NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

# Personalization of product rankings in e-commerce

Josefine Frederike Kuka

Dissertation presented as partial requirement for obtaining the Master's degree in Advanced Analytics

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


# Personalization of product rankings in e-commerce

**Josefine Frederike Kuka**

November 2018

Dissertation presented as partial requirement for obtaining the Master's degree in Advanced Analytics

Nova IMS - Universidade Nova de Lisboa
Student number: M2016010
Supervisor: Miguel de Castro Neto

**Personalization of product rankings in e-commerce**
Josefine Frederike Kuka

# Abstract

Consumers face a large number of choices while shopping online. Studies have shown, that they are already expecting to be targeted with content addressing their personal needs. In a web shop, products are presented as lists based on a selected category or as results of a product search. To support the users in their decision making, they can be provided with a personalized product ranking fitted to their current interests.

In this piece of work, three levels of personalized product rankings are proposed: explicit personalization, cluster-based personalization and individualization. To estimate the potential effect of the personalization and its required effort, two prototypes for the second and third level are developed and evaluated. The prototypes are based on a previously existing non-personalized ranking, which ranks the products in descending order according to a sales prediction. The cluster-based prototype enhances this product ranking by determining customer clusters beforehand using both situative and behavioural data. The individualized product rankings rely on the combination of the ranking with a recommendation system realized as a matrix factorization. In doing so, the concept of learning to rank is considered.

By evaluating the cluster-based and individualized prototype on a sampled data set in comparison to the non-personalized ranking, it is shown that the created personalized rankings are in fact closer to the users' needs. Furthermore, a subjective evaluation confirms that the cluster-based rankings can reflect the users' interests in a better way.

**Keywords:**
Personalization, Ranking, Machine Learning, Learning to Rank, Clustering, Recommendation system, Matrix Factorization, E-commerce

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Consumers face a large number of choices when shopping online. Many web shops are providing different sorting options for product lists: descending and ascending price or rating, newest products or product popularity, whereas the latter is often the default sorting. However, the question arises what this popularity actually means: popularity with the customer or rather with the seller (Huke, 2011)?

To provide the users with more relevant products in a web shop, a personalization can be very useful. Personalized product rankings support the decision making of online costumers and help to improve their satisfaction (Zhang et al., 2016).

This thesis outlines opportunities of personalization in web shops especially with regards to product rankings. Different personalization techniques and levels are elaborated. Furthermore, two prototypes are realized and evaluated in cooperation with an online shop.

## 1.1 Context and relevance

Personalization is a trending topic in the context of data science and e-commerce. Different approaches are already widely used in web shops, e.g by using recommendation systems to offer similar or related products the customer might be interested in, or by creating a customer segmentation to optimize marketing campaigns (Kim and Chan, 2003). Further applications for personalization are internet search engines capturing the users search history, location or other aspects to provide individualized search results (Dou et al., 2009).

In the context of web mining, the personalization of product rankings can be related with all three sub-areas: content mining, usage mining and structure mining. Web mining is the application of data mining techniques to discover and extract information from web documents and services (Etzioni, 1996). The processing of user data to build personalized services is part of usage mining, whereas the ranking techniques can be related to structure mining. By integrating recommendations based on text or image similarity, content mining can also be used for personalized rankings.

In literature, there are various studies about the effects of personalized websites

both outside, as well as within the context of online shopping. In a field experiment using a news aggregation website, it was shown that content personalization does have an effect on the websites persuasive power and the users' willingness to pay (Benlian, 2015).

Salesforce's recent "State of the Connected Customer" report found, that 72% percent of consumers say that they expect companies to understand their unique needs and expectations. 66% of consumers even say, they are likely to switch brands if they feel treated like a number, not like an individual. Therefore, personalized services do not only have great potential, but are obligatory to retain customers to the brand, with 70% of consumers saying a company's understanding of their individual needs influences their loyalty (Salesforce Research, 2017).

## 1.2 Description of current situation

Among the different ways of personalizing web shops, product recommendation is the most widely used application (Shuk Ying and Bodoff, 2014). Whereas the personalization of the whole product ranking instead of providing separate product recommendations, is rather rare in web shops to the author's knowledge.

This thesis relies on an already existing product ranking algorithm, which predicts the expected sales in the upcoming days and ranks the products accordingly. Building on this, the personalization will be implemented for a retailer, which is already adopting the baseline product ranking.

## 1.3 Research objectives

This thesis aims to identify methods for the personalization of product rankings in e-commerce. Product ranking in this case refers to the order in which products are shown to a user in an online shop across categories and filters.

The first aspect of this research project is to identify different personalization methodologies. These will be classified in different levels of personalization and rated in their possible effects as well as the required effort to implement these in practice.

Furthermore, this project includes the implementation of two prototypes, which prove the practicability and effect of the personalization. The personalized product rankings are evaluated using a common ranking evaluation metric. In addition to the quantitative method, the rankings will be evaluated by experts based on their qualitative impression.

More specifically, the prototype development will be used to answer the following hypotheses:

1. Using machine learning procedures, product rankings can be improved to fit the customers' future purchasing behaviour.
2. Personalized rankings are perceived as being of better quality and can therefore improve the shopping experience.

Within the scope of this work it is not possible to evaluate the changes of the actual purchasing behaviour or in the web shops sales, usually proven by implementing a statistical A/B test providing half of the users with the transformed and the other half with the original ranking. Instead, the quantitative metric and qualitative evaluation will indicate the effects of personalized rankings.

## 1.4 Outline

Chapter 2 provides a literature review on personalization concepts in commerce and for web applications in general. Based on the findings, a categorization of personalization techniques for product rankings is developed resulting in a definition of different levels of personalization. In chapter 3 all relevant machine learning methods are presented, beginning with the concept of learning to rank and an evaluation metric for rankings. Furthermore, gradient boosting is introduced for prediction tasks, k-means for segmentation and matrix factorization for recommendation tasks.

Chapter 4 describes the currently applied modeling for the baseline product ranking, which is based on a sales prediction for each product. In Chapter 5 two prototypes are presented and compared: The first prototype personalizes the rankings for a defined number of customer clusters, whereas the second prototype creates a personalized ranking for each individual user. The results of this thesis are summarized in chapter 6.

# Chapter 2

# Literature review on personalization concepts

This chapter summarizes personalization concepts in research and in commerce. The theoretical personalization process and the 4R personalization framework provide guidance for the implementation of a personalized service. Additionally, it is explained, which factors should be considered to estimate the potential outcome of a service beforehand. Analyzing common personalized applications in the web and e-commerce, as well as existing personalized product ranking vendors, helps to identify and categorize possible variations of the rankings in the end of this chapter.

## 2.1 Personalization for web applications

### 2.1.1 Definition and aspects of personalization

The concept of personalization is intuitive, but an encompassing definition is rather difficult as there are many aspects and use cases. To provide an overview, this section summarizes several definitions coming from e-commerce and computer science view.

**E-commerce view**

1. "Personalization is the combined use of technology and customer information to tailor electronic commerce interactions between a business and each individual customer" (Personalization Consortium, 2003).
2. "Personalization is about building customer loyalty by building a meaningful one-to-one relationship; by understanding the needs of each individual and helping satisfy a goal that efficiently and knowledgeably addresses each individual's need in a given context" (Riecken, 2000).
3. "Personalization is the capability to provide users, customers, partners, and employees, with the most relevant web experience possible" (Kasanoff, 2001).

**Computer science view**

4. "Personalization is a toolbox of technologies and application features used in the design of an end-user experience" (Kramer et al., 2000).
5. Personalization is an "explicit user model that represents user knowledge, goals, interests, and other features that enable the system to distinguish among different users" (Brusilovsky and Maybury, 2002).

A thematic analysis of the definitions above shows, that most definitions consider a purpose or goal of personalization, the object or application, that is personalized (e.g. interface, content) and the target of personalization (e.g. user, consumer). Some definitions also include a statement of the means by which personalization is implemented, but as personalization is applied in many different fields, a general definition should not consider a specific approach (Fan and Poole, 2006). Therefore, Blom's general concept (Blom, 2000) can be adapted and personalization defined as a process that changes the functionality, interface, information access and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals (Fan and Poole, 2006).

### 2.1.2 General personalization process

The personalization of web applications can be divided into three phases, as shown in figure 2.1. The basis is built in the first phase by collecting and storing information about the user. The data is prepared and transformed in a so called user profile (Jablonski et al., 2004; Mobasher, 2007). In the second phase, the collected information is analyzed and filtered based on the relevance for the targeted personalization. The aim of the second phase is the recognition of patterns to enable the delivery of personalized content or service (Jablonski et al., 2004; Mobasher, 2007). In the final phase, the extracted information and patterns are then used to provide a personalized service (Gauch et al., 2002).

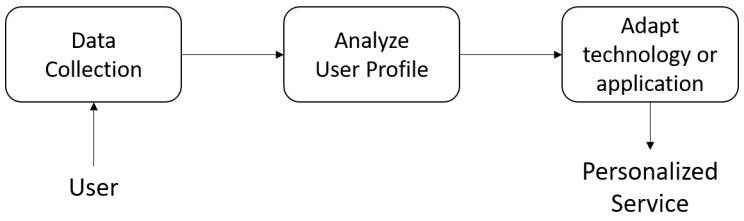



Figure 2.1: Process of user profile based personalization (cf. Gauch et al., 2002)

The described process is giving a simple overview of the technical development steps, that are necessary for any personalization service. The product ranking

is thereby the application to be personalized for each user or identified user segment. Other web applications, that are common for personalization, are listed in section 2.2. Multiple variations to implement the first two phases are explained in section 2.5.

### 2.1.3 4R personalization framework

In addition to the study of Salesforce explaining the relevance of personalization in e-commerce (see section 1.1), Accenture also developed a framework with the aim to replicate the requirements from the best offline personalized experience to a digital environment. Complementing the three phases of personalization, the 4R framework provides a construct to understand the technologies and capabilities needed to enable personalization (Accenture, 2018; Zoghby et al., 2016).

According to Accenture's study, 56% of the consumers are more likely to shop at a retailer in a store or online that recognizes them by name. Therefore, the first element of the framework is Recognize: Consumers and prospects have to be identified and addressed using the already known information including demographics, geography and expressed interests. The second "R" is about remembering the customer's history. This does not only include to remember the order, view and consume history, but also to understand, why they made each decision (Zoghby et al., 2016).

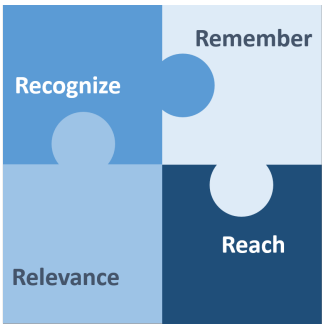



Figure 2.2: The four Rs of personalization (cf. Zoghby et al., 2016)

Approximately three out of five user are more likely to make a purchase in a store or online, when a retailer recommends options for them based on their past purchases or preferences. Therefore, the study identifies Recommend or Reach as third "R" of the developed framework. The online user should be reached with the right marketing, offer, content or product recommendation based on their profile and previous actions. Additionally to the usage of collected information about the user to provide personalized services, the frameworks also considers the Relevance as the fourth "R". Relevance can be determined with the user profile, but also his current situation, e.g. location, season or phase of life. Two-thirds of

consumers are more likely to make a purchase in a store or online from a retailer that sends them relevant and personalized promotions (Zoghby et al., 2016).

In context of this piece of work the first two aspects of the framework are already given: the visitors of the web shop are identified and tracked using web cookies and a customer number in case of a purchase. Furthermore, both tracking and transaction data are stored and can be used to build the personalization service. The other two aspects of reaching the customer with relevant content is the purpose of the personalized product ranking.

### 2.1.4 Estimation of the personalization potential

Before implementing any personalization service or more generally any elaborate project, it is useful to estimate the potential outcome. As the potential outcome is not observable, the concept is quite ambiguous. There are different definitions for this concept, but they can be summarized as follows: The potential outcome is the maximum level of output associated with the full utilization of existing resources (Campanelli et al., 1999).

To estimate the potential outcome of a personalized application following dependencies have been identified (Company, 2017):

- available data
- audience size
- relevance for the customer

The first dependency is the available data in terms of amount, depth, correctness and accessibility. For the implementation of a personalization service both historical and current data is necessary. The historical data is used to identify patterns and to train the personalization engine. Equally, the current data is applied to generate the personalized content (Progress Software Corporation, 2016; Mobasher, 2007; Company, 2017).

The amount and depth of the available data is in a direct relationship with the second factor for the potential analysis: the audience size. The audience size decreases in the process of the customer life cycle. There are different applications, which can be personalized for each stage. In the acquisition phase there is a great potential in terms of the audience size, e.g. using a personalized marketing campaign. However, in this phase the knowledge (implicit data depth) about each potential customer is very low. By having only a low effect per customer in the desired target figure can have a bigger impact through the immense audience size. On the other hand, it is possible to create a personalized experience for very well-known returning customers. But the effect has to be large in order to scale


Figure 2.3: Relationship between audience size and data depth (Progress Software Corporation, 2016)

with the smaller audience size. This relationship is shown in figure 2.3 presenting different audiences in the customer life cycle and the related increasing data depth (Progress Software Corporation, 2016; Company, 2017).

In the estimation of the personalization potential it is also important to consider the relevance for the customer (Company, 2017), as already mentioned in section 2.1.3 describing relevance as part of the 4R personalization framework. In addition to the known user specific information, the current situation of the customer has to be considered. This includes e.g. the location, season, probable current need or interest and phase of life (Accenture, 2018). Therefore, the usefulness of the desired personalization should be rated for each visitor or customer based on its relevance.

To estimate the potential in the desired target figure these three described dependencies have to be considered and evaluated. To achieve a similar effect, either the planned personalization is relevant for a bigger audience size and the data depth can be lower, or the audience size is smaller, but deeper knowledge about the targeted persons is available.

## 2.2 Common personalized applications

The phenomenal growth of the internet has resulted in the availability of huge amounts of online information, a situation that can be overwhelming for the end-user. To overcome this problem, personalization technologies have been extensively employed across several domains to provide assistance in filtering, sorting, classifying and sharing online information (Uchyigit and Ma, 2008).

### 2.2.1 Personalized applications in e-commerce

E-commerce offers many possibilities for personalization using different applications and communication channels. To understand the different opportunities, the overall customer journey should be considered. There are different customer journeys for different purposes, but the customer journey presented in figure 2.4 summarizes the overall process for e-commerce customers appropriately.

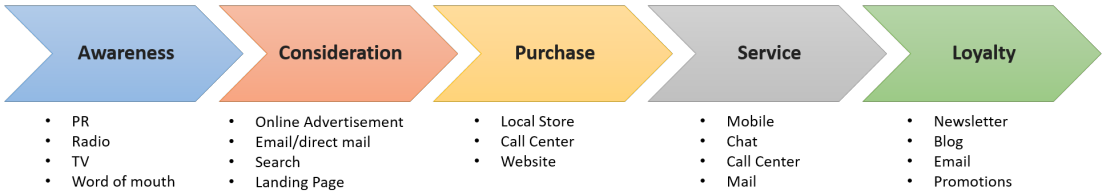



Figure 2.4: Customer journey (cf. Davies, 2015)

The customer journey starts with awareness and consideration. A typical use case in this phase is the usage of personalized online marketing campaigns. Marketing campaigns are usually shown off-site and present products or vouchers to achieve awareness for the own online shop. The advertisement can be done through multiple channels, e.g. online advertisement banners on other websites or in social media, as well as personalized emails or printed media.

In regard to the purchase phase of the customer journey, an internally conducted competitor analysis has shown, that the most common on-site personalization applications in e-commerce are recommendation systems and the auto completion in product search engines.

Recommendation systems support online customers to find products and services suiting their wishes and needs. There are two main approaches to the implementation of recommendation, see also section 3.6.1. First, collaborative filtering uses the preferences of a large set of customers. Assuming that human preferences are correlated, recommendations given to a customer are derived from preferences of customers with similar interests. Second, content-based filtering uses preferences of a specific customer to infer recommendations. In this context, products are described by their characteristics (e.g. category, color, and brand). Based on the past interaction with products an user profile is built. The stored preferences for each user are then used to offer similar products (Felfernig et al., 2008).

The second common on-site personalization application is an auto-completion of the search engine input. Personalized search terms are displayed to the users based on previous searches and their on-site behaviour. The first level of implementation is to store and display previous search terms. The second personal-

ization step is the auto-completion by suggesting personalized keywords for user entered search terms (Company, 2017).

Other personalized applications along the customer journey are e.g. a personal main feed or even a personalized layout, assortment banners, personalized search results or product rankings in categories.

### 2.2.2 Rankings as personalized application

This section will focus on rankings as the targeted application for a personalization, giving the user the information, products or services he is interested in.

A *ranking* of items is a rank-ordered list of items. Thus, a ranking vector is a permutation of integers: 1 to n. In contrast, a *rating* of items assigns a numerical score to each item. Therefore, every rating list creates a ranking list, when sorted, but not vice versa (Langville and Meyer, 2012).

There are two predominant paradigms, how web users can access information: browsing or searching by query (Gasparetti and Micarelli, 2008). In the first paradigm, users analyze the content by navigating sequentially through the items (e.g. web pages or products), following hyperlinks. This approach is useful if the aim is the exploration of items, but it is not suitable for locating a specific item. Therefore, the second paradigm is to search using queries specifying the user's interest. The search engine directly retrieves documents from an index of many items by applying a common information retrieval (IR) model. Documents and queries are both processed and converted into representations, which are then used as input of a similarity function resulting in a list of relevant documents (Gasparetti and Micarelli, 2008).

Commonly known and used internet rankings are generic search engine results. There is a number of approaches that address the personalization task based on the traditional content-based IR model combined with explicit feedback techniques, e.g. ref. Joachims et al. (1997) and Moukas and Maes (1998). Other approaches are reducing the effort for explicit user input by content-based and collaborative filtering techniques. Naive Bayes classifiers creating representations of the user needs were used to build some of the first prototypes based on implicit feedback techniques. To construct the user profile search histories, as well as user created, copied, or viewed content can be analyzed. Furthermore, the user actions can be monitored and used to predict the prospective user needs (Gasparetti and Micarelli, 2008).

Google is the most popular search engine, according to NetMarketShare (2017) with over 80% market share worldwide. One of the reasons for success of Google is its personalization. Besides the users' behaviour and situative data, like their

location, Google is using so called micro moments. By identifying or predicting the users' need and purpose Google differentiates between four different moments: I want-to-know moment, I want-to-go moment, I want-to-do moment or I want-to-buy moment. Based on this classification they decide, which content is displayed first: e.g. websites, Google maps, places of interest or products (Ramaswamy, 2015).

This piece of work is concentrating on product rankings in e-commerce, more specifically ordering products in a web shop in order to improve the user experience and eventually to increase sales. The main differences between a search result in a generic internet search engine like Google and a product ranking are on the one hand the type of content and on the other hand the amount of items to be ranked. On-site, there are two types of product rankings: coming from an on-site product search engine (searching paradigm) or showing a category product list (browsing paradigm). A product ranking inside a category view does not contain any relevance to a search term, whereas a product search ranking is more similar to a general search engine result. A ranking of products without using the relevance related to a search term is usually done by ranking the most popular items to the top. The way how to define popularity can be done differently. The baseline product ranking in this project, described in chapter 4, uses a prediction of sales of each product in the upcoming day. Various possibilities to personalize the product rankings are described in the following two sections.

## 2.3 State of the art in personalized product rankings

This section analyzes industry implementations in the field of personalized product rankings. The insights are then used for the categorization of personalization methodologies and the conceptual design of the planned prototypes.

The shopping website About You, which is especially designed for younger generations, uses celebrities and influencers wearing the clothes of the online shop to present their styles. Customers can like and follow their favourite celebrities and styles. In doing so, the user can explicitly define his preferences in fashion. About You uses this information to personalize the customer experience including the product rankings. Therefore, About You is an example of explicit personalization, where the customer is asked to define his preferences (Stücke et al., 2016).

Odoscope is a company providing personalization services especially for the e-commerce industry. The personalization of product rankings is based on static on-site data, e.g. the user agent (browser, device, etc.), location, time and weather. As their method does not use any customer specific data, it is called situation-

based personalization. The website states that each visitor is categorized by using a real-time clustering based on the previously mentioned features. For each cluster and product, a sales probability is predicted. The main advantages of this method are on the one hand having less privacy problems with customer data and on the other hand the opportunity to provide also new customers with a personalized ranking even before their first on-site action (Odoscope GmbH, 2018).

