Universiteit Leiden

ICT in Business

Recommender Systems in Practice:

A Business Value Driven Approach for Startups

Name: S.B. Mulders

Date: 18/09/2017

1st supervisor: M.C.A. de Rijk
2nd supervisor: Prof.dr.ir. W. Kraaij

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Recommender systems are a type of solution to overcome the information overload problem our society is experiencing. The value of recommender systems is proven by many of the leading online businesses such as Amazon, eBay and Spotify. The field of recommender systems is vastly studied resulting in thousands of different publications, articles and theories. Ironically, this amount is a form of information overload by itself.

Organizations face significant challenges in designing a recommender system that truly delivers business value. The main finding of this thesis is that specifically startups are affected by this because of their unique circumstances: extreme uncertainty, low time-to-market and unknown customer demands. Inspired by the leading online businesses, some startups have the idea to offer personalized recommendations too. However, when diving into the field of recommender systems, startups might get overwhelmed, do not know where to get started and face certain challenges.

This thesis outlines distinct capabilities of recommender systems and links these with challenges startups face. Ultimately, the goal of this research is to help startups in the process of designing, operating and maintaining a recommender system.

To achieve this goal, a framework is proposed that consists of 7 steps: from business analysis to the development and monitoring of a recommender system. Tools are provided to help startups make the right decisions in designing a recommender system that truly delivers business value. To assess the complexity, required effort and risk involved in certain recommendation ideas, a *dimension model* was developed. This model consists of eight dimensions: user model, item model, domain model, performance, quality of outcomes, architecture and infrastructure, interface & user experience and privacy & security.

Acknowledgements

First and foremost, I would like to express my gratitude for my supervisor, Tino de Rijk. Your passionate commitment, helpful discussions, and insightful guidance have been very useful. I am also thankful for my second supervisor, Prof. dr. ir. Wessel Kraaij. Your dedication in a busy period and your valuable comments are much obliged.

Furthermore, Tim de Kraker deserves a special mention. Your input and insights during the entire process have greatly benefitted this study. A word of gratitude is also reserved for my close friend and business partner Diederik van den Burger. Our brainstorm sessions and your proofreading have helped me a great deal. Another word of thanks to Menno Schellekens for his time for testing the research outcomes in the form of a case study.

I would also like to thank Niels Basjes for helping me validate the outcomes of this research and Gino Bazzoni for motivating me throughout the process and bringing me into contact with Niels Basjes.

Naturally, I want to gratefully thank my parents, family, and friends for believing in me and providing me with unfailing support and encouragement. Finally, special thanks goes out to my good friends Kaylesh Ajodhia, Hugo Brouwer and Dylan te Lindert. We were in the same boat together, and we managed to keep each other motivated when times got hard.

Contents

List of Figures

List of Tables

1 Introduction

The last decade our society has experienced a rapid transformation in many aspects of our lives, mostly driven by the rise of the internet. We are online every single day. We shop online and we live a large part of our social life online. The exponential growth of online eco-systems has as a consequence that individuals are being "bombarded by information whether or not they actively seek it" (Edmunds and Morris, 2000). This growth of information availability and variety makes it difficult to find and abstract relevant information.

For many online business, this issue has serious consequences: customers have too many options to choose from and may lack time and/or knowledge to check and consider each available option. Francesco et al. (2011) state that "while choice is good, more choice is not always better". Recommender systems aim to solve this issue by providing personalized recommendations based on an individual's past behavior and behavior of similar users. Recommender systems are software tools and techniques that use several sources of input to predict potential likes and interest of people to better fit their needs, provide a richer experience and filter out abundant information. The provided suggestions support individuals in the decision making process on, for example, what news to read, what products to buy, what movies to watch, etc.

The enormous value of effectively providing personalized recommendations is already demonstrated by many of the leading online businesses such as Amazon, Spotify, eBay, and Netflix. Visitors are turned into customers, customers spend more, and an intimate customer relationship is built. Enough reason for other businesses (and startups) to also provide personalized recommendations to their customers.

1.1 Innovation gap

One of the observations on the many studies conducted on recommender systems and information filtering is that many different approaches and algorithms exist. Currently, application designers and software developers have the task to choose the best approach amongst a large variety of algorithms. Most of the times, this results in experiments which compare the performance of different methods. Small companies, especially startups, often focus on creating minimum viable products (MVP) (Ries, 2011) due to the short time-to-market, unknown customer demands, and environment of high uncertainty. The process of designing a recommender system consumes substantial resources and does not result in short-term benefits. Also, most startups fail to understand the concept of providing personalized recommendations by not knowing what is involved in designing, operating and maintaining a recommender system. For these reasons, a recommender system is likely not part of an MVP. Startups focus rightfully on surviving and adapting to the continuously changing environment, while not choosing for implementing a recommender system or failing to implement one in the right way, thereby missing business value potential.

Designing, operating and maintaining a recommender system introduces several challenges and problems that startups have to face. First, many different approaches and algorithms exists. But there is more to a recommender system than a "simple" algorithm. Arguably the most important factor is the ensuring that the recommender system is aligned with the business model to have it ultimately deliver business value. Currently, there is little research on how recommender systems deliver business value. For startups specifically, the cold start problem (the initial lack of useful data) might introduce certain challenges as well. Other aspects that need to be considered are, for instance, the performance, the capabilities and the scalability of the recommender system. Naturally, the operational cost and the cost of maintaining a recommender system are key factors in the design process too.

1.2 Research purpose

This study is mainly focused on the practical side of recommender systems. By providing helpful tools for startups wishing to provide personalized recommendations to their users, startups are supported in the process of designing, operating and maintaining a recommender system. More specifically, the tools are captured in a reference model and help startups from the generation of recommendation ideas to business model alignment, to cost-benefit analysis up to the task of building and integrating the system. This research does not focus on improving specific recommender system algorithms or techniques. It also does not focus on the mathematics behind certain algorithms. The relevance of this study is two folded: academic and business, which are separately described in the following sections.

1.2.1 Academic relevance

The last decade, the topic of recommender systems has been vastly studied. Many of these studies focused on the algorithms of recommender systems (Adomavicius and Tuzhilin, 2010; Bao, Zheng, and Mokbel, 2012; Panniello et al., 2014; Safran and Che, 2017; Zheng et al., 2009), while the practical side is often underexposed. Books have been written on how to get started with recommender systems (Falk, 2017; Gorakala, 2016) but they do not take into account the noisy, uncertain and uncontrollable environment of startups, the considerations that underlay formulating a recommendation idea and justifying the right approach.

Ozok, Fan, and Norcio (2010) argue that current literature on web-based recommender systems is merely focused on algorithms and techniques, rather than design, implementation and usability issues. Further research on automated collaborative filtering by Herlocker, Konstan, and Riedl (2000) pointed out that current e-business recommender systems are black boxes which do not provide any transparency on how a recommendation is calculated. While their conclusion relates more to the algorithms and structural recommender system design, it does also point out that there is little transparency on how recommender systems should be implemented.

This research aims to describe distinct capabilities of different recommender system approaches and identify key problems and challenges that startups face when they want to offer personalized recommendations. Moreover, solutions to overcome or deal with these problems and challenges are given.

1.2.2 Business relevance

The benefits of successfully operating and maintaining a recommender system are widely researched. Chen, WuShin-yi, and Yoon (2004) proved that the number of recommendations and the number of reviews on an item have a significant impact on the sales of that particular item. By suggesting additional items to users, recommender systems are also known to promote and increase cross-sell opportunities. Furthermore, recommender systems can improve customer loyalty because when the needs of users are served well, they are more likely to return to the e-business website.

Clearly, recommender systems offer great potential for e-business businesses. For this reason, startups are likely highly interested in reaping the same benefits. However, as stated earlier, startups wanting to offer personalized recommendations to their customers have to deal with different challenges and problems. Startups are overwhelmed with the available options and need guidance in finding the most suitable option. And besides that, startups need to balance their resources and potential benefits carefully. The real challenge is to design the best possible recommender system with the least amount of resources required.

This study aims to solve this issue for startups by providing insights into the practical side of recommender systems and offering valuable guidelines and tools for startups, so that startups can rapidly and effectively work on designing, operating and maintaining a recommender system to ultimately enhance business growth.

1.3 Thesis outline

The next section, section 2, covers the methods used during this research. The research questions, the research scope, the design-science process and the literature review process are described in this section. The research method section also contains a list of definitions.

Section 3 outlines background information and relevant literature. The concepts of startups and business value are described in detail. Furthermore, two fields of research, information retrieval and information filtering, that are closely related to and interconnected with recommender systems are briefly described.

The fourth section aims to outline several fundamental concepts and theories on recommender systems. This chapter is closed by a section on the specific usage of recommender systems within the startup domain.

Section 5 consist of a high level description of the proposed framework. This framework, aiming at closing the previously described innovation gap, consists of several steps that provide guidance in designing, operating and maintaining a recommender system. Next, in section 6, the practical usage and the test and validation of the framework are described.

The following sections, sections 7 and 8 contain case studies that were used to test and validate the outcomes of this research. These chapters each contain a subsection on what their specific impact on the research outcome (the guidelines) were. Chapter 7 contains a summary of an interview with an expert in the field of recommender systems. The outcome of this interview was also used to validate the research.

The final two chapters respectively contain a discussion and a conclusion. In the conclusion, the research questions are answered and suggestions for future research are given.

2 Research methods

This section describes the methods that were used in an attempt to close the previously described *innovation gap*. First, the research questions are described. Next, the scope and the research methodology are explained. A description on how literature is reviewed is written in the following section. Lastly, a list of different terminologies and definitions used in this research is given.

2.1 Research questions

Primary question: how can startups be helped in overcoming problems and challenges when designing, operating and maintaining a recommender system that truly delivers business value?

- What type of techniques or algorithms do organizations use to provide personalized recommendations?
- What criteria can be used to describe the distinct capabilities of different recommender system approaches?
- How do organizations currently create (business) value from offering personalized recommendations?
- What problems and challenges do startups specifically face in offering personalized recommendations to their customers?
- What solutions can help address the problems and challenges that startups face?

2.2 Scope

This research is focused on closing the *innovation gap*. In other words, this research aims to make the process of designing, operating and maintaining a recommender system easier, more effective and less time-consuming for commercial startups. Not any small business is a startup, as will be further explained in section 3.1.1. The outcome of this research will primarily help businesses that *are born to be big* and operate in circumstances of extreme uncertainty (unknown customer demands, unknown markets, etc.).

The outcomes of this research might be useful for any other type of business. Yet, the focus of this research is primarily on startups.

2.3 Research methodology

Given the exploratory nature of this research, the need to identify and address certain challenges and issues and the need to search for solutions in an existing knowledge base, this research followed the design-science approach (Hevner et al., 2004). The authors of this framework have proposed an information systems (IS) research framework for defining business problems, finding solutions in a knowledge base, constructing artifacts and evaluating and refining these artifacts. This framework is displayed in figure 1.

The conceptual framework of the design science approach provides clear guidelines for understanding, executing and evaluating a research. Usually, design-science research is initiated by the need for a solution for a business or organizational problem. The framework uses that need to develop/build and justify/evaluate an IT artifact using a knowledge base consisting of foundational theories, frameworks, models and other literature. The business problem in the light of this research is the overall recommender system design process and the problems and challenges that arise during this process. Ultimately, the design-science research approach aims to create and evaluate meaningful IT artifacts that solve business problems. The IT artifact in this research is a solution that helps startups in the process of designing, operating and maintaining a recommender system.

This research contains four main phases: define, develop, evaluate and conclude. These phases are based on the IS-research framework. The first phase, **define**, focused on understanding the environment and the knowledge base to ultimately identify challenges and issues for startups and challenges and issues when designing, operating and maintaining a recommender system. During this phase, existing literature on recommender systems, startups and business value was reviewed. A detailed description on how this literature study was carried out is described in the following two sections. Also, in the first phase of this research, the research problem was described in more detail. A list of definitions was also created (section 2.5).

