St. Pölten University of Applied Sciences

/media & digital technologies

Recommender Systems

Investigation of Recommender Systems and their potential and applications with Machine Learning methods

Research Paper

Bachelor's degree programme Media Technology
at the St. Pölten University of Applied Sciences

by:
Lukas Ganster
1810261031

Supervising tutor: FH-Prof. Dipl.-Ing. Dr. Markus Seidl, Bakk.

St.Pölten, 26.06.2020

Declaration

- The attached research paper is my own original work undertaken in partial fulfilment of my degree.

- I have made no use of sources, materials or assistance other than those which have been openly and fully acknowledged in the text. If any part of another person’s work has been cited, this either appears in inverted commas or (if beyond a few lines) is indented.

- Any direct citation or source of ideas has been identified in the text by author, date, and page number(s) immediately after such an item, and full details are provided in a bibliography (reference list) at the end of the text.

- I understand that any breach of the fair practice regulations may result in a mark of zero for this research paper and that it could also involve other repercussions.

St.Pölten, 26.06.2020	
....	....
Place, Date	Signature

Abstract

In the last two decades, technology has dramatically reduced the barriers of publishing and distributing information. In the digital age, users no longer have to laboriously search for suitable products, but are supported by innovative systems. A Recommender System is a computational mechanism for information filtering, where users provide recommendations (in the form of ratings or selecting items) as inputs, which the system then aggregates and directs to appropriate suggestions. In developing such systems, they have identified various approaches, all of which have their advantages and limitations. More efficient techniques from Artifical Intelligence take the possibilities of pattern recognition, which can play an essential role in generating recommendations, to a new level. Most of the more successful Internet services today, such as Netflix or YouTube, have integrated these technologies in some form. And the success to be recognized, for example concerning sales numbers or music plays, proves them right. This scientific work gives an overview of how recommendations are generated and explains the appropriate methods of Artifical Intelligence. Furthermore, practical examples are also used to identify the direct benefits.

Table of Contents

1 Introduction

Recent developments in electronic media especially in computer science and the rapid growth of the Web have allowed the creation of large and distributed available data sets. The age of information overload and the possibility to immediately access various resources gives rise to completely new challenges.

Recommender Systems (RS) are data analytics applications and are widely discussed in literature as they provide a solution to problems of information overload in a variety of contexts and application areas (Myrén and Neto 2017, p. 1). They help consumers to have proper choices regarding personal preferences and past behavior.

In the era of Big Data, the decision-making processes are much more difficult than ever. It is hard to evaluate the overwhelming number of alternatives, which comes with an increasing number of possible choices. This problem is commonly referred to as the “Paradox of Choice” (Schwartz, 2005).

For this scientific work only secondary research was conducted. The literature used to write this thesis was discovered by searching the Internet, specifically by utilizing the search engines Google Scholar and Google. To obtain the articles, papers, studies and books cited in this paper, libraries, journals and databases such as Google Books, Springer, Research Gate, Project MUSE and OpenThesis were searched. The aim of this paper is to give an overview of what the current state of the art methods for RS are. Important keywords that were viable to conduct this research paper were “Recommender System”, “recommendations”, “Collaborative Filtering”, “Machine Learning”, “recommender algorithm” and “Deep Learning”.

This research paper begins by introducing the basics of *Artificial Intelligence (AI)* and examining its relevance in the technical area in Section 2. Followed by an overview of advantages, limitations, and common approaches for the implementation of RS in Section 3. Of particular interest in Section 4 is the design of recommendations with the use of AI which highlights important new opportunities. After this, Section 5 will demonstrate the application of recommendations using examples from industry. Section 6 summarizes the results of the previous Sections and discusses possible further uses of the presented techniques.

2 Artificial Intelligence

The discipline of AI in general is gaining more and more attention in many applications such as *Computer Vision*, *Decision Making* and *Speech Recognition*. This Section explains the basic terms of AI and gives a brief historical overview to understand how fast this field is moving forward.

2.1 Brief historical overview

In order to gain an understanding of the historical context of this topic and how fast it is developing, a short historical outline is given. Commonly known as the official birth is the year 1956, where John McCarthy organized a two-month workshop at Dartmouth. The workshop did not bring any new findings or breakthroughs, but it introduced the attendees, ten in total, to each other. For the next two decades these people were the pioneers in the field of AI (Russell et al., 2010, p. 17). During the so-called “golden years”, there was a lot of enthusiasm, as Frank Rosenblatt’s Perceptron proved that machines could simulate the human brain and Joseph Weizenbaum developed the first chatbot, which was named as ELIZA in 1966 (Calomme, 2016).

After the first few successes from researches, the industry endured two AI winters from 1974 to 1980 and from 1987 to 1993, which refers to a period of reduced interest of the public and less subventions as companies failed to deliver innovations. In 1997 IBM developed DEEP BLUE, the first chess-playing system to win a match against the reigning world champion Gary Kasparov (Reynoso, 2019).

From 2011 until now the concepts of big data, deep learning, and data science compared with more powerful computing systems brought AI to an exceptional level. Nowadays, for companies like Facebook, Amazon and Google it is usual to incorporate any kind of artificial component in their products.

2.2 Machine Learning

In order to mimic the human intelligence, to act independently and intelligently, machines have to learn like humans. The art of making decisions needs a fundamental comprehension of the surrounding world. When it comes to computer science, this approach is called *Machine Learning (ML)*, which is a simple way of achieving AI (Jordan and Mitchell, 2015, p. 255).

With advances of enormous computing power, today's computers have the competence to store, process and extract large amounts of data, which is essential in ML. Using these datasets, a computer program can detect certain patterns and learn showing behavior that the programmer or author may be completely unaware of. The data needs to get processed by the program after a metric indicates the distance between the current behavior and the ideal behavior. Subsequently, a feedback mechanism is necessary that uses the distance to guide the program in order to produce a result with better accuracy (Joshi, 2020, p. 4). In literature there are several models of ML discussed, but generally they can be divided in three sub-categories: *Supervised Learning, Unsupervised Learning* and *Reinforcement Learning* (Heidenreich, 2018; Jordan & Mitchell, 2015). A basic understanding of how these intelligent methods actually work is vital to follow the later Sections of this research paper.

Supervised Learning knows the input variables and uses an algorithm to learn how to adjust to the output variables that are also known. This is usually done with large training data sets and aims to align the ML function so that the result can be predicted as accurately as possible. The learning mechanism is therefore based on a pre-defined output to be learned, the result of which is known. In this approach, learning is done iteratively until an acceptable result is obtained (Russell et al., 2010, p. 695).