The e-commerce companies or service providers described below are using a technique for their personalized product rankings that combines the previously existing ranking with weighted factors, e.g. customer affinities or product recommendations.

Hausl, a software developer of the online shopping platform Asos, explains in a talk, how they developed personalized search results. In addition to the search relevance, the ranking is transformed with the following weighted factors (Hausl, 2017):

- product recommendation regarding previously bought items (item-item collaborative filtering)
- brand and colour affinity
- user-item affinity using matrix factorization
- gender of customer

Although this ranking was developed for search results, it can be also applied to a product ranking in a category view.

Rich Relevance is a software and service provider in the area of interest. The website states that the weighted model is based on following factors (Rich Relevance, 2018):

- brand, category and product affinity
- price quantile
- affinity to new products

The software vendor Apptus developed a similar concept inter alia with Intersport, a worldwide sports retailer. The personalization is based on the product relevance, sales figures and the customer on-site behaviour (Apptus Technologies AB, 2018). Apart from Rich Relevance and Apptus, there are several other specialized software vendors, e.g. Dynamic Yield, PureClarity, Qubit and Episerver.

The analysis of state-of-the-art solutions has shown, that personalized product rankings are an emerging topic in e-commerce. There are several vendors and approaches for this problem. However, to the author's knowledge there are only

few companies that are already using it productively. The creation of rankings from the perspective of machine learning is outlined in section 3.1. The different kinds of personalization are categorized in section 2.5.

## 2.4 Personalization and privacy

Since firms started to realize that data can generate value for them and for their customers, they are collecting, storing, and using more data about consumers (Beke et al., 2018). Every year about 16 trillion gigabytes of data are recorded, and forecasts indicate a growth to 163 trillion gigabytes by 2025 (Reinsel et al., 2017).

Personalization is an important factor in the internet and especially e-commerce, because it helps users find exactly the information, products, and services they need. However, controversial revelations regarding the expansion of information collection and privacy in general (e.g. Edward Snowden's disclosures about data collection and surveillance programs) have resulted in a worldwide surge of privacy concern (Beke et al., 2018).

### 2.4.1 Privacy risks

Toch et al. discuss potential privacy risks in three domains of personalization: social-based personalization, behavioural profiling, and location-based personalization.

Social networks obtain real names, email addresses, list of friends, demographics, personal photos, geographic location history, inter-personal communications, and more, which can be used to personalize the social network's feeds as well as advertisement presented on the platform. The privacy concerns in social-based personalization is mainly based on the highly sensitive information available through social networks. The usage of the described data does not only affect the privacy of an individual user, but can also affect their friends and other contact's privacy (Toch et al., 2012). A recent example is the Facebook–Cambridge-Analytica data scandal, that involves the collection of personally identifiable information of more than 50 million Facebook users. In 2014, Cambridge Analytica asked a number of users to take a personality test, who also agreed to have their Facebook data collected for academic use. However, the personality test app also collected the information of the test-takers' Facebook friends. The data was then allegedly used to attempt to influence elections (Graham-Harrison and Cadwalladr, 2018).

Behavioural profiling is the practice of collecting data about a user's online activities and using this information to tailor the user experience to each individual.

The main concerns about this type of personalization are the data tracking without consent or knowledge by the user and the sharing of collected information with third parties (Toch et al., 2012).

Location-aware services become more popular with the usage of GPS-enabled phones and the adoption of WiFi-positioning technologies, as well as the increase in mobile data bandwidth. Therefore, the ability of service providers to continuously track the location of their users and to offer services based on the exact physical location also grows (Toch et al., 2012).

### 2.4.2 EU General data protection regulation

To strengthen the rights of individuals, the European Union (EU) developed the general data protection regulation (GDPR), which modernizes and uniforms the legal situation in the EU with regard to personal data and privacy. The GDPR takes most of the privacy concerns in consideration, which have been described in the previous section.

Personal data is all information that refers to an identified or identifiable natural person, i.e. any form of data that directly or indirectly allows conclusions to be drawn about a person, such as a name, an e-mail address or a telephone number among others (European Union, 2016).

Technical and organizational measures must ensure that the assignment of any collected data to a person is not possible or only with considerable effort. Data encryption and pseudonymization are explicitly recommended by the GDPR to ensure an appropriate level of protection of users. Web cookies and mobile advertising identifiers, encrypted e-mail addresses, as well as other technical identifiers that serve to display customized, personalized advertising messages to users are regarded by the GDPR as personal data in pseudonymized form - and thus as suitable to ensure secure data processing (European Union, 2016). Therefore, the same rules and conditions apply as for other pseudonymized information defined by the GDPR.

The GDPR provides two principles of central importance, out of which at least one must be fulfilled. The first principle is, that an effective, i.e. unambiguous consent of the user is available, e.g. an explicit opt-in. For pseudonymous, non-sensitive data, other forms of unambiguous consent are also sufficient. However, the user has to be informed about the pursued usage of his data and must have the option to control the usage. The second principle states, that there is a legitimate interest in data use: This is the case when the data is to be used for direct marketing purposes and the interests of the data subject are protected (European Union, 2016).

With regard to this piece of work, the introduction of the GDPR means that users must be informed about the use of cookies, as well as storage and processing of data. The personalization is then limited to the users who can be recognized and have given their consent to the processing of their data. Personalization methods that are not based on users' personal data can be applied to all customers.

## 2.5 Categorization of personalized product ranking implementations

Personalization techniques can be categorized along three dimensions (Fan and Poole, 2006):

1. Application: What is personalized?
2. Target: To whom?
3. Automation: Who does the personalization?

As the application is clearly defined as a product ranking and therefore a personalization of the content order, the dimensions are complemented with a fourth aspect in the context of this thesis: the real-time ability subject to the type of input data, which is used to provide relevant products to a visitor. Based on the competitor analysis, literature review and an interactive workshop with e-commerce and business intelligence professionals from the partnering retailer, the author of this thesis collected the following aspects for personalized product rankings and categorized them.

### 2.5.1 Target and depth of personalization

In consideration of the depth of personalization, the target can either be a segment of individuals or a specific individual. In the first case the implementation is personalized for different categories, which can be statically defined such as women, single-child families or members of a club, or dynamically defined, e.g. clustering based on user behaviour. If an individual can be identified within one of predefined categories, the system adapts to the segment's needs (Fan and Poole, 2006).

Clustering aims to divide a data set into groups or clusters where inter-cluster similarities are minimized while the similarities within each cluster are maximized. As noted earlier, the primary motivation behind the use of clustering is to improve the efficiency and scalability of a personalization tasks. In the context of web mining, there are two typical approaches for a segmentation: clustering

users or clustering items. In a user-based clustering, users are grouped together based on the similarity of their user profile. In an item based clustering, items are clustered based on the similarity of the interest scores for these items across all users, or based on similarity of their content features or attributes. The user-based clustering can be used to reduce the personalization task from individual users to groups of users with similar interests. Both cluster and item-based segmentation are often used to reduce the dimensions in personalization tasks, like recommendation systems (Mobasher, 2007)

The second option is to design systems in order to adapt to the needs of a single user. The goal is to deliver goods, services or information targeting each individual. The actual implementation of individuated personalization nevertheless can be based on a categorical analysis. In case, that the users cannot be captured in the unique individuality of a person, the profile can be defined as the unique intersection of a variety of categories representing the most important characteristics of an individual (e.g. gender, nationality, job, place, age, children in household, etc.). If, in theory, enough categories are utilized to define each individual uniquely, this system works for all intents and purposes as an individuated personalization system, even though simplified categories are used. In general, a personalization for each single individual takes more system resources than a categorical personalization (Fan and Poole, 2006).

### 2.5.2 Degree of automation

The collection of data about the users is the first phase of each personalization project and thus, the most important problem. The acquisition can be partitioned by the type of feedback used to build a user profile: explicit or implicit data (Gasparetti and Micarelli, 2008; Fan and Poole, 2006).

Explicit feedback systems rely on direct user intervention, e.g. by specifying keywords of interest or answering questions about the personal needs (Gasparetti and Micarelli, 2008). Explicit personal information can also be collected during the shop checkout process including demographic information such as birthday, marriage status and address (Gauch et al., 2002). As described in section 2.3, the web shop "About You" uses a system, where each user can follow his favourite celebrities and styles, which is used to personalize their product feed. Therefore, it can be taken as an example of explicit personalization. Even though explicit feedback techniques have shown to improve retrieval performance, other studies have found that these techniques are not able to considerably improve the user model (White et al., 2001). A main problem is, that the effort to specify the own interests is often too high since users are not assuming a great benefit for themselves. Therefore, the effectiveness of explicit techniques are limited (Gasparetti and Micarelli, 2008).

In contrast, implicit feedback systems collect information about the users unobtrusively while browsing. The user's behaviour is tracked and monitored to build a unique user profile (Gasparetti and Micarelli, 2008). The variety of data sources that can be used were mentioned in the competitor analysis and are categorized in section 2.5.3. In comparison to the explicit feedback, implicit data can be biased in the users' decisions and already influenced from previous retrieval functions. The fact that implicit feedback is readily available in large quantities overcomes this problem and therefore results in an overall good quality in the personalization (Gasparetti and Micarelli, 2008; Gauch et al., 2002). With the introduction of the GDPR, however, privacy must be observed (see 2.4.2). If customers and website visitors do not give their consent, the data may not be used for personalization (European Union, 2016).

An implementation of personalization can also include both, explicit and implicit information about the user, which can result in an even better user profile than only using one type of data (Gauch et al., 2002).

### 2.5.3 Real-time ability subject to input data

The real-time ability of a personalized system depends on the system's properties, but is particularly subject to the available input data, on the one hand to train a model and on the other hand to predict and provide a personalized product ranking. As part of this thesis, several feature variables, that are useful for a personalized product ranking, are collected and categorized. The result is presented in figure 2.5 and is described in the following.

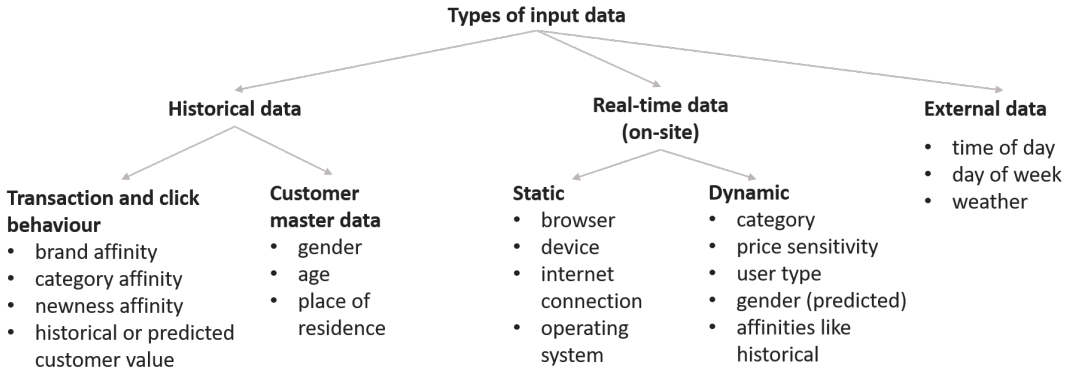



Figure 2.5: Types of input data with exemplary features

Typical features for any personalized application are gender and age, which can be summarized in customer hard facts. This information is usually stored historically in a web shop during a user account creation or during the purchase checkout. If a

customer is recognized in a later session, this information could be used and thus, it is not a real-time personalization. Another possibility is to predict each user's gender and age based on the online behaviour. In this case the personalization can be applied in real-time.

Besides customer hard facts, the historic transaction data of each customer and the historic clicking behaviour (tracking data) of each visitor can be used to build a personalized application. The advantage of the historically saved data is that the product rankings can be pre-calculated and there is no dependency to a data storage, which handles real-time data. In comparison, transaction data gives more detailed information about the interest of a user, but is only available for existing customers. While the clicking behaviour can be tracked and stored for every shop visitor, e.g. identified by a cookie, it is less meaningful than data from purchases, but available for much more users. Typical features for this category of data are the interest in specific product categories or brands, and the affinity to new products or price.

If the system properties allow a real-time analysis and usage of the users' clicking behaviour, a personalization of the product ranking can be adapted during a session. Previously unknown visitors get a personalized experience after browsing through the web shop and the experience improves incrementally for each user with each click. The main disadvantage is that real-time tracking data is not always available and the effort to adapt the product ranking in real-time is very high. To implement a real-time personalization based on user data, system dependencies with several systems may need to be solved and simplified beforehand. A solution can be the usage of on-site static data, e.g. browser, internet connection and operating system. The advantages are that necessary data is available for every shop visitor and if not stored permanently, does not need the consent of the user in regard to the GDPR.

In addition to the described features, the data can be enriched with external data, like the user's location, the weather, day of week and current time.

### 2.5.4 Identified levels of personalization

Based on the three described dimensions relevant for a personalized ranking, the author of this thesis has identified different degrees of personalization, summarized as broad levels in figure 2.6.

The first level of personalization is using explicit feedback of the user to generate personalized content. The data is gathered by explicitly asking the user about his characteristics or interests (see section 2.5.2). The implementation effort can range from a very simple rule based personalization to a medium complex system. The effects are immediately apparent to the user and therefore transparent and

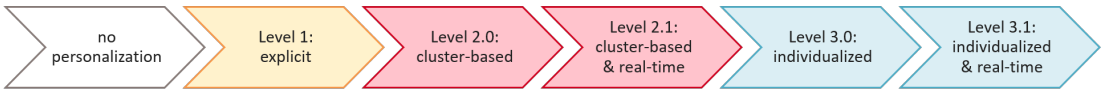



Figure 2.6: Levels of personalization

not biased by wrong interpretation of implicit user data. The main disadvantage is that the personalization is dependent on the user's willingness to provide information.

The second level of personalization uses implicit information about the users, eventually enriched by explicit feedback, to generate user segments with similar characteristics. These are used to provide a ranking with more relevant products for each specific user segment. By using only historic data for the personalization, the technical complexity is rather small for the integration in web-shops. A real-time personalization is considerably more complex as the web shop has to provide the user data in real-time to expand the user profile and to determine the user segment. The product rankings could be updated incrementally, but don't have to, if the system properties are not given. A cluster-based approach is therefore useful, if the web shop can handle only a finite number of rankings.

The third level of personalization is a fully individualized ranking for each visitor, assuming to know something about the user before or during the session. The first step is to provide a ranking for every user, that is historically known, based on his unique characteristics and interests. The most extensive personalization in a product ranking is individualized and in addition adapts during a session in real-time. In comparison the individualized and especially the real-time approach is much more complex. From the author's experience, the ranking system in the most web shops cannot yet handle this. Therefore, the implementation potentially requires a new IT infrastructure.

There are various possibilities for the implementation for personalized rankings of each level. The complexity is mostly depending on the used data, its availability and amount. This piece of work includes the conceptual design and development of two prototypes with the second and the third level of personalization, see chapter 5. These prototypes will be compared in their implementation effort and effect.

# Chapter 3

# Relevant concepts in machine learning

A variety of machine learning techniques are required to develop personalized product rankings. This chapter introduces the concept of learning to rank as well as an evaluation metric, which is designed to rate rankings on their ability to retrieve relevant items. Furthermore, machine learning techniques are categorized based on the characteristics type of supervision, training input and generalization. The gradient boosted decision trees are introduced for the prediction of sales data in the prototypes. K-means is used for the segmentation of users based on their previous behaviour. The matrix factorizing is proposed as recommendation engine for individualized product rankings.

## 3.1 Learning to rank

Learning to rank refers to machine learning techniques, which are training a model in a ranking task. The models can be differentiated in three approaches: pointwise, pairwise and listwise (Liu, 2009).

Pointwise methods assume that each item in the training data has a numerical or ordinal score. Therefore, the ranking problem is reduced to a regression or classification problem (Freno et al., 2015; Liu et al., 2018). The assigned score typically refers to the relevance of the item. The predicted scores are therefore used to rank the items in the list (Freno et al., 2015). Linear or logistic regression are examples of scoring functions used in pointwise approaches (Freno et al., 2015), as well as Perceptron Ranking (PRanking) and McRank (Liu et al., 2018).

In pairwise approaches, the ranking is formulated as a pairwise classification problem. Given a pair of items the aim is to learn the order correctly. The relevance score in the training data set determines the order in each pair, whereas the value of this score is not learned directly (Freno et al., 2015). Some of the most popular approaches are RankSVM and RankNet (Liu et al., 2018; Freno et al., 2015). Bayesian product ranking (BPR), which is described in section 3.6.2, also belongs to the pairwise approaches.

Finally, listwise approaches consider the training examples as a list of ranked items and attempt to minimize a loss function that is defined for the whole list. AdaRank and ListNet are examples for listwise methods (Liu et al., 2018).

Gradient-boosted trees (see section 3.4.2) are another popular learning to rank method (Freno et al., 2015; Mohan et al., 2010). The normalized discounted cumulative gain is a common listwise metric to evaluate the rankings, elaborated in section 3.2.

The three described approaches can be applied for all kind of ranking problems and thus, also for the pursued product rankings. As example, Freno et al. (2015), researchers from the MIT, Amazon and Zalando examine different learning to rank methods to improve product recommendations on Amazon. Liu et al. (2018) apply i.a. BPR to personalize recommendations.

## 3.2 Evaluation of ranking techniques

Since all products are not of equal relevance to the users, highly relevant items should be identified and ranked first in a ranking. In order to develop ranking techniques in this direction, it is necessary to use evaluation methods that credit the rankings for their ability to retrieve highly relevant products. The normalized discounted cumulative gain (nDCG) is particularly suitable as a measure of ranking quality. It is commonly used to measure the effectiveness of web search engine algorithms and related applications (Järvelin and Kekäläinen, 2002).

The nDCG is based on the cumulative gain, which is the sum of gain until a certain position. The gain values are usually given by a function resulting in ad-hoc relevance judgments associated with the items in the ranking (Moniz et al., 2016).

$$CG_p = \sum_{i=1}^{p} rel_i \tag{3.1}$$

The discounted cumulative gain (DCG) considers in addition to the gain factor also discount factors. Based on the fact that the probability of a user seeing an item decreases with a rear position in the ranking, the gain is discounted at lower ranks (Moniz et al., 2016; Järvelin and Kekäläinen, 2002):

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{3.2}$$

This metric is commonly used in its normalized form, the normalized discounted cumulative gain (nDCG), as shown in the following equation (Järvelin and Kekäläinen, 2002):

$$nDCG_p = \frac{DCG_p}{IDCG_p}, \tag{3.3}$$

where $IDCG_p$ represents the maximum possible $DCG_p$ given by the optimal ranking order (Järvelin and Kekäläinen, 2002). The main advantage of the nDCG is the possibility to optimize the predictive model directly on this metric.

## 3.3 Types of machine learning systems

Machine learning systems can be classified in broad categories using different criteria, which are outlined in the following (Géron, 2017):

- Type and amount of supervision: supervised, unsupervised, semi-supervised or reinforcement-learning
- Type of training input: incremental (online) or batch (offline) learning
- Type of generalization: instance-based or model-based learning

### 3.3.1 Type and amount of supervision

A major difference in machine learning systems is the amount and type of supervision they receive during training. Algorithms that learn from input/output pairs given from an external teacher are called supervised learning algorithms. Whereas in unsupervised learning, the training data is unlabeled and the system tries to learn without any teacher (Géron, 2017; Müller and Guido, 2016).

There are two typical tasks for a supervised machine learning system: classification and estimation (regression). The main difference is the target variable, which is categorical for classification and numerical for estimation problems. Both tasks can be solved with common algorithms, e.g. decision trees, neural networks, linear or logistic regression (Larose, 2005). The typical unsupervised problems and corresponding algorithms are segmentation, e.g. with hierarchical clustering or k-means, dimension reduction and visualization, e.g. with a principal components analysis or embeddings, and association rule learning, e.g. with apriori (Géron, 2017). While there are many successful applications of unsupervised methods, they are usually harder to understand and evaluate in comparison to supervised algorithms (Müller and Guido, 2016).

Some machine learning problems rely on partially labeled training data, usually with a very little proportion of labeled data. This is called semi-supervised learning and involves in the most cases a combination of unsupervised and supervised algorithms (Géron, 2017).