Once a certain level of understanding was gained and all relevant literature was reviewed, the

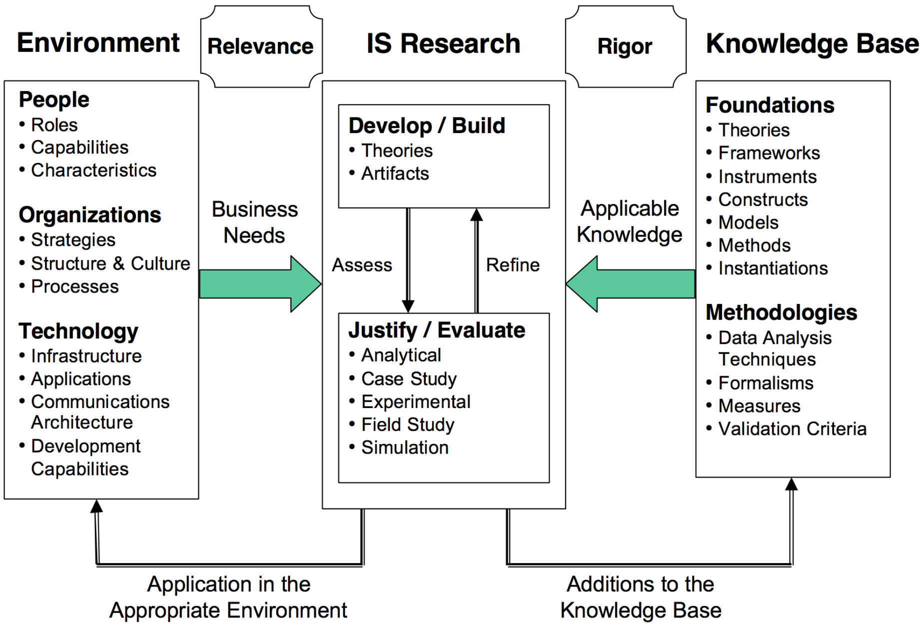

Figure 1: Information Systems Research Framework (Hevner et al., 2004)

research reached the second stage: **develop**. The goal of this stage was to design guidelines and tools for startups, as a first step in solving the business problem or making the innovation gap smaller. To achieve this goal, different challenges and issues for both startups and the process of designing, operating and maintaining a recommender system were listed and existing recommender system approaches were analyzed. Using this knowledge, a first version of the guidelines was created.

The third stage, **evaluate**, consisted of testing and evaluating the newly developed guidelines using two case studies. The goal of this stage was to test the usefulness and relevance of the guidelines. First, the guidelines were tested during the BarDoggy case study. Second, the guidelines were tested using the Analytics for Learning case study. The outcomes of the case-studies were used to adjust the guidelines.

After the guidelines were tested and validated using the case studies, the research was finalized. To officially finalize the research, the fourth stage, **conclude**, consisted of activities such as reporting and finalizing the research results, summarizing lessons learned, writing a discussion, explaining limitations and providing future research directions.

2.4 Literature review

To establish to what extent current research has progressed towards identifying and clarifying challenges and problems for both startups and recommender systems, a literature review was carried out. This section describes how literature on recommender systems, startups, and business value was reviewed.

2.4.1 Startups & business value

To gain insights in the what startups are, what challenges they face, what the definition of business value is and how business value is created, a brief literature review was conducted. First, Google Scholar was queried for keywords on startups and business value to find relevant publications and electronic journal databases. The keywords have been defined as follows: "startup strategy", "challenges for startups", "startup failures", "business value" and "business value creation". The search on Google Scholar resulted in 34 relevant papers regarding startups and business value. Besides browsing Google Scholar, regular Google searches were conducted to find more relevant literature. After having searched Google and Google Scholar, different electronic journal databases were searched for more publications and literature. The following electronic databases were searched:

- SpringerLink
- ScienceDirect
- ACM Digital Library
- ResearchGate

While searching for and reviewing relevant literature, master's and doctoral dissertations, non-English papers and unpublished papers were ignored. The main rule was to find papers that were published in academic journals since those papers are peer reviewed which increases the reliability of the authors and the overall quality of the papers.

2.4.2 Recommender systems

The topic of recommender systems has been and currently still is studied many times. Therefore, it is necessary to review existing literature and to have an up-to-date view of recommender systems.

An extensive literature study on recommender system research has been conducted by Park et al. (2012). This research reviewed a total of 210 research papers from 46 journals and classified each paper according to their self-made classification framework. The results of this research are useful but do not provide insights in recent challenges, problems, and trends. Assumably, since 2012 many new articles have been published. Following the exact same literature review to find more recent work (2010 and later) results in many thousands of search results. Therefore, the keywords have been slightly modified to only focus on challenges, problems, and trends.

The literature review was carried out as follows. First, certain keywords have been defined: "recommender system challenges", "recommender system problems", "recommender system trends". Google Scholar was queried for these keywords to find relevant literature and relevant electronic journal databases. Later, these keywords were used to find articles and publications on recommender systems in selected electronic journal databases. Recommender systems can be used in many different domains, such as marketing, computer science, information technology, statistics, and data analytics. Therefore, it is necessary to search different electronic journal databases to get a clear picture of all relevant literature. For this part of the literature review the same electronic databases as for the startup & business value review were used.

Searching these databases with the different keywords resulted in many thousands search results. Querying the keyword "recommender system challenges" on SpringerLink resulted in 11 results, in ScienceDirect in 10 results and in ACM Digital Library in 46 results. The abstract, introduction and conclusion of all the papers found by querying the different databases with the various keywords were studied. Research papers that were often cited in the found research paper results were also studied.

The review processes was conducted by one person. Articles and publications that were not directly related to recommender systems were not taken into account. And again, master's and doctoral dissertations, non-English papers and unpublished papers were ignored. The focus was on finding papers that were published in academic journals.

2.5 Definitions

To ensure a common understanding of terminologies and key concepts, it is essential to define important terminologies. This section aims to define the most common terminologies and key concepts used in this thesis. The definitions related to recommender systems are based on the definitions of Beel et al. (2016). Their definitions clearly outline key differences among terminologies which are often intertwined. Figure 2 displays the relationships between the different terminologies described in this section.

On the highest level, when an ***recommendation idea*** is discussed, this refers to a hypothesis on how recommendations might be created. An idea consists of four components which differentiate how detailed the idea is: classes, approaches, algorithms, and implementations. The least detailed idea is defined as a ***recommendation class***, a class is a high-level view on how recommendations could be generated. Examples of recommendation classes are collaborative filtering and content-based filtering, which are fundamentally different in their way of generating recommendations. Collaborative filtering uses the preferences of users to generate recommendations, while content-based filtering uses the properties of similar items. Both classes are described more detailed in section 4.2.

An idea which contains more details on how a recommendation class could be put into practice is referred to as a ***recommendation approach***. An approach builds upon the fundamental recommendation class, by simply explaining how the idea could be realized. Yet, recommendation approaches are meant to be rather vague, to leave actual details on how recommendations are to be calculated out. This leaves room for speculation on the exact calculation of recommendations.

Naturally, the next step of outlining the details of an idea does contain information on the calculation of recommendations. The ***recommendation algorithm*** accurately describes where certain user and item characteristics come from, and how they are processed to calculate recommendations. This might also include weighting schemes. A ***recommendation implementation*** is the actual source code of an algorithm and could even be the compiled source code. Key in this level of detailedness is that there is no room for speculation on how recommendations are calculated: every aspect is in there.

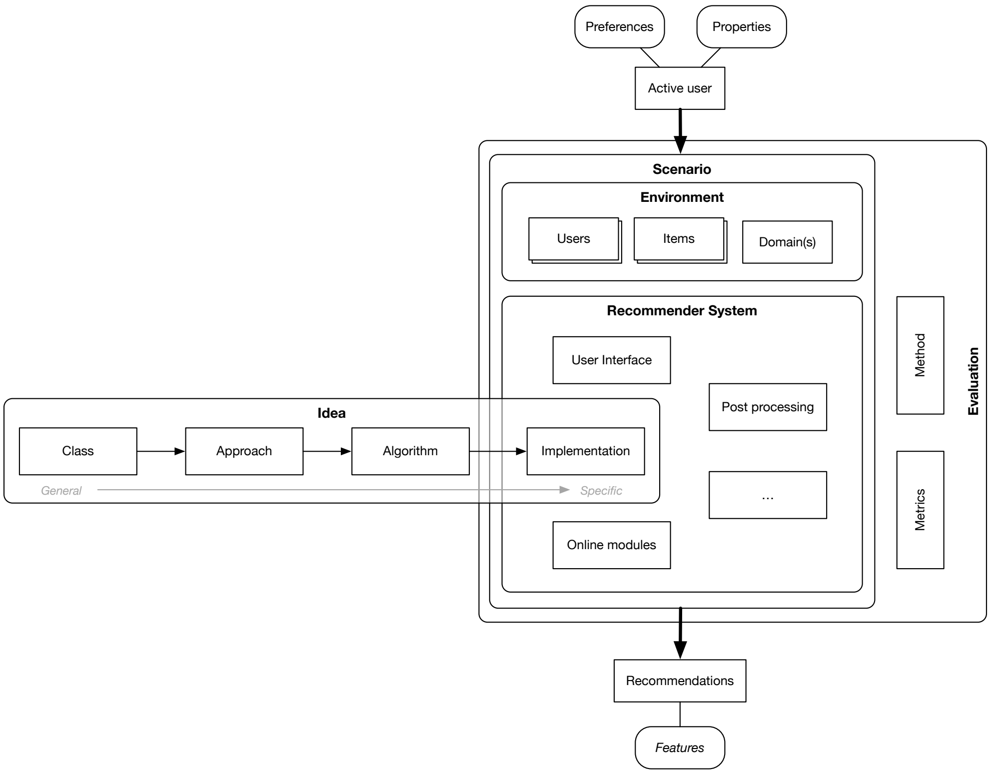

Figure 2: Relationships between definitions (based on Beel et al. (2016))

A ***recommender system*** is a fully functioning piece of software capable of generating recommendations via at least one implementation. A recommender system can consist out of multiple features, e.g., a feature that is responsible for generating personalized recommendations on a domain and a feature that is responsible for sending notifications when new items are available. Next, other components fit in this definition too: a user interface, post processing and other online modules. Nowadays, a trend can be spotted that recommender systems more and more use multiple implementations (or algorithms) to generate recommendations.

The entire setting and context of a recommender system are what is described as the ***recommendation scenario***, which includes the recommender system and the ***recommendation environment***. The recommendation environment contains "external" aspects such as the whole user- and item-space and the different domain(s). An ***item*** is an entity in the recommender system and can represent any kind of digital or physical product: a book, a voucher, a bar, a restaurant, a news article etc.

The ***active user*** is a single user that is currently interacting with the recommender system. The active user might have certain preferences and (demographic) properties. The ***recommendation process*** is the series of steps needed to calculate recommendations, from a user-perspective. The recommendation process takes certain input (e.g. about a user and his/her preferences) to *produce* certain recommendations. A ***recommendation*** is a single item fitting the user's preferences and interest best.

Quality metrics and methods assessing the quality of the recommender system fit the ***recommender evaluation*** block. Different quality aspects such as accuracy, user interaction, and satisfaction are defined and described in detail in section 4.7.

3 Background information & relevant literature

The following section aims to describe, explain and summarize relevant literature, theories and concepts to create a basic and common understanding of startups, business value and recommender systems. First, a description on startups and business value is written. The following section contains a part on information retrieval and information filtering, which form the foundation of recommender systems. Next, the concept and different approaches (both traditional and modern) of recommender systems are described. Lastly, the specific appliance of a recommender system in a startup context is written.

3.1 Startups & business value

The most basic definition of a business is an organization where products and services are exchanged for money or for other products and/or services. Starting a new business is not necessarily the same as starting a startup. Not every new business is a startup, while every startup is a business. Traditional businesses can often be started by having a business plan and requesting a bank loan. A startup is more challenging to set up. Many startups do not reach proper establishment, and when they do, many do not survive their first year (Evers, 2003). This section aims to define startups and outline the differences between a startup and a "normal" business. It also contains a description of challenges for startups and a definition of business value.

3.1.1 What are startups?

Ries (2011) defines a startup as "a human institution designed to create a new product or service under conditions of extreme uncertainty." The word startup does not say anything about the size of the company, the industry, the target markets or the type of goods or services offered. The key element is purely the creation of a product or service under conditions of extreme uncertainty. Startups are characterized by a short time-to-market, unknown customer demands, extreme uncertainty and often a limited amount of resources.

The word *innovation* is often used in the same context as startups. Innovation can be defined as the process in which a **new** idea is transformed into a good or service that creates value for the final customer. Naturally, the creation of new products or services by a startup is innovation. All kinds of innovations are possible: new scientific inventions, transforming existing technology for a new purpose, business models that target unexplored markets or bringing existing products or services to a new market or new target customer group. Innovation is key in all these examples.

3.1.2 Differences between startups and small businesses

The most fundamental differences between a startup and a small business are the driving force and the mindset of the founder(s). The founder of a startup is often dreaming about being big: gaining ground, growing and becoming a hit. Right from the first day, the startup has the intention to be a disruptive, large company. The founders believe that the "next big thing" is their idea. An idea that will shake up existing industries, steal customers from existing companies or even create new markets.

