaller Beauty & Appliances domain. Consequently, the examination of the model's performance on the Office & Movies domain gains greater significance, as it allows for a direct comparison with the state-of-the-art original DARec model.

The right-hand figure shows the superiority of the AutoEncoder model over the original DARec I-Autorec model. This advancement can be partially attributed to the decreased sparsity of the experiment dataset, owing to the application of a higher criteria in data pre-processing. The mitigation of sparsity permits the model to access more significant information and generate improved predictions, thus resulting in heightened performance compared to the original model. The diminished sparsity empowers the model to leverage richer data, leading to more accurate and effective recommendations.

7.1.2 DARec

The experimental investigations concerning the DARec model were undertaken to establish a robust baseline. This process aimed to facilitate a precise assessment of subsequent experiments involving the content-based features. The primary objective was to train an adept rating pattern extractor that would serve as a benchmark for further evaluations. Particularly in the Office & Movies domain, the comparison with the original DARec model was informative in gauging the effectiveness of the developed model. Figure 15 illustrates that an embedding size of 410 in the Rating Pattern Extractor yielded superior outcomes in the Office & Movies domain compared to the original DARec model using the same embedding size. Impressively, with an RMSE of 0.957, this performance surpassed even the best feasible outcome of the original model (RMSE of 0.9582 for embedding size 400), shown in Figure 9.

7.1.3 Hybrid model

The interpretation of the results is split over the two distinct domain sets and addresses the influence of stemming at the end.

7.1.3.1 Beauty & Appliances

Upon examination of the RMSE scores of the experiments, Table 11 shows that a discernible pattern emerges: within the context of this domain, the integration of content features offers no apparent enhancement. Rather, the results exhibit a deterioration when compared to instances where content features were omitted. This phenomenon could potentially be attributed to the introduction of an abundance of novel information. This hypothesis finds some tentative support in the correlation between the newly introduced content features and those extracted using the Rating Pattern Extractor.

As presented in Figure 17, the correlation between the two feature sets tends to be minimal or virtually non-existent. This observation hints at the possibility that the newly introduced content features encapsulate distinctive information previously not considered. Particularly in the context of a relatively small dataset, this influx of distinct information could potentially lead to model confusion: the additional diverse information introduced to the model might make it harder for the model to accurately learn and generalise patterns from the data. This can result in the model being uncertain or less accurate in its predictions, leading to a decrease in performance. Nonetheless, it is important to note that while the data could possibly explain the phenomenon, arriving at definitive conclusions in this regard would require a more extensive and rigorous investigation and proof.

Additionally, when looking at the parameter settings in Table 11, the recorded hyperparameter values among the various experiments highlight a notable level of sensitivity and variability. These tendencies could potentially be linked to the complex layered structure of the model, and in part, to the challenges posed by data sparsity that the hyperparameter optimisation algorithm faces.

7.1.3.2 Office & Movies

The experiments on this domain set show that the incorporation of content features does yield a slight enhancement in model performance (Table 12). However, the effect is notably marginal, reflected in only a subtle change in the achieved outcomes. This marginal effect can be attributed to the features are not contributing useful information. The experiments show that the positive influence increases marginally with more features, but this is an extremely small increase.

Figure 18 displays the correlation between the features extracted by the RPE and the 200 content features for the Office & Movies domain. Similar to the Beauty & Appliances domain set, a minimal to non-existent relationship between these features is observed. Given the larger scale of the Office & Movies domain, the addition of content features may provide a marginal assistive effect instead of confusion.

Additionally, Table 12 highlights substantial variations in hyperparameter values for the optimal models of each experiment. This could imply that the model's sensitivity is not overly pronounced. Nonetheless, the marginal improvement in results, while possibly stemming from the introduction of novel features, remains of such small magnitude that its definitive causation is challenging to ascertain.