The last type of supervision is reinforcement learning. The learning system, often called agent, is not told which actions to take, but instead must discover which actions yield a reward or a penalty (as negative reward). However, the actions may affect not only the immediate reward but also later ones. Therefore, the main characteristics of reinforcement learning are a trial-and-error search and the delayed reward system. Typical applications for this kind of problem solving are robot control, elevator scheduling or solving board games. (Sutton and Barto, 1998).

For this piece of work, both supervised and unsupervised machine learning algorithms are applied. The algorithms used for a customer segmentation is explained in section 3.5. The basis for the product ranking is a sales prediction, which is introduced in section 3.4. Furthermore, the second prototype consists of a recommendation system, which is explained in section 3.6

### 3.3.2 Type of training input

Machine learning systems can be differentiated, whether they can learn incrementally from a stream of data or only from one single batch. If the system is incapable of learning incrementally, it is trained using all the available data. After training the model it is then launched into production to apply it to previously unknown data. As the model runs without learning anymore, it is called offline or batch learning. The process of training, evaluating, and launching the machine learning model can be automated, even a batch learning system can adapt to change by updating the data and training a new version of the system from scratch as often as needed (Géron, 2017).

In online learning, the system can be trained incrementally by feeding through new data instances sequentially, either individually or using small groups called mini-batches. The sequential training is usually fast and the model adapts as soon as new data is provided. Another advantage of the incremental method is, that the whole data does not have to be stored for a long term to retrain the model like in batch training. Furthermore, online learning algorithms can also be used to train systems on huge data sets that cannot fit in one machine's main memory, then called out-of-core learning (Géron, 2017).

### 3.3.3 Type of generalization

Many machine learning tasks are about making predictions. If a model, trained on a set of data, is able to make accurate predictions on other unseen data, it is able to generalize from the training set to the test set. Models are usually build and evaluated with a performance measure. A good performance on the training data is insufficient, but the goal is to perform well on new instances (Müller and Guido, 2016; Géron, 2017).

In terms of the applied generalization approach, machine learning systems can be separated in an instance-based and a model-based learning. In case of an instance-based learning the system estimates the target based on the similarity of the new data with the training instances. So the system learns the given instances by heart and then generalizes by using a similarity measure. In contrast, a system can generalize from a training data set by building a model using each samples characteristics and then use that model to make predictions, called model-based learning (Géron, 2017).

The explained types are a non exclusive categorization of machine learning systems. In the following sections, the specific machine learning methods will be introduced, which are important for the developed prototypes.

## 3.4 Introduction to gradient boosted decision trees

### 3.4.1 Classification and regression trees

Decision trees are widely used models for classification and regression tasks. Essentially, they consists of a hierarchy of if-else-decisions, which can be outlined in a tree structure (Müller and Guido, 2016).

The basic concept of a decision tree is to partition the data in more homogeneous groups with respect to the problems target. There are different techniques for the construction, whereas one of the most common was proposed as classification and regression tree (CART) by Breiman et al. (1984) (Kuhn and Johnson, 2013).

The model begins to classify the data set $S$ by searching a distinct predictor with its exact split value, that splits $S$ in two sets $S_1$ and $S_2$ and minimizes the overall sum of squared errors:

$$SSE = \sum_{i \in S_1} (y_i - \overline{y}_1)^2 + \sum_{i \in S_2} (y_i - \overline{y}_2)^2, \tag{3.4}$$

where $\overline{y_1}$ and $\overline{y_2}$ are the averages of the training set outcomes within the sets $S_1$ and $S_2$ (Kuhn and Johnson, 2013).

The model is optimized by repeating the process of finding the best split point in both sets. This recursive process yields a binary tree of decisions. There are different stopping criteria, that can be defined. If the target is a binary or categorical target, the most obvious stopping criterion is when every subset contains only one target category. But in practice, this would probably lead to an over-fitted model. Common stopping criteria are the definition of a maximum depth of the tree, the limitation of the number of leaves or requiring a minimum number of points in a node (Müller and Guido, 2016).

Models based on a single tree have some weaknesses. One well known limitation is the model instability: small changes in the data set can result in a completely different structure of the tree. Thereby also the interpretation of the decision tree is not stable. A second limitations is a weaker predictive performance. As each test concerns only a single feature, the data set is split in straight lines. The prediction result of a decision tree can be visualized as rectangular regions that contain more homogeneous outcome values. If the relationship between predictors and the target cannot be adequately defined by rectangular subspaces, the tree-based model will then have a larger prediction error than other kinds of models (Kuhn and Johnson, 2013).

To overcome these limitations, different tree-based models can be combined into one model, called ensemble (Kuhn and Johnson, 2013).

### 3.4.2 Gradient boosting

The idea of a boosting algorithm is to combine several weak learners into a strong learner. The difference in comparison to other ensemble methods is that the single predictors are trained sequentially, each trying to correct its predecessor. The most popular boosting methods are AdaBoost and Gradient Boosting (Géron, 2017). As this project is using a gradient boosted decision tree model, only the concept of this algorithm will be explained in detail.

The gradient boosting relies on an additive model of weak learners, e.g. regression trees. Given a loss function (e.g. squared error for regression), each learner results in some residuals to the actual target data. The residual (also called gradient) is then used as target value for the following learner. By adding the new model to its predecessor the loss function is minimized. The procedure continues for a user-specified number of iterations. Usually the algorithm is initialized with a best guess, e.g. mean target value of the regression (Kuhn and Johnson, 2013).

Gradient boosted decision trees are very powerful supervised machine learning

models. They are widely used in the industry and are also frequently the winning entries in machine learning competitions. Their main drawback is the necessity of a careful parameter tuning and a quite long training duration. Similarly to other tree-based models, the algorithm works well with binary as well as continuous features and without having to scale them (Müller and Guido, 2016).

The main parameters of gradient boosted tree models are the number of trees and the learning rate, which controls the degree to which each tree is allowed to correct the mistakes of the previous trees. If the learning rate is rather small, the number of trees has to be higher to build models with similar effect. So these two parameters are highly interconnected. However is must be noted that increasing the number of estimators leads to a more complex model and may lead to overfitting. Another important parameter is the maximum depth (or alternatively maximum number of leaf nodes) to reduce the complexity of each tree. Usually the maximum depth is very low as the basic concept relies on weak learners (Müller and Guido, 2016).

## 3.5 Introduction to segmentation

Segmentation or clustering refers to the grouping of records or observations into classes of similar objects. A cluster is composed of records that are similar to each another, but different from records assigned to other clusters. In contrast to classification, the clustering problem has no target variable and is therefore unsupervised. The goal of a clustering algorithm is to segment a data set into relatively homogeneous subgroups or clusters. It is often performed as a preliminary stage in a data mining process or project. The resulting clusters can be used to reduce the search space for other algorithms or as an additional input variable (Larose, 2005; Mobasher, 2007).

### 3.5.1 Clustering strategies

Generally speaking, clustering methods can be divided into three strategies (Mobasher, 2007), which are briefly explained in the following:

- **Hierarchical clustering** distinguishes between the bottom-up (agglomerative) and top-down (divisive) approach with the aim to create a treelike cluster structure (dendrogram). In the initialization of agglomerative clustering each observation is considered as a cluster, and then combined iteratively into bigger clusters by aggregating the two closest clusters. Divisive methods start from the whole data set of items as a single cluster and partition this data with the most dissimilar records being split off recursively

into a separate clusters (Larose, 2005; Mobasher, 2007).

- In **partitioning methods**, the number of clusters ($k$) is defined beforehand. The algorithms partitions the given data set in $k$ partitions, where each partition represents a cluster (Mobasher, 2007; Müller and Guido, 2016). The shape and allocation of the clusters is improved iteratively, but the number of clusters does not change. The most widely used partitioning method is the k-means algorithm (Mobasher, 2007), which is described in section 3.5.2.
- **Model-based methods** discover the best fit between data points given a mathematical model, usually based on the distribution and density of the observations (Mobasher, 2007).

### 3.5.2 K-Means

K-means clustering is one of the simplest and most commonly used clustering algorithms (Müller and Guido, 2016). The algorithm tries to find cluster centers that are representative of a subset of the observations, given a user-defined number of clusters (Mobasher, 2007).

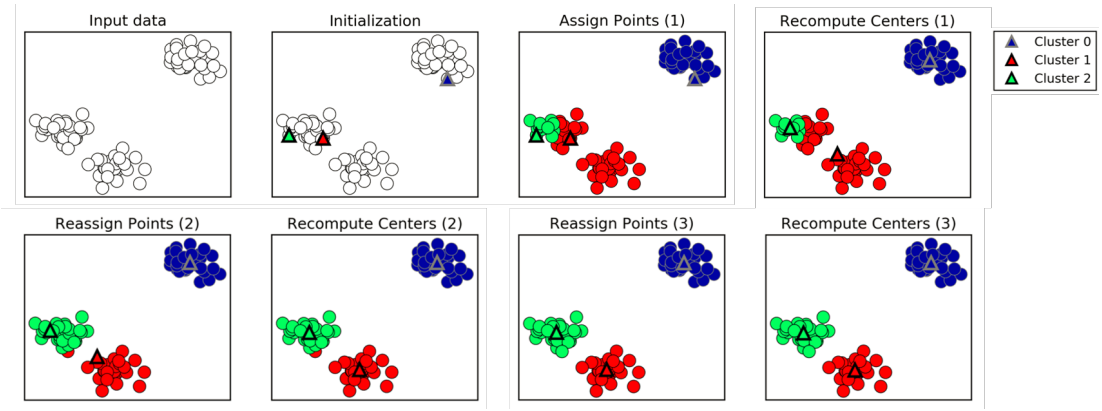



Figure 3.1: Evolution of k-means algorithm (Müller and Guido, 2016)

Therefore, the algorithm proceeds as follows:

1. The number of clusters $k$ has to be defined by the user.
2. The algorithm is initialized by randomly appointing k observations to be the cluster center locations.
3. Each data point is then assigned to the closest cluster center.
4. The new center of each of the k clusters is estimated as the mean of the data points that are assigned to it.
5. Steps 3 to 5 are repeated until convergence or termination (Larose, 2005).

This process is also illustrated in figure 3.1. For step 3 it is necessary to define a “nearest” criterion. Usually the Euclidean distance is used to find the nearest cluster centroid, although other criteria can be applied as well (Larose, 2005). When using the distance measure it becomes clear that differently scaled features in the data set will affect the distance to different degrees. Therefore, a scaling of the features in the data set should be considered during data preparation.

The algorithm convergence is reached when the assignment of any observation to clusters no longer changes (Müller and Guido, 2016).

## 3.6 Introduction to recommendation systems

Recommendation systems can be defined "as any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful items in a large space of possible options" (Burke, 2002). Recommendation systems are typical personalized applications (see section 2.2), which are often presented as separate short product list in a web site. But the approach and methods guiding the user to relevant items can be adapted for all product rankings. By integrating the recommendations into the standard product list, a single point of contact is created and more relevant products are presented personalized to the web shop visitors.

### 3.6.1 Recommendation strategies

The usage of recommendations as personalized application can be categorized in three well-known strategies in the field of recommendation systems (Blanco-Fernández et al., 2008; Jannach et al., 2010):

- Content-based methods
- Collaborative filtering
- Hybrid strategies

#### Content-based filtering

Content-based filtering uses preferences of a specific customer to suggest objects similar to those the customer liked in the past (Felfernig et al., 2008; Blanco-Fernández et al., 2008). The method is based on user and product profiles. In this context, the products are described by keywords (e.g. categories), that are also stored in the user’s profile in the case that the user in interested in the

product. In the next session, the stored preferences are used for offering additional products associated to the keywords (Felfernig et al., 2008). This technique requires a metric to quantify the similarity between the users' profiles and the product attributes. The most used metrics have a critical weakness related to their syntactic nature, which only detects similarity between items sharing the same attributes. Therefore, content-based recommendation systems have the limitation to suggest only items, that are very similar to the items already known by the user. This results in a recommendation system with a limited diversity in the recommended objects (Blanco-Fernández et al., 2008). Therefore, content-based filtering is useful to present similar products on a product detail page in a web shop giving the user alternative products. This approach is less feasible in presenting the customer with other inspiring and relevant products.

### Collaborative filtering

In contrast, collaborative filtering techniques are based on recommending those objects appealing to other like-minded viewers, named neighbors of the user (Blanco-Fernández et al., 2008). Therefore, the preferences of a large set of users are stored. Assuming that human preferences are correlated, recommendations given to a user are derived from preferences of visitors with similar interests (Felfernig et al., 2008).

In collaborative filtering, there are two different approaches for the neighborhood proposed in the literature:

- User-based collaborative filtering: Two users are similar if they have rated or shown interest the same items in their profiles.
- Item-based collaborative filtering: Two items are assumed to be similar if the users, who have rated one of them, tend to rate the other one with similar ratings.

As collaborative filtering is based on the experience of the user's neighbors, the provided recommendations are more diverse than in content-based filtering. Furthermore, this technique does not require elaborately maintained content descriptions (Blanco-Fernández et al., 2008)

However, collaborative filtering also has drawbacks. The "gray sheep problem" names the limitation to provide relevant content to users with unusual preferences, very different from the remaining visitors. As their neighborhood is reduced, the recommendations are not accurate (Blanco-Fernández et al., 2008). The second problem relates to the sparsity of information about the interest of each user in each single object or product. The techniques using nearest neighbor algorithms may be unable to make many product recommendations. This problem is also

called reduced coverage (Sarwar et al., 2000). Furthermore nearest neighbor algorithms require a computation that grows exponentially with both the number of customers and number of products. Therefore, these techniques are limited in their scalability (Sarwar et al., 2000).

To overcome the scalability problem of neighborhood-based collaborative filtering, there are alternative approaches, e.g. latent factor models. A latent factor model tries to explain the interest or rating by characterizing both items and users with not directly observable properties, called latent features. Matrix factorization is one common possibility to identify these latent factors, see section 3.6.2.

#### Hybrid strategies

Hybrid strategies combine both content-based methods and collaborative filtering using the synergetic effects between recommending similar and additional objects to the user based on past preferences (Blanco-Fernández et al., 2008). In this way, the users are provided with more accurate recommendations compared to using only one strategy individually (Burke, 2002). One usual approach for a hybrid recommendation system is the usage of the content descriptions of the products defined in their profiles (content-based), as well as the levels of interest assigned to the product (collaborative filtering) to compute the similarity between users (Blanco-Fernández et al., 2008).

### 3.6.2 Matrix factorization for collaborative recommendation

By creating recommendation systems, matrix factorization methods can be used to derive a set of latent (hidden) factors from the relation between products and users. Both users and items are then characterized by the derived vectors of factors. In the domain of a web shop, the automatically observed factors can correspond to obvious aspects of a product such as the category or price level, but usually they can not be easily interpreted. The derived factors can be used in two ways: First, an item can be recommended to an user, if it is similar to the user with respect to the latent factors. Secondly, the matrix multiplication of the item and user matrix are an approximation of the original interest or rating matrix filling previous blanks (Jannach et al., 2010).

#### Formal definition

The initial matrix can be created using either explicit feedback, e.g. user ratings, or implicit feedback, e.g. interest using the browsing or transaction history of

the user (Koren et al., 2009; Jannach et al., 2010). By using explicit feedback, the matrix is usually sparse, as every user has rated only a small percentage of possible items. Implicit feedback will result in a more densely filled matrix (Koren et al., 2009).

In matrix factorization models, both users and items are mapped to a joint latent factor space of dimensionality $f$. The user-item interactions are therefore modeled as inner products in that space. Each item $i$ is associated with a vector $q_i \in \mathbb{R}^f$, and each user $u$ is associated with a a vector $p_u \in \mathbb{R}^f$. The vector $q_i$ measures the extent to which the item possesses those factors, whereas $p_u$ measures the extent of interest the user has in items, that comply with those factors. The scalar product of both vectors captures the interaction between each user and item resulting in an approximated relevance of an item (Koren et al., 2009):

$$\hat{r}_{ui} = q_i^T p_u \tag{3.5}$$

A common technique for capturing latent relationships between users and items is singular value decomposition (SVD) (Koren et al., 2009; Jannach et al., 2010). In simple terms, the SVD theorem by Golub and Kahan (1965) states that a given matrix $M$ can be decomposed into a product of three matrices as follows, where $U$ and $V$ are called left and right singular vectors and the values of the diagonal of $\Sigma$ are called the singular values:

$$M = U\Sigma V^T \tag{3.6}$$

The previously described matrix factorization model is closely related to SVD, where M represents the relevance matrix and $Q$ can be derived from $U\Sigma$ and $P$ from $\Sigma V^T$. Applying conventional SVD requires a full item-user matrix, which is not given in a recommendation problem. Therefore, the problem can be reduced by modeling and optimizing the observed values directly, while avoiding over-fitting through a regularized model (Koren et al., 2009). To learn the factor vectors ($p_u$ and $q_i$), the system minimizes the regularized squared error on the set of known ratings

$$min_{q*,p*} \sum_{u,i \in \kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(2|q_i| + 2|p_u|), \tag{3.7}$$

where $\kappa$ is the set of each known rating or interest. The system learns the model by fitting the previously observed values with the overall approach to generalize the relevance between items and users in a way that predicts unknown values. To avoid over-fitting of the model, the learned parameters are regularized. The constant $\lambda$ controls the extend of regularization, usually determined by using cross validation (Koren et al., 2009).

### Stochastic Optimization

To minimize the equation 3.7, there are different optimizing functions. As an example for a learning algorithm, the stochastic gradient descent optimization proposed by Simon Funk (2006) is described. The algorithm loops through all known relevance values in the training data set. The system predicts the relevance $\hat{r}_{ui}$ and calculates the prediction error by:

$$e_{ui} = r_{ui} - q_i^T p_u \tag{3.8}$$

The parameters are then modified by a magnitude proportional to $g$ in the opposite direction of the gradient:

$$q_i \leftarrow q_i + g \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \tag{3.9}$$

$$p_u \leftarrow p_u + g \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \tag{3.10}$$

On one hand, this approach is relatively easy in the implementation and on the other hand efficient in run time.

An enhancement of stochastic gradient descent is the Adam optimizer. Whereas stochastic gradient descent maintains a single learning rate for all weight updates, Adam computes individual adaptive learning rates for different parameters during the training (Kingma and Ba, 2014). Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods. Therefore, it became a popular algorithm in the field of machine learning.

### Bayesian personalized ranking loss

The previously described optimization problem is referring to the pointwise learning to rank approach. Even though matrix factorization can be used for the prediction of each item's relevance necessary for a personalized ranking, the approach is not directly optimized for rankings (Rendle et al., 2009).

Bayesian personalized ranking (BPR) uses a pair-wise interpretation of the implicit feedback matrix. The overall approach is to learn the personalized ranking by considering the known order of item pairs. The concept reconstructs the input matrix containing only positive feedback from the user into pairwise positive or negative preferences: given a pair of items $(i, j)$, the user either prefers i over j (positive) or dislikes i over j (negative) (Rendle et al., 2009). The model is thus trained through negative sampling: for any known user-item pair, one or more items are randomly sampled to act as negatives (expressing a lack of preference by

the user for the sampled item). The paper "BPR: Bayesian personalized ranking from implicit feedback" by Rendle et al. provides a detailed description of this pairwise learning to rank method.

In summary, matrix factorization can be used to recommend items to users by imputing the relevance using two matrices with latent factors, representing items and users. The matrix factorization can be implemented in an iterative approach by minimizing the loss between the predicted and actual relevance in the training data set (pointwise approach) or by considering the order of item pairs (pairwise approach). This technique is used to build individualized product rankings.

This chapter summarized all techniques, which are relevant for the implementation of the proposed prototypes. The methods include a predictive task with gradient boosted decision trees, a segmentation task using k-means and a recommendation task by applying a matrix factorization. Furthermore an evaluation metric was presented, which is used to compare ranking in their quality.

# Chapter 4

# Initial product ranking with predictive approach

This chapter describes the product ranking project, which is the baseline for this piece of work and was developed beforehand. All advancements to the implementation are presented in chapter 5. The product ranking in the initial project follows a predictive, but non-personalized approach. The underlying idea is, that products with a high probability to be bought should be ranked on top of products with a lower expected sales. Therefore, the model provides a sales prediction for each product and ranks the items accordingly. The overall goal is a performance-based product ranking finally leading to better conversion rates and to an increased sales volume.