On the other hand, small businesses are independently owned businesses that are operated and organized for profit, are not particular dominant in their market or industry and do not aim to take over the world. Therefore, one of the main differences is the driving force: startups truly want to *disrupt* with a scalable and high impact business idea; whereas the intentions of a small business owner are often to be independent, to be an own boss and secure a place in a local and existing market. Both startup founders and small business owners can be considered entrepreneurs. Ultimately, the motivation behind a startup is fundamentally different from that of a startup.

Furthermore, a small business is a permanent organization designed to *execute* a repeatable and scalable business model. Small businesses often avoid being in conditions of (extreme) uncertainty and business models generally stay the same over long periods of time. A startup is a temporary organization designed to provide a vision of a product or vision, to create a series of hypotheses about virtually all pieces of a business model (who are customers? what are their demands? how can we be funded? what are the distribution channels?) and to rapidly validate whether the hypotheses are correct. If the hypotheses turn out to be incorrect, the business model gets changed immediately. The need for rapid validation results in a short time-to-market.

Naturally, both startups and small businesses are focused on generating revenue and ensuring future existence. However, there is a nuance difference. Startups tend to focus more on *gaining ground* rather than *gaining profit*. For startups, gaining market share is more important than gaining short term profits. In fact, losses are initially acceptable if market share is gained. For small businesses, this is the other way around. Small businesses often operate in local markets and value profit over market share.

Another fundamental difference is the way risks are handled. Small businesses usually avoid taking risks. Startups have risk involved in almost every aspect of the business model. This, of course, has to do with the extreme uncertainty startups face. Figure 3 outlines both differences and similarities between startups.

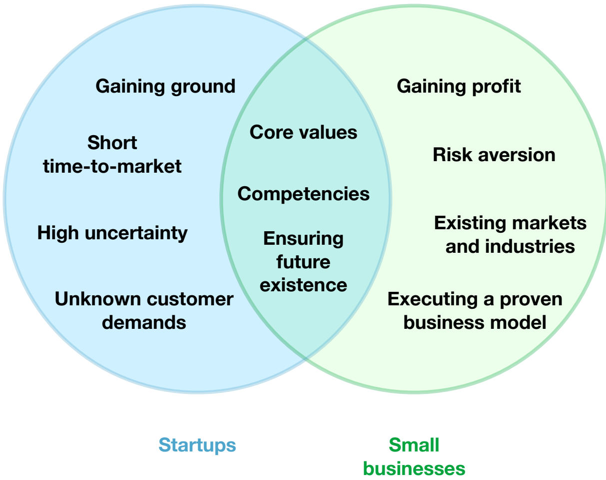

Figure 3: Startup and business differences and similarities

Another difference between startups and small businesses is funding. Most likely, both will initially be funded by savings of the founder, money from family and friends or (simple) bank loans. If a startup is proven to be successful, it will likely need and receive additional series of funding from either venture capitalists, crowdfunding or angel investors. Eventually, the startup might get more funding via an initial public offering (IPO). Each type of funding is likely to shrink the equity of the founder. The ownership of the company gets diversified by every round of funding.

Ultimately, a startup might cease to exist as an entity that operates independently because of a merger or acquisition. For most startups, this is acceptable as it might be needed to scale growth. For small business owners, a merger or acquisition would defeat the purpose of their existence, which likely is being an independent business owner.

3.1.3 Challenges for startups

While this research is not focused on the general challenges that startups face, it is important to briefly outline the circumstances startups function in. In fact, the general challenges that startups face do have impact in almost every aspect of the new born business.

Evers (2003) identified common challenges for all startups: finding sources of capital, coping with (extreme) uncertainty and having the right knowledge. Furthermore, the experience and the network of the founder(s) is considered a key enabler for a the success of a startup as is shown by empirical evidence (Storey, 1985).

The challenge of finding investors and the managing in financial and project terms can be considered key challenges. Founders of startups are good at thinking outside of the box and envisioning a big picture, but are often less skilled in 'thinking in numbers', that is, planning, estimating time and costs, revenue streams, cash flows, profit margins etc.

3.1.4 Business value & value creation

A business always starts with value creation. Porter (1985) defines value as "the amount buyers are willing to pay for what a firm provides them." The purpose of any business is to create and deliver value in such a way that a positive number is present when costs are subtracted from revenue, resulting in the so-called economic value. Economic value is a part of business value, where business value also includes other types of value which are difficult or impossible to express in monetary terms. Examples of these types of values are the value of employees, management, intellectual capital, strategic partnerships, environmental and societal impact. Naturally, the intangible synergies that make the business as a whole greater than the sum of its parts is also a part of business value.

When referred to value or business value in this research, value means the total value created regardless of whether a customer, the business itself or any other stakeholder initiated the value transaction. Traditionally, business value can be measured in terms of revenue growth, customer satisfaction, market share or profit.

Based on different theoretical models and frameworks, Radhakrishnan, Zu, and Grover (2008) identified the sources of value creation in online business and created a model containing these sources. This model, displayed in figure 4 is based upon well-known theories and literature such as the value chain framework by Porter (1985), the theory of economic development and new value creation by Schumpeter (1934), resource-based view (RBV) literature, and strategic network theories.

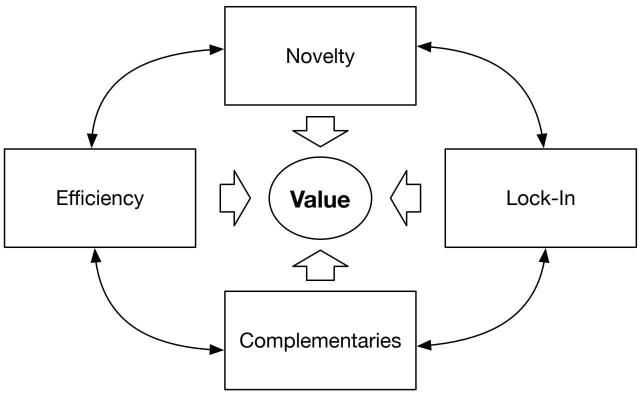

Figure 4: Sources of value creation in online business (Radhakrishnan, Zu, and Grover, 2008)

The value creation potential of online businesses is, according to the authors, based on four interconnected dimensions: efficiency, complementarities, lock-in, and novelty. These four dimensions are brought together in a model, which is displayed in figure 4. The **novelty dimension** refers to the value that is created when new products or services, new business models or other innovations are introduced. This creates value since no other players offer this yet.

Whenever products or services are bundled together and provide more value than having each item separately, complementaries are present. This value gained from this strategy is placed under **complementaries dimension**. The **lock-in** domain refers to the value that can be gained from recurring customers.

3.2 Information retrieval & Information filtering

One of the earliest types of recommender systems were information retrieval systems designed to deal with the issue of information overload. These systems were mainly designed to filter out items of low interest, to only display elements of high interest. An example of an information retrieval task is a student who is searching a digital library for a specific article. A certain query containing keywords in line with the current information need is given, which a search engine then uses to browse its index to retrieve the items that match the keywords in the given query. Typically, information retrieval is used in situations where information is stored in relatively large and static databases. Nowadays, information retrieval is widely implemented on the web. Websites and online shops use information retrieval to enable users to search (real-time) through a product catalog database. Information retrieval can use boolean operators between words to combine search words (e.g. "and", "or" and "not"). Figure 5 outlines these boolean operators. The green parts represent the search results that will be shown. For the *AND* operator, online the results containing both words will be shown. The *OR* operator shows results where either one of the words is present. *NOT* excludes results containing a certain word.

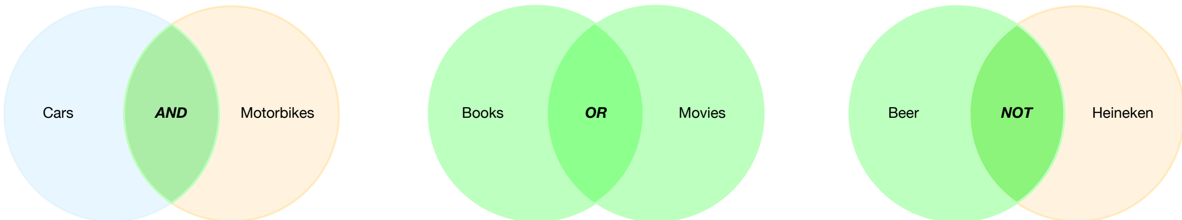

Figure 5: IR boolean operators

The field of information filtering does not necessarily use a query to find items of high interest, it is mainly focused on using the filtering technologies to identify items that match specific information needs. Information filtering is often based on descriptions of information needs of an individual user or a group of users. According to Belkin and Croft (1992) information retrieval and information filtering are fundamentally different tasks. The key difference between information retrieval and information filtering is that filtering removes information from a source, where retrieval aims to find information in a full source.

4 Recommender systems

This section aims to outline the core concepts of recommender systems. The following subsection contains a high-level definition of a recommender system. Next, traditional recommendation algorithms are explained. In the following subsection, different challenges and limitations are described. Modern recommendation algorithms are briefly addressed in the next subsection. The following three subsections touch the topics of the degree of personalization, common application models and quality metrics. Lastly, this section contains a brief conclusion on the usage of recommender systems in the context of a startup.

4.1 What is a recommender system?

A recommender system is a technology that is capable of recommending certain items to users. Very often, there are many items as well as many users, which makes the recommendation process complex. Both users and items have attributes: users might have personal preferences and items might have certain properties. The more attributes an item or user has, the better the recommendation results could be.

According to Burke (2002), there are typically three aspects required for a recommendation process to start:

1. background data: data that is present before the recommendation process begins
2. input data: a user's information and preferences which are needed the system to provide recommendations
3. an algorithm that combines the first two aspects to calculate the personalized recommendations

Let's assume that U contains the set of users whose preferences are present, I represents the set of items over which recommendations can be given, u represents the user whom requested recommendations and i is an item for which the preference of u needs to be calculated. The next section uses this nomenclature to explain and outline how different recommendation classes function.

4.2 Traditional recommendation algorithms

To understand the problems and challenges startups face when dealing with recommender systems, it is necessary to first understand and analyze the different algorithms and recent developments of recommender systems in general. This section aims to explain how different algorithms work. Based on the work of Burke (2002), "core" techniques as content-based, collaborative filtering, demographic, utility-based and knowledge-based techniques are described.

Content-based algorithms Content-based algorithms are built upon techniques from the field of information retrieval research. The general concept is that the system tries to recommend similar items based on a user's past preferences, while not taking into account the preferences of other users. Basically, the system aims to build a predictive model of the user preferences. For example, if the user has watched a TV-show in the drama genre, the RS will likely recommend a different TV-show in the same genre.

The recommendation that a content-based algorithm provides is solely based on a profile which consists of an analysis of the content of items which a user has rated previously. Often, user feedback is collected to make more accurate recommendations. Feedback can be collected both explicitly and implicitly:

- **Explicit** feedback is provided by users when they rate an item. This can be either by giving a certain amount of stars (1 to 5), like/dislike (binary) buttons or similar techniques. Often, it is difficult to obtain explicit feedback because when users do not like an item, they are likely to not choose the item instead of giving negative feedback.
- **Implicit** feedback is collected when a user's behavior is monitored. For instance, this is data that can be collected when a user watches a movie, listens to a song or views an item. This kind of data is easier to obtain than explicit feedback.

The background data for a content-based algorithm is a list of available features of the items I. The input data is the user's rating of certain items in I. The algorithm uses the background and input data to estimate the likelihood whether the user will like a certain item. Only the items that have a high similarity degree are recommended.

Collaborative filtering-based algorithms The most widely implemented and familiar algorithm probably is the collaborative filtering-based algorithm. A collaborative filtering algorithm aims to predict a user's preference based on the preference of a whole group of other users. Collaborative filtering approaches are based on the assumption that if user A has the same item preference as user B, then user A is also likely to have the same preference on a different item. A recommender system using collaborative filtering basically automates the 'word-of-mouth' principle. The opinions of other users with similar interests are used to help the user to make a choice (Lu et al., 2015). An example: if user A likes item 1, item 2 and item 3 and user B likes item 2, item 3 and item 4, then their item preference is similar. In this case, the algorithm should recommend item 4 to user A and item 1 to user B.

A collaborative filtering-based algorithm uses the ratings of users U on the items I as background data. The individual ratings of user u on the items I are used as input data. The algorithm tries to identify users in U that are similar to the user u, and use the ratings of those users to predict a certain score on item i.

One of the preferred algorithms for collaborative filtering is the k Nearest Neighbors (kNN) classifier algorithm (Francesco et al., 2011). The kNN classifier is the most widely used algorithm for collaborative filtering according to Bobadilla et al. (2013). First, the algorithm determines the k user neighbors for the user u. Then, the ratings for the items which are not rated by user u are aggregated by using ratings of the neighbor users. Lastly, the predictions are extracted and the top N recommendations are provided to the user u.