7.1.3.3 Stemming

In both domain sets, the effects of stemming appear to be negligible and inconclusive. The outcomes exhibit striking similarities across the experiments, demonstrating that stemming does not consistently lead to either superior or inferior results. In scenarios where the integration of content features yield negative effects, applying stemming would not possess the inherent ability to counteract the adverse effects. Similar for the scenario where the hybrid model performs marginally better, applying stemming does not amplify or negate that effect. Stemming is a technique intended to enhance word alignment, especially in situations where words hold significant weight in improving recommendations. Nevertheless, in this context, the variations in terms that stemming seeks to address do not hold responsibility for the performance of the hybrid model.

7.2 Limitations

The mentioned outcomes are set against a backdrop of certain inherent challenges and complexities. The model's instability and sensitivity to hyperparameters are notable considerations. The highly contingent behaviour of the model in response to parameter variations could be attributed, at least in part, to its stacked architecture, which potentially amplifies the influence of hyperparameter changes. Furthermore, the utilisation of the hyperparameter optimisation algorithm reveals potential struggles with data sparsity. This aspect raises the question of whether the algorithm's efficacy is hindered when faced with sparse datasets.

An additional factor impacting the experiment results is the inherent data shuffling in the absence of a seedable data loader. Despite consistent and seeded splits, the variability in data ordering in the dataloader introduces an element of randomness that might influence the model's performance.

It is crucial to acknowledge that the results are based on certain assumptions, randomness, and initialisation conditions. The convergence of the model, particularly in the context of the stacked architecture, might be sensitive to these aspects, and such variability could introduce noise into the results. While the results appear promising and warrant further investigation, it's crucial to approach these findings with caution.

7.3 Business Case

Embedded within the research's foundational business rationale, the experimental insights hold significant relevance. Across different domain sets, these experiments have yielded valuable findings.

For the Beauty & Appliances domain set, the results underscore that even with access to small, sparse datasets, the bare DARec model (without content features) remains proficient in generating commendable recommendations. The AutoEncoder experiments reveal the efficacy of a well-trained encoder in filling missing ratings. Additionally, they demonstrate that supplementing content features in situations with limited data could, paradoxically, lead to inferior outcomes.

In the context of the Office & Movies domain set, the experiments indicate that while the inclusion of content features contributes positively to model performance, the marginal improvement achieved does not seem to warrant the substantial resource investment. The intricate process of sourcing, extracting, re-training, and optimising hyperparameters may not be proportionate to the modest enhancement in recommendation accuracy. In summary, these outcomes shed light on the delicate balance between resource investment and potential gains. Companies seeking to fine-tune their recommendation systems can benefit from these insights when deciding whether to incorporate content features, particularly in scenarios where the incremental enhancement in accuracy might not outweigh the associated complexities and costs.

7.4 Future Work

The realm of future research offers several intriguing avenues for further exploration. One such direction for future exploration involves broadening the scope of investigation to include a diverse array of domain sets. This approach would yield a more holistic understanding of how content features interplay with different domains and dataset sizes. The current selection of two domain sets was influenced by the potential for meaningful comparison and considerations of computational efficiency and focused on domain sets that were extremely dissimilar. Subsequent research could concentrate on domains sharing more similar content characteristics, or explore scenarios with varying sets of mutual users, including intermediary sizes or significantly larger user sets.

Due to temporal limitations, the scope of the experiments was constrained. Extending the experimental framework holds promise in yielding more comprehensive and dependable results. This expansion could shed further light on the discussed limitations, possibly offering insights to mitigate their impact, and potentially fortify the validity and reliability of the outcomes.

Lastly, a deeper exploration of alternative content-based transfer techniques could shed light on potential enhancements to the recommendation accuracy. The extraction of content-based features was undertaken using a straightforward approach, without undergoing extensive optimisation. The simplicity of the method and the availability of the data was a driving factor for the decision. While the chosen method offered an initial insight into the influence of content features on the recommendation accuracy of a CDR system, further investigations should encompass more advanced and optimised content feature extraction methods. This exploration has the potential to illuminate the various facets of content-based transfer learning and highlight how the selection of content-based transfer techniques can significantly influence recommendation accuracy.