## 4.1 Data understanding and preparation

For this piece of work, three different data sources are available and used: product, transaction and tracking data. The product data includes information about each product itself on a daily basis, e.g. product name, price and availability. The transaction data contains each purchased items with detailed information connected with an order and customer identifier. To comply with the GDPR, these identifiers are encrypted. Furthermore, no personal customer data, e.g. name or address, is available. The third data source is the web tracking from the web site. Different kinds of interactions with the online shop are tracked and collected in a database. The visitor identifier is relying on the user's cookie. All personal information in the tracking data is also encrypted. In this piece of work, only product detail page view events are used from the set of tracking data. The transaction data from online purchases can usually be connected with the corresponding tracking information. However, the transaction data also includes purchases in a local store, via phone or mail.

As the ranking of products is based on their sales prediction, the model must be trained on sales data. The target variable is defined as the average sales in the upcoming eight days for each product. The reason for not only using one day is that most products are not bought every day. To keep the importance of the weekday the target considers eight days and therefore, the prediction date's

weekday twice.

The features used in the initial project are combined from multiple resources:

- Product specific data, e.g. products category and availability
- Transaction data, e.g. number of ordered units in the past days
- Tracking data, e.g. number of product detail page views
- Other data, e.g. day of the week.

These features are processed every day and stored in a database as they are necessary for a daily training and prediction.

## 4.2 Modeling

By predicting the sales, a continuous variable, this implementation is an estimation problem. Therefore, a supervised learning method is used (see section 3.3). In this project, modeling with XGBoost has proven to be very efficient.

XGBoost is short for "Extreme gradient boosting" and is a scalable and portable library. It relies on the basic concept of gradient boosting with an ensemble of classification and regression trees (see section 3.4.2). Thereby, XGBoost is an implementation of gradient boosted decision trees designed for speed and performance (Chen and Guestrin, 2016). XGBoost became popular by being extensively used to create state of art data science solutions resulting in a long list of winning solutions in machine learning competitions (Chen and Guestrin, 2016).

The model is trained by using 21 days historical data with the known target, the actual average sales on the day and the seven subsequent days. During the development of new feature variables the algorithm is fitted multiple times in a so-called back testing and evaluated on a metric described in the next section. Furthermore, the model is trained using different parameters (grid search). By comparing the average result for each parameter set, the model is optimized.

The product ranking is then created by ranking the products according to their sales prediction in descending order.

## 4.3 Evaluation of model quality

To compare different product rankings and to optimize the predictive model, an evaluation metric is used. For this problem the normalized discounted cumulative gain (nDCG) is particularly suitable as it is a measure of ranking quality. The

nDCG is introduced in section 3.2. The relevance of the items is defined by the actual sales value of this product. By evaluating the models performance with the nDCG measure, this project follows the concept of a listwise learning to rank approach (see section 3.1).

In general, one global ranking is provided for the whole web shop, but a web visitor is always seeing a sub-list related to a topic, more specifically to a product category. Therefore, the nDCG is adapted and calculated as average nDCG for each sub-list. Although an item can be displayed in several product categories, the metric calculation considers each item only once in its main category path.

Additionally, the adapted metric only examines the products until the certain position, as over 90 percent of the clicks from the product list to a product detail page occurs on the first two pages. In this way, the order of the products in the position below the threshold do not have any impact on the score of the metric used in this project. Products with a high relevance ranked in the top positions let the score increase. If an item with lower relevance displaces the optimal ranking, the score will decrease.

## 4.4 Deployment

The deployment process of this initial project is shown in figure 4.1. A data flow is displayed as a black arrow, whereas the order of execution steps is market with grey arrows.

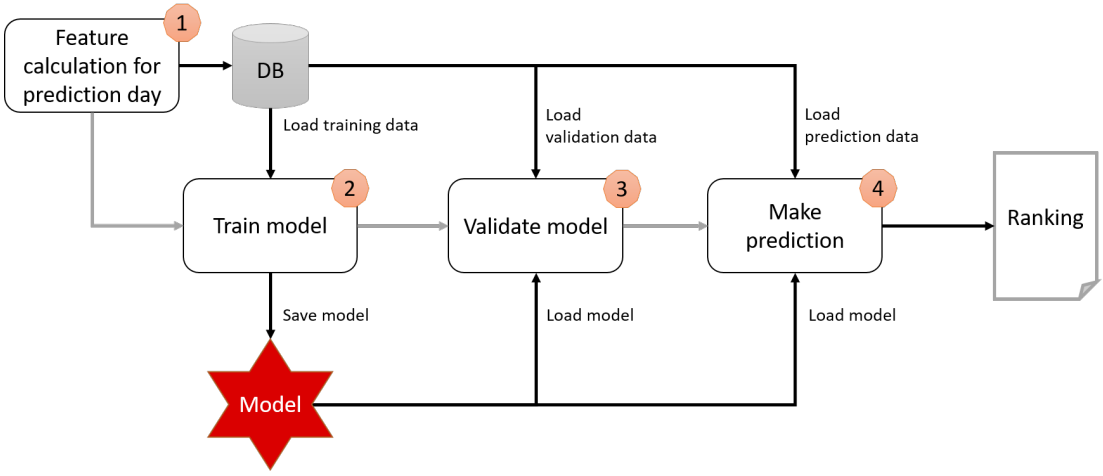



Figure 4.1: Deployment process of productive system

The first step is the feature calculation for the prediction date. Furthermore, the target for historical days is updated and stored with all features in a database.

As this calculation is performed daily, the features are stored long term and can be used as training data afterwards. In the second step, the model is trained using 21 days of historical data. The model is saved to be accessed later. To guarantee consistent quality, the model is validated on other historical data. If the evaluation score is below a predefined threshold, the new model is discarded and an older model is reactivated for the prediction. The last step is the actual prediction by loading the features and applying the active model. The estimated ranking is exported and uploaded to be displayed in the web shop. In case of any failure, e.g. there is no or not enough data for the prediction, the ranking of the previous day is used.

The exported ranking gives a score and thereby a position for each product. In the web shop, there is no page displaying all available products. A customer can only see sub-lists of items which belong to a selected product category. These sub-lists are ordered accordingly to the provided ranking.

## 4.5 Limitations of the current solution

The main limitation of the implementation is, that the optimal ranking is not known. It is only an assumption, that products with a higher sales volume should be ranked above products with lower sales figures. Furthermore, the actual sales, which is used as target, is already influenced by the displayed ranking on that day. The resulting cross effects are not known. An opportunity is to overcome this problem by considering the position of each product as a feature or as a weight factor in the target.

The current solution provides only one ranking per day. Besides any kind of personalization for customer, there could be multiple rankings depending on the time of the day. This piece of work will concentrate on the enhancement of the product ranking in terms of personalization and does not try to overcome the described limitations, but it is important to keep the limitations in mind.

Chapter 5

# Implementation of product rankings with personalized approach

This chapter describes the implementation of two prototypes for personalized rankings. The first prototype is based on a customer clustering, resulting in a personalized product ranking for each identified customer cluster. The second prototype considers each customer as an individual and enriches the original ranking with personalized recommendations. Both approaches have advantages and disadvantages, which will be pointed out by describing the implementation process and evaluation results in this chapter. Although the implementations are only prototypes, the author takes into account, that a productive realization is possible and practical.

## 5.1 Cluster-based personalization

### 5.1.1 Prototype overview

The implementation of the cluster-based product ranking involves two machine learning tasks: on the one hand the customer clustering and on the other hand the prediction of the ranking for each customer cluster, as shown in figure 5.1.

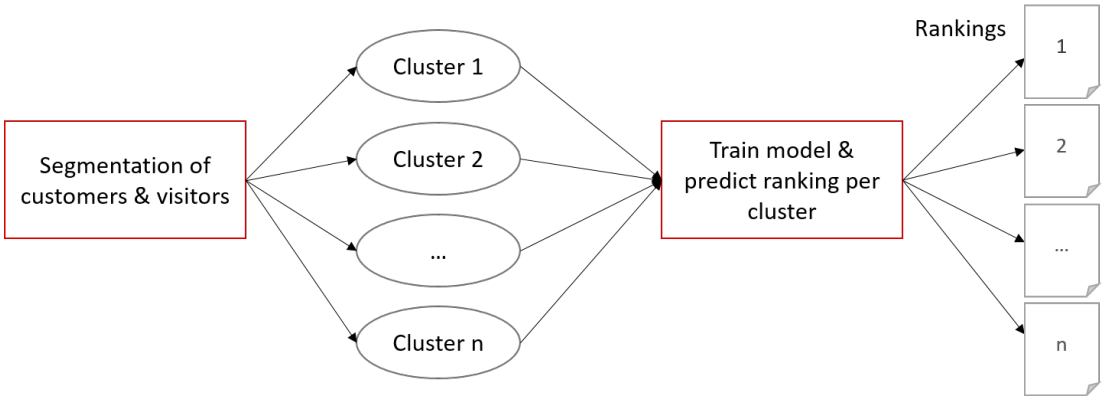



Figure 5.1: Overview of cluster-based approach

The approach is to identify customers with similar interests or behaviour, which

are likely to be interested in similar products in the future. It is expected that the rankings are closer to the customer expectations and therefore of better quality.

In a productive environment the customer clustering is trained once or on a regular basis, e.g. once a quarter. However, the model is applied on a daily basis to assign the customers with updated information to a cluster. The predictive model is then trained and applied resulting in a product ranking for each customer cluster. The rankings are provided to the web shop, which will import all rankings into the sort index. When a user enters the web shop, the personalized ranking of the user's cluster should be displayed. For this purpose, a web service can be implemented, that returns the daily estimated *cluster_id* when called with a visitor or a customer identifier. In case of only using situative data, the web shop can select the suitable product ranking by itself.

### 5.1.2 Implementation of customer clustering

This piece of work follows two approaches for the customer clustering using on the one hand situative data and on the other hand historic implicit data. Situative data is given by the current situation of the customer, e.g. time, location, user-agent or traffic sources. The historic implicit data describes the past behaviour of the shop's users. An overview of the different data types is given in section 2.5.3.

#### Situative data

The usage of situative data, in this case the purchase time, was a by-product of this thesis. In order to test the general functionality of the overall ranking calculation, the customers are split into deterministic groups by the time of their purchase. The resulting product rankings performed better than expected and the author continued to pursue this approach.

| Cluster id | Description | Time range | Number of orders |
|---|---|---|---|
| 0 | Night | 0:00 AM - 5:59 AM | 2.5% |
| 1 | Morning | 6:00 AM - 10:59 AM | 20.4% |
| 2 | Lunch time | 11:00 AM - 1:59 PM | 21.4% |
| 3 | Afternoon | 2:00 PM - 4:59 PM | 20.7% |
| 4 | Early evening | 5:00 PM - 7:59 PM | 17.5% |
| 5 | Late evening | 8:00 PM - 11:59 PM | 17.4% |

Table 5.1: Clustering using the order time

The time clusters are built by defining usual time periods during a day (morning, lunch time, afternoon, etc.) resulting in groups with approximately equal

sizes, see table 5.1. Only the cluster of purchases in the night (cluster id 0) is considerably smaller.

An analysis of the ordered products at different times during the day did not show significant differences in the corresponding categories, see figure 5.2. The only noticeable exception is a considerable higher share of purchases in the first category during the night between 0 AM and 6 AM. However, the absolute amount of orders in that period is lower than during the day periods.

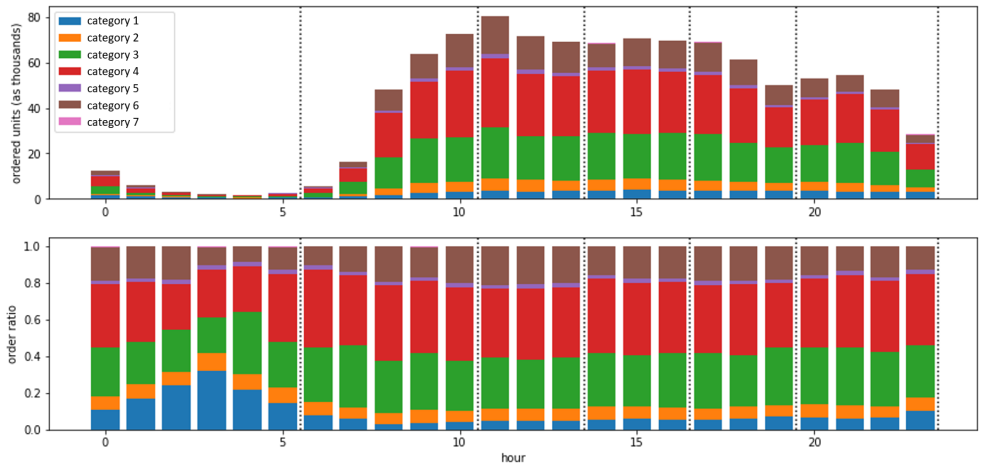



Figure 5.2: Amount and share of purchases per category

Although the analysis did not show major differences in the main categories, it is plausible, that a person, who orders in the morning or early afternoon has different interests than persons, who are browsing and ordering in the evening. Therefore, this piece of work will show, that different product rankings for each defined time period can reflect the interests of the web-shops visitors in a better way.

### Customer behaviour clustering - User identification

The customer clustering is based on both web-tracking and transaction data. To calculate the product rankings, each known customer with a *customer_number* has to be assigned to a cluster. Each user, visiting the web-shop, is being allocated by a *visitor_id*. To display the correct product ranking, it is essential to connect each *visitor_id* with its corresponding *customer_number*. Visitors, who did not order anything yet, do not have a customer identifier. Nevertheless, the returning visitors should be assigned to a cluster in a similar way as known customers.

Figure 5.3 shows a schema, where a single user (left icon) interacts with the shop in several ways. The available data sets include transaction and tracking

data from a number of retailers, that can be matched by identical device or web cookie identifiers resulting in a global customer number. An analysis has shown, that 78% of the partnering retailer's customers are not interacting with any other online shop in the data set. Furthermore, the product assortment is quite different from other shops. Therefore, the gain of using the transaction or tracking data from other retailers is very low and the decision was made to use only the data from the partnering retailer to build the prototype.

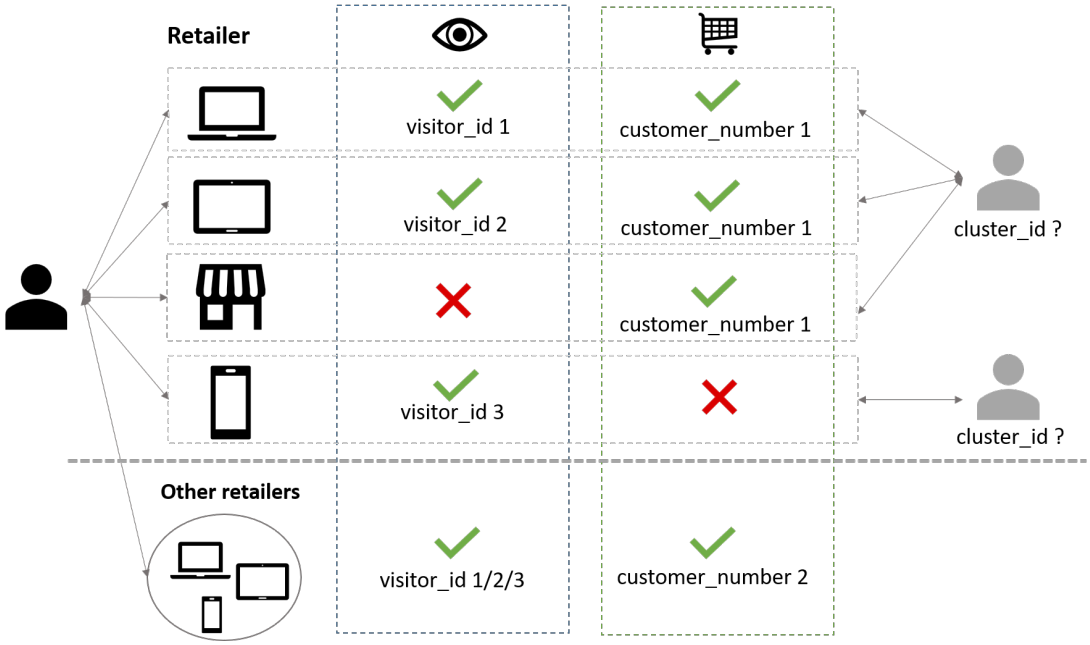



Figure 5.3: Schema for the customer definition and handling

The same user can still interact with the shop through multiple channels, e.g. a computer, tablet, mobile phone or if he is going to a local store, as presented in the figure above. If a user is going to a local store, there is clearly no tracking data available. But if he is using a loyalty card, his purchases are stored together with his customer number in the transaction data. If the user is visiting the web shop on different devices he is tracked with a different visitor identifier. But if the user is logged into his account or purchases something online, the visitor identifier can be connected with the unique customer number. In case of the user displayed in figure 5.3, the shop interactions from the computer, tablet and local store can be joined to one internally used technical user by using the known customer identifier. The tracking data from the mobile phone belongs to the same physical user, but can not be assigned to the same technical user, as no similar identifier is known. Therefore, the same person is represented by two technical users in this implementation. Both technical users have to be assigned to a customer cluster and if the behaviour of the physical user is similar in all channels, both technical users should be assigned to the same cluster.

All available data points are assigned to a technical user following the process described above. The following sections will refer to the technical users defined here, when using the term user.

### Customer behaviour clustering - K-Means

The features for each user can be composed by both, tracking and transaction data, or only one of these data sources. If a user only visited the web site and did not buy anything, it is still possible to assign a cluster.

In general, the following features are used: affinity to product categories, price sensitivity and affinity to new products. Using transaction and tracking data as different features next to each other has some disadvantages. The users, that only visited the shop and did not purchase anything, will have a high distance to actual customers, although they might have very similar interests. The sparsity of the purchase-based features results in undesirable distances for the clustering algorithm. Instead the same features for both tracking and transaction data are defined. Both, a product detail page view and the purchase is defined as interest in a product. In this way it is possible to aggregate all data points from different data sources belonging to a user into the same features. For reasons of topicality, in each case only the interactions from the last year are considered.

The following features are calculated and used:

- Affinity to product categories:
  Proportion of product detail page visits or purchases in this category
  - first level category (described by numbers 1-7)
  - second level category (described by letters, e.g. A, W, F)
- Price sensitivity:
  Sensitivity averaged over viewed and purchased products
  - product price lower than average price in category
  - product price ratio to highest price in category
  - lowest price in ratio to product price
- Affinity to new products:
  Proportion of new products viewed or purchased
  - newer than 90 days
  - newer than 30 days
  - newer than 10 days

Due to the large number of categories, the listed features result in about 100 columns. The same scaling of all features is important for a clustering algorithm.

As all features are proportions, the values range between 0 and 1, and no further scaling is necessary.

The algorithm k-means (see section 3.5.2) is used as clustering method, as it is easy to apply and results in reasonable customer clusters. In the implementation the *MiniBatchKMeans* class of the python package *sklearn* is used. The *MiniBatchKMeans* is a variant of the k-means algorithm, which uses sampled batches to reduce the computation time, while still attempting to optimize the same objective function (Pedregosa et al., 2011). The clustering model can be trained iteratively with new visitors and customers.

The number of clusters is determined using the elbow method, displayed in figure 5.4. With an increasing number of clusters ($k$), the intra-cluster distance decreases. Beyond a certain number of cluster the distance is not decreasing appreciably and all customers close to each other are already assigned to the same cluster.

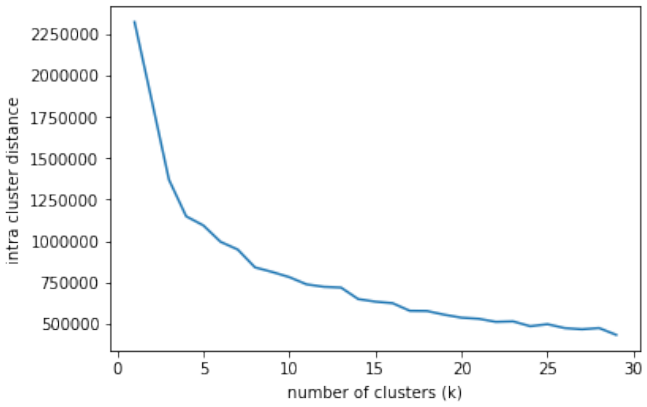



Figure 5.4: Elbow graph to define number of clusters

Based on this elbow graph, the author decided to proceed with three versions of the customer behaviour-based clustering:

- Version 1: Clustering with 17 clusters
- Version 2: Clustering with 17 clusters and more focus on newness features
- Version 3: Clustering with 25 clusters

A detailed description of all three clustering versions is added to the appendix, see page 74. These three versions as well as the time-based clustering are used to predict personalized product rankings in the following sections.