Unsupervised Learning also knows the input variables, but has no information about associated output variables. It operates with data that is not labeled or classified in any way, and then tries to find certain structures in these data sets. The algorithm tries to identify commonalities from the available data and reacts to new data depending on the presence or absence of these commonalities (Joshi, 2020, p. 11).

Reinforcement Learning is a separate type and must be considered independently of the supervised and unsupervised approach. No labelled data is required in advance, instead, the system interacts with the environment and uses that feedback for its learning. Thus, during the training phase, a trial-and-error procedure is used to continuously improve the capability of this algorithm (Joshi, 2020, p. 11; Kaelbling, Littman, & Moore, 1996).

3 Recommender Systems

This Section deals with the definition of RS and how the user information is collected. Furthermore, it will point out what the goals, limitations, and problems when it comes to the development of such systems.

3.1 Introduction

When we are looking at today's society, it is often inevitable to make choices without enough experience of the alternative possibilities. In daily life, people rely on recommendations from other people by spoken words, guides, reviews, recommendation letters or similar.

RS, or Recommendation Systems are software tools that recommend a selection which could be the most interesting out of a huge amount of data. To find interesting content, the systems gather information about the users and subsequently uses them to generate proper recommendations (Muno, 2008, p. 3). In Figure 1 the functionality of producing the recommendations is illustrated.

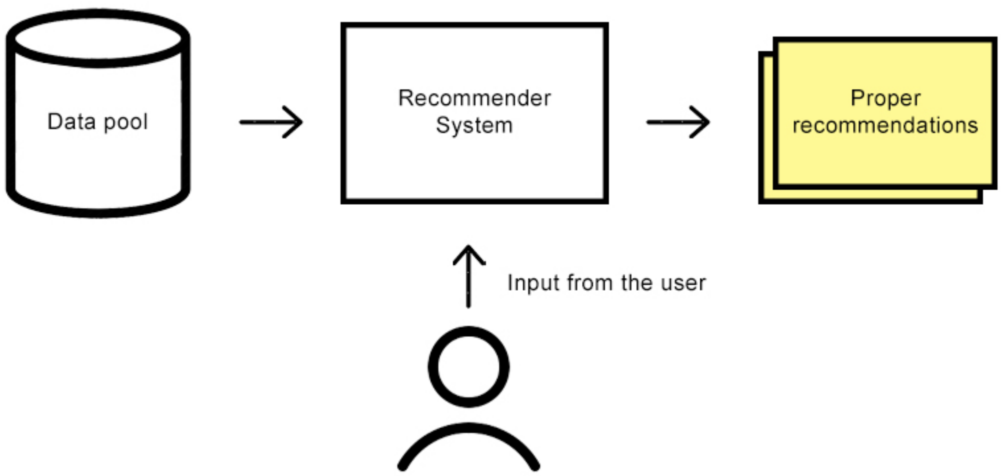

Figure 1. Basic functionality of RS (data based on Muno, 2008, p. 3)

However, first the system must gather and save enough information about the user preferences. There are many points that can be seen as information for the RS such as personal interests, hobbies, browser history as well as age, size, and weight of the user.

3.2 Collect user information

In order to generate meaningful and personalized recommendations based on the user behavior, the system needs, as mentioned in the previous Section, to gather information about the user. In general, this process can be split into three different feedback categories: *implicit approach, explicit approach* and *mixing approach* (Lampropoulos and Tsihrintzis, 2015, p. 5).

Implicit approach

When a RS uses the implicit approach, it does not require attendance from the user, instead, the behavior of the user gets recorded. The system tries to learn steadily and therefore notices how the user reacts to each incoming piece of data, which makes it easy to collect in large quantities without any effort from the user. An example could be how long the user remains on a certain website and which ones he visits afterwards. Browsing history, frequently visited websites and the count of the number of times a song is played could be beneficial for generating tailored recommendations (Aggarwal, 2016, p. 109).

Explicit approach

When it comes to the explicit approach, users must explicitly state their preference for a specific item. Explicit feedback requires additional input from the users, which usually happens by ratings or like and dislike options. A well-known form of user-provided ratings is a 1-to-5-star scale. Commonly positive or high ratings indicate that the user likes an item, while negative or lower ratings are expressing few interest. However, explicit feedback may be spare as such ratings involve additional work by the user who probably does not see the benefit of it (Jannach, Lerche, and Zanker, 2018, p. 5).

Mixing approach

A system that uses the mixing approach combines implicit and explicit feedback. Users must rate a small number of items by their relevance for the purpose of generating meaningful output. The ratings are used as a training set for ML algorithms that are executed in order to generate user interest profiles (Lampropoulos and Tsihrintzis, 2015, p. 6).

3.3 Goals and purposes

In the previous Section the term RS was defined as software tools that are simply built to provide suggestions of certain items to the users. This definition has to be refined as the purposes are endless and the possible roles played by the RS are more diverse than we previously thought. To better understand the motivations why RS were introduced, we must distinguish between the roles of the service provider and the service consumer.

The most important function for a RS is probably to increase the number of sales. Especially for providers with systems in a commercial context, the rise of the conversion rate is a primary goal. Another major purpose is to sell more diverse items along with improving the experience of the user with the application. An essential part of a high acceptance of the recommendations consists of a combination of accurate recommendations and a well-designed interface (Ricci, Rokach, & Shapira, 2011, p. 5).

From a consumer's point of view, a RS should focus on finding a suitable item. In most cases, this is not always just one, but several, for example with the recommendation of music playlists. Some users like the idea of contributing ratings in the system because they assume that the community benefits from their input (J. L. Herlocker, Konstan, Terveen, & Riedl, 2004, p. 9).

3.4 Limitations and problems

In the last Section, the different usage scenarios of RS were presented, while the current Section is about the issues and limitations that go along. The literature discusses a wide variety of problems that come with different approaches when talking about tailored recommendations. Since listing all possible challenges would go beyond the scope of this work, the most important problems are briefly addressed in the following Section.

Cold start

One of the most discussed problems with RS, especially with collaborative based systems, is the cold start problem. It refers to a situation when almost nothing is known about a user or an item. Basically, this problem can be seen as a new-user problem or a new-item problem (Schein, Popescul, Ungar, & Pennock, 2002, p. 3).

- New-user problem describes the situation when almost nothing is known about the user preferences.
- New-item problem describes the situation when ratings are required for items that have not been rated yet.

Sparsity problem

The data sparsity problem describes the phenomen that users in general rate only a small number of items. The unavailability of a large number of ratings leads to less accurate recommendations results (Adomavicius & Tuzhilin, 2005, p. 740).