Both the content-based and collaborative filtering algorithms face the *cold start problem* which occurs when not enough meaningful data is present. The cold start terminology is borrowed from a similar problem with cars. When the engine of a car is cold, it usually has problems starting. However, when the engine is started and later runs at a normal temperature, it runs perfectly smooth. For the content-based and collaborative filtering algorithms, this simply means that there is insufficient data available to calculate the recommendations optimally. For startups, the cold start problem is a typical and often occurring situation, as they often start with zero customers.

Demographic algorithms Another way of providing a user the best possible recommendations could be by looking at personal attributes. Based on these personal attributes, demographic recommender systems try to categorize a user. The exact personal attributes can vary greatly, as the most important thing is to be able to group users. The demographic class of a user is then used to identify similar users, and extrapolate the preferences of those users to the user.

The demographic technique is based on the premise that users with similar personal attributes, will also have similar preferences. One of the benefits of the demographic technique is that a new user does not need to have rated items previously to get a recommendation. Thus, this technique does not face the cold start problem.

The demographic information about all users U and the ratings of each user on items in I is the background data in this case. The demographic attributes of user u are input data. The algorithm simply tries to identify users in U which are similar to user u. Next, the ratings on I by similar users are used to predict the rating of item i.

Utility-based & knowledge-based algorithms In contrast with content-based, collaborative filtering and demographic algorithms, utility- and knowledge-based algorithms provide recommendations based on the user's specific needs and the set of available items. For example, a utility-based algorithm could take item stock or delivery times into account as the user might prefer a fast delivery over having the best item. Knowledge-based algorithms do know precisely how a particular issue the user faces can be solved by an one item. The simplest example of a knowledge-based recommendation might be a search query, e.g. "books about WOII". A knowledge-based solution could then offer an editor's choice list: "Best books. about WOII". In this case, the system contains *functional knowledge* about how a certain item could meet the user's requirements: it knows that the search query should show the editor's choice list on that particular topic.

The background data for utility- and knowledge-based systems is information about the features of items in I. Naturally, the functional knowledge within knowledge-based systems is considered background data as well. For utility-based systems, the input data is a utility function for each of the available items in I, which describes the preferences of user u. For knowledge-based systems the input data simply is a description of the preferences or needs of user u.

A knowledge-based system uses the background data and the input data simply to find a match between user u and item i. The utility-based system applies the utility function and by doing that, it lists the best fitting items in I.

Table 1 contains a brief summarization of the previously described recommendation approaches.

Class	**Background**	**Input**	**Process**
Collaborative filtering	Ratings from U of items in I.	Ratings from u of items in I.	Identify users in U similar to u, and extrapolate from their ratings of i.
Content-based	Features of items in I	u's ratings of items in I	Generate a classifier that fits u's rating behavior and use it on i.
Demographic	Demographic information about U and their ratings of items in I.	Demographic information about u.	Identify users that are demographically similar to u, and extrapolate from their ratings of i.
Utility-based	Features of items in I	A utility function over items in I that describes u's preferences.	Apply the function to the items and determine i's rank.
Knowledge-based	Features of items in I. Knowledge of how these items meet a user's needs.	A description of u's needs or interests.	Infer a match between i and u's need.

Table 1: Recommendation algorithms (Burke, 2002)

4.3 Challenges & limitations

The previously described recommendation algorithms have strengths and weaknesses. Besides that each algorithm has certain weaknesses, the specific use-case of a recommender system often introduces various challenges and limitations too. Almost every recommender system implementation is unique, but there are challenges that are relevant for most implementations. The most common challenges and problems that were encountered during this research are:

1. The cold start problem
2. Small datasets, large datasets & the sparsity problem
3. The portfolio effect

4. The “gray sheep”-problem
5. Scalability
6. Proactive recommendations
7. Context-aware recommendations
8. Diversity of recommendations
9. Short- and long-term preferences
10. Generic user models and cross domain appliances
11. Crowd-avoidance
12. Mobile based recommendations
13. Group recommendations and generalizations
14. Providing explainable recommendations

These challenges are separately described in detail in attachment A: Recommender System Challenges. Challenges 1 and 2 seem to be highly relevant for start ups, as they both concern the size of the user- and item-space. The other challenges are dependent on the specific use-case and how the recommendation idea is defined. Context-aware recommendations might, for instance, not be needed when a demographic algorithm is sufficient. Or, mobile based recommendations, might not be needed when the recommender system only outputs to a (desktop) website. Figure 6 outlines what challenges are relevant for specific algorithms.

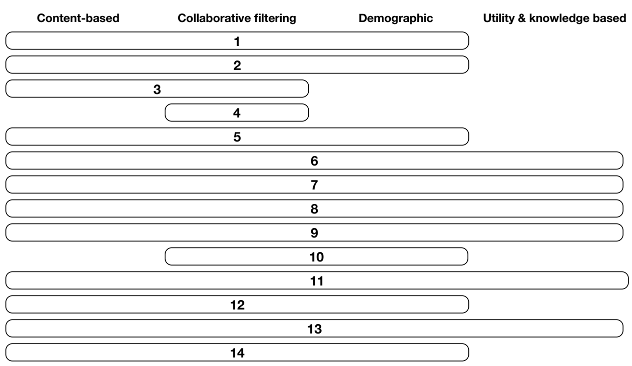

Figure 6: Challenge/algorithm classification

The list of challenges and limitations combined with the individual strengths and weaknesses of the different traditional algorithms results in a table that outlines the tradeoffs between the traditional algorithms. Table 2 outlines these tradeoffs, partially based on the findings of Burke (2002).

Algorithm	**Strengths**	**Weaknesses**
Collaborative filtering	A. Capable of identifying cross-genre niches (generic user models and cross domain appliances) B. No domain knowledge needed C. Adaptive: quality improves over time D. Runs using implicit feedback only E. No information about item features needed	F. Cold-start problem G. "Gray sheep"-problem H. Small datasets, large datasets & sparsity problem I. Short- and long-term preferences problem J. Group recommendations and generalizations difficulties
Content-based	B, C, D	F, H, I, J
Demographic	A, B, C	F, G, H, I, K. Demographic information must be present
Utility-based	E, L. Capable of handling I M. Minimal user input needed	N. User input on utility function required O. Not adaptive
Knowledge-based	E, L, M,	O P. Requires knowledge engineering

Table 2: Recommendation algorithms tradeoffs

4.4 Modern recommendation algorithms

The individual weaknesses can be partially or fully overcome by combining different algorithms. Currently, this is a trend in the field of recommender systems. Algorithms are combined in an attempt to solve individual drawbacks and achieve peak performance. Many researchers choose to combine algorithms in different ways, tailored to their specific needs. The combination of different algorithms is often referred to as an *hybrid approach*.

Hybrid approaches

Hybrid approaches can be organized in different ways (Burke, 2002). The *weighted* hybrid approach uses the outcomes of different algorithms to calculate a single recommendation. Another possibility is the *switching* of algorithms depending on the current situation. For example, when no ratings are present, a demographic algorithm is used instead of a collaborative filtering one. Outcomes of different algorithms might also be presented at the same time, which is called the *mixed* approach.

Furthermore, features of different algorithms might be combined into a single recommendation algorithm (*feature combination*). Algorithm can be organized as chains, using an algorithm to refine the outcomes of a different algorithm (which is called *cascade*). *Feature augmentation* can be used when certain output of one algorithm is used as an input feature for another. Lastly, *meta-level* uses the user or item model of one algorithm as input for a different algorithm.

For the weighted, mixed, switching and feature combination approaches, the exact sequence of the algorithms employed does not matter. Content-based and collaborative filtering algorithms will always need user ratings, even when combined with demographic, utility based or knowledge based algorithms.

Computational intelligence & social network based approaches

Another trend is the usage of computational intelligence-based recommendation approaches (Kunegis, Said, and Umbrath, 2009; Lu et al., 2015; Schafer, Konstan, and Riedl, 2001; Yera, Castro, and Martínez, 2016). These approaches make use of sophisticated data analysis and data mining techniques, such as, artificial neural networks, genetic algorithms, clustering techniques and fuzzy set techniques. These techniques are used to 'learn' from data, spot patterns and extrapolate those findings to generate recommendations.

Also, social-network based recommendation techniques are gaining more popularity (He and Chu, 2010; Lu et al., 2015). Probabilistic models can be developed to utilize information from social networks such as: user preferences, influence from friends, and an item's general acceptance. According to He and Chu (2010) friends have a tendency to select the same products and give similar ratings. Experiments by the same researchers prove that the prediction accuracy improves when social network data is used. Not only does the prediction accuracy increases, social network data also serves as a remedy for the cold start problem and data-sparsity issues.

4.5 Degree of personalization

Recommender system algorithms are capable of producing recommendations at different levels of personalization. Schafer, Konstan, and Riedl (2001) identify three levels of personalization: non-personalized, ephemeral personalization and persistent personalization. The non-personalized level contains recommender systems that provide identical recommendations to each user. These kinds of recommendations can be provided via manual creation, statistical methods or other techniques. In the e-business domain recommendations are often non-personalized. Examples of this kind of recommendations are lists such as top 10 best selling, editor choices and featured items. Key is that these recommendations are the same for each single user.

Recommender systems that use current user data as input are classified as ephemeral personalization. This level is closely related to the non-personalized level but differs since current user input is used to give more personalized recommendations. Examples of recommender systems in this level are systems which use user data of the current session or shopping cart to provide recommendations. Recommendations are personalized in a 'short-term' way: only the visited items in a current browsing session are considered.

Persistent personalization, the highest level of personalization, is used in recommender systems which provide different and unique recommendations for different users. When different users look at the same item, the given recommendations are different based on their own profile. User-to-user correlation or collaborative filtering methods are often used to provide this degree of personalization. These kinds of recommendations can be seen as 'long-term' as the recommendation is not particularly bound to, for example, one browsing session.

4.6 Common application models

The business model and business goals largely define how the recommender system has to be integrated into existing processes and products. Clearly, the different recommendation algorithms, challenges and limitations result in many different ways of designing and integrating a recommender system. On top of the integration, the algorithm and certain design considerations make each recommender system implementation unique. Yet, there are different design patterns that can be observed when looking at existing recommender systems. Research by Schafer, Konstan, and Riedl (2001) has revealed five common application models of recommender systems used in an e-commerce setting: broad recommendation lists, customer comments and ratings, notification services, product-associated recommendations and deep personalization. In line with the definitions used in this thesis, these application models can be seen as recommendation ideas: high level conceptualizations.

Broad recommendation lists are mainly used by businesses that offer thousands to millions of different products. Not only do they try to avoid users from getting lost and frustrated, they also aim to engage users as early as possible. Nowadays, because of the ever-increasing amount of online shops and the easiness to browse from one site to another, it is important to keep users on a website. Broad recommendation lists contain non-personalized suggestions and are lists such as best rated items, best sellers, editor choices etc. This kind of lists help (new) users to find their way on a website and familiarize themselves with the content offered. The level of personalization is low, which is exactly one of the advantages of this application model as no data about a specific user is needed. Another advantage is the fact that the business can influence the content of such recommendation lists. This is helpful for promoting items that need broad attention because of high inventory levels or inventory that needs to be sold. Having said that, this application model results in higher business value in contrast to customer value.

Customer comments and ratings allow users to write and read ratings and reviews on certain items. According to Schafer, Konstan, and Riedl (2001), users often feel that e-commerce businesses are focused too much on selling products only. Of course, it is true that businesses are mainly focused on maximizing revenue and profit, users do not need to experience this. It must be avoided that a recommendation gives the user the feeling that the recommendation is only there to induce them to buy an item. This has to do with the level of credibility of a business, as it needs to satisfy both the feeling and demands of the user, as well as achieving business goals (sustaining growth, generating revenue). A possible way of overcoming this problem, to fulfill and satisfy users and build credibility, is to provide a "community center", which allows users to write and read ratings and reviews. Basically, this enables users to communicate and provide each other with advice. Based on both negative and positive ratings and reviews, a user can form its own opinion. One of the advantages of a community center is that the user-base provides the content by itself, the users rate items and write reviews themselves. This contributes to a higher credibility since users are more likely to trust the opinion of other users rather than the content creators of the website. In the eyes of a user, content-creators can be biased, as they only focus on selling products and making profit. Another advantage is that the community helps in engaging users, as the sense of a community helps the website in being different from competitors. This application model is results in both higher customer and business value.