### 5.1.3 Implementation of product ranking

Referring to the implementation overview in section 5.1.1, the first machine learning task is finished with the clustering. The second step, the estimation of the actual product rankings, is based on the previously described baseline implementation of product rankings with a predictive approach, see chapter 4.

#### Data understanding & preparation

The target of the baseline implementation is the actual sales per product in the upcoming eight days. The predictions are then used to rank the items accordingly.

In the cluster-based prototype this target has to be calculated for each cluster separately. Therefore, the target generation has to be adjusted and the sales for each day, product and cluster is aggregated. The number of required predictions increases to the number of products in the baseline multiplied by the number of clusters. The identifier in the data set are thus the product number, date and cluster identifier.

The same way as the target, the features have to be calculated depending on the cluster. In the baseline model, the sales data (ordered value and units) of each product in the last 30 days is used. In the personalized prototype these features are again provided per product and cluster identifier. Nevertheless, the global sales data can also be used next to the cluster-specific features. All other features (e.g. the products availability or newness) are used unchanged, as they are not dependent on the customer clusters.

The target and feature preparation is done by extracting the data from a Hadoop cluster and transforming it using Hive and Pyspark functions. The results are then stored in a database for model training and prediction.

#### Modeling

The modeling technique is not changed in comparison to the baseline implementation. The model is trained by an Extreme gradient boosting using the python package XGBoost.

To create a prediction for each product and cluster, there are two different approaches: On the one hand the baseline implementation could be applied for each cluster only using the cluster-specific data. In this case $N$ models have to be trained and applied, where $N$ is the number of clusters. On the other hand, one model can be trained, where the input data is a observation for each product, date and cluster identifier. The second approach was preferred in this piece of

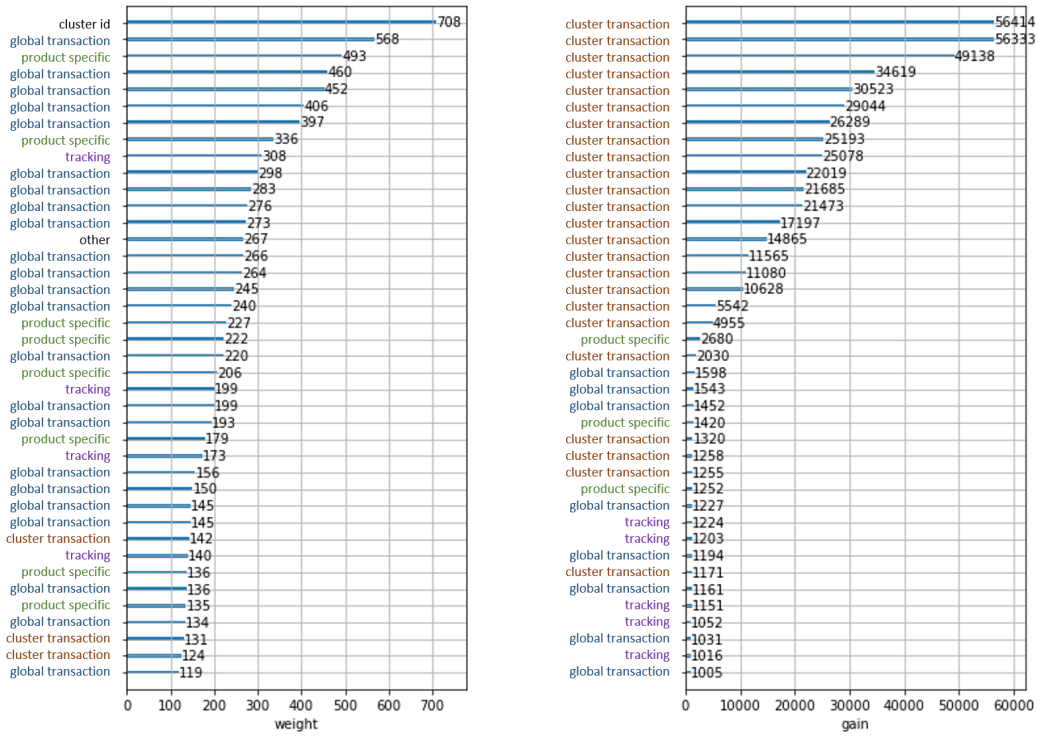



Figure 5.5: Feature importance in gradient boosted tree model

work, because the model can learn the effects from a larger amount of input data and profit from the cross effects between different customer clusters. In this case, the cluster is not only an identifier, but can also be used as a feature giving the model the chance to learn customer cluster specific characteristics.

To train one model and predict the target for different clusters, several code changes are made. The row identifiers are enhanced with the cluster identifier and the configuration of the features and target source tables is updated. The way the rankings are saved into a data base or file is also adjusted.

Due to the changes in the features and especially the significantly higher number of observations, the parameters of the model have to be regulated to improve its performance. This parameter tuning is conducted using a cross validation over a longer period of days and a grid search testing a variation of different parameters. It is also tested whether adding or removing some features (e.g. the global transaction history) has an advantage. In the case of the order value, it was an advantage to use both as features, the sum of order value in the specific cluster and the total for all clusters.

The fitted model gives the opportunity to evaluate the feature importance. In figure 5.5, the 40 most relevant features in terms of weight and gain are displayed, encoded by their feature type: global transaction, cluster transaction, product

specific, tracking or other feature data. The importance type "weight" is defined as "the number of times a feature is used to split the data across all trees" and "gain" as "the average gain of the feature when it is used in trees" (DMLC, 2016). In terms of weight, the most important features are the cluster identifier and global order values in the transaction history. In terms of gain, the features that are related to the single clusters have a higher importance: recently ordered units and ordered value in this cluster. Therefore, both global and cluster-related features are important to fit the model.

The implementation of this prototype allows to create product rankings independent from the number of clusters or products, as well as the number of training days, the selected features or the target variable. These information are provided in configuration files and can be easily varied.

#### Product ranking

After predicting the sales value in the upcoming eight days for each product and cluster, the prediction is scaled between 0 and 1. The final ranking file, which is generated on a daily basis, contains one column per cluster and a row per product. The first rows of an exemplary ranking file with six time-based clusters is presented in table 5.2. The higher the score, the higher the product will be displayed in the shop. In the example, product "PN-100004" seems to sell better for cluster 2, 3 and 5 than 0, 1 and 4.

| **product_number** | **0** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|---|
| PN-100000 | 0,04449 | 0,04506 | 0,04506 | 0,04506 | 0,04506 | 0,04506 |
| PN-100001 | 0,13791 | 0,56948 | 0,17536 | 0,17536 | 0,17536 | 0,57015 |
| PN-100002 | 0,12410 | 0,16042 | 0,16042 | 0,16042 | 0,45363 | 0,16042 |
| PN-100003 | 0,11172 | 0,13862 | 0,13862 | 0,38733 | 0,13862 | 0,13862 |
| PN-100004 | 0,12487 | 0,15990 | 0,50682 | 0,50706 | 0,15990 | 0,50601 |
| ... | ... | ... | ... | ... | ... | ... |

Table 5.2: Exemplary ranking file for six clusters

### 5.1.4 Evaluation

To evaluate the different versions of the clustering, the normalized discounted cumulative gain (nDCG) is used, as described in section 3.2. This metric calculates the gain based on the actual sales per position in comparison to the perfect ranking in each sub category. The position in the ranking is thereby based on the predicted sales. In the perfect ranking, the products are sorted descending with their actual sales in this cluster, which results in an nDCG of 1.

This gain is calculated for each cluster and averaged weighed by the total sales in each cluster. Furthermore the rankings are calculated and evaluated for every sixth day in the time period between August 15th and October 31th, 2017. The same evaluation is done by using the baseline ranking of each day for all clusters and the resulting nDCG values are presented in table 5.3.

| **Clustering** | time-based | V1 | V2 | V3 | V3 optimized |
|---|---|---|---|---|---|
| **Cluster-based nDCG** | 0.544 | 0.516 | 0.493 | 0.537 | 0.66 |
| **Baseline comparison** | 0.500 | 0.394 | 0.390 | 0.380 | 0.380 |

Table 5.3: Ranking evaluation at cluster level

The results can be interpreted as follows: If the cluster-based nDCG is higher than the baseline, the personalized rankings are closer to the target ranking per cluster. This is valid for all clustering versions. By optimizing the parameters in the same clustering version, the ranking is better by an increasing nDCG. This is can be observed between column "V3" and "V3 optimized", where the last mentioned was optimized in the model parameters. However, this metric has two major disadvantages: It is not possible to compare the nDCG values of the different clustering versions as the optimal ranking is dependent on the definition of the clusters. Furthermore the differences in the metric are not easily interpretable.

In order to achieve a comparable evaluation metric, the evaluation using the nDCG can also be applied on a single customer basis. In this case, the perfect ranking is very sparse, as every customer orders only some products per day, but the methodology is the same. Therefore, 10000 random purchases in the period from August to October 2017 were selected. About the half of the customers are known, which are approximately 55% of the purchases (in some cases multiple purchases per customer on different days). The product rankings calculated for the cluster-based evaluation can be used to evaluate the ranking for each single customer in this subset of purchases. In this way the metric is comparable with different kinds of the customer clustering.

To solve also the problem of interpretability, a second metric has been defined: the average position of purchased products. The position of each product is defined by ranking the prediction score in each subcategory. The positions of all purchased products are averaged per customer and then over the whole sample of customers. The results of both metrics are presented in table 5.4.

Using the described evaluation sample, the non-personalized baseline achieves the smallest nDCG and thus it is shown, that both the situative and behaviour-based customer clustering are closer to the interests of the customers. The parameter optimized third version of the customer clustering achieved the best results. The

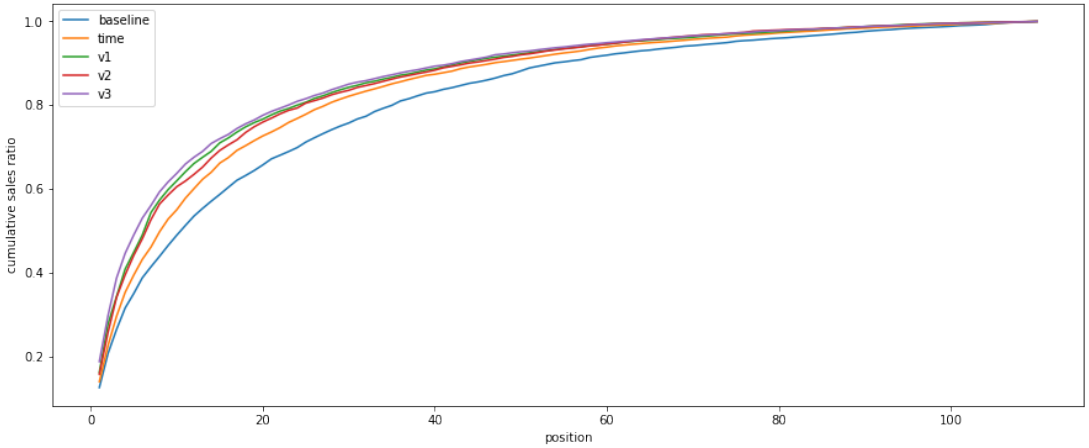



Figure 5.6: Cumulative sales value distribution with increasing position

same outcome can be seen in the average product position metric. In the baseline ranking the products that have been bought by each single customer are on average on position 27.6 for known customers and 28.6 for all customers. Already by using different lists during the day, the interests of the customers are better addressed. In this way, purchased products are no longer in place 28.6 on average but in position 23.9 for all customers (22.4 for the known customers). The best customer-behaviour clustering improves the position of the purchased products to 20.0. The main advantage of the time-based clustering is, that it is applicable for all users, whereas the behaviour based version is only working for about 55% users, which are already known, can be identified and have given their consent to data usage.

| **Ranking** | Baseline | time-based | V1 | V2 | V3 | V3 optimized |
|---|---|---|---|---|---|---|
| **Customer nDCG metric** | 0.245 | 0.285 | 0.326 | 0.314 | 0.353 | 0.365 |
| **Avg. product position** | 27.60 | 22.36 | 22.06 | 22.04 | 20.96 | 20.01 |

Table 5.4: Ranking evaluation at customer level

The effect of moving the historic sales value into higher positions is also pictured in the cumulative plot in figure 5.6. The blue line represents the baseline, summing up the ratio of the total sales per position (referring to the same customer sample as before). Using the personalized rankings improves the sales ratio in the front positions, which moves the line of the cumulative sales ratio into the upper left corner.

This prototype was evaluated on a theoretical basis using historic data. The

products, that are purchased, have been moved to the upper positions by creating customer clusters, which had similar interest beforehand. By having a personalized ranking with more interesting products in the upper positions, the expectation would be, that the customers experience improves and therefore also the sales volume increases. These effects can only be proven e.g. by implementing an A/B test, where some users receive the baseline ranking and the others will be provided with a personalized ranking. A proposal for the implementation of an A/B test is presented in chapter 7. In doing so, a positive effect in the future sales could be shown, but this is not part of this piece of work. Nevertheless, the subjective perception of the rankings is evaluated by shop experts and described in the following section.

### 5.1.5 Description of clusters and rankings

Based on the evaluation of the clustering versions, the third behavioural clustering version can be identified as the best variant in the evaluation setting. This clustering divides the known customers and visitors into 25 different segments. The clustering is based on the affinity to first- and second-level categories, the newness and price of purchased or viewed products. This allows a description of the user clusters by their main features. A complete overview of these features per cluster can be found in the appendix on page 74. No cluster is extremely large having sizes ranging from 2-7 percent of the users.

Four out of the 25 clusters are related to the main category 4. These four user clusters have the highest proportion of female users: almost 90% in cluster 1 and 6, 50% in cluster 20 and 18% female users in cluster 11. All other clusters have a considerable higher proportion of male than female users. It is also remarkable that these clusters have an older customer base with an average age of 50-53.2 years. In comparison, the average over all clusters is 45 years. The gender and age of the users are not used for the clustering, but offer interesting information in this subsequent evaluation.

Eight clusters are mainly focused on the category 3, showing more than 75% of the interests in products of this category. The clusters differ in their second-level category and their price sensitivity. Whereas all of the users in this eight clusters will mainly browse through the main category 3, the finer division allows to rank the products within this main category closer to the users interests.

Similar to the clusters of category 3, there are six clusters with a main focus in category 6 differing in the second-level category. All six clusters are characterized by a share of over 97% of male users. Furthermore, there are two customer segments that have a similar strong interest in both main categories 3 and 6. One of this clusters has a strong focus on the subcategory A.

In category 1, three clusters have been identified. Whereas the two clusters show 90% of interest in this main category, the third user cluster also shows interest in category 6 (14%), category 3 (8.6%) and category 5 (8.1%). Therefore, it is the cluster with the highest share of interest in products of the main category 5.

The remaining two user clusters have the highest share of interest in products on the landing page (category 2), where a mix of products from different categories is presented. The main category 7 accounts for a very small proportion of total sales and is not included as focus in any of the clusters mentioned.

In addition to the theoretical analysis of the personalized product rankings for the described user clusters, the aim of this thesis is to assess the subjective perceived quality of the rankings. Therefore, nine shop managers of the retailer are asked to give their feedback on the noticed differences and consistency of the rankings.

The main problem at this point of the thesis is, that the web shop's current implementation does not allow to import the 25 different rankings. To evaluate the ranking's subjective quality, it is essential to display the products in the desired order. Therefore, the author implements a script, that generates HTML files given a ranking file and a product category. In this way, it is possible to display the different rankings similar to the presentation in the official web shop. The main difference is, that the products in this implementation occur only in one category, which is stored in the available product data. In the official web shop, a product can be displayed in multiple categories, but these further categories are not provided in the available data sets.

Based on the identified user clusters six categories are selected and overall 19 exemplary product ranking generated to be evaluated. In doing so, care was taken to maintain an even ratio of common categories and clusters. At the same time, the number of lists should not be too large to minimize the effort for the respondents. The six category overview pages for two to six relevant clusters each are summarized and displayed in one document. In addition, the baseline ranking is also provided for each category. For comparability, August 14, 2017 was selected as the prediction day for all rankings. The experts are asked about each personalized list of products:

1. whether they see a clear difference between this list and those from other clusters and the baseline, and
2. whether the products are suitable for the described user cluster.

These two statements are rated on a scale from 1 to 4:

1: I do not agree
2: I rather do not agree
3: I rather agree
4: I agree

In addition to the rating of the two statements, the experts were asked to comment their thoughts about each presented ranking and the overall impression. The answers of the experts are averaged for both statements and summarized in table 5.5.

| Cluster id & description of its focus | | Noticeable difference[1] | Suitable for the cluster[1] |
|---|---|---|---|
| Products ranking for subcategory 6.W | | | |
| 19 | Cat. 6.W expensive | 3.3 | 2.8 |
| 22 | Cat. 6.W cheaper | 1.7 | 2.4 |
| 2 | Cat. 6.W and mix with others | 2.1 | 2.4 |
| Products ranking for category 3 | | | |
| 16 | Cat. 3.O | 3.8 | 2.9 |
| 9 | Cat. 3.A | 4 | 3 |
| 7 | Cat. 3.W & 3.E | 3.9 | 2.8 |
| 23 | Cat. 3.W cheaper | 4 | 3.3 |
| 15 | Cat. 3 mix, expensive | 4 | 2.6 |
| 24 | Cat. 3 mix, cheaper | 4 | 1.9 |
| Product ranking for category 1 | | | |
| 3 | Cat. 1.G | 3 | 2.8 |
| 8 | Cat. 1.C | 2.8 | 2.9 |
| 14 | Cat. 1 mix | 2.7 | 3 |
| Product ranking for subcategory 3.K | | | |
| 0 | Cat. 3.K cheaper | 2.8 | 2.7 |
| 18 | Cat. 3.K expensive | 2.8 | 2.8 |
| Product ranking for subcategory 4.F | | | |
| 6 | Cat. 4.F expensive | 3.3 | 3 |
| 1 | Cat. 4.F cheaper | 3.3 | 2.9 |
| Product ranking for subcategory A | | | |
| 9 | Cat. 3.A | 3.8 | 3.1 |
| 10 | Cat. 6.A | 3.2 | 3.2 |
| 4 | Mix of 3.A & 6.A | 3.1 | 3.1 |

Table 5.5: Subjective perception of the rankings by nine shop experts

In 15 out of the 17 presented lists, the experts can notice a difference to the other clusters and to the baseline. The effect is most clearly recognizable in the main category 3. The experts also rated the products in the rankings as rather suitable for the clusters. In the comparison of cluster 15 and 24 in this category it can already been seen that an affinity for rather cheap or expensive products is not well reflected. The same problem occurs for the categories 6.W and 3.K.

[1] 1 - I do not agree, 2 - I rather do not agree, 3 - I rather agree, 4 - I agree

The different product rankings in the categories 1, 4.F and A are rather fulfilling the examined statements. The experts' comments explain, that there are noticeable differences. In general, more products are presented, which are suitable for each described user cluster. However, there is still more overlap between the different lists than expected.

The feedback on the overall impression also indicates that there are partially strong differences, but also similarities in the product rankings. Some bestseller products keep coming up in all clusters. The clearest difference was achieved in the clusters with focus in category 3 and overall the the cluster descriptions match the lists quite well. It has also been noticed that the price of a product seems to be more important than the purchase frequency, which is caused by the selected target. In this case, another target is already intended, which is not only based on sales but also on newness and availability. The biggest weakness of the presented product rankings is the too weak difference in the price affinity, which should therefore be considered in more detail during the further development. Overall, the shop experts are very interested in the solution and would like to continue the cooperation for personalized product lists.

### 5.1.6 Discussion

The past section of this thesis presented an opportunity to personalize product rankings in a web shop by differentiating between different clusters of users.

In a simple approach the users can be segmented by the time of their purchase. In the more complex behavioural clustering, the users are joined from multiple sources (transaction and tracking data) with partly different identifiers (e.g. different visitor ids from multiple devices). The category, newness and price information of the purchased and viewed products form the features for clustering.