Scalability

Due to the fact that RS have to deal with ever increasing amounts of data, more and more computing power is required. The problem of scalability describes the situation that RS always have to deliver an output in a certain time, even with a huge number of users and items. (Xin, 2015, p. 18).

Gray sheep

Users that have an unusual taste in comparison to the rest of the users stand out and are often called “gray-sheep” in literature. Since these users have little in common with the others, it is usually difficult to produce tailor-made recommendations (Falk, 2019, p. 132).

3.5 Basic models of Recommender Systems

A RS can use many different techniques and methods to calculate a meaningful recommendation for a user. The simplest approach is based on simple “If-rules” that are entered into the system. Basically, similarities can be found either in the content of the items themselves, which is used in *Content-based Filtering (CB),* or among the users, which is used in *Collaborative Filtering (CF).* In order to remain within the scope of the work, only the two methods that are used most in practice are briefly introduced in the following Section.

3.5.1 Content-based Filtering

RS that go the content-based way only determine recommended items based on information about the active user. This is often done with the help of attributes that describe the items - therefore these systems are called content-based recommenders. The approach of CB tries to find similarities between the various items and then uses them to generate suitable recommendations (Adomavicius & Tuzhilin, 2005, pp. 735–737). Figure 2 shows this in a simplified way using a consumer who reads an article and gets another one provided by the system.

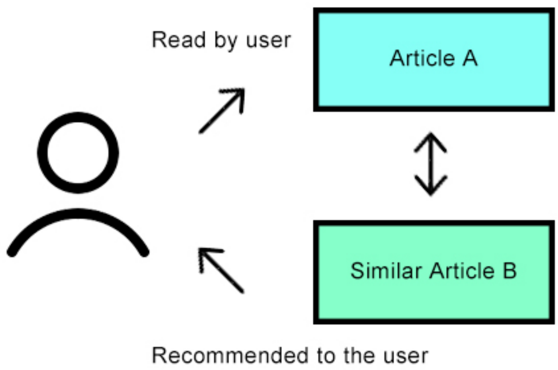

Figure 2. Content-based approach (data based on Mohamed, Khafagy, & Ibrahim, 2019)

Every single content that the system includes is provided with one or more keywords and thus classified into one or more categories. The set of the keywords is strictly domain-dependent: a RS for movies may describes each item with its director, its genre, the cast and so on, while a RS for music may describes each item with its artist, its length and the genre. These keywords or features are then used for trying to obtain the same or similar classifications. The classifications are then identical or may even complement each other. Based on the interest in a certain item, the system then indirectly learns which items could still be useful for the user. Since the similarity search is based only on the comparison of objects, this technique is called “Item-to-item-correlation” (Burke, 2002, p. 4).

Table 1. Example of a movie recommendation scenario

Film	**Comedy**	**Fantasy**	**History**	**Action**
A	2	9	0	6
B	0	1	8	2
C	7	3	0	2
D	1	8	0	6

Table 1 shows a simplified example of different movies evaluated in different categories. A recipient who would have interest in “Film A” would probably also, due to the similar categorization, have interest in “Film D”. He would then have “Film D” proposed by the RS because of the similar orientation of the two films.

So fundamentally, CB is dependent on two sources of data to work properly: items that are adequately described with keywords and a profile that describes information about the interests of the user, which was already addressed in Section 3.2. In a book recommendation scenario, for example, a user profile may store the preferred authors, genres and et cetera. Subsequently, the generated interest profile can then be compared with unseen items using the available characteristics (Terveen & Hill, 2001, pp. 7–9). Fundamentally, one can assume that the greater the overlap, the greater the relevance for the consumer.

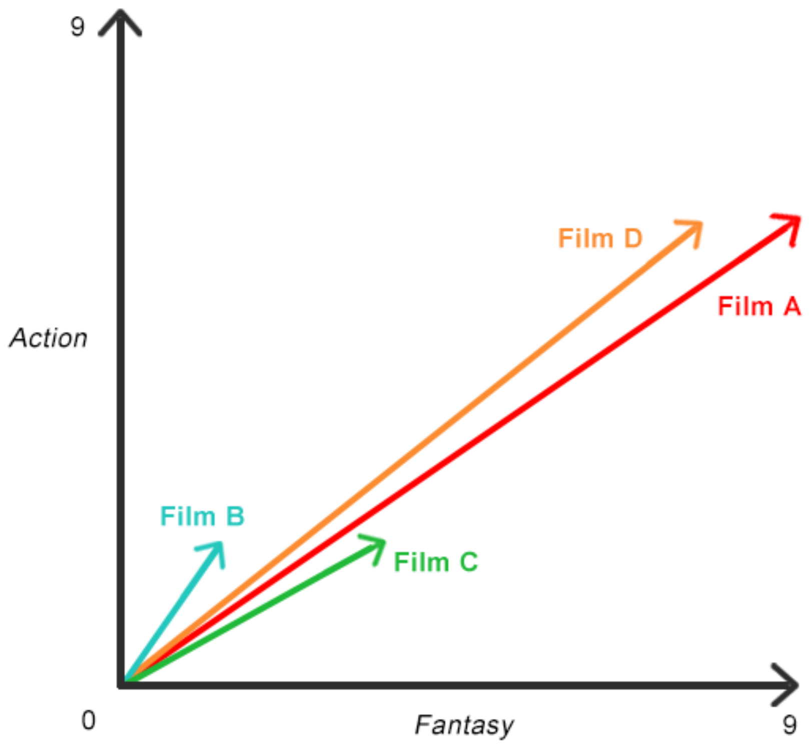

Figure 3. Simplified Vector Space Model

From a technical point of view, they work with feature vectors that either represent the interests of the user or the keywords of the items. In order to subsequently find similarities, the feature vectors are compared using *Vector Space Models*. The smaller the distance between two feature vectors, the greater the similarity of the objects (Salton, Wong, & Yang, 1975). To illustrate this, Figure 3 shows the previous example, which has been simplified with only two dimensions. Each vector represents one certain film.

3.5.2 Collaborative Filtering

In reality, advices are often based on the opinions of other people who are considered trustworthy or like-minded. This idea is also used in the implementation at the CF approach, where the recommendations are based on the preferences of similar people, the so-called nearest neighbours. RS who take a collaborative approach look for similarities among users. CF is a method in which the behaviour of user groups is evaluated and as a result the interests of individuals are inferred. The underlying idea here is that it can be assumed that people who have the same preferences for one product will also agree on other products.