8 Conclusion

The research explored the integration of content-based and rating pattern-based transfer learning techniques to improve the recommendation accuracy of cross-domain recommender systems. The findings reveal a nuanced picture of the impact of content features on different domain sets. In the case of the smaller Beauty & Appliances domain set, the addition of content features had a negative influence on the model's performance. This points to the possibility that, within such a constrained domain, the addition of new information may lead to confusion and potentially outweigh the benefits. Conversely, for the larger Office & Movies domain set, a slight positive impact was observed. However, these improvements, though intriguing, were marginal. Hence, providing a straightforward answer to the research question posed in Section 1.2 ("To what extent does the combination of content-based and rating pattern-based transfer learning strategies positively influence the recommendation accuracy of cross-domain recommendation systems?") is not feasible. Depending on the domain set, size of the data, and the number of the added content features, the impact on the recommendation accuracy differs.

Furthermore, it is important to acknowledge the model's inherent instability and sensitivity to hyperparameters, which could partly be attributed to the complex stacked architecture and the challenges posed by data sparsity. Likewise, the inability to seed the dataloader introduced variability in the data order for training and testing, impacting the consistency of results.

Future research holds promise in diverse directions. Exploring varied domain sets and optimising the content feature extraction steps could enhance understanding and results' reliability. Investigating alternative content-based transfer techniques could unveil avenues for boosting recommendation accuracy in cross-domain systems.

In conclusion, it is imperative to approach the findings with caution. While the incorporation of content features holds promise for potential improvements, their impact on recommendation accuracy varies from minimal to potentially adverse. The small enhancements observed within domain set Office & Movies additionally prompt consideration regarding the cost-benefit analysis of content feature integration and hybrid model training.

References

Accenture. (2016). *Consumers welcome personalized offerings but businesses are struggling to deliver, finds accenture interactive personalization research*. https://newsroom.accenture.com/news/consumers-welcome-personalized-offerings-but-businesses-are-struggling-to-deliver-finds-accenture-interactive-personalization-research.htm

Agarwal, N. (2021, March 31). *The impact of intelligent recommendations on sales* [LinkedIn]. Retrieved August 22, 2023, from https://www.linkedin.com/pulse/impact-intelligent-recommendations-sales-neeraj-agarwal/

Bell, R. M., & Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 43–52. https://doi.org/10.1109/ICDM.2007.90

Berkovsky, S., Kuflik, T., & Ricci, F. (2007). Cross-domain mediation in collaborative filtering. *User Modeling 2007: 11th International Conference, UM 2007, Corfu, Greece, July 25-29, 2007. Proceedings 11*, 355–359.

Bobadilla, J., Ortega, F., Hernando, A., & Gutierrez, A. (2013). Recommender systems survey [ISBN: 0950-7051 Publisher: Elsevier]. *Knowledge-based systems*, *46*, 109–132.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. in proceedings of fourteenth conference on uncertainty. *Artificial Intelligence, Madison, Wisconsin. Morgan Kaufmann.*

Cantador, I., Fernández-Tobıas, I., Berkovsky, S., & Cremonesi, P. (2015). Cross-domain recommender systems. *Recommender systems handbook*, 919–959.

Cheng, Z., Ding, Y., Zhu, L., & Kankanhalli, M. (2018). Aspect-aware latent factor model: Rating prediction with ratings and reviews. *Proceedings of the 2018 world wide web conference*, 639–648.

Cooper, J. (2018, September 3). *How amazon generate 35% of their revenue*. Retrieved August 22, 2023, from https://www.linkedin.com/pulse/how-amazon-generate-35-revenue-james-cooper/

Cremonesi, P., Tripodi, A., & Turrin, R. (2011). Cross-domain recommender systems. *2011 IEEE 11th International Conference on Data Mining Workshops*, 496–503.

Desrosiers, C., & Karypis, G. (2010). A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, 107–144.