Four different clustering versions are then used to build the personalized ranking. These rankings have been compared with the baseline implementation, which is a prognostic product ranking without any personalization. All versions of the clustering have shown better results in all applied metrics in comparison to the baseline. While the non-personalized list ranks the actually purchased products on average to position 27.6, a behaviour-based personalization could achieve an improvement up to position 20.0. For unknown users, the author recommends using situative clustering, e.g. by time, as a strong improvement in product positions could also be observed there.

Based on these observations, it can be said that this prototype complies to the first hypothesis: By implementing a clustering based on previously known customer behaviour, and using these clusters for the generation of personalized product

ranking, the resulting rankings are closer to the customers' future purchasing behaviour.

To verify the second hypothesis about the perceived quality, the product rankings of the best behavioural clustering are displayed in a mock-up of the actual web shop and rated by experts of the collaborating shop. The experts did recognize a noticeable difference between the personalized rankings in comparison to the baseline or other clusters. The presented products are rather suitable for most of the associated and described clusters. The biggest weakness is identified in the differences in price affinity, which are too small. Overall the personalized rankings were observed as being of better quality. The experts think that the use of the personalization can bring an advantage in the customer experience and therefore the second hypothesis can be confirmed.

The cluster-based prototype complies to both hypotheses. The perceived benefits, but also limitations, will be used to implement a productive system and measure the actual effect in an A/B test, see chapter 7.

## 5.2 Individualized product ranking

### 5.2.1 Prototype overview

The aim of the second prototype is to provide each known user with a unique product ranking aligned to his interests. The previously used baseline ranking is combined with individual factors, as illustrated in figure 5.7.

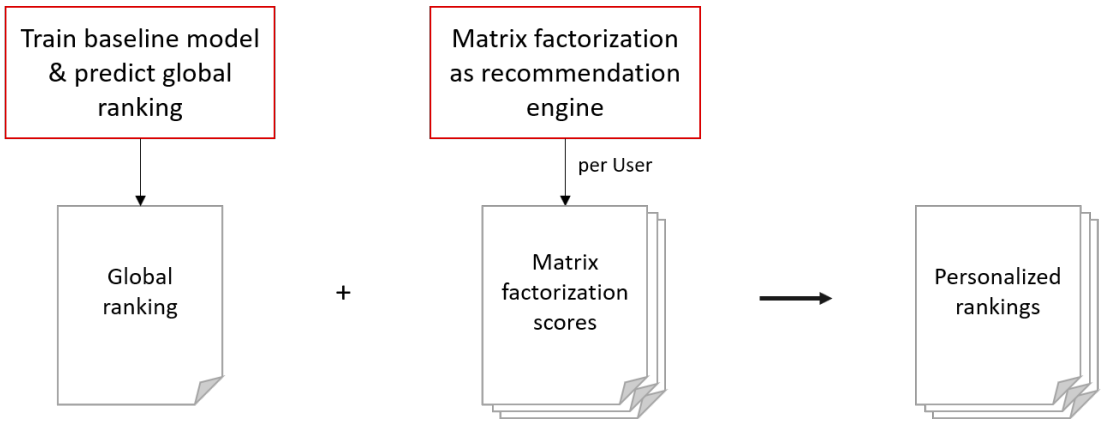



Figure 5.7: Overview of individualized approach

The baseline calculation of a predictive product ranking stays unchanged in this prototype. The ranking learns overall trends, recognizes bestseller products and considers the availability of products as well as the day of the week. This global applicable ranking is predicted on a daily basis.

For the personalized rankings, individual factors of each customer are considered additionally. To estimate the recommendations a matrix factorization is used, whereas other options are presented in section 3.6.1. The method is chosen in relation to the realizabilty in a productive environment as follows. The matrix factorization results in user and item factors. The baseline ranking and the item factors could be provided to the shop on a daily basis. The user factors could then be transmitted using a web service, to calculate the individualized ranking.

Overall the implementation is rather a proof of concept, whereas the first prototype based on customer clusters is closer to a productive system. So far, this prototype is limited to transaction data only, considering each customer as a user. The target vision for a productive system of the individualized product rankings would include an implementation for customers and visitors combining transaction and tracking data like in the cluster-based prototype.

### 5.2.2 Implementation of matrix factorization

The individual scores for the personalized ranking are estimated using a matrix factorization. The general concept is presented in figure 5.8, whereas a detailed description can be found in section 3.6.2. The technique basically reduces a sparse item-user-matrix into two usually smaller matrices: user factors and item factors.

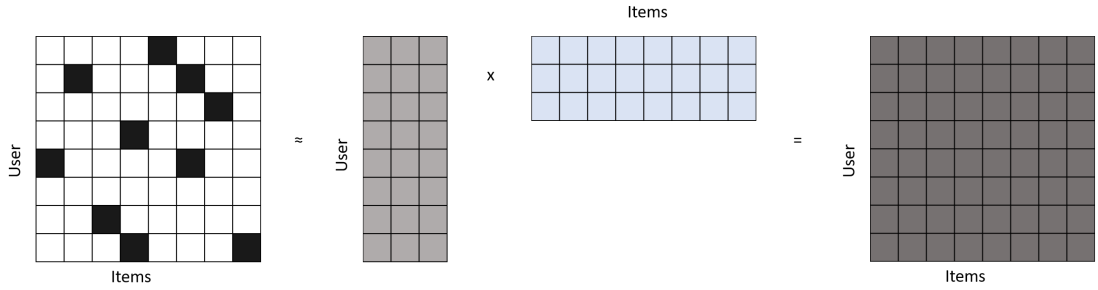



Figure 5.8: Representation of matrix factorization

The scalar product of the user and item matrix approximates the original user-item-matrix, but also fills all empty cells. These scores can be used to build a new, or transform an existing product ranking.

#### Data preparation

In the first step, the user-item matrix has to be defined. To generate the recommendations for a specific prediction date, the data from the previous year until the day prior to the prediction date is considered. This prototype only relies on the transaction data (customer relevant data) in contrast to the cluster-based prototype, which also considered tracking data from all visitors.

The transaction data includes all purchases including the customer and product identifier, order date and information about the number of ordered units as well as their value. For the input matrix the data is aggregated per user and product. Users, who purchased one item only, are excluded because of insufficient data depth. Furthermore, items with less than five purchases are excluded in order to concentrate on more relevant products.

For the input data two approaches are examined: binary and continuous values. In general, only positive samples are known, when a customer purchased an item. For the other user-item pairs it is not known, if the user is not interested or if the user did not purchase the item yet. Nevertheless, during the development process it became apparent that the model also requires negative examples to learn effectively. For this reason, for each user $k_u$ random items are assigned as

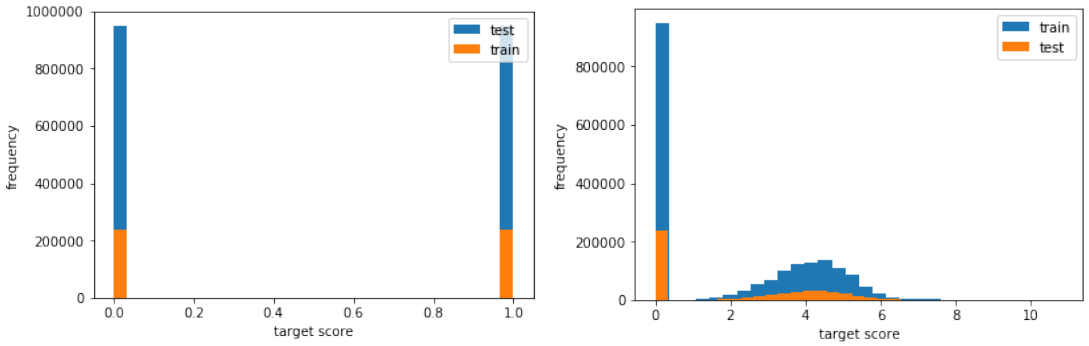
negative sample with the value zero, where $k_u$ is the number of items purchased per user. Therefore, the binary input matrix $R_{ui}$ is defined as:

$$R_{ui} = \begin{cases} 1, & \text{if the user u purchased item i,} \\ 0, & \text{for } k_u \text{ random items, that have not been purchased by user u,} \\ \emptyset, & \text{in all other cases.} \end{cases} \tag{5.1}$$

The resulting matrix has the same number of positive and negative samples, but is still very sparse with over 99% of empty cells. The input matrix with continuous values is defined in a similar way, as presented in equation 5.2.2. The aggregated order value per item and user is quite diverse and therefore logarithmized. The resulting target values are between 0 and 10 and close to a normal distribution.

$$R_{ui} = \begin{cases} log(v), & \text{if the user u purchased item i with order value v,} \\ 0, & \text{for } k_u \text{ random items, that have not been purchased by user u,} \\ \emptyset, & \text{in all other cases.} \end{cases} \tag{5.2}$$

For the actual input of the model, only the positive and randomized negative samples are used. To train and validate the matrix factorization, the input data is furthermore split into 80 percent of training and 20 percent of test data. Both input data sets are presented in figure 5.9




(a) Binary target

(b) Continuous logarithmic target

Figure 5.9: Input data for matrix factorization

### Modeling

The implementation of the matrix factorization is build with PyTorch (Torch Contributors, 2017), which allows a relatively easy and flexible definition of the model. Furthermore, PyTorch supports the model training using the GPU. This prototype follows the implementation of the Github repository Spotlight by Kula, 2017.

The overall process to fit the matrix factorization model is presented in pseudo code as follows:

**Algorithm 1:** Execution structure to fit matrix factorization model

```
Data: User-Item pairs with target score
Result: Fitted model containing user and item factors
Initialize model with number of factors;
Select loss function and optimizer;
Split data in test and training data set;
Create batches of the training data with a previously defined size;
foreach epoch do
    foreach batch in training batches do
        Predict scores for batch with model;
        Estimate loss between prediction and actual scores;
        Backpropagation of loss using optimizer;
    end
end
Evaluate model;
```

In the first step the latent representations for the factorization model is defined as a neural network module in PyTorch. Both users and items are encoded as an embedding layer. In the initialization, the number of factors valid for both user and item embedding has to be defined. With an increasing amount of factors, the model becomes more accurate, but also more complex. Furthermore, user, item and global biases are defined. The predictive score for a user-item pair is given by the scalar product of the item and user latent vectors summed with the mentioned biases. This calculation is the forward function of the defined module. If the binary target is used, the described formula for the score is furthermore embedded in a sigmoid function.

Secondly the optimizer and loss function have to be defined. Common optimization functions are stochastic gradient descent (SGD) or the Adam optimizer (see section 3.6.2), which are both directly provided as PyTorch optimizer classes. In a grid search Adam has proven to result in a better model.

For the definition of the loss function, two approaches are applied: a pointwise and a pairwise learning to rank method (see section 3.1). For pointwise losses, PyTorch provides multiple loss functions, like the mean squared error for continuous targets (torch.nn.MSEloss) or the binary cross entropy for binary targets (torch.nn.BCEloss). For the pairwise loss, the bayesian product ranking (BPR) implementation of the Spotlight project (Kula, 2017) is used.

The training is done batch-wise with a training and test split of 80:20. The current factors are used to estimate the prediction for the current batch of user and item pairs. The loss function determines the gap between the prediction and the actual score. Using the loss, the optimizer performs the backward step and thus improves the latent factors in the representation. This is done for all batches and iteratively for multiple epochs. The main parameters for the training are: number of factors, batch size, learning rate and weight decay for generalization. A grid search is used to optimize the parameters and shows that the models performance is highly dependent on generalization and learning rate. The evaluation of the models performance on the unseen training data set is a good indicator for the generalization ability, exemplary shown in figure 5.10.

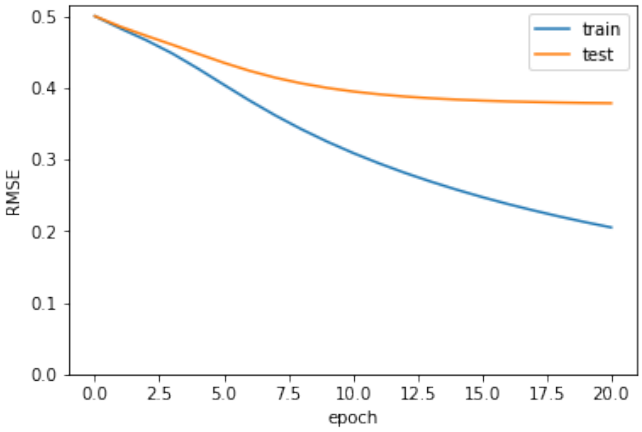



Figure 5.10: Root mean squared error trend

Nevertheless, the performance of the model based on pointwise or pairwise evaluation measures is not sufficient for the aim to create personalized product rankings and compare it with the baseline and cluster-based product rankings. The transformation in product rankings and the evaluation of these are described in the following sections.

### 5.2.3 Implementation of product ranking

The ranking score is a value between 0 and 1, where a higher score represents a higher relevance. The scores from the matrix factorization with pointwise approaches and binary targets, are also ranging from 0 to 1. If a pairwise approach is applied, the matrix factorization is only optimized on the order of the items. Therefore, the scale can be very different and has to be adjusted. A scaling is also necessary for the scores which are based on the continuous logarithmized target.

The final personalized product ranking is then based on a combination of the baseline product scores $s_b$ with the individual scores $s_i$. To achieve the best performance, different versions are tested, e.g.

$$s = (s_b + s_i)/2 \tag{5.3}$$

$$s = (s_b + s_b * s_i)/2 \tag{5.4}$$

$$s = (s_b + s_b * \sqrt{s_i})/2 \tag{5.5}$$

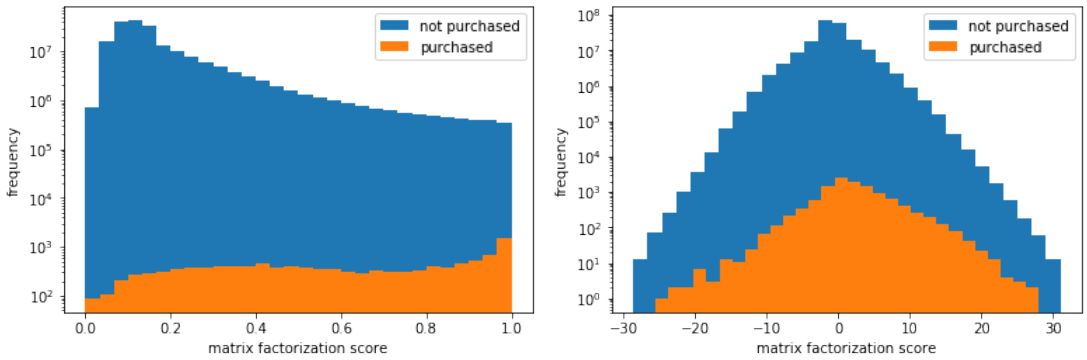
Furthermore, an ensemble of the baseline scores with a pairwise and pointwise matrix factorization was implemented:

$$s = (s_b + s_b * s_{pointwise} + s_b * s_{pairwise})/3 \tag{5.6}$$

In comparison, the equation 5.5 and 5.6 performed better than other formulas and are therefore considered in the detailed evaluation. The web shop is only provided with the numeric relevance scores, which are then used to generate the ranking by ordering the products by their score in descending order for each product category.

### 5.2.4 Evaluation

To compare the individualized rankings with the cluster-based approach, the same user sample is used as described in section 5.1.4. The matrix factorization is trained for each sixth day in the period from August to October 2017. In a productive environment the model should be trained on a daily basis, but for simplicity and comparability with the cluster-based prototype, the model training was reduced. Every sixth day is chosen to avoid a dependency on the weekday. However, the prediction is done for every day by using the latest trained model to estimate the recommendation scores of all products for each user-date pair in the sampled data set. The resulting scores are evaluated against the actual purchases of the customers.




(a) Binary & pointwise

(b) Continuous & pairwise

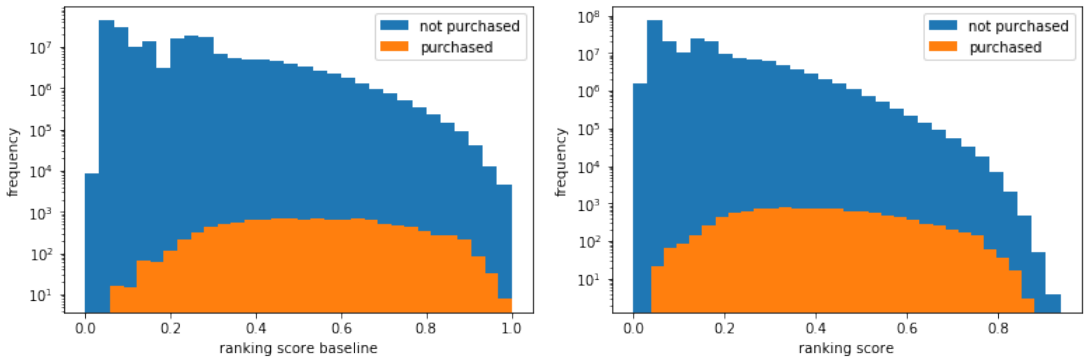
Figure 5.11: Histograms of matrix factorization scores

The figure 5.11 shows two histograms, where the purchased products are displayed in orange and not purchased products in blue. The y-axis contains the number of products and is logarithmized. Every product receives a recommendation score per customer-date pair, and therefore, the number of not purchased products is much higher. The graph 5.11 (a), referring to the binary target and pointwise approach, shows that on the prediction date not purchased products tend to get a score that goes towards zero. The proportion of actually purchased products increases with an increasing score. The recommendation can thus predict the relevance of an item. Graph (b) is referring to the continuous target and pairwise approach and also shows slightly higher recommendation scores for purchased products in comparison to not purchased products. Due to the pairwise method, these scores cannot be evaluated directly, but should be regarded in relation to the scores for each customer.

The two presented scores are estimated for each user, item and date and then combined with the baseline prediction score of the corresponding prediction date using the equations 5.5 and 5.6. The baseline already provides a very good basis, for example by assigning a low score to currently unpopular or unavailable products. Products that are generally interesting for all customers receive a high score.

The histograms in figure 5.12 show the distribution of the baseline scores (a) and final scores for the personalized ranking (b) of all customers in the evaluation sample. The number of products that was not purchased is considerably smaller for higher ranking scores in comparison to the baseline. The final score (see equation 5.6) does not include values up to 1, as no product received in all three combined scores a value of 1.

The rankings are also analyzed on the previously used evaluation metrics, summarized in table 5.6. The baseline metrics are unchanged with an average product




(a) Baseline scores (b) Final scores

Figure 5.12: Histograms of baseline and personalized ranking scores

position at 27.6 for known customers. The personalized rankings using only the pointwise matrix factorization score improve the metric about 1 position to an average product position of purchased items at 26.59. When also using the pairwise matrix factorization scores, the result slightly improves to the average position of 26.47 in the product rankings.

| **Ranking** | Baseline | with pointwise | with point- & pairwise |
|---|---|---|---|
| **Customer nDCG metric** | 0.245 | 0.253 | 0.253 |
| **Avg. product position** | 27.60 | 26.59 | 26.47 |

Table 5.6: Ranking evaluation at customer level

Therefore, the application of a matrix factorization as enhancement in the product ranking is improving the product ranking. The relevant products have been moved up in average by 1.11 position for known users. As already described in the evaluation of the cluster-based personalization, the metrics are estimating the effect on historic data, where the customers might already be biased by the actually presented rankings on that day.

### 5.2.5 Discussion

The presented model and transformations are only one opportunity to personalize product rankings on an individualized basis. A variety of other options to include the assumed preferences of a customer are presented in section 2.3 about state of the art implementations of product rankings.

For each user the past purchases are aggregated and used to create individual recommendation scores for all products. A matrix factorization is applied as modeling technique using both pointwise and pairwise approaches. The resulting scores are then used to personalize the existing baseline ranking.

The evaluation with the sampled data set verifies that the usage of personalized rankings improves the evaluation metrics and presents more relevant products in the upper positions. The individualized ranking is closer to the customer's future purchasing behaviour and therefore, this prototype also complies with the first hypothesis. A subjective evaluation of the individualized rankings is not pursued for this prototype. The experts would have to evaluate a large number of different rankings to rate their quality in respect of the past interests of each customer. In this case, an A/B test would be necessary for conclusive results.

## 5.3 Comparison and discussion

In comparison to the cluster-based implementation, the individualized prototype did show considerably less improvement in the evaluation metrics. The implementation of the individualized prototype is, at this time, less sophisticated and uses only transaction data, whereas the cluster-based personalization also includes tracking data. Therefore, the approach would need to be enhanced in order to make an appropriate comparison based on the evaluated metrics.