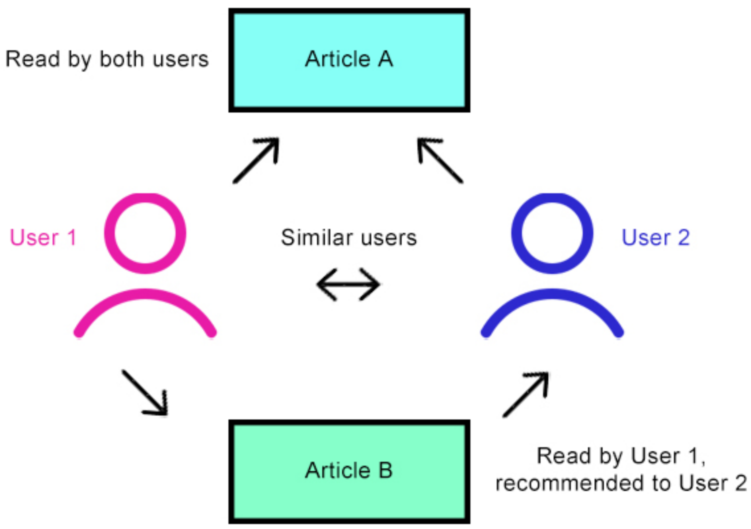

Figure 4. Collaborative approach (data based on Mohamed et al., 2019)

Figure 4 illustrates a scenario where User 2 gets an article provided by the RS. In the example, only the resemblance to a single user is established, while in practice many users with similar behavior are involved. So, if you want to predict what User 1 thinks about a certain article, the opinion of other users about this article or certain properties of the article is considered. However, only those users should be selected who also share the opinion of User 1 for other items. So instead of analyzing and describing the content of the items, these systems are based on the participation of the members in the community (J. Herlocker, Konstan, & Riedl, 2002, pp. 287–310).

In principle, the CF methods are classified as either *Model-based* or *Memory-based* (Adomavicius & Tuzhilin, 2005, pp. 734–749).

3.5.2.1 Model-based versus Memory-based

Model-based algorithms are mainly developed with techniques known from ML, such as *Cluster Model* or *Bayesian Network Model* for example (Yi Zhang & Koren, 2007, pp. 47–54). These approaches will be examined again in a later Section.

Memory-based algorithms, also referred to as *Neighborhood-based*, are the traditional CF methods that assume that comparable users would rate certain items similarly. These approaches are further divided in the literature into *Item-based* and *User-based* (Sarwar, Karypis, Konstan, & Riedl, 2001, pp. 285–295).

The literature mentions many possibilities to calculate the similarity of users or items. Two of the most important metrics are the *Pearson correlation* and the *cosine similarity* (Deshpande & Karypis, 2004, pp. 143–177; Lang, 1995, pp. 331–339).

Formula 1. Pearson correlation

$$\mathrm{simil}(x,y) = \frac{\sum\limits_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum\limits_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2} \sqrt{\sum\limits_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

In Formula 1 the similarity of users *x* and *y* is calculated using the Pearson correlation coefficiently. The set of items rated by both users is described as *Ixy* and the individual ratings as *r.*

Formula 2. Cosine similarity

$$\mathrm{simil}(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum\limits_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum\limits_{i \in I_x} r_{x,i}^2} \sqrt{\sum\limits_{i \in I_y} r_{y,i}^2}}$$

Formula 2 demonstrates how the cosine similarity calculation works. The variables for users, items and ratings are used here in the same way as in the previous formula. Many factors, such as cultural habits, could lead some people to be more critical or better at evaluating different items, which is something that is taken into consideration to some extent in cosine similarity.

3.5.2.2 User-based versus Item-based

In the following, the different aspects of item-based and user-based CF approaches are briefly described.

User-based Collaborative Filtering

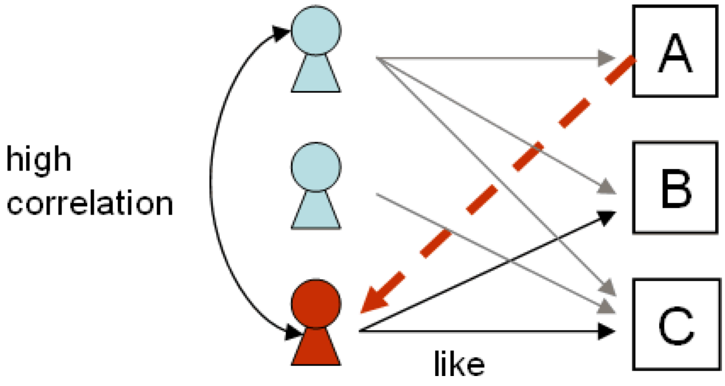

Figure 5. Drawing of User-based CF algorithm (Kim, 2006)

With the user-based filtering approach, different users are correlated via their (similar) ratings for specific products shown in Figure 5. In other words, users are classified into specific groups according to their preferences. This means that if a user in this group decides on an item, the RS can immediately recommend it to all users in this group (Aggarwal, 2016, p. 34; Pinela, 2017).

Item-based Collaborative Filtering

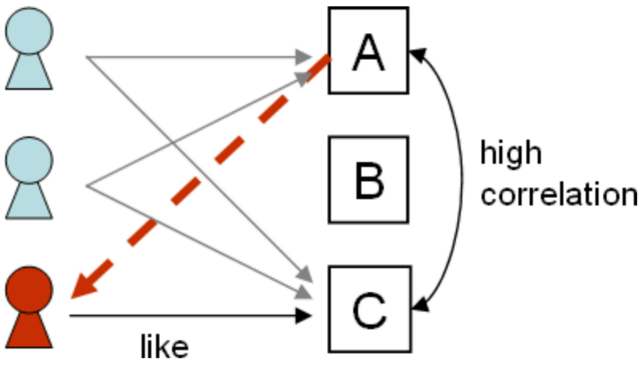

Figure 6. Drawing of Item-based CF algorithm (Kim, 2006)

With the item-based filter approach, different items are correlated by their (similar) ratings and afterwards similar items are recommended, as shown in Figure 6. In contrast to the content-based approach, which was described in Section 3.5.1, this approach only dispenses with any information that describes the item. A trivial example could be a restaurant that has been rated highly by some customers and is therefore recommended to the user (Aggarwal, 2016, p. 40; Sarwar et al., 2001).

4 Recommendations with Artificial Intelligence

Understanding how recommendations are generated for users with RS from scratch, it is interesting to see how this can be accomplished with the assistance of AI. Furthermore, this Section deals with the added value of using intelligent methods and briefly discusses some of these approaches.