Notification services are of great value when it comes to building customer relationships. Businesses use notifications to reach users and inform them about the arrival of new items and when certain promotions are active. To be able to do this, data about a user's preference needs to be present, as is it is only relevant to notify users about items of their preference. When such data about a user is present, users can be targeted accurately, which benefits both the user and the business. According to Schafer, Konstan, and Riedl (2001), a notification is effective for having users visit the website again and increasing the customer retention. A slight disadvantage of this application model is that user data is needed. Yet, this data does not have the be highly detailed, as high-level knowledge on a user's preference is already enough to notify them about relevant content.

Product-associated services are comparable with what physical shops and retailers do: arranging products in such a way that it enhances cross-selling. Complementary items for a product are placed in the same aisle to have customers buy more products. The advantage for e-business websites is that there are no physical limitations: recommendations can be directly shown when a user is looking for or viewing an item. The degree of personalization is at ephemeral level: current user input is needed and used to give 'short-term' personalized recommendations. Many e-business websites use this application model, for the reason that a high variety of data sources can be used to calculate the best recommendations. Examples of this kind of data sources are ratings, item attributes, purchase histories and expert opinions. Then, content-based algorithms are used to calculate item similarity to passively provide recommendations to the targeted user. Mostly, product-associated recommendations are integrated on a product detail page.

Deep personalization is the most desirable objective of most e-business websites: creating long-term customer relationships that lead to customer loyalty, to eventually create higher competitive barriers. Deep personalization is already widely implemented in the web advertisement business, as on different websites ads are displayed while still fitting a user's individual interest. More and more e-business websites are working with deep personalization techniques as well. Using a user's (browsing) behavior, preferences and purchase history it becomes possible to do collaborative filtering. The ability to use other users' behavior and preferences to predict an individual's preference is the most difficult, but also the most valuable way of recommending items. The algorithms used for deep personalization are the traditional recommendation algorithms as well as hybrid approaches.

Interestingly, the more advanced the application model, the higher the expected customer value. When comparing *deep personalization* with *broad recommendation lists*, the customer value is substantially higher for *deep personalization* while the business value does not increase significantly.

These five application models each address different business goals. Businesses are not necessarily tied to one application model. In fact, many businesses already employ different application models, using one or more recommender system implementations. In attachment B: Recommender System Design Framework for Startups section 3, these application models are linked to certain business objectives to illustrate how recommender systems are capable of achieving business goals.

4.7 Quality metrics

One might think that there has to be a 'one size fits all' solution for recommender systems: a perfect system using the latest technologies, capable of making recommendations in every single use case. Unfortunately, that is not the case. The different knowledge sources, different contexts, and system requirements make almost every recommendation implementation unique. A valid question to be asked is: knowing that every recommender scenario is unique, what are the most promising approaches for my use-case?

Beel and Langer (2015) state that "an identification for the most promising approaches for recommending research papers is not possible, and neither is a replication for most evaluations". While their research focused on recommender systems for research papers, it proves that it is difficult (if not impossible) to select the best approach for a certain use-case. Assumably, the main reason for this is that, while it is relatively easy to evaluate a single recommender system, it is challenging to compare different approaches based on these evaluations due to different circumstances and context. For evaluating a single recommender system, different options are available:

- **Objective statistical metrics** using techniques developed in the fields of information retrieval and machine learning. These metrics are evaluated algorithmically, do not require user involvement and address a single RS quality criteria: relevance. Objective statistical methods can be categorized in: error metrics and accuracy metrics. The capability of a recommender system to estimate ratings users would give to an item can be measured by certain **error metrics**:
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)

  To measure the capability of a recommender system to accurately select a small set if items matching the preferences of the user, certain **accuracy metrics** can be used:
  - Precision
  - Recall
  - F-measure
  - Hit rate
  - False positive rate
  - Mean Average Precision (MAP)
- **Subjective quantitative metrics** are explored by different recommender system researchers in recent years. Subjective quality metrics use a user-centric approach to measure and improve the quality of recommender systems. Objective statistical metrics might point out that a certain approach results in the "best" recommendations, but users do not necessarily have to find these "best" recommendations satisfactory and useful. Cremonesi et al. (2011) concluded that objective statistical metrics are not necessarily a good predictor of the quality as it is perceived by users. This conclusion is based on comparing the objective statistical quality against the following subjective quality metrics:

  - *Perceived accuracy* determines to what extent the recommendation matches the preferences, interests, and tastes of a user.
  - *Novelty* outlines the degree to which the user receives new or surprising recommendations. In RS literature this is sometimes also called *serendipity*.
  - *Global satisfaction* measures a users' overall "feeling" with the recommender system.

  Other researchers emphasize the need for diverse recommendations, recommendations that recommend items out of, for example, different product groups or other domains (Hou et al., 2017). Diversity and serendipity are seemingly subjective metrics but are in fact quantifiable and can be seen as subjective quality metrics. The diversity concept is further described in attachment A: Recommender System Challenges.

- **User- and item-space quantitative metrics** or user/item-space coverage can be calculated using the number of users, items, and generated recommendations. Knowing how many users are receiving recommendations and how many items are actually used for recommendations can determine the quality of the RS in terms of coverage. For calculating the coverage, the recommender system has to function in a production environment.

- **User interaction quantitative metrics** or online metrics are capable of measuring the interaction of a user with the recommender system. Different variables can be measured: click rates, click-through-rates, conversion rates etc. This particular metric also requires that the recommender system functions in a production environment.

Further research by Beel et al. (2016) points out that evaluation results are rarely reproducible, making it even harder to define the quality of a recommender system in terms of quantitative metrics. Evaluation results are seldom reproducible because the reproducibility is difficult to ensure since small deviations might lead to large differences in the effectiveness and performance of a recommender system approach. The same authors suggests that recommender systems should be *effective*.

Effectiveness can be defined as the extent to which a recommender system is capable of achieving its objective(s). Typically, one of the objectives is to provide "good and useful" (Gunawardana, Shani, and Il, 2009) recommendations that make users happy (Ge, Delgado-Battenfeld, and Jannach, 2010). Needs and interests of users vary, resulting in different items making users happy. Closely related to providing good and useful recommendations that make users happy, is the overall quality of the software. To make users happy, the recommender systems needs to function reliable to avoid potential downtimes. The system needs to be efficient when deployed to production: performance and scalability are paramount, users do not want to wait long for a recommendation. The recommendation needs to be visible instantly otherwise users might leave.

Less related to making users happy are the quality aspects of security and maintainability. Potential security breaches as a result of a poorly designed architecture or poorly coded software play a role in the overall recommender system quality. When the system is poorly maintainable, e.g. not adaptable, not portable or not transferable (between different software engineers), it is hard to further improve the system. Naturally, this impacts the overall quality too. The reliability, performance efficiency, security and maintainability are software quality characteristics defined by the Consortium for IT Software Quality (*Code Quality Standards*, n.d.).

Another quality factor that is often overseen by various researchers in the field is that a good recommender system also satisfies the needs of the business. Most of the times, it is assumed that when users are satisfied, the business is satisfied too. But that does not necessarily have to be the case in every situation. Most businesses, especially startups, aim to keep costs low. Costs concerning a recommender system could be computational costs such as disk storage, CPU power, data traffic etc. Thinking from a business/cost perspective, a "good" recommender system might be defined as one that is easy to develop, to operate and to maintain while the costs are kept low. Also, when thinking from a business perspective, the recommender system needs to deliver business value, enabling the business to achieve business goals. The recommender system might be perfect in terms of objective statistic metrics, subjective quality metrics a user- and item-space coverage, but if it does not deliver business value, it might not be that good after all.

To ensure that a recommender system delivers business value, business value has to be defined and relevant business goals have to be identified. When these to aspects are known, the recommender system can be assessed on whether it is capable of delivering business value. A/B tests can be used to determine if recommender systems genuinely result in business value. Examples of goals and metrics that can be used when doing A/B tests:

- **Number of items viewed and or sold.** Measure how many recommended items are actually viewed and/or purchased.
- **Item diversity**. Measure the diversity of items being sold.
- **Customer satisfaction**. Ask customer feedback to identify the satisfaction level (customer satisfaction metrics).
- **Customer intimacy/loyalty**. Identify the extent to which customers are recurring.
- **Increase market share**. Introduce novel personalization features, or increase customer satisfaction and/or customer loyalty

4.8 Startups & recommender systems

To find out if customers really appreciate personalized recommendations, a recommender system has to be part of the startup's MVP. When a recommender system is part of a MVP, the startup can quickly learn if and how customers want to see personalized recommendations. According to Schafer, Konstan, and Riedl (2001), recommender systems have the potential to enhance a business (and startup) in several ways:

- **Converting browsers into buyers** most visitors browse a website without purchasing anything. Recommender systems can personalize the user experience by suggesting items that fit the active user's preferences and interests.
- **Increasing cross-sell opportunities** by showing personalized or related items next to the item that the active user is currently viewing. Additional items might, for example, also be shown during the check-out process.
- **Building loyalty** by learning about users and their preferences, an intimate customer relationship can be built. When a website is able to personalize the user experience, users tend to be more loyal in comparison with websites that do not offer a personalized experience. A personalized user experience helps in retaining customers.

Businesses often want to influence the behavior of users, by persuading them to view, consider or even purchase an item. Research by Garfinkel et al. (2006) proves that objective and credible recommendations are capable of influencing the behavior of a user. Recommender systems allow businesses to practice mass customization and leverage information about users to create a more personalized experience for users. For users, this feels light the business “knows them” and is capable of serving them in an effective way. This also enhances customer loyalty and intimacy as customers are not treated as random strangers.

While these two studies cover most of the recommender system potential, startups have to find out themselves if and how a recommender system can be leveraged to achieve business value. For some startups, certain off-the-shelf solutions might be sufficient for generating both business and customer value. Yet, for other startups, customized recommendation algorithms are required to reap benefits.

The turbulent context of a startup makes the process of designing, operating and maintaining a recommender system more challenging when compared to the context of an established and ‘normal’ business:

- Most startups do not have many customers yet. Also, these newly founded businesses likely do not offer many different items or products. This means that most startups are highly likely to face the cold start problem.
- Startups do not have the right knowledge to design and develop a recommender system. And with that, most theories and literature available do not focus on the practical side of recommender systems. This results in startups spending valuable time and money on exploring, discovering and testing recommender system approaches without getting anything in return.
- The often unknown customer demands force startups to continuously gather customer feedback to improve their value proposition. Spending too much time and resources on creating a supposedly ‘good’ recommender system is dangerous: do customers eventually value the recommender system and does it result in the expected business value?
- The low time-to-market forces startups to rapidly develop their product or service offering. Business decisions have to be justified, but the pressure to deliver products/services in a fast pace might result in certain decisions not being justified. Assumptions can be easily made, which might have serious consequences.

- Lastly, although not applicable in all cases, most startups only have a limited amount of resources (time and money) available. Spending these resources in a wrong way (e.g. not making an educated guess) can be fatal for the startup.

Clearly, designing, operating and maintaining a recommender system introduces certain problems and challenges. For startups, these problems and challenges are even more striking. When designed properly, a recommender system offers huge potential. A poorly designed recommender system might not result in substantial benefits, the benefits might not even outweigh the costs. A good justification and an educated guess is needed before the design and development process can start.

5 Framework for designing, operating and maintaining a RS

To help startups in designing, operating and maintaining a recommender system, a framework consisting of basic steps was built. These steps guide startups in the whole recommender system creation process. The full framework can be found in attachment B: Recommender System Design Framework for Startups. The framework offers several tips & tricks, tools and models to help understand recommender systems and outline what decisions underly designing the right approach. Ultimately, the goal of the guidelines is to design a fully functioning recommender system that truly delivers business value. This section aims to explain how the framework was built. Furthermore, this section contains a high-level summary and a justification for why certain steps, tools and models can be useful.

5.1 Framework summary

The general approach that might be taken to designing, operating and maintaining a recommender system is outlined in figure 7. The first step is to set a foundation by analyzing different business aspects. In order to have a recommender system that truly delivers business value, it must first be pointed out what the business value for the particular startup is. The first step does not provide guidelines for creating a business model, it rather aims to outline relevant aspects of the business model and business objectives to align the recommender system with.