Dias, M. B., Locher, D., Li, M., El-Deredy, W., & Lisboa, P. J. (2008). The value of personalised recommender systems to e-business: A case study. *Proceedings of the 2008 ACM conference on Recommender systems*, 291–294.

Elkahky, A. M., Song, Y., & He, X. (2015). A multi-view deep learning approach for cross domain user modeling in recommendation systems. *Proceedings of the 24th international conference on world wide web*, 278–288.

F. Maxwell Harper, J. (2005). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *5*, 4.

Falk, K. (2019). *Practical recommender systems*. Simon; Schuster.

Fernández-Tobıas, I., Cantador, I., Tomeo, P., Anelli, V. W., & Di Noia, T. (2019). Addressing the user cold start with cross-domain collaborative filtering: Exploiting item metadata in matrix factorization. *User Modeling and User-Adapted Interaction*, *29*, 443–486.

Fraihat, S., & Shambour, Q. (2015). A framework of semantic recommender system for e-learning. *Journal of Software*, *10*, 317–330. https://doi.org/10.17706/jsw.10.3.317-330

Fürnkranz, J., & Kliegr, T. (2015). A brief overview of rule learning. *Rule Technologies: Foundations, Tools, and Applications: 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings 9*, 54–69.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. *International conference on machine learning*, 1180–1189.

Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., & Guo, J. (2013). Cross-domain recommendation via cluster-level latent factor model. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part II 13*, 161–176.

Gogna, A., & Majumdar, A. (2015). Matrix completion incorporating auxiliary information for recommender system design. *Expert Systems with Applications*, *42*(14), 5789–5799.

Gope, J., & Jain, S. K. (2017). A survey on solving cold start problem in recommender systems. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 133–138. https://doi.org/10.1109/CCAA.2017.8229786

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, *5*(4), 1–19.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 230–237. https://doi.org/10.1145/312624.312682

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (2017). An algorithmic framework for performing collaborative filtering. *SIGIR Forum*, *51*(2), 227–234. https://doi.org/10.1145/3130348.3130372

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 89–115.

Hofmann, T., & Puzicha, J. (1999). Latent class models for collaborative filtering. *IJCAI*, *99*(1999).

Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., & Zhu, C. (2013). Personalized recommendation via cross-domain triadic factorization. *Proceedings of the 22nd international conference on World Wide Web*, 595–606.

Isinkaye, F., Folajimi, Y., & Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/https://doi.org/10.1016/j.eij.2015.06.005

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: An introduction*. Cambridge University Press.

Jiang, Y., Shang, J., & Liu, Y. (2010). Maximizing customer satisfaction through an online recommendation system: A novel associative classification model. *Decision Support Systems*, *48*(3), 470–479. https://doi.org/https://doi.org/10.1016/j.dss.2009.06.006

Kanagawa, H., Kobayashi, H., Shimizu, N., Tagami, Y., & Suzuki, T. (2019). Cross-domain recommendation via deep domain adaptation. *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part II 41*, 20–29.

Khan, M. M., Ibrahim, R., & Ghani, I. (2017). Cross domain recommender systems: A systematic literature review. *ACM Computing Surveys (CSUR)*, *50*(3), 1–34.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. https://doi.org/10.1109/MC.2009.263

Koren, Y., Rendle, S., & Bell, R. (2011). Advances in collaborative filtering. *Recommender systems handbook*, 91–142.

Kumar, A., Kumar, N., Hussain, M., Chaudhury, S., & Agarwal, S. (2014). Semantic clustering-based cross-domain recommendation. *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 137–141.

Kumbhare, T. A., & Chobe, S. V. (2014). An overview of association rule mining algorithms. *International Journal of Computer Science and Information Technologies*, *5*(1), 927–930.

Leung, C. W.-k., Chan, S. C.-f., & Chung, F.-l. (2007). Applying cross-level association rule mining to cold-start recommendations. *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, 133–136.

Li, B. (2011). Cross-domain collaborative filtering: A brief survey. *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, 1085–1086. https://doi.org/10.1109/ICTAI.2011.184

Li, B., Yang, Q., & Xue, X. (2009). Transfer learning for collaborative filtering via a rating-matrix generative model. *Proceedings of the 26th annual international conference on machine learning*, 617–624.