4.1 Advantages compared to traditional methods

Just because the principles and practices of AI are used in many areas of computer science, does not mean that they are suitable for use at RS. In practice, methods from ML, but especially from its subset, *Deep Learning (DL),* are used to generate tailor-made recommendations for customers. However, before discussing individual methods in more detail, it should be made clear what advantage the application of these methods has for the system. As already discussed in a previous Section, most RS are driven by the goal of maximising revenues or increasing user satisfaction, which also poses a certain learning problem.

The majority of known approaches at RS that use techniques from DL are classified as model-based CF algorithms. In contrast to conventional approaches, Deep Neural Networks are able to model the non-linearity of data with nonlinear activation functions like sigmoid, hyperbolic tangent (tanh), Rectifier (ReLu) or others. This is particularly important when working with complex behaviour patterns (He et al., 2017). Modern frameworks such as Tensorflow, Keras, or DeepLearning4j are modular and offer a very high flexibility (S. Zhang, Yao, Sun, & Tay, 2019). Another major reason is that deep neural networks have delivered truly promising results in understanding natural language, speech recognition or machine translation. In short, the approaches of DL can lead to a paradigm shift in addressing recommendations in an age of huge data volumes (Yongfeng Zhang, Ai, Chen, & Croft, 2017).

4.2 Common methods

The approaches on how RS can be implemented with the help of a deep neural network are manifold in the literature. A complete overview of all state-of-the-art methods for deep learning based recommendations is provided by the surveys of

Zhang et. al. (2019) and Mu (2018). In the following two well-known representatives are picked out and briefly illuminated.

4.2.1 Restricted Boltzmann Machine

In the context of RS, one of the best-known applications of artificial neural networks are the *Restricted Boltzmann Machines (RBM)*. They are stochastic in nature, a special type of Boltzmann machine, and connect visible and invisible neurons. The visible units are often referred to as input layer, while the invisible ones are called hidden layer. The name-giving restriction is that the neurons within the individual layers are not connected to each other. The name-giving restriction is that the neurons within the individual layers are not connected to each other and that, compared to many other architectures used in machine learning, there is only one hidden Layer. The connections of the visible and invisible neurons form a bipartite graph and depending on the task the training can be supervised or unsupervised (Abdollahi & Nasraoui, 2016). Figure 7 illustrates a simple RBM.

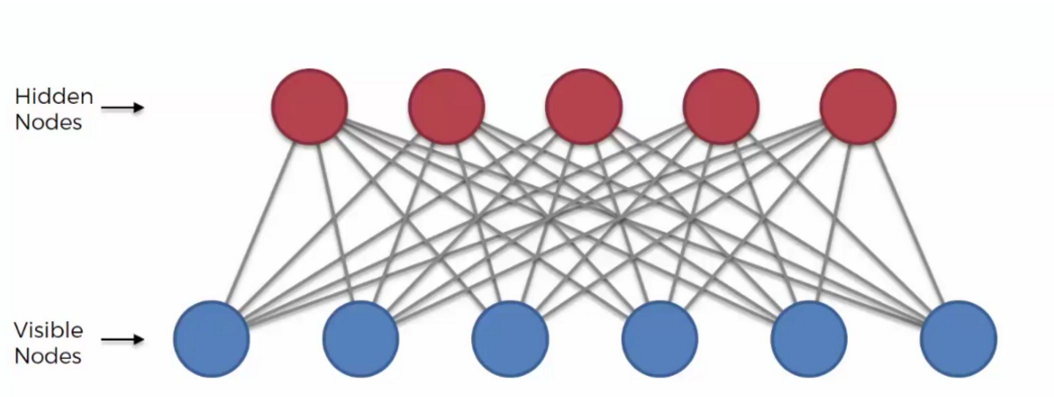

Figure 7. Restricted Boltzmann Machine (Nayak, 2019)

If the RBM is then fed with data throughout its life cycle, it will continuously improve and gain a better understanding of relationships between the data the more it receives. As an example, consider a case with several users and films where the system is supposed to identify commonalities in the data. Assuming that the data from Table 2 is our input, a 1 indicates that a user liked the movie and a 0 indicates that he did not like the movie. Cells that are empty mean that the user has not seen the movie.

Table 2. Data record that the RBM receives as input (data based on Thapliyal, 2018)

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0		1	1	1
User 2	0	1	0	0	1	0
User 3		1	1	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1		0	1
User 9		0	1	1	1	1
User 10	1		0	0		0
User 11	0	1	1	1	0	1

RBM manages to make visible the coherence of the data and concludes that users who like movie 3 and 4 will almost certainly also like movie 6. Furthermore, one can also assume that users who do not like movie 3 and 4, almost certainly do not like movie 6. This is a simplified representation of how similarities are found in the data. In the scenario of the movies, an RBM would also be able to identify the relationship of the movies in terms of genre or actors and can, for example, recommend movies with the same genre, protagonists or directors. This is visualized again in Figure 8.

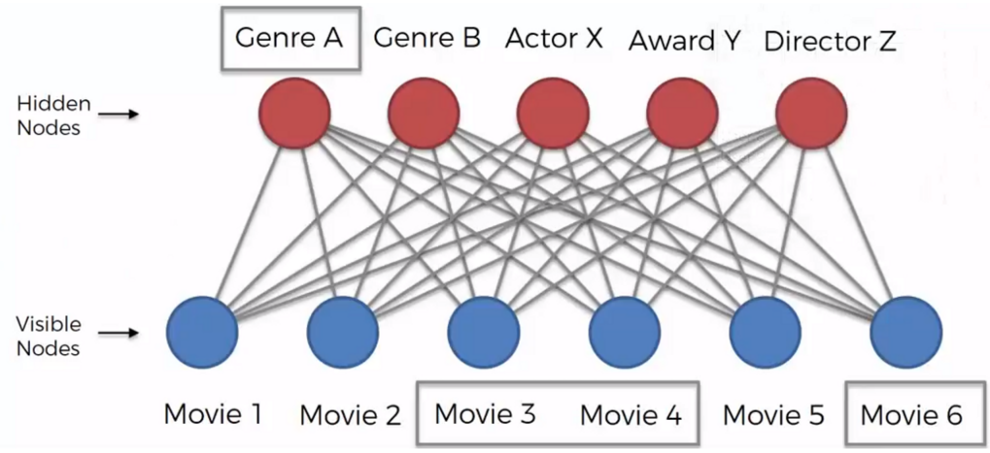

Figure 8. Restricted Boltzmann Machine in a movie scenario (Thapliyal, 2018)

4.2.2 Autoencoder

Another technique that is often applied in CF is the so-called *Autoencoder (AE).* It is an unsupervised model that was originally used to learn the representation of a set of input data, also known as encoding. In terms of architecture, AEs are a feedforward neural network and consist of an input layer, a hidden layer and an output layer, which is shown in Figure 9. The intention of learning is to use the output signal and the input signal as similarly as possible. (Bacuet, 2019; Mu, 2018).