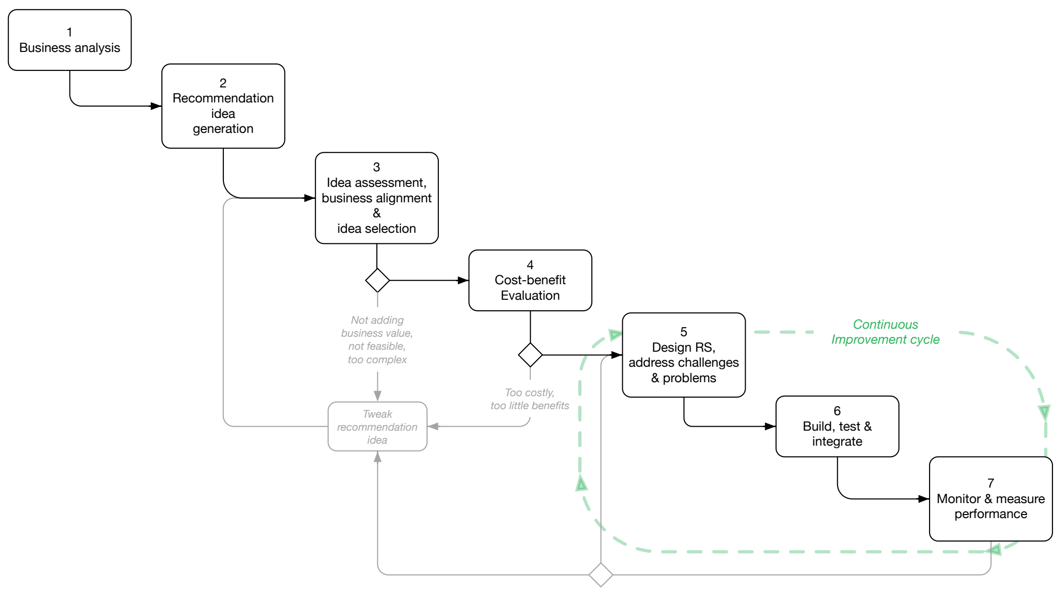

Figure 7: Framework for designing, operating and maintaining a recommender system

Next, different recommendation ideas have to be generated (step 2). A recommendation idea should make clear what types of recommendations are given in what way. To generate different ideas, various input sources can be used: business ideas, systems of competitors, theories, literature and existing application models as described in 4.6.

After multiple recommendation ideas have been created, each idea has to be assessed on required effort, complexity and risk involved (step 3). Using the *dimension model*, these three aspects can be assessed. Furthermore, the dimension model outputs certain focus areas: areas that need extra attention during the design phase. It can occur that an idea is not feasible, in which case the idea has to be adjusted or canceled. Ultimately, this step leads to one 'best' recommendation idea to further develop.

The estimations on required effort, complexity and risk involved have to be translated to "business language" in the next step, step 4. This step comprises a cost-benefit analysis of the recommendation idea. Ideally, both the costs and benefits are expressed in monetary terms. Naturally, only ideas capable of delivering benefits should be further developed.

Once the cost-benefit analysis has been carried out and the benefits are expected to outweigh the costs, the actual design process can start (step 5). The goal of this phase is to work on the *recommendation algorithm*, to precisely define how recommendations are generated, what data sources are used and how problems and challenges are addressed. Next, the algorithm can be build (or compiled) and implemented (step 6). The last step, step 7, is aimed at monitoring the recommender system and measuring its performance. Results of the performance test can result in changes to either the recommender system or the recommendation idea.

5.2 Framework building blocks

The seven steps of the framework have been designed with the goal of creating a recommender system that delivers business value. The framework is a combination of existing project management, software development, requirement engineering approaches, and recommender system literature with some adjustments to create an optimal fit for the particular startup recommender system use case. These adjustments are partially described in this section, but also in the next section where the steps are separately described in detail. In other words, existing approaches, theory and literature with minor modifications form the foundation of the framework.

The whole process, meaning all the steps combined, is an iterative process partially inspired by the well-known software engineering process Rational Unified Process (RUP) (Kruchten, 2002) and the ISO12207 for software lifecycle processes. RUP has as goal to ensure the production of high-quality software that meets the needs of both the business and the end-users where the ISO12207 is the international standard for software lifecycle processes that defines tasks related to developing and maintaining software.

More specifically, the first two steps (business analysis and recommendation idea generation) are built upon the work of Arendsen (2010) on identifying both business and project objectives, formulating requirements and strategic alignment and planning. The third step (idea assessment, business alignment & idea selection) is also built on the work by the same author, but work by Hughes (2009) was used to clarify further how ideas could be assessed and evaluated (estimating effort and risk involved). This work by (Hughes, 2009) is partially inspired by the generally known PRINCE2 project management standard. Recommender system specific challenges and problems have an array of literature sources, which are referred to appropriately in the framework itself.

The cost-benefit evaluation, the fourth step, is based on the technique of Hughes (2009) for identifying and expressing software development costs. For the step regarding the design of the recommender system and addressing problems and challenges, multiple sources of literature and theory have been consulted. For building the recommender sytem architecture, theory on enterprise architectures by Ross, Weill, and Robertson (2006) was employed, as well as work by Hristakeva and Jack (2015) on identifying core recommender system components. Other relevant authors have been consulted in formulating interface design guidelines and guidelines for functional design (Ozok, Fan, and Norcio, 2010; Wax, 2008). Again, recommender specific problems and challenges are referred to in the framework.

Lastly, the build & implement and monitor & measure performance are built upon concepts from RUP, ISO12207 and theory by Hughes (2009) on development methods, testing and defining and measuring software quality.

5.3 Framework philosophy

The framework is based on the previously described building blocks, but it is also designed to the specific way most startups operate: creating a minimum viable product (MVP). The MVP way of thinking can also be used when designing a recommender system. Figure 8 briefly outlines this approach. Rather than creating partial products that do not serve any purpose on its own, meaningful smaller versions of a product should be created.

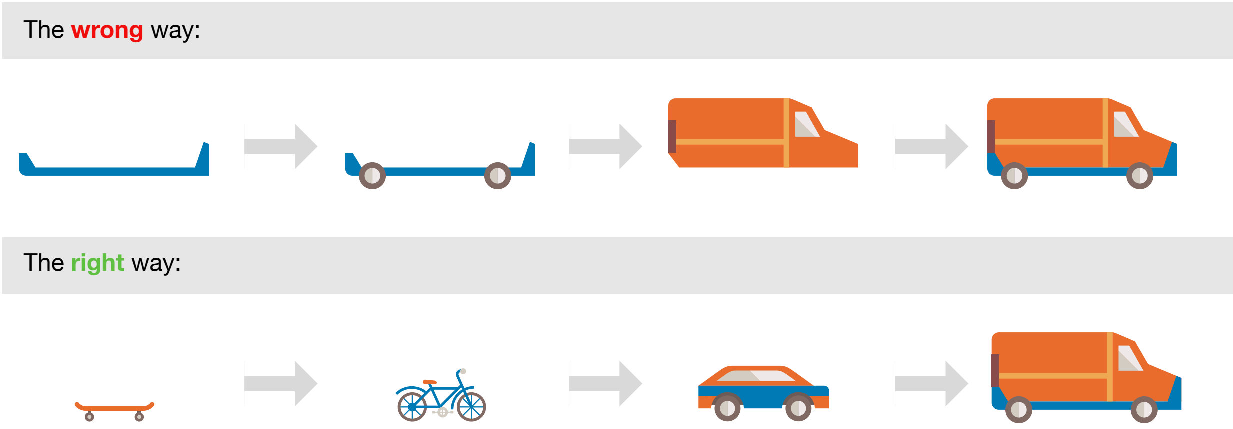

Figure 8: MVP approach

5.4 Outline of steps

This section contains a brief explanation of each step in the framework. For each step, the input, the actual process and the output are described, as well as the goal of the particular step.

5.4.1 Step 1: Analyzing the business model, goals & objectives

The first step does not require any input. The goal of this step is to analyze the business model in such a way that a solid foundation for the recommender system can be built. To ensure that the recommender system delivers business value, it has to be pointed out what business value is. Linking the strategy and relevant business goals and objectives is a way to assess if the recommender system delivers business value. Before that becomes possible, the strategy, business goals and business objectives need to be clear. A business goal is a statement of achievement, where a business objective usually is a specific action to achieve a goal. This step aims to support the startup in clarifying the business model by offering some exercises. These exercises are based on work by Ross, Weill, and Robertson (2006) but are tailored to the specific context of a startup:

- Identify the business model elements that distinguish the startup competitively
- Envision the (end) customer experience as it ought to be
- Decide in what terms the startup should grow
- List business goals and objectives

The sources of value creation as described in section 3.1.4 are explained in the guidelines and helps the startup to identify the sources for their business model. Also, examples of business goals are given to support the startup in finding and/or formulating relevant business goals. Examples of these business goals are: providing visitors the right information, create and maintain a high level of customer satisfaction or increase profitability per customer. The full list of examples and more details on what to exactly do in this step can be found in attachment B.

This first step results in a business foundation to further built upon. More specifically, this step helps startups in identifying relevant aspects of their business model, strategy, goals and objectives that will be used in other steps.

5.4.2 Step 2: Recommendation idea generation

The foundation built in the first step is used in the second step for generating recommendation ideas. The goal of the second step is formulate multiple recommendation ideas, regardless of their actual (financial) feasibility. During this step, ideas do not yet have to contain details on the exact algorithm or the exact implementation. By doing this, startups are challenged to think out-of-the-box and are forced to find out how recommender systems work and what potential they offer.

In the detailed description of this step in attachment B, startups are guided in the process of generating recommendation ideas. Different potential sources of ideas are given. For example, an exercise for the startup is to create a taxonomy of knowledge sources (Burke and Ramezani,

2011) as displayed in figure 9, which possibly identifies sources of data that can be exploited for a particular idea.

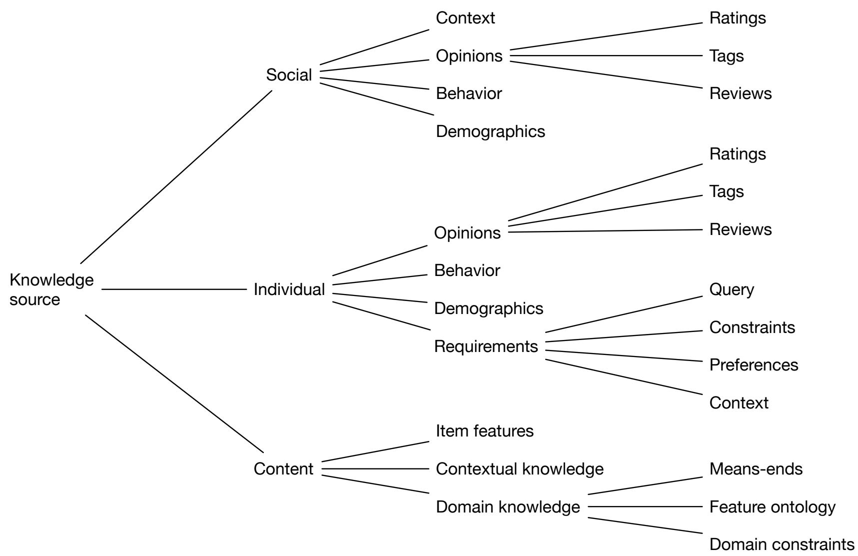

Figure 9: Taxonomy of knowledge sources (Burke and Ramezani, 2011)

Other possible sources of ideas are given as well. Different application models, as explained in section 4.6, can be a valuable source, since these application models are ready to be 'copied'. Existing literature and theories have to be considered and can form a large source of inspiration. The advantage of studying existing literature is that the level of understanding recommender systems and different approaches get increased. The same goes for existing examples or implementations by other (online) businesses: good for both inspiration and understanding. The detailed description of step 2 contains a list of relevant literature and theories, as well as a list of existing (online) examples of recommender systems.

Figure 10 graphically outlines the recommendation idea generation and already shows that only a few ideas can be taken to the next step. Realistically, having formulated multiple recommendation ideas, a basic level of understanding recommender systems is gained. A high-level assessment on what ideas are realistic and feasible should be carried once a sufficient amount of ideas has been generated. Then, only the ideas that might seem realistic and feasible are taken to the next step. As a result of this brief assessment, ideas might get adjusted or merged to become more realistic and feasible.