Nevertheless, both prototypes show an improvement in comparison to the non-personalized baseline. Each approach has advantages and disadvantages, which are summarized in table 5.7. Although the cluster-based approach was more promising in this study, it might neglect marginal groups of users. The clustering will be trained infrequently as a consistency is necessary to provide accurate predictions without recalculating all training features. Nevertheless, the training will be done regularly, e.g. on a monthly basis. Hence, new clusters are not instantly considered, even though they might be reasonable. The advantage of an individualized approach is that it considers each user individually and can also be trained to detect associations between purchased products, eventually in sequential relationships.

Looking at the data preparation, the individualized prototype is less complex in the current state. The clustering of the first approach requires a data preprocessing of purchased and viewed products for each user according to defined characteristics, e.g. product category or price. In contrast, the input data of the matrix factorization does not require product related information, as it learns latent factors from the user-item-matrix itself. Furthermore, the data processing for the ranking prediction is much more expensive for the cluster-based prototype. The features relying on transaction data do not only have to be provided globally,

but for each cluster individually. The number of rows for training and prediction increases by a factor of $N$, where $N$ is the number of clusters. In contrast, the individualized rankings builds on the baseline ranking without modification and is therefore less expensive in terms of data preparation. The same can be concluded for the modeling of the predictive task in both approaches. The cluster-based approach is more expensive by creating not only one global ranking, but a ranking for each cluster.

However, the clustering task is less complex in comparison to the matrix factorization. The clustering has to be trained less frequent with the whole base of users. The cluster should stay constant for a defined period of time, but can still be improved by an incremental training with new incoming customers using the proposed model. For a daily prediction, the model is stored and applied to classify new or returned users in the existing clusters. In contrast, the matrix factorization has to be trained more often. New users can not be populated into the existing factors. Therefore, the matrix factorization is more complex in terms of modeling and prediction.

By using behavioural data for personalization, the user must be informed about its purpose and consent to data tracking to comply with the EU GDPR. This limits both implementations to be applied only on a subset of users. However by using only situative data, the model can be trained with users, that gave their consent to data tracking, whereas the personalization can still be applied to all users.

Both implementations consider the integration ability into web shops, but require an additional web service to provide the user with his personalized rankings. In the first approach, the rankings for each cluster can be imported by the web shop on a daily basis. The web service responds with the cluster id given a user identifier. In the second approach, the shop stores only the item factors, whereas the web service responds with corresponding user factors. This requires a computation of the actual ranking in the shop back end. Therefore, the individualized prototype is more complex in terms of shop integration. Other possibilities, e.g. transferring a complete ranking for a requested user, would be even more difficult.

The maintainability of the cluster-based prototype is also better in comparison to the individualized as the rather small number of resulting rankings can be validated more easily.

If applying the findings of this work in a productive environment, the author would recommend to start with situative clustering. This personalization provides a considerable improvement in comparison to the daily provided baseline ranking. It can then be enhanced with a behaviour-based personalization for known users, which should be either cluster-based or individualized. More details about future work and research are presented in chapter 7.

| | **Cluster-based personalization** | **Individualization** |
|---|---|---|
| **Performance** | [+] in this use case more promising<br>[–] marginal groups are not targeted<br>[–] new clusters are not instantly considered | [–] effect was smaller in historic evaluation<br>[+] each individual can be targeted<br>[+] can be enhanced with sequence based recommendations |
| **Data prep.** | [±] similar efforts for user definition<br>[–] clustering needs separate feature engineering<br>[–] cluster-based features for prediction are needed | [±] similar efforts for user definition<br>[+] input data for recommendation is simpler<br>[+] input data for prediction is unchanged |
| **Modeling** | [+] clustering less expensive<br>[–] prediction more complex (more data) | [–] matrix factorization more expensive<br>[+] baseline prediction less complex |
| **Privacy** | [±] user must consent to data usage for personalization with behavioural data<br>[+] personalization using situative data is not limited by privacy regulations | [±] user must consent to data usage for individualization with behavioural data |
| **Integration** | [+] in general possible<br>[±] web service is necessary for id transmission<br>[+] $N$ lists can be stored in sort index of shop | [+] in general possible<br>[±] web service is necessary for user factors<br>[–] rankings have to be calculated given item& user factors |
| **Maintenance** | [+] validation of results is easier with finite number of created lists | [–] more complex with individual ranking for each user |

Table 5.7: Advantages and disadvantages of personalization approaches

# Chapter 6

# Conclusions

Personalization is an emerging topic in e-commerce. Studies have conducted, that users are not only used to it, but they are already expecting to get offers fitting their interests and needs. To provide an automatic personalization, machine learning techniques can be applied. The personalization process includes the collection of data and building a user profile to finally apply the personalization to an application.

Whereas common personalization applications in e-commerce are product recommendations and online marketing campaigns, the whole product ranking can also be personalized. Product rankings, in this context, refer to the order in which products are displayed in web shops.

The concept of ranking items in the perspective of machine learning is called learning to rank differentiating between three approaches: pointwise, pairwise and listwise. In pointwise approaches the relevance of an item is learned directly. In pairwise approaches, pairs of items with a known order are used to train the overall ranking. Listwise approaches are similar to the pointwise approach, but the model is optimized by estimating the model's error on the whole ranking.

In preparation for the development of personalized product rankings, different levels of personalization have been identified. The first level has the aim to provide an explicit personalization, where the user can directly state their preferences or characteristics. The second level considers each user as part of a defined group. Implicit data is used to cluster users with similar interests and provide them with the same personalization. The third level considers the personal interests of each user and thus, creates individual product rankings. The personalization approaches can be enhanced with a real-time ability, depending on the available data sources and system properties.

Based on the findings, two prototypes were conceptualized and developed in this piece of work: The first personalization approach is based on a customer segmentation and creates a product ranking for each identified cluster. The second approach relies on a recommendation engine, that is combined with the existing ranking, resulting in individual rankings.

Both implementations rely on a baseline product ranking, which predicts the sales for each product in the upcoming eight days. The predicted scores represent the

relevance of each item and build in descending order the product ranking for the web shop. The desired improvements in the rankings were evaluated using the common ranking metric normalized discounted cumulative gain (nDCG) and the average position of purchased items in the ranking.

The cluster-based prototype requires two machine learning tasks: segmentation and prediction. The segmentation uses both transaction and tracking data, involving the affinity to product categories, newness and price of the products. Additionally, a simple clustering was developed by defining time periods during the day and assigning the customers to specific segments based on the time of their purchase.

All clustering versions were then used to calculate personalized rankings. A prediction of the sales value for each product and cluster was generated and sorted in descending order to create the ranking. By integrating the nDCG measure to optimize the model, it is a listwise learning to rank method. The prediction was created using features based on previous transaction data in the desired cluster as well as in the overall customer set. Furthermore, product and tracking data is included.

The second prototype combines the unchanged baseline prediction with individual factors, which were retrieved by a matrix factorization. The input data are the purchased items per customer with the same number of randomly selected negative samples. A pointwise and pairwise learning method was applied and the resulting scores were then combined with the original score.

The non-personalized baseline ranking has an nDCG of 0.245 and the purchased products are on average at position 27.6, taking into account the known customers in a random sample with 10000 observations in the period of August to October 2017. The best performing cluster-based personalization improved the metrics to an nDCG of 0.365 and an average position of 20.01. The nDCG is higher, if the ranking is closer to the perfect ranking for each user. The perfect ranking consists of all products, where the purchased products are in the first positions ordered by their sales value. The position of the purchased products improves by over seven positions. The cluster-based ranking therefore better reflects the interests of the customers. Additionally, the customers have been clustered by the time of their purchase, as a simple example of situative clustering. This time-based clustering was able to improve the metrics to an nDCG of 0.285 and an average position of 22.36. The main advantages of situative clustering are on the one hand a considerably easier implementation and integration into the web shop, and on the other hand the possibility to provide all users with an improved ranking and not only known users. Furthermore, situative clustering is not as limited by the GDPR in comparison to personalization that is based on the user's behaviour (see section 2.4.2).

The individualized rankings also led to an improvement in comparison to the baseline version, resulting in an nDCG of 0.253 and an average product position of 26.47. The rather small improvement of the nDCG can be explained by the learning approach. With regard to the binary target, the model learned, which products are likely to be purchased, but did not take the products' value into account. However, the nDCG considers each product's value as relevance in the ranking. In contrast to the cluster-based personalization, this prototype only considers transaction data. At the current stage of development, the individualized prototype is not able to keep pace with the cluster-based prototype. However, cluster-based personalization approaches are not generally better than individualized methods. More research would have had to be invested in the development of the individualized prototype to provide an appropriate performance comparison.

Nevertheless, both personalization methods achieve measurable improvements in the product rankings. The future interests of the customers are reflected in a better way. The implementation and evaluation of the prototypes contribute to the the first hypothesis of this thesis: Using machine learning procedures, product rankings can be improved to fit the customers' future purchasing behaviour.

The second hypothesis considers the subjective impression of the product rankings, which was examined by experts of the web shop. This evaluation was only conducted for the best performing cluster-based ranking. The experts were provided with a selection of product lists associated with a customer cluster, which are described by their main characteristics. The experts were asked to assess the difference between the different product rankings and their fit to the described clusters. Overall, the experts were able to recognize differences in the rankings that fit the characteristics of the customer clusters. The main limitation is the lack of recognizability of price affinity in most of the presented rankings. This can be attributed to the target definition, which is based on the products order value and therefore benefits more expensive products. In summary, the rankings were perceived as good reflections of the customer clusters interests. The second hypothesis is thus supported for the cluster-based personalization implemented in this thesis.

# Chapter 7

# Future Work and Research

In terms of applied machine learning techniques, there are many other options, that can be used to provide personalized product rankings. To learn the ranking relevance, which is the sales value per product, gradient boosted trees were used. This was adapted from the baseline implementation of the product ranking and is in combination with the optimization using the nDCG measure a listwise learning to rank approach. In research, there is a variety of algorithms to learn rankings, which could also be applied and evaluated. An overview is given in section 3.1. However, the aim of this piece of work was to enhance and improve the existing implementation with personalization approaches.

Considering the cluster-based prototype, the clustering can be tested both with other features and different clustering algorithms. A selection of possible features for personalization is presented in figure 2.5. The situative clustering can be enhanced with static web site information like the browser or traffic source. In the behaviour based clustering the affinity to specific brands might be interesting. Alternative clustering algorithms can be found in section 3.5. In regard to the actual ranking prediction, an enhanced target variable and further features are already intended, e.g. price trends. These changes might be helpful to achieve larger differences for clusters with high or low price affinity.

The individualized personalization offers many possibilities for testing other recommendation procedures. Building on the existing prototype, the input data can be enhanced with tracking data. Furthermore, if real-time data will be available in the future, it can also be enhanced in terms of real-time ability.

Candidates for further evaluations and realization as productive system are the time-based clustering as a preliminary stage and the cluster-based personalized rankings. It is planned to evaluate any personalization of the product rankings using an A/B test. A proposal for the structure of an A/B test is presented in figure 7.1.

In general, all web shop visitors are assigned to either the group with personalization or the control group with a split of 50:50. During the whole testing period, a visitor stays in the same group. If the user is assigned to the test group with personalization, it must be determined whether the user is already known and enough information is available. If the user is known and gave his consent to personalization, a behaviour based personalization can be displayed. In the

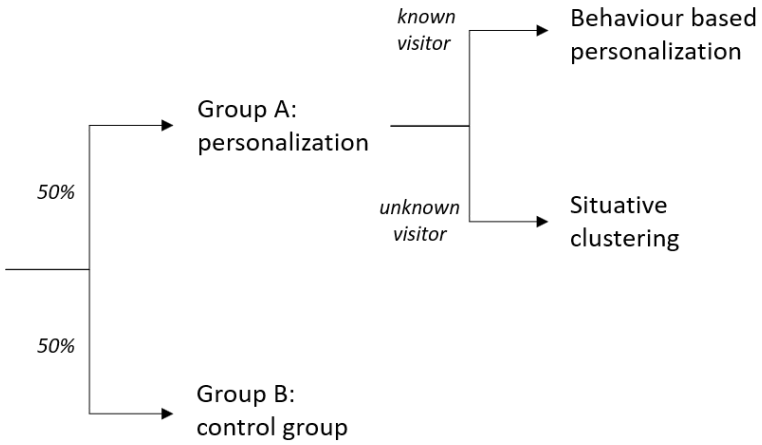



Figure 7.1: A/B test proposal

other case, a situative personalization should be applied. The A/B test allows the determination of the actual effects of personalized product rankings.

# Acronyms

| | |
|---|---|
| BPR | bayesian product ranking. |
| CART | classification and regression tree. |
| DCG | discounted cumulative gain. |
| GDPR | general data protection regulation. |
| GPS | global positioning system. |
| GPU | graphics processing unit. |
| HTML | hypertext markup language. |
| IR | information retrieval. |
| nDCG | normalized discounted cumulative gain. |
| SGD | stochastic gradient descent. |
| SVD | singular value decomposition. |
| XGBoost | extreme gradient boosting. |

# Glossary

| | |
|---|---|
| A/B test | In web analytics, A/B testing is a randomized experiment with two variants, A and B. It includes application of statistical hypothesis testing to compare two versions of a single variable, typically by testing a subject's response to variant A against variant B, and determining which of the two variants is more effective. |
| Hadoop | Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. |
| Hive | The Apache Hive data warehouse software facilitates reading, writing, and managing large data sets residing in distributed storage using SQL. |

| | |
|---|---|
| pseudonymization | Pseudonymization is a procedure by which personally identifiable information fields within a data record are replaced by one or more artificial identifiers, or pseudonyms. A single pseudonym for each replaced field or collection of replaced fields makes the data record less identifiable while remaining suitable for data analysis and data processing. Pseudonymization can be one way to comply with the European Union's new General Data Protection Regulation demands for secure data storage of personal information. Pseudonymized data can be restored to its original state with the addition of information which then allows individuals to be re-identified, while anonymized data can never be restored to its original state. |
| Pyspark | PySpark is the Python API for Spark. Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Scala, Java, and Python that make parallel jobs easy to write, and an optimized engine that supports general computation graphs. |
| tracking data | Tracking data in this piece of work refers to website visitor tracking (WVT), which is an aspect of Web analytics and deals with the analysis of visitor behaviour on a website. Analysis of an individual visitor's behaviour may be used to provide that visitor with options or content that relates to their implied preferences; either during a visit or in the future. |
| transaction data | Transaction data is data describing an event (the change as a result of a transaction) and contains always a time dimension. In e-commerce typical transactions are orders, invoices, payments and return shipments. |

| | |
|---|---|
| web cookie | A web cookie is a small piece of data that a server sends to the user's web browser. Cookies are mainly used for three purposes: session management, personalization and web tracking. |

# Appendix

## Description of clustering results

The analyzed shop divides between seven main categories, encoded by the numbers 1 to 7. Furthermore, each main category splits up in several sub-categories, described by letters (A-Z) in the following tables. These categories, as well as features about price sensitivity and the affinity to the newness of products are used for the clustering.

The resulting clusters for three selected clustering versions are presented on the next page. The description of each cluster refers to the main focus of this group of customers.

## Customer behaviour-based clustering - Version 1

| ID | user | proportion | cat 1 | cat 2 | cat 3 | cat 4 | cat 5 | cat 6 | cat 7 | price_low | new_prod | Description of cluster focus | Age | Female | Male |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 251764 | 11,5% | 1,3% | 5,8% | 80,2% | 5,9% | 2,9% | 3,8% | 0,1% | 65,2% | 0,7% | category 3 - subcat. K, E | 45,1 | 12,6% | 87,4% |
| 1 | 195859 | 8,9% | 2,9% | 2,5% | 8,4% | 0,2% | 0,2% | 85,8% | 0,0% | 32,4% | 0,3% | category 6 - subcat. W expensive | 45,8 | 1,3% | 98,7% |
| 2 | 6670 | 0,3% | 0,1% | 0,9% | 1,9% | 96,0% | 0,7% | 0,3% | 0,1% | 64,7% | 0,8% | category 4 - subcat. I | 56,6 | 81,0% | 19,0% |
| 3 | 116038 | 5,3% | 79,1% | 4,7% | 5,2% | 0,6% | 1,2% | 9,1% | 0,0% | 55,0% | 0,5% | category 1 - mix | 45,1 | 5,5% | 94,5% |
| 4 | 187786 | 8,6% | 1,6% | 3,6% | 83,2% | 1,0% | 0,7% | 9,9% | 0,0% | 65,0% | 0,6% | category 3 - subcat. W, A, O | 41,8 | 2,5% | 97,5% |
| 5 | 145048 | 6,6% | 0,1% | 0,9% | 1,9% | 96,5% | 0,5% | 0,1% | 0,1% | 79,0% | 0,4% | category 4 - subcat. F cheaper | 53,2 | 88,8% | 11,2% |
| 6 | 141799 | 6,5% | 0,2% | 1,6% | 5,8% | 90,4% | 1,4% | 0,3% | 0,2% | 34,2% | 0,6% | category 4 - subcat. F expensive | 51,6 | 84,9% | 15,1% |
| 7 | 59526 | 2,7% | 0,7% | 1,2% | 93,2% | 0,4% | 1,1% | 3,4% | 0,0% | 59,8% | 1,0% | category 3 - subcat. O | 42,2 | 10,8% | 89,2% |
| 8 | 98108 | 4,5% | 0,5% | 0,9% | 92,6% | 0,3% | 0,2% | 5,5% | 0,0% | 61,3% | 0,6% | category 3 - subcat. A | 41,6 | 2,2% | 97,8% |
| 9 | 86984 | 4,0% | 0,5% | 1,8% | 90,9% | 3,2% | 1,9% | 1,6% | 0,0% | 29,4% | 0,4% | category 3 - subcat. K, expensive | 45,0 | 12,9% | 87,1% |
| 10 | 61764 | 2,8% | 3,2% | 80,5% | 8,6% | 2,7% | 1,7% | 3,0% | 0,3% | 64,9% | 0,9% | category 2 - subcat. W, E | 45,8 | 17,0% | 83,0% |
| 11 | 125372 | 5,7% | 4,6% | 5,3% | 38,5% | 0,9% | 1,5% | 49,1% | 0,1% | 59,1% | 0,5% | category 3 & 6 - subcat. W | 44,2 | 0,8% | 99,2% |
| 12 | 162722 | 7,4% | 2,8% | 5,3% | 49,9% | 2,5% | 6,1% | 33,1% | 0,2% | 67,1% | 0,6% | mix of categories 3-A, 6-A, 7 | 43,6 | 3,8% | 96,2% |
| 13 | 100678 | 4,6% | 0,4% | 0,7% | 95,9% | 0,1% | 0,1% | 2,8% | 0,0% | 71,1% | 0,4% | category 3 -subcat. W | 42,6 | 5,8% | 94,2% |
| 14 | 66767 | 3,0% | 0,7% | 1,5% | 7,3% | 88,5% | 1,2% | 0,7% | 0,1% | 49,4% | 0,7% | category 4 - subcat. M | 52,3 | 16,9% | 83,1% |
| 15 | 60798 | 2,8% | 89,5% | 1,2% | 4,0% | 0,1% | 0,1% | 5,0% | 0,0% | 53,1% | 0,3% | category 1 - subcat. C | 49,3 | 6,9% | 93,1% |
| 16 | 228751 | 10,4% | 2,2% | 2,0% | 13,6% | 0,3% | 0,2% | 81,6% | 0,0% | 64,0% | 0,3% | category 6 -subcat. A | 45,5 | 1,5% | 98,5% |
| 17 | 93155 | 4,3% | 2,1% | 1,6% | 5,2% | 0,1% | 0,1% | 91,0% | 0,0% | 80,6% | 0,2% | category 6 -subcat. W cheaper | 48,1 | 2,2% | 97,8% |