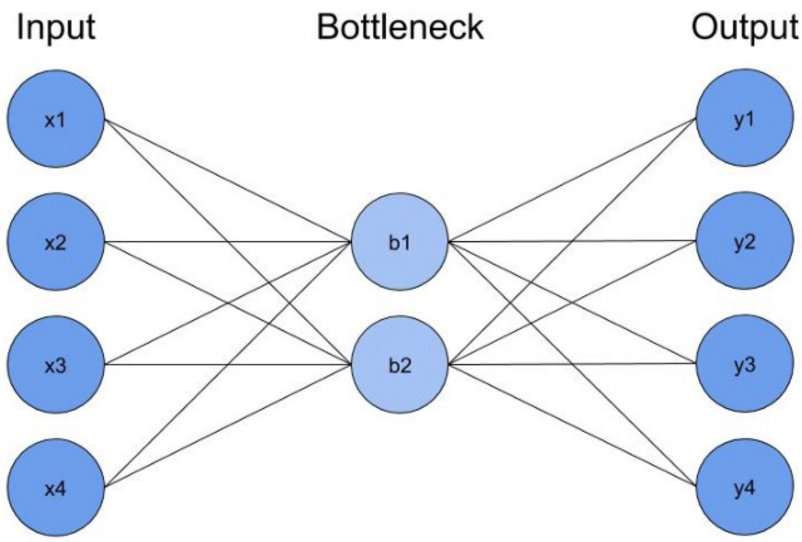

Figure 9. Architecture of an Autoencoder (Bacuet, 2019)

The data is encoded from the input layer into the hidden layer, which leads to a reduction of the shape. This reduction of dimension gives the hidden layer the name bottleneck. The bottleneck layer is usually smaller than the input layer, so the system can learn correlations between your individual data. Afterwards the information is decoded from this layer and returned to its original form.

Unfortunately this model, like many others in the DL field, is a black-box model, where it is not directly visible how the recommendations are generated (Haghighi, Seton, & Nasraoui, 2019).

Formula 3. Autoencoder decoding and encoding steps (Oppermann, 2020)

$$\phi : X \longrightarrow Z : x \mapsto \phi(x) = \sigma(Wx + b) := z$$
$$\varphi : Z \longrightarrow X : z \mapsto \varphi(z) = \sigma(\tilde{W}z + \tilde{b}) \ := x'$$

Mathematically, the transitions of the layers, i.e. steps of encoding (X to Z) and decoding (Z to X), are described as in Formula 3. The input data is described as vector x which is multiplied by a weight matrix and added with a bias term. The result is then applied to non-linear operations σ such as sigmoid, tanh or ReLu.

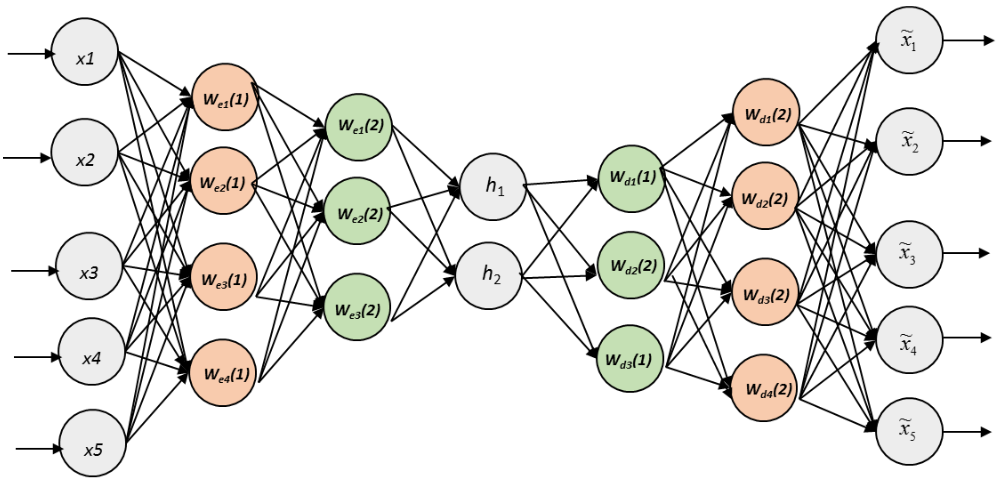

Figure 10. Deep Autoencoder architecture (Oppermann, 2020)

A more advanced form of the simple AE would be the *Deep Autoencoder (DA)*, which is shown in Figure 10. The only difference here is the additional number of hidden layers, which allows the method to process more complex data. This is especially useful in CF, where the implicit and explicit collected ratings of the users are processed together (Oppermann, 2020).

5 Usage in real life applications

In today's industry, an ever-increasing number of companies want to use the potential of RS as best as possible in their online presence. As we have heard in a previous Section, most RS are driven by a desire to maximize revenue or increase user satisfaction. In this Section, using examples from Netflix and YouTube will demonstrate how and for what these practices are used in real life applications.

5.1 Netflix

In late 2006, Netflix launched a public competition, widely known as the Netflix Prize, with the intention of creating a better CF algorithm for predicting user ratings for movies. The various proposed systems and approaches were measured using the *root mean squared error (RSME)* which is better the lower the value. At the start of the challenge, Netflix's in-house RS reached an RSME of 0.9529 and offered a million-dollar prize to anyone who could beat this benchmark by 10%. After the research field of RS had been brought into the medial spotlight, developers all over the world began to combine the most diverse approaches and methods with the aim of achieving the specified goal. One year later, a team achieved a major interim success by achieving an RSME of 0.88 with their approach. Their implementation was later integrated into the Netflix source code in a modified form and included a linear combination of *Matrix Factorization*, which is a type of CF, and RBM. It is also important to mention that the approach of the later winner in 2009 with a score of 0.85 was not adapted by the company because the integration would not have been profitable due to such a small difference (Amatriain & Basilico, 2015).

According to a paper published by Netflix Research, about 80 percent of streamed hours are traced back to their RS. The remaining 20 percent refer to individual searches. To achieve these enormously good values, the present Netflix recommender algorithm utilizes a wide range of implicit feedback. This includes data about the output device, the day of the week, time of day, how intensively the platform is used or even which recommendations the user has disregarded. With every piece of information that the system receives from the consumer, it is continuously retrained and improved in terms of accuracy. At the right time, the streaming provider realized that well-tuned and personalized recommendations combined with a good user experience offer incredible business value (Gomez-Uribe & Hunt, 2016).