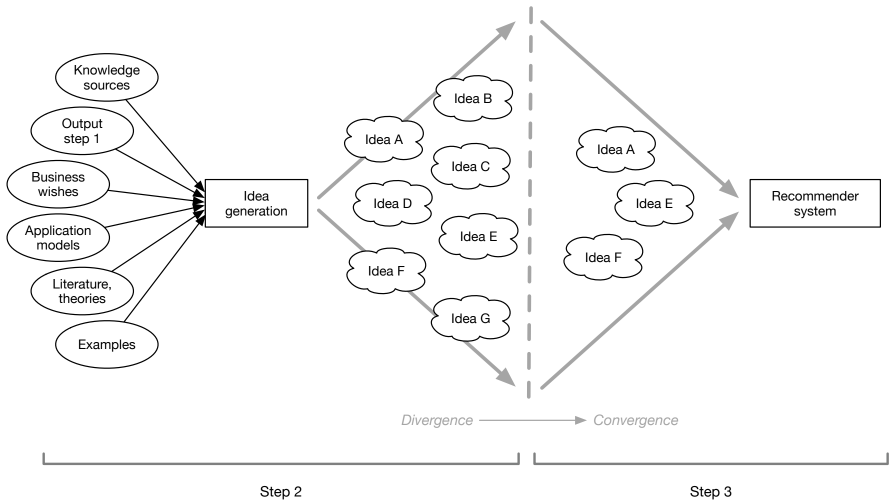

Figure 10: Recommendation idea generation

5.4.3 Step 3: Idea assessment, business model alignment & idea selection

The goal of the third step is to select the 'best' idea out of the list with recommendation ideas. It is important to select just one idea at this point. Developing multiple ideas at the same time might lead to an unbalanced focus resulting in multiple ideas that receive equal limited attention rather than one idea receiving all attention. Moreover, the high uncertainty surrounding startups, the low time-to-market, limited resources and the MVP way of working make it too risky to work and focus on more ideas at once.

To help the startup in finding the 'best' idea tools are offered. The definition of 'best' is dependent on the specific context of the startup, but ideally 'best' represents the optimal balance between added business value, effort, complexity and risk involved. Naturally, the costs involved plays a role here too but the financial aspects are handled in the next step.

First and foremost, the idea must be ensured to be truly adding business value. Therefore, the idea has to be aligned with the business model. Aligning the recommendation idea directly with the business model results in several benefits: reduced costs, increased responsiveness and agility and a higher valuation of the recommender system. Costs are reduced since the recommender system does not have to be (re-)adjusted later on in the process to ensure it is aligned with the business model (first-time-right, as early as possible, in terms of business model alignment). Responsiveness and agility are increased since the idea is aligned on a high level, resulting in a foundation that ensures that details are also aligned with the business model. The valuation of the recommender system is higher since it is directly known what business goals the system helps achieve, instead of finding out later what value it delivers.

A simple tool can be used for aligning recommendation ideas with business goals is a matrix as displayed in 3. In this example, the application models as described in section 4.6 are used as recommendation ideas.

	AM1 **Broad recommendation lists**	AM2 **Customer comments & ratings**	AM3 **Notification services**	AM4 **Product associated services**	AM5 **Deep personalization**
Goal 1	X				
Goal 2	X	X		X	
Goal 3		X			X
Goal 4		X	X		
Goal 5			X	X	
Goal 6	X			X	
Goal 7		X	X		X
Goal 8	X			X	X

Table 3: Business goal/recommendation idea (application model) matrix

To estimate the required effort, complexity and risk involved, a model was developed: the dimension model. The dimension model consists of eight dimensions, each consisting a set of criteria. Each criterium has to be considered and can be given a certain rating (or score) based on the level of relevance of that particular criterium. Once all criteria in a set have been considered and scored, the total score is calculated by summing each score. The total score determines the score of the dimension. This score provides insight in the complexity of a dimension. This complexity can then be used to base estimations on effort and risk on, on top of the regular estimations on effort and risk involved. The dimension model consists of the following eight dimensions:

- **Dimension 1: User Model (D1)**

  The user model partially defines the recommender system approach: whether it could be a content-based algorithm, a collaborative filtering algorithm or a hybrid approach. This dimension helps in finding factors related to users (data, features) that might increase the overall complexity. Key criteria in this dimension are: the amount of users in the dataset, information about user features, the expected willingness of users to rate items, the frequency of changing user preferences and the need for group recommendations (as opposed to individual recommendations).

- **Dimension 2: Item Model (D2)**

  Both the user model dimension and the item model dimension are the most vital parts of the dimension model. This dimension aims to estimate the complexity that comes from the items that the recommender system deals with. The item model also defines the recommender system approach and algorithm. Criteria used in this dimension are: amount of items in the dataset, information about item features, portfolio effect impact, item availability & scarcity and crowd avoidance impact.

- **Dimension 3: Domain Model (D3)**

  The domain model dimension represents factors that do not strictly fit the user and item dimensions but are important for generating accurate recommendations. Mostly, these factors come from the domain or the context of the recommender system. Examples of these factors could be: time, location, weather. Also, different domains could require different recommender system approaches: recommending music is different from recommending restaurants when using a content-based algorithm. Some domains might require explicit and specific domain knowledge that is unsuitable for a recommender system to use. The criteria in this dimension are: domain knowledge needed, (cross-)domain appliances and contextual information types.

- **Dimension 4: Performance (D4)**

  Any software solution needs to have a a certain level of performance: users do not want to wait too long for a system to respond. There are certain factors that could potentially decrease the performance of the recommendation generation, such as multiple algorithms and real-time recommendations. This dimension aims to clarify to what extent the performance could decrease beyond a reasonable level by checking if features are present that usually decrease performance. Examples of criteria in this dimension are: need for real-time generated recommendations, live/offline recommendation generation and the degree of adaptability importance.

- **Dimension 5: Quality of outcomes (D5)**

All recommendations need to have a certain level of quality, as they must at least satisfy the expectations of a user. This dimension aims to point out the importance of the quality of the recommendations, as in some cases, the quality needs to be different than usual. To achieve this goal, this dimension contains the following criteria: perceived level of accuracy, novelty & diversity, overall users' satisfaction and the amount of salience recommendations wanted.

- **Dimension 6: Architecture & Infrastructure (D6)**

  In order to enable an efficient and effective recommender system, both the architecture and infrastructure have to be designed carefully and thoughtfully. When designing this, certain decisions have to be made on for example scalability and agility. This dimension helps in finding factors that might make the architecture & infrastructure more complex. This dimension contains the following criteria: required level of scalability, amount of source databases & other components, impact of calculations on production database, need for data warehouses, centralized or decentralized recommender system and the amount of devices to be supported.

- **Dimension 7: Interface & User Experience**

  The interface and user experience play an important role in recommendation acceptance. This particular dimension has as goal to determine to what extent the interface and user experience play a role in the design process and the recommender system. Key concepts as sufficiency, transparency and flexibility are explained in the detailed description. Criteria in this dimension are: importance of ease of use, required level of sufficiency, the need for transparent and explainable recommendations, the need for flexibility after initial recommendation and the need for accessibility features.

- **Dimension 8: Privacy & Security**

  Privacy and security are becoming increasingly important in any software solution. When speaking of personalized recommendations, it might become a tad extra important since personal information about users is involved. This dimension helps in finding out if privacy and security factors play a role beyond the standard and reasonable level of protecting privacy and having certain security measurements. To achieve this, the dimension contains the following criteria: level of required privacy preservation, perceived level of exposure risk, perceived level of trust and social/environmental responsibility.

Attachment B: Recommender System Design Framework contains a comprehensive table containing all the eight dimensions and the corresponding criteria. Table 4 outlines the criteria that belong to the first dimension (the user model dimension), to illustrate how the dimension scores are generated. The framework also gives a textual explanation of each criterium, making it as easy as possible for startups to give scores.

1	Amount of users in dataset	low (<1K users) (+3), medium (>1K, <10K users) (+1), high (>10K users) (+0)
2	Information about user features	none (+2), limited (<2 features) (+1), detailed (>2 features) (+0)
3	Expected willingness to rate items	low willingness (+2), basic willingness (+1), high willingness (+0)
4	Frequency of changing user preferences	long-term stable (+0), periodic changes (+1), highly frequent (+2)
5	Need for group recommendations rather than individual recommendations	not needed (+0), needed (+2)

Table 4: Dimension 1: User Model criteria

The dimension scores in total do not have an actual importance, meaning that it can not be said that if scores are higher than, for example, 3, the idea is not feasible. The different scores rather aim to outline the level of complexity and focus a dimension should receive. The required effort, complexity and risk involved are important factors as they can be used to compare different ideas. The scores can be plotted in a radar chart, as figure 11 outlines.

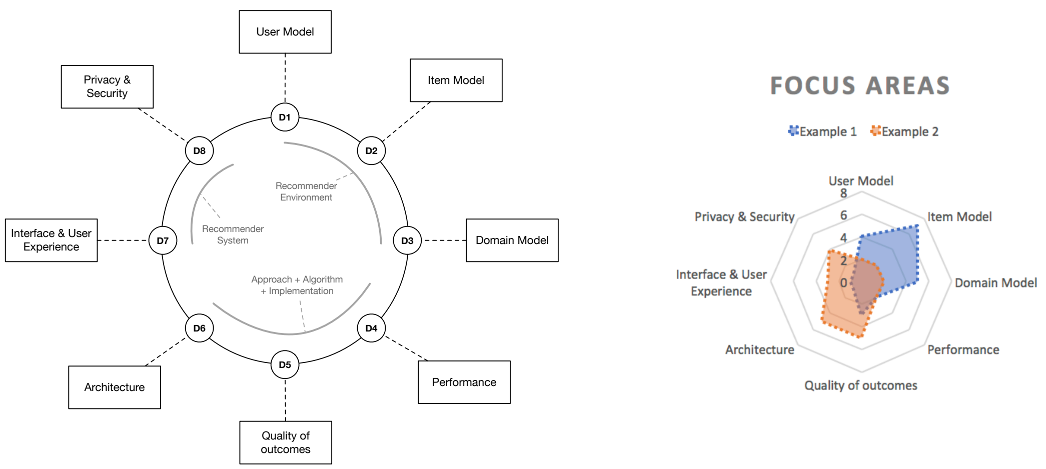

Figure 11: Dimension model and radar chart

Once the ideas are assessed separately, it is possible to select the best idea: the idea that has the best balance between added business value, effort, complexity and risk involved.

5.4.4 Step 4: Evaluation: cost-benefit analysis

Before the selected idea can be developed, the financial feasibility has to be tested using a cost-benefit analysis. This step emphasizes that while a certain idea might be good, if it is too costly to develop, it might be worth postponing it. The goal of this step is to find out whether an idea truly results in benefits that outweigh costs, to avoid finding that out during development. The cost-benefit analysis is not part of the previous step because it is too labour intensive to perform this analysis for each single idea.

In this step, startups are challenged to create a cost-benefit analysis consisting out of two steps. First, the task is to identify all the costs and benefits of designing, operating and maintaining a recommender system. Examples are given to make this process easier for the startup. Second, the most challenging task has to be carried out: expressing all the costs in monetary and quantitative terms. This is particularly challenging since potential benefits might be hard to express in monetary terms. How does one measure increased customer intimacy for example? Again, examples of these benefits are given. Also, suggestions for standard packages too keep costs limited are given.

Ultimately, this step produces enough evidence to make a decision on whether to further develop the idea or not. Purely from a financial perspective, costs might outweigh the financial benefits, as long as the non-financial expected returns are worth the investment. In other words: the recommender system might cost money if benefits in other terms are gained.

5.4.5 Step 5: Design & address problems and challenges

When the expected benefits of the recommendation idea outweigh the costs, the system can be designed. This step provides guidance in designing the recommender system and addressing problems and challenges. Part of this step is to justify the chosen filtering methods in the recommendation idea. The startup is offered several tools that help build a solid foundation to ensure the recommender system helps achieving the business goals.

Startups are asked to define at least one *central question* for the system to ensure focus. This might sound trivial, but it is tempting to add features that, when looked at properly, do not really add value. Especially within the context of startups and MVP-thinking, only features need to be added that truly add business value. Next, three angles on design are explained: functional, interface and architectural.

- **Functional design** focuses on the functionalities or tasks the system will perform. At the root of every piece of design lies a function or task. Guidelines are provided for optimizing the offered functions. Examples of these guidelines are: considering the goal of the system, considering the target audience, clear how-to-use, ensuring the recommender system is engaging.
- **Interface design** provides ten guidelines by Ozok, Fan, and Norcio (2010) that helps in considering four interface elements: *sufficiency*, *transparency*, *flexibility*, and *accessibility*. These elements need to be satisfied for users. Examples of the guidelines for interface design are: present the name, price and thumbnail picture of the product, present short and concise product information, only display recent comments etc.
- **Architectural design** is mainly focused what kind of components the recommender system has and needs, and how these components are interrelated. Components can be business objects, application objects, information objects or technology objects. Five core components (Hristakeva and Jack, 2015) are explained that help in building a basic architectural design of the recommender system. These core components are displayed in figure 12.