## Customer behaviour-based clustering - Version 2

| ID | user | proportion | cat 1 | cat 2 | cat 3 | cat 4 | cat 5 | cat 6 | cat 7 | price_low | new_prod | Description of cluster focus | Age | Female | Male |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16818 | 0,8% | 2,4% | 1,4% | 6,0% | 0,3% | 0,2% | 89,7% | 0,0% | 54,2% | 0,014629 | category 6 - subcat. E | 46,7 | 12,6% | 87,4% |
| 1 | 2333 | 0,1% | 0,8% | 2,7% | 87,1% | 0,8% | 4,8% | 2,5% | 1,3% | 65,0% | 0,802447 | category 3 - new products | 43,6 | 1,3% | 98,7% |
| 2 | 249366 | 11,5% | 4,1% | 4,4% | 47,2% | 2,1% | 4,9% | 37,1% | 0,2% | 65,0% | 0,011787 | category 3 & 6 - mix | 43,7 | 81,0% | 19,0% |
| 3 | 199112 | 9,2% | 3,2% | 2,2% | 9,5% | 0,2% | 0,2% | 84,7% | 0,0% | 36,7% | 0,007738 | category 6 - subcat. W expensive | 46,7 | 5,5% | 94,5% |
| 4 | 92376 | 4,3% | 0,0% | 0,4% | 0,8% | 98,5% | 0,2% | 0,0% | 0,0% | 86,1% | 0,01061 | category 4 - subcat. F cheaper | 53 | 2,5% | 97,5% |
| 5 | 220 | 0,0% | 4,1% | 8,9% | 69,2% | 0,7% | 8,1% | 1,0% | 7,9% | 67,0% | 0,996883 | mix of new products | 44 | 88,8% | 11,2% |
| 6 | 322808 | 14,9% | 1,2% | 2,5% | 87,5% | 0,7% | 0,5% | 7,6% | 0,0% | 73,1% | 0,010512 | category 3 - subcat. W cheaper | 42,7 | 84,9% | 15,1% |
| 7 | 147653 | 6,8% | 0,7% | 2,7% | 88,7% | 4,1% | 1,9% | 1,8% | 0,0% | 56,3% | 0,009499 | category 3 - subcat. K | 45,2 | 10,8% | 89,2% |
| 8 | 160298 | 7,4% | 86,0% | 2,7% | 4,1% | 0,3% | 0,4% | 6,5% | 0,0% | 54,2% | 0,011325 | category 1 | 45,8 | 2,2% | 97,8% |
| 9 | 111913 | 5,2% | 0,3% | 2,5% | 8,5% | 86,1% | 1,9% | 0,5% | 0,3% | 61,0% | 0,028266 | category 4 - subcat. F cheaper | 52,7 | 12,9% | 87,1% |
| 10 | 330997 | 15,3% | 2,9% | 2,4% | 12,3% | 0,3% | 0,2% | 81,9% | 0,0% | 66,5% | 0,006694 | category 6 - subcat. W, A | 46,4 | 17,0% | 83,0% |
| 11 | 71024 | 3,3% | 0,8% | 1,6% | 8,6% | 86,8% | 1,3% | 0,8% | 0,1% | 49,7% | 0,018202 | category 4 - subcat. M | 52,5 | 0,8% | 99,2% |
| 12 | 30475 | 1,4% | 7,1% | 60,5% | 15,3% | 3,6% | 3,9% | 9,1% | 0,4% | 69,5% | 0,020436 | category 2 - subcat. W, E | 45,5 | 3,8% | 96,2% |
| 13 | 599 | 0,0% | 7,5% | 6,0% | 6,9% | 0,9% | 7,3% | 67,5% | 3,9% | 44,9% | 0,862772 | category 3, 6 - mix | 44,9 | 5,8% | 94,2% |
| 14 | 1097 | 0,1% | 0,0% | 0,1% | 0,1% | 99,5% | 0,1% | 0,0% | 0,2% | 44,0% | 0,986841 | category 4 - F, new, expensive | 52 | 16,9% | 83,1% |
| 15 | 300898 | 13,9% | 1,0% | 3,8% | 85,0% | 3,2% | 1,9% | 5,0% | 0,1% | 49,6% | 0,014169 | category 3 - mix | 44,2 | 6,9% | 93,1% |
| 16 | 43230 | 2,0% | 3,2% | 82,9% | 8,2% | 2,2% | 1,2% | 2,2% | 0,3% | 62,7% | 0,022163 | category 2 - subcat. E | 45,6 | 1,5% | 98,5% |
| 17 | 81671 | 3,8% | 0,1% | 0,4% | 1,1% | 97,9% | 0,5% | 0,1% | 0,1% | 21,0% | 0,018021 | category 4 - subcat F expensive | 51,1 | 2,2% | 97,8% |

## Customer behaviour-based clustering - Version 3

| ID | user | proportion | cat 1 | cat 2 | cat 3 | cat 4 | cat 5 | cat 6 | cat 7 | price_low | new_prod | Description of cluster focus | Age | Female | Male |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 101182 | 4,7% | 1,2% | 4,1% | 83,4% | 4,0% | 2,4% | 4,8% | 0,1% | 78,4% | 1,2% | category 3 - subcat. K - cheaper | 44,3 | 13,3% | 86,7% |
| 1 | 158438 | 7,3% | 0,1% | 1,1% | 2,2% | 95,7% | 0,7% | 0,1% | 0,1% | 77,4% | 2,0% | category 4 - subcat. F - cheaper | 53,2 | 88,3% | 11,7% |
| 2 | 151002 | 7,0% | 4,4% | 3,8% | 19,2% | 0,4% | 0,4% | 71,8% | 0,0% | 53,9% | 0,9% | category 6 - subcat. W | 45,8 | 0,5% | 99,5% |
| 3 | 44062 | 2,0% | 89,9% | 2,6% | 2,3% | 0,3% | 0,1% | 4,8% | 0,0% | 49,8% | 1,5% | category 1 - subcat. G | 44,6 | 4,4% | 95,6% |
| 4 | 108909 | 5,0% | 2,3% | 2,6% | 42,5% | 0,9% | 0,8% | 50,8% | 0,0% | 59,0% | 1,1% | mix of categories 3-A, 6-A | 42,6 | 1,1% | 98,9% |
| 5 | 50643 | 2,3% | 5,8% | 48,7% | 21,9% | 2,6% | 6,9% | 13,5% | 0,7% | 65,2% | 2,3% | category 2 & mix | 44,4 | 11,1% | 88,9% |
| 6 | 103368 | 4,8% | 0,1% | 0,6% | 1,5% | 97,1% | 0,6% | 0,1% | 0,1% | 26,5% | 2,8% | category 4 - subcat. F - expensive | 51,6 | 89,0% | 11,0% |
| 7 | 86462 | 4,0% | 0,8% | 1,6% | 93,1% | 0,4% | 0,3% | 3,8% | 0,0% | 46,7% | 1,1% | category 3 - subcat. W, E | 41,6 | 4,0% | 96,0% |
| 8 | 58858 | 2,7% | 90,7% | 1,3% | 3,5% | 0,1% | 0,1% | 4,4% | 0,0% | 53,7% | 0,9% | category 1 - subcat. C | 49,2 | 9,0% | 91,0% |
| 9 | 104343 | 4,8% | 0,5% | 0,9% | 91,5% | 0,3% | 0,2% | 6,6% | 0,0% | 62,2% | 1,6% | category 3 - subcat. A | 41,6 | 2,0% | 98,0% |
| 10 | 62434 | 2,9% | 0,8% | 0,7% | 5,4% | 0,1% | 0,1% | 92,9% | 0,0% | 63,3% | 0,5% | category 6 - subcat. A | 46,3 | 2,0% | 98,0% |
| 11 | 55913 | 2,6% | 0,4% | 0,8% | 3,2% | 94,5% | 0,6% | 0,4% | 0,1% | 48,8% | 2,0% | category 4 - subcat. M | 52,6 | 18,7% | 81,3% |
| 12 | 129998 | 6,0% | 2,4% | 3,7% | 63,2% | 0,8% | 0,8% | 29,2% | 0,0% | 69,4% | 1,2% | category 3 & 6 | 44,2 | 1,0% | 99,0% |
| 13 | 55665 | 2,6% | 1,7% | 1,5% | 9,0% | 0,4% | 0,3% | 87,2% | 0,0% | 60,5% | 1,1% | category 6 - mix | 46,3 | 1,3% | 98,7% |
| 14 | 82977 | 3,8% | 63,9% | 4,4% | 8,6% | 0,6% | 8,1% | 14,2% | 0,1% | 57,4% | 1,2% | category 1 -mix | 45,6 | 5,9% | 94,1% |
| 15 | 106732 | 4,9% | 1,5% | 6,5% | 78,2% | 4,1% | 3,8% | 5,7% | 0,1% | 40,1% | 1,7% | category 3 - mix, expensive | 44,9 | 6,2% | 93,8% |
| 16 | 53607 | 2,5% | 0,6% | 0,9% | 93,7% | 0,3% | 0,9% | 3,5% | 0,0% | 59,0% | 2,4% | category 3 - subcat. O | 42,1 | 10,6% | 89,4% |
| 17 | 38923 | 1,8% | 2,6% | 86,8% | 5,9% | 2,2% | 1,0% | 1,2% | 0,2% | 65,1% | 2,3% | category 2 - subcat. E | 46,6 | 18,6% | 81,4% |
| 18 | 78032 | 3,6% | 0,4% | 1,6% | 92,6% | 2,6% | 1,6% | 1,3% | 0,0% | 30,9% | 0,9% | category 3 - subcat. K, expensive | 45,1 | 12,5% | 87,5% |
| 19 | 70819 | 3,3% | 1,1% | 0,8% | 3,0% | 0,1% | 0,1% | 95,0% | 0,0% | 14,1% | 0,8% | category 6 - subcat. W, expensive | 46,3 | 1,7% | 98,3% |
| 20 | 61276 | 2,8% | 1,6% | 5,2% | 31,3% | 53,7% | 5,0% | 2,8% | 0,4% | 57,6% | 1,9% | mix of subcategory K | 50 | 51,2% | 48,8% |
| 21 | 127129 | 5,9% | 2,7% | 2,7% | 15,0% | 0,3% | 0,3% | 79,1% | 0,0% | 60,2% | 0,8% | category 6 - mix | 45,3 | 1,6% | 98,4% |
| 22 | 68981 | 3,2% | 1,2% | 0,9% | 2,9% | 0,1% | 0,0% | 94,8% | 0,0% | 84,6% | 0,6% | category 6 -subcat. W, cheaper | 48,2 | 2,7% | 97,3% |
| 23 | 69344 | 3,2% | 0,4% | 0,8% | 96,0% | 0,1% | 0,1% | 2,6% | 0,0% | 90,2% | 1,0% | category 3 -subcat. W, cheaper | 43,7 | 5,7% | 94,3% |
| 24 | 133791 | 6,2% | 1,3% | 4,2% | 82,4% | 1,2% | 1,7% | 9,1% | 0,0% | 74,0% | 1,7% | category 3 -mix, cheaper | 41,8 | 4,1% | 95,9% |

# Bibliography


Accenture (2018). Propel growth & value: Dynamically curating experiences to each individual and context in a seamless manner across channels. `https://www.accenture.com/us-en/service-propelling-growth-through-personalization`. Accessed on Mar 05, 2018.

Apptus Technologies AB (2018). Navigation - optimise your navigation. `www.apptus.com/product/navigation`. Accessed on Feb 09, 2018.

Beke, F. T., Eggers, F., and Verhoef, P. C. (2018). Consumer informational privacy: Current knowledge and research directions. *Foundations and Trends in Marketing*, 11(1):1–71.

Benlian, A. (2015). Web personalization cues and their differential effects on user assessments of website value. *Journal of Management Information Systems*, 32(1):225–260.

Blanco-Fernández, Y., José J. Pazos-Arias, A. G.-S., Ramos-Cabrer, M., and López-Nores, M. (2008). Personalization strategies and semantic reasoning: Working in tandem in advanced recommender systems. In Uchyigit, G. and Ma, M. Y., editors, *Personalization Techniques And Recommender Systems. Series in Machine Perception and Artificial Intelligenc*, volume 70, chapter 8, page 191ff. World Scientific, Vigo, Spain.

Blom, J. (2000). Personalization: A taxonomy. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '00, pages 313–314, New York, USA. ACM.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees (Wadsworth Statistics/Probability)*. Chapman and Hall/CRC.

Brusilovsky, P. and Maybury, M. T. (2002). From adaptive hypermedia to the adaptive web. *Commun. ACM*, 45(5):30–33.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.

Campanelli, G., Sardoni, C., and Kriesler, P. (1999). Part II: Economic theory and applied analysis: Chapter 26: A modified trend through peaks approach

to measuring potential output. *Keynes, Post-Keynesianism & Political Economy*, pages 469–483.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, USA. ACM.

Company (2017). Personalization & individualization (P&I) compass. Unpublished internal document (Originator is anonymized and can be provided to the reviewing committee).

Davies, A. (2015). Better customer service through content consumption insights. `www.salesforce.com/blog/2015/08/better-customer-service-through-content-consumption-insights.html`. Accessed on Mar 10, 2018.

DMLC (2015-2016). Xgboost: Python API Reference. `http://xgboost.readthedocs.io/en/latest/python/python_api.html`. Accessed on June 11, 2018.

Dou, Z., Song, R., Wen, J. R., and Yuan, X. (2009). Evaluating the effectiveness of personalized web search. *IEEE Transactions on Knowledge and Data Engineering*, 12:8:1178–1190.

Etzioni, O. (1996). The world-wide web: Quagmire or gold mine? *Commun. ACM*, 39(11):65–68.

European Union (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88.

Fan, H. and Poole, M. (2006). What is personalization? Perspectives on the design and implementation of personalization in information systems. In *Journal of Organizational Computing and Electronic Commerce - J ORGAN COMPUT ELECTRON COMME*, volume 16, pages 179–202. ACM.

Felfernig, A., Teppan, E., and Gula, B. (2008). User acceptance of knowledge-based recommenders. In Uchyigit, G. and Ma, M. Y., editors, *Personalization Techniques And Recommender Systems. Series in Machine Perception and Artificial Intelligence*, volume 70, chapter 10, page 249ff. World Scientific, Singapur.

Freno, A., Saveski, M., Jenatton, R., and Archambeau, C. (2015). One-pass ranking models for low-latency product recommendations. In *Proceedings of*

*the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1789–1798, New York, USA. ACM.

Gasparetti, F. and Micarelli, A. (2008). A deep evaluation of two cognitive user models for personalized search. In Uchyigit, G. and Ma, M. Y., editors, *Personalization Techniques And Recommender Systems. Series in Machine Perception and Artificial Intelligence*, volume 70, chapter 2, page 33ff. World Scientific, Singapur.

Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2002). User profiles for personalized information access. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web. Lecture Notes in Computer Science*, volume 4321, pages 54–89. Springer, Berlin Heidelberg.

Golub, G. and Kahan, W. (1965). Calculating the singular values and Pseudo-Inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224.

Graham-Harrison, E. and Cadwalladr, C. (2018). Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. *The Guardian*. `https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election`. Accessed on August 22, 2018.

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Hausl, S. (2017). Leveraging recommender systems to personalise search results. `www.pyvideo.org/pydata-london-2017/leveraging-recommender-systems-to-personalise-search-results.html`. Accessed on Feb 09, 2018.

Huke, S. (2011). Der unterschätzte Umsatzfaktor: Produktlisten im Web-Shop. `www.internethandel.de/blog/Der-unterschatzte-Umsatzfaktor-Produktlisten-im-Web-Shop/`. Accessed on Dec 11, 2017.

Jablonski, S., Petrov, I., Meiler, C., and Mayer, U. (2004). *Guide to Web Application and Platform Architectures*. Springer Science & Business Media, Berlin Heidelberg.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems - An Introduction*. Cambridge University Press, Cambridge.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Joachims, T., Freitag, D., and Mitchell, T. M. (1997). Webwatcher: A tour guide for the world wide web. In *Proceedings of the 15h International Conference on Artificial Intelligence (IJCAI1997)*, pages 770–777.

Kasanoff, B. (2001). *Making It Personal: How to Profit from Personalization Without Invading Privacy*. Perseus Publishing.

Kim, H. R. and Chan, P. K. (2003). Learning implicit user interest hierarchy for context in personalization. *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 101–108.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Kramer, J., Noronha, S., and Vergo, J. (2000). A user-centered design approach to personalization. *Commun. ACM*, 43(8):44–48.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer, New York, USA.

Kula, M. (2017). Spotlight. `https://github.com/maciejkula/spotlight`. Accessed on June 12, 2018.

Langville, A. N. and Meyer, C. D. (2012). *Who's #1? - The Science of Rating and Ranking*. Princeton University Press, Kassel.

Larose, D. T. (2005). *Discovering Knowledge in Data - An Introduction to Data Mining*. John Wiley & Sons, New York.

Liu, H., Wu, Z., and Zhang, X. (2018). Cplr: Collaborative pairwise learning to rank for personalized recommendation. *Knowledge-Based Systems*, 148:31–40.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Mobasher, B. (2007). Data mining for web personalization. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web. Lecture Notes in Computer Science*, volume 4321, pages 90–135. Springer, Berlin Heidelberg.

Mohan, A., Chen, Z., and Weinberger, K. (2010). Web-search ranking with initialized gradient boosted regression trees. In *Proceedings of the 2010 International Conference on Yahoo! Learning to Rank Challenge - Volume 14*, YLRC'10, pages 77–89. JMLR.org.

Moniz, N., Torgo, L., and Vinagre, J. (2016). Data-driven relevance judgments for ranking evaluation. *CoRR*.

Moukas, A. and Maes, P. (1998). Amalthaea: An evolving multi-agent information filtering and discovery system for the WWW. In *Autonomous Agents and Multi-Agent Systems*, volume 1(1), pages 59–88. Kluwer Academic Publishers.

Müller, A. C. and Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.

NetMarketShare (2017). Search engine market share. `www.netmarketshare.com/search-engine-market-share.aspx`. Accessed on Mar 08, 2018.

Odoscope GmbH (2018). The power of odoscope operational intelligence. `www.odoscope.com/en/solution/`. Accessed on Feb 09, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Personalization Consortium (2003). What is personalization. `www.personalization.org`. Accessed on Dec 11, 2017.

Progress Software Corporation (2016). Sitefinity: The ultimate guide to website personalization. `www.sitefinity.com/campaigns/tutorial-guides/ultimate-guide-to-website-personalization`. Accessed on June 15, 2018.

Ramaswamy, S. (2015). Outside voices: Why mobile advertising may be all about micro-targeting moments. *The Wallstreet Journal*. Accessed on Mar 10, 2018.

Reinsel, D., Gantz, J., and Rydning, J. (2017). Data age 2025. `http://www.seagate.com/files/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf`. Accessed on August 18, 2018.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States. AUAI Press.

Rich Relevance (2018). Accelarate product discovery with personalized browse and navigation. `www.richrelevance.com/solutions/personalized-browse-and-navigation/`. Accessed on Feb 09, 2018.

Riecken, D. (2000). Introduction: Personalized views of personalization. *Commun. ACM*, 43(8):26–28.

Salesforce Research (2017). State of the connected customer: Insights from 6,700+ consumers and business buyers on the intersection of experience, technology, and trust. `www.salesforce.com/form/conf/service-cloud/state-of-connected-customer`. Accessed on June 18, 2018.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. T. (2000). Application of dimensionality reduction in recommender system – a case study. In *ACM WEBKDD Workshop*.

Shuk Ying, H. and Bodoff, D. (2014). The effects of web personalization on user attitude and behavior: an integration of the elaboration likelihood model and consumer search theory. *MIS Quarterly*, 38(2):497–520.

Stücke, J., Lembcke, T., and About You Tech (2016). „Jedes Teil dein Style" - Personalisierte Beratung mit ArangoDB. `www.medium.com/about-developer-blog/4d6f5b6f5525`. Accessed on Feb 09, 2018.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning - An Introduction*. MIT Press, Cambridge, 1 edition.

Toch, E., Wang, Y., and Cranor, L. F. (2012). Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction*, 22(1):203–220.

Torch Contributors (2017). Pytorch documentation: torch.nn. `https://pytorch.org/docs/stable/nn.html`. Accessed on June 16, 2018.

Uchyigit, G. and Ma, M. Y. (2008). *Personalization Techniques And Recommender Systems. Series in Machine Perception and Artificial Intelligence (Vol. 70)*. World Scientific, Singapur.

White, R. W., Jose, J. M., and Ruthven, I. (2001). Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001*.

Zhang, M., Guo, X., and Chen, G. (2016). Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking. *Decision Support Systems*, 83:10–21.

Zoghby, J., Tieman, S., Cimino, X., and Lim, I., editors (2016). *Orchestrate, Organize, and Operationalize. Delivering on the Promise of Personalization Scale*. Accenture Interactive (Accenture Digital). Accessed on Mar 05, 2018.