Netflix leaves nothing to coincidence and places the top picks for the customer using a *Top N* video ranker at the top of the page. The objective of this algorithm is to provide the best proposals from the entire product catalogue (Gomez-Uribe & Hunt, 2016).

Personalization not only takes care of what content is presented to the consumer, but also how they sees it. Depending on the taste and individual preferences for certain actors, different thumbnails are suggested. Figure 11 exemplifies this with the movie “Pulp Fiction”. Depending on whether the user has seen more films with John Travolta or Uma Thurman, a different artwork is suggested for one and the same movie. Obviously, this technique is also applied to other factors such as category, which is also visualized in Figure 12. Users who tend to watch movies of the "romance" genre will see the upper thumbnail of "Good Will Hunting", while those who prefer to watch "comedy" will see the lower one (Amat, Chandrashekar, Jebara, & Basilico, 2018).

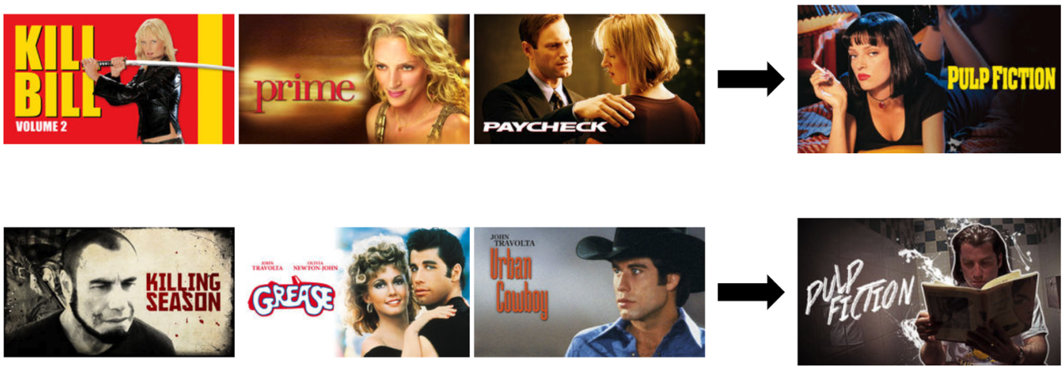

Figure 11. Personalized artworks at Netflix depending on actors (Amat et al., 2018)

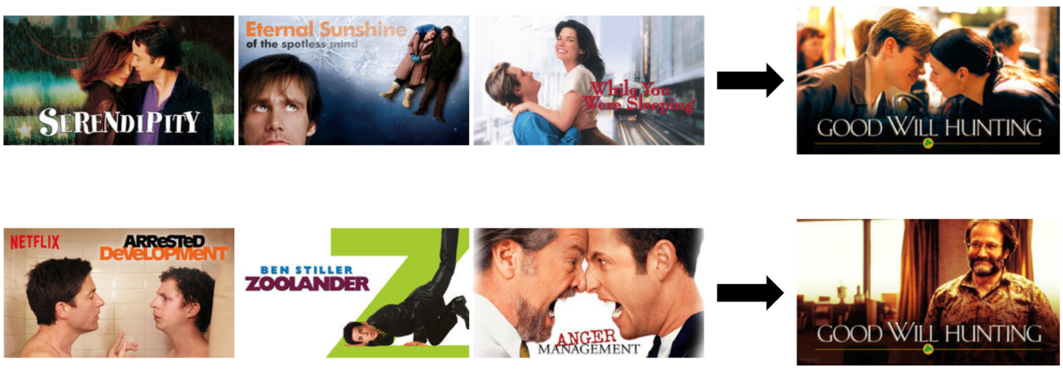

Figure 12. Personalized artworks at Netflix depending on genres (Amat et al., 2018)

5.2 YouTube

A further well-known example from practice would be the recommendation algorithm of YouTube. The parent company Google published a paper at the *10th ACM Conference on Recommender Systems* explaining how the platform manages to captivate its users with its content. The driving force behind the success was Google Brain, which was later opensourced as Tensorflow. With this framework it was possible for the whole world to train and test deep neural networks in an easy way. With the possibilities of DL it was possible to tackle many problems in the context of RS. The research paper addresses three major challenges for the YouTube algorithm. The first and perhaps most important problem has already been addressed in Section 3.4 and deals with scalability. Another challenge is the dynamic of the system and the associated balance between old proven content and new video uploads. And finally, the fact that data about the user may be rare, against which the system must be robust to some extent (Covington, Adams, & Sargin, 2016).

To summarize, the system consists of two neural networks that are responsible for *candidate generation* and *ranking*, which is shown in Figure 13. First, the history of the user's video views in the candidate generation network is used as input. CF-based methods are used to find out user similarities via features such as the videos viewed, search queries or user demographics. In the next network the videos are then ranked by relevance. This is accomplished with a rating function that gets a wide range of features that represents videos and users. Using this two-step approach, the system can filter out a manageable amount of videos that might interest the user from a huge number of available videos. In this sense, YouTube combines traditional approaches and extends them with methods of AI, which finally led to *Deep Collaborative Filtering* (Covington et al., 2016).

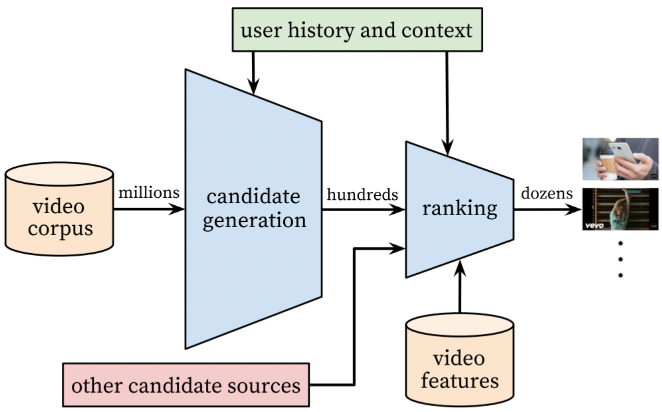

Figure 13. Recommendation system of YouTube (Covington et al., 2016)

6 Conclusion

The primary function of RS is to help the user finding content in an age of information overload by generating recommendations for the person's further search from information collected about the user. In recent years, they have become more and more popular, especially due to their application in e-commerce, where they help the user to identify the most relevant items to their personal preferences.

Practical examples have demonstrated that the implementation of systems combining the advantages of several approaches, in combination with ML techniques, raises quality to a completely new level. Fundamentally, it can be assumed that ML with the mechanisms of pattern recognition will influence many sub-areas of technology in the long term.