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, *7*(1), 76–80.

Loni, B., Shi, Y., Larson, M., & Hanjalic, A. (2014). Cross-domain collaborative filtering with factorization machines. *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings 36*, 656–661.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, 73–105.

MacKenzie, I., Meyer, C., & Noble, S. (2013, October 1). *How retailers can keep up with consumers* [McKinsey & company]. Retrieved August 22, 2023, from https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

McAuley, J., Targett, C., Shi, Q., & Hengel, A. v. d. (2015, June 15). Image-based recommendations on styles and substitutes. Retrieved February 20, 2023, from http://arxiv.org/abs/1506.04757

Melville, P., Mooney, R. J., Nagarajan, R., et al. (2002). Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, *23*, 187–192.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Mongia, A., Jhamb, N., Chouzenoux, E., & Majumdar, A. (2020). Deep latent factor model for collaborative filtering. *Signal Processing*, *169*, 107366.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, *13*, 393–408.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web: Methods and strategies of web personalization* (pp. 325–341). Springer.

Porter, M. F. (1980). An algorithm for suffix stripping [ISBN: 0033-0337 Publisher: MCB UP Ltd]. *Program*, *14*(3), 130–137.

Resnick, P., & Varian, H. R. (1997). Recommender systems [ISBN: 0001-0782 Publisher: ACM New York, NY, USA]. *Communications of the ACM*, *40*(3), 56–58.

Ricci, F., Rokach, L., & Shapira, B. (2010, October 21). Recommender systems handbook [Journal Abbreviation: Recommender Systems Handbook]. In *Recommender systems handbook* (pp. 1–35). https://doi.org/10.1007/978-0-387-85820-3_1

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for idf. *Journal of documentation*, *60*(5), 503–520.

Sahebi, S., & Walker, T. (2014). Content-based cross-domain recommendations using segmented models. *CBRecSys@ RecSys*, 57–64.

Salesforce. (2017). *Personalization in shopping*. https://www.salesforce.com/products/commerce-cloud/resources/personalized-shopping/

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*, 285–295.

Sedhain, S., Menon, A., Sanner, S., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering, 111–112. https://doi.org/10.1145/2740908.2742726

Serrano, S. (2023, June 8). *Personalized product recommendations tactics for profits* [Barilliance]. Retrieved August 22, 2023, from https://www.barilliance.com/personalized-product-recommendations-stats/

Shapira, B., Rokach, L., & Freilikhman, S. (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, *23*, 211–247.

Tan, S., Bu, J., Qin, X., Chen, C., & Cai, D. (2014). Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, *127*, 124–134. https://doi.org/10.1016/j.neucom.2013.08.034

Tang, J., Wu, S., Sun, J., & Su, H. (2012). Cross-domain collaboration recommendation. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1285–1293.

Wang, J., & Lv, J. (2020). Tag-informed collaborative topic modeling for cross domain recommendations. *Knowledge-Based Systems*, *203*, 106119.

Yuan, F., Yao, L., & Benatallah, B. (2019, May 26). DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. Retrieved February 22, 2023, from http://arxiv.org/abs/1905.10760

Zhang, Q., Hao, P., Lu, J., & Zhang, G. (2019). Cross-domain recommendation with semantic correlation in tagging systems. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Zhang, X., & Wang, H. (2005). Study on recommender systems for business-to-business electronic commerce. *Communications of the IIMA*, *5*(4), 8.

Zhao, F., Xiao, M., & Guo, Y. (2016). Predictive collaborative filtering with side information. *IJCAI*, 2385–2391.

Zhu, F., Wang, Y., Chen, C., Zhou, J., Li, L., & Liu, G. (2021, March 2). Cross-domain recommendation: Challenges, progress, and prospects. Retrieved February 9, 2023, from http://arxiv.org/abs/2103.01696

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555