It is hard to overlook how indispensable a well-functioning system has become for the generation of personalized and tailored suggestions. With examples from industry such as Netflix and YouTube, you can see the tremendous growth and business value that a well-built RS can bring. The years of investing so much time and resources in research and development of RS and AI are now paying off.

The outstanding development of recommendation services has changed the way we think about them. As users we are aware that the algorithm almost knows us better than our partners or families do. The potential that can be associated with having a system that knows exactly what needs and requirements you need is sufficiently well known. Forthcoming research may bring even better systems than those we know today. The entire subject area will certainly be the focus of much discussion in the future and will generally be subjected to constant change.

References

Abdollahi, B., & Nasraoui, O. (2016). Explainable Restricted Boltzmann Machines for Collaborative Filtering. *ArXiv:1606.07129 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1606.07129

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook* (1st ed.). Springer Publishing Company, Incorporated.

Amat, F., Chandrashekar, A.,s Jebara, T., & Basilico, J. (2018). Artwork personalization at netflix. *Proceedings of the 12th ACM Conference on Recommender Systems*, 487–488. https://doi.org/10.1145/3240323.3241729

Amatriain, X., & Basilico, J. (2015). Recommender Systems in Industry: A Netflix Case Study. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 385–419). https://doi.org/10.1007/978-1-4899-7637-6_11

Bacuet, Q. (2019, April 2). How Variational Autoencoders make classical recommender systems obsolete. Retrieved June 2, 2020, from Medium website: https://medium.com/snipfeed/how-variational-autoencoders-make-classical-recommender-systems-obsolete-4df8bae51546

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, *12*. https://doi.org/10.1023/A:1021240730564

Calomme, Valentin. 2016. “THE HISTORY OF ARTIFICIAL INTELLIGENCE.” *Welove.At*. Retrieved April 30, 2020 (https://www.welove.ai/en/blog/post/history-of-artificial-intelligence.html).

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198. https://doi.org/10.1145/2959100.2959190

Deshpande, M., & Karypis, G. (2004). Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems*, *22*(1), 143–177. https://doi.org/10.1145/963770.963776

Falk, K. (2019). *Practical recommender systems*. Shelter Island, NY: Manning.

Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System:

Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems*, *6*(4), 1–19. https://doi.org/10.1145/2843948

Haghighi, P. S., Seton, O., & Nasraoui, O. (2019). An Explainable Autoencoder For Collaborative Filtering Recommendation. *ArXiv:2001.04344 [Cs]*. Retrieved from http://arxiv.org/abs/2001.04344

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, 173–182. https://doi.org/10.1145/3038912.3052569

Heidenreich, H. (2018, December 4). What are the types of machine learning? Retrieved May 9, 2020, from Medium website: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval*, *5*(4), 287–310. https://doi.org/10.1023/A:1020443909834

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

Jannach, D., Lerche, L., & Zanker, M. (2018). Recommending Based on Implicit Feedback. In P. Brusilovsky & D. He (Eds.), *Social Information Access* (pp. 510–569). https://doi.org/10.1007/978-3-319-90092-6_14

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Joshi, A. V. (2020). *Machine Learning and Artificial Intelligence*. Springer Publishing Company, Incorporated.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *ArXiv:Cs/9605103*. Retrieved from http://arxiv.org/abs/cs/9605103

Kim, J. (2006). *What is a Recommender System?* Bilbao.

Lampropoulos, A. S., & Tsihrintzis, G. A. (2015). *Machine Learning Paradigms: Applications in Recommender Systems*. Springer Publishing Company, Incorporated.

Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *In Proceedings of the 12th International Machine Learning Conference (ML95*, 331–339.

Mohamed, M., Khafagy, M., & Ibrahim, M. (2019, February 19). *Recommender*

Systems Challenges and Solutions Survey. https://doi.org/10.1109/ITCE.2019.8646645

Mu, R. (2018). A Survey of Recommender Systems Based on Deep Learning. *IEEE Access*, *6*, 69009–69022. https://doi.org/10.1109/ACCESS.2018.2880197

Muno, S. (2008). *Seminararbeit zum Thema: Recommender-Systeme*.

Myrén Adam, and Piotri Skupniewicz Neto. 2017. “Evaluation of Machine Learning Algorithms in Recommender Systems: Candidate Recommender Systems in the Staffing Industry.”

Nayak, M. (2019, April 18). An Intuitive Introduction Of Restricted Boltzmann Machine (RBM). Retrieved June 1, 2020, from Medium website: https://medium.com/datadriveninvestor/an-intuitive-introduction-of-restricted-boltzmann-machine-rbm-14f4382a0dbb

Oppermann, A. (2020, April 29). Deep Autoencoders For Collaborative Filtering. Retrieved June 2, 2020, from Medium website: https://towardsdatascience.com/deep-autoencoders-for-collaborative-filtering-6cf8d25bbf1d

Pinela, C. (2017, November 6). Recommender Systems—User-Based and Item-Based Collaborative Filtering. Retrieved June 22, 2020, from Medium website: https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f

Reynoso, Rebecca. 2019. “A Complete History of Artificial Intelligence.” Retrieved April 28, 2020 (https://learn.g2.com/history-of-artificial-intelligence).

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 1–35). https://doi.org/10.1007/978-0-387-85820-3_1

Russell, S. J., Norvig, P., & Davis, E. (2010). *Artificial intelligence: A modern approach* (3rd ed). Upper Saddle River: Prentice Hall.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

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. https://doi.org/10.1145/371920.372071

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '02, 253. https://doi.org/10.1145/564376.564421

Schwartz, B. (2005). *The Paradox of Choice*.

Terveen, L., & Hill, W. (2001). *Beyond Recommender Systems: Helping People Help Each Other*. 21.

Thapliyal, M. (2018, December 13). Restricted Boltzmann Machine. Retrieved June 1, 2020, from Mc.ai website: https://mc.ai/restricted-boltzmann-machine/

Xin, Y. (n.d.). *Challenges in Recommender Systems: Scalability, Privacy, and Structured Recommendations*. 128.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, *52*(1), 1–38. https://doi.org/10.1145/3285029

Zhang, Yi, & Koren, J. (2007). Efficient bayesian hierarchical user modeling for recommendation system. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 47–54. https://doi.org/10.1145/1277741.1277752

Zhang, Yongfeng, Ai, Q., Chen, X., & Croft, W. B. (2017). Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1449–1458. https://doi.org/10.1145/3132847.3132892

List of figures

List of tables