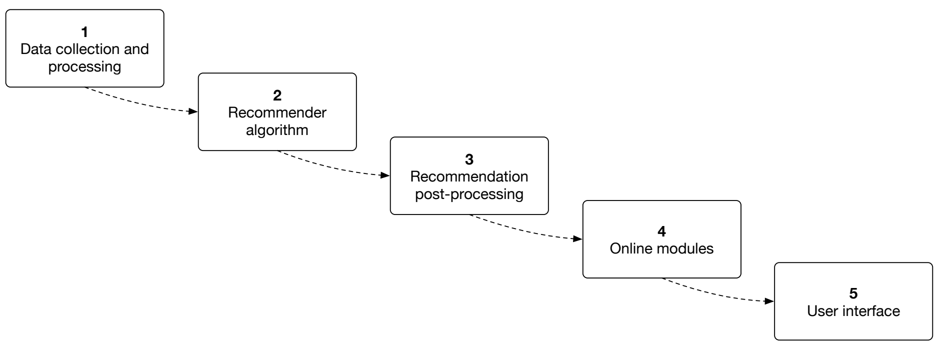

Figure 12: Recommender system core components (Hristakeva and Jack, 2015)

The outcomes of the dimension model are considered in this step too. It is likely that some of the problems have to be addressed in either the functional, interface or architectural design aspects. This step results in a comprehensive package of information and designs that can be used to start the actual development of the recommender system.

5.4.6 Step 6: Build, test & integrate

After designing the recommender system in functional, interface and architectural terms, the system is ready to be built. Once it has been built, the system has to be tested and integrated into the existing infrastructure. Startups might not have an proper infrastructure already, but the recommender system has to be linked with the data sources at least.

Different options are explained: finding an off-the-shelf package, utilizing proven code libraries or writing code from scratch. Obviously, the latter seems a bad idea since it might cost too much. Either way, the goal of this step is to have a recommender system that is capable of providing personalized recommendations. Not only does this include the development of the recommender system, it also mentions the fact that other software components might need updates too (website, app, etc.).

Startups are also reminded of the fact that dependencies of the recommender system have to be ready too. This might include things such as access to the right data and/or knowledge sources. If a certain API needs to be utilized, the access to this API needs to be allowed. Another example is when users are asked for explicit ratings to avoid the cold-start problem. It might take a while before these ratings are collected. During this time, the recommender system might not be ready to fully function.

The testing aspect of this step is merely focused on the software running properly, without any errors. Tests in terms of statistical accuracy or effectiveness using dummy data should only be carried out to see if the system does not behave unexpectedly. Individual components have to be unit tested, as well as the recommender system as a whole using e2e (end-to-end) tests.

5.4.7 Step 7: Metrics & monitoring

Once the system has been built and ready to use in production, the system should be tested using one or more of the described quality metrics in section 4.7. The guidelines aim to clarify to some extent what metrics should be used in different cases. The goal of this step is two folded. On the one hand, the quality of the recommender system needs to be tested. Recommendations have to be accurate and meet the expectations of users. On the other hand, a baseline for future improvements has to be set.

Ideally, the recommender system functions properly at the first run: users are given accurate recommendations. But in practice, recommender systems need to be gradually tested and improved. Recommendations might not be that accurate in the beginning and modifications to either the algorithm or the design have to be made.

An exceptional suitable method for testing recommender systems is A/B testing. A/B testing, or split testing, is a method where two versions of a software or system are compared against each other to determine which one performs better. Minor changes to the recommender system can be tested on different (smaller) user groups. When these changes lead to positive results (e.g. higher accuracy, higher user satisfaction), the change can be rolled out to the whole user base.

5.5 Continuous improvement cycle

The last three steps in the framework can and should ideally be repeated based on the performance of the system. As already written in the description of the last step, a recommender system is likely to need adjustments once it first has been launched. Recommendations might need to become more accurate or more diverse, the performance needs to be improved or users might provide certain feedback that requires changes to the system.

After releasing the first version, startups are asked to create a baseline. This baseline can be used to test whether future improvements actually have a (positive) result. Quality metrics that can be used for the baseline are described in the quality metrics section (section 4.7).

6 Framework validation & calibration process

This section aims to answer the questions on how the framework can be used in practice and what potential benefits the framework offers. This section also aims to outline how the framework was tested and validated.

6.1 Usage of the framework

Ideally, one or two persons at a startup read attachment B: Recommender System Design Framework for Startups and use it as a reference during the whole design process. Ultimately, when the steps in the framework are followed appropriately, a fully functioning recommender system or certain features are set in place. The intention of the framework is that it does not require external consultancy or other forms of help to design the recommender system. Naturally, other knowledge sources (theories, literature, etc.) can and should be employed, but no other help in terms of experts or professionals is needed.

6.2 Benefits of the framework

The proposed framework has been designed to make the process of designing, operating and maintaining a recommender system more efficient and less time consuming. First of all, in line with the innovation gap, most startups do not have the knowledge to get started with offering personalized recommendations. By following the steps of this framework, a basic level of understanding is gained.

Using the proposed framework helps in keeping the design process organized. Since the framework is partially based on proven software project management practices, the benefits that these practices deliver are also gained. Examples of these kind of benefits are transparency, early and predictable delivery, predictable planning and costs and focus on both business and customer value. Furthermore, by breaking down the design process into manageable steps, the startup can focus on high quality development.

6.3 Framework tests & validation

First, a 1.0 version of the framework was developed using the outcomes of the literature study. After this initial version was developed, it was tested and validated using the BarDoggy case study. This case study was carried out by the author of this thesis and by the founder of BarDoggy. The framework was further improved while it was being tested and validated. After three months, when BarDoggy went through several framework iterations, a 1.1 version of the framework was defined. The next chapter, chapter 7, contains a summary of the test and validation process and the outcomes of this case study. This chapter also contains a *changelog* on what was changed as a result of the case study. This *changelog* is written in the *framework adjustments* section (section 7.3).

Next, the 1.1 version of the framework was further tested and validated using the Analytics for Learning case study. This case study, carried out by two persons that were not directly involved in the development of the framework, is summarized in chapter 8. As with the BarDoggy case study, this chapter contains information on the case study test and validation process, the outcomes and the framework adjustments. These adjustments were implemented in the framework and the version was updated to 1.2.

The most recent version of the framework at time of writing is version 1.2. This version is the version that can be found in the previous chapter. This version was further validated using an interview with an expert in the field of the practical usage of recommender systems. The person interviewed was asked specifically on his opinion on the framework, especially regarding accuracy, completeness and usefulness. A summary of this interview is written in chapter 9.

9 Expert interview

Next to the case study, the proposed framework has been validated through an expert validation. The combination of use cases and expert validation can be considered triangulation. The expert that was interviewed is Niels Basjes, Lead IT-Architect Scalable Solutions at Bol.com. Bol.com is the biggest online retailer in the Netherlands in terms of revenue (2015). Niels Basjes plays an important role in developing and maintaining the recommender system used by Bol.com. The full summary of the interview can be found in attachment D: Expert Interview. This section contains the main conclusions of the interview.

Regarding the innovation gap, while it is true that many literature is written on recommender systems and different approaches exist, it is also true that there are quite a few open source options and off-the-shelf packages available. Organizations, and startups in particular can simply pick one of these options and have a recommender system up and running without any hassle or trouble. By simply picking an existing solution, all steps in the framework can be skipped.

The true usability of the proposed framework is dependent on the type of startup. Offering personalized recommendations has to be a core business process, or the business has to be dependent on personalized recommendations to a large extent. When the startup needs a custom algorithm right from the beginning, the proposed framework can be helpful. The high level outline seems to offer sufficient guidance in the process of designing a recommender system. However, when no custom algorithm is needed, or a recommender system is merely a small 'nice to have' feature, the startup could and should choose a free, open-source or off-the-shelf solution.

Regardless of whether the startup uses an open-source, off-the-shelf of in-house developed approach, the system should be decoupled from other components. The reason for this is that when the chosen approach is not capable of supporting the recommendation process anymore, it can be replaced easily.

Also, the usage of objective statistical metrics is particularly useful for academic settings. Bol.com uses A/B tests to check whether improvements to their recommender algorithm has the desired impact. Usage of objective statistical metrics in practical terms for startups does not add value. The business value approach of this research looks promising.

10 Discussion & limitations

Currently, one minor limitation is that the framework is focused on startups specifically. Other businesses could likely benefit from this framework too. Yet, the framework has to be tested on non-startup use cases.

Furthermore, this research is partially based on the assumption that startups have difficulties in designing, operating and maintaining a recommender system. The process costs too much valuable time and money. But is it really true that startups need guidance in this process? The validation, especially the case studies, prove that the framework is valuable: it helped the startups during the design process and in justifying certain design choices. The interviewed expert questions the true usability for all use cases, but acknowledges that for some specific cases, the framework might be of value. In his opinion, most startups could simply use an existing and off-the-shelf solution. However, startups that require a customized recommender system, could benefit from following the proposed approach.

Assuming that most startups do have a hard time in designing, operating and maintaining a recommender system, is the proposed framework the only solution to overcome their struggles? The answer to this question is likely no. There are other project or software management tools and theories that could be used. However, this framework is specifically tailored to startups wanting to offer personalized recommendations. For startups, the framework offers an objective methodology to design, operate and maintain the most correct recommender system for their use case. The framework justifies certain design choices and eliminates possible biases and prejudice.

At time of writing this thesis, there are no other methods or approaches known for helping businesses develop a recommender system purely from a business value perspective. Combining the business value with the specific context of a startup is unique. Naturally, startups could employ any regular software project management approach (Prince2, Scrum, RUP) but these approaches do not explicitly take recommender systems and the specific startup context into account.

Another discussion point is that the framework, the dimension model in particular, is likely to need continuous fine tuning. The more the model is used, the better it can be tuned.

The dimension model needs constant updates in order to support new theories and concepts. Future research is definitely needed to provide a full, up-to-date and sound recommendation idea estimation tool for startups.

On the one hand, the framework provides guidance on the recommender design process from start to end. On the other hand, the framework can serve as a learning tool for the persons involved in the design process. Arguably, not using the framework and 'finding things out' from scratch could be a great lesson too. However, as assumed earlier, startups do not always have time and resources to base a recommender system implementation on trial and error. Furthermore, the framework serves as a justification: it eliminates potential biases. Biases are likely to be present when the persons involved start 'finding things out' themselves.

11 Conclusion

Over the last twenty years, recommender systems have been vastly studied and nearly revolutionized the marketing and delivery of content by providing personalized recommendations over a broad range of product offerings. The ever increasing amount of online eco-systems and more demanding customers force businesses to rethink the way products and services are offered. Personalized recommendations might become the standard for every (online) business.

Startups are newly established businesses mostly focused on disrupting a market or industry while working in conditions of extreme uncertainty. This results in the need for startups to deliver products or services to its customers as fast as possible via minimum viable products, to get feedback and further improve the product or service.

Startups that recognize the potential of offering personalized recommendations could use recommender systems. This research identified certain challenges and problems that startups face when wanting to offer personalized recommendations. Besides the challenges and problems, designing, operating and maintaining costs a certain amount of resources. This issue is what is in this thesis referred to as the *innovation gap*. To close this gap, the following main question was raised: how can startups be helped in overcoming problems and challenges when designing, operating and maintaining a recommender system? In order to answer the main questions, the subquestions have to be answered first.

1. What type of techniques or algorithms do organizations use to provide personalized recommendations?

In able to understand and analyze the problems and challenges regarding designing, operating and maintaining a recommender system, the question on what type of algorithms or techniques exist first has to be answered. This thesis has identified five traditional algorithms that recommender systems might use: content-based, collaborative filtering, demographic, utility-based and knowledge-based algorithms.

In recent years, these five traditional algorithms are combined to overcome individual drawbacks and achieve peak performance. The combination of two or more algorithms is what is called a hybrid approach. Hybrid approaches exist in different forms: weighted, switching, mixed, feature combination, feature augmentation and meta-level. Computational intelligence and social network based approaches also gain more popularity in recent years. Computational intelligence uses sophisticated data mining techniques such as artificial neural networks and genetic algorithms to calculate similarities.

Currently, organizations use different recommender system implementations and algorithms, but the most common and effective are hybrid approaches.

2. What criteria can be used to describe the distinct capabilities of different recommender system approaches?

The decision on which of the five algorithms or a combination of those for a specific use case is dependent on several different factors. For example, when geographical or physical space limitations are present, the issue of crowd-avoidance has to be considered. The most important deciding factor is the available knowledge and data sources. Yet, all recommender systems share the same goal: providing personalized recommendations to have the end-user find interesting or useful products or services in a space of possible options. Therefore, outlining the distinct capabilities in terms of algorithmic accuracy or usefulness is less relevant in the context of helping startups designing, operating and maintaining a recommender system.

Distinct capabilities that truly help startups in the process of designing, operating and maintaining a recommender system are captured in the *dimension model*. This model contains eight dimensions: (1) user model, (2) item model, (3) domain model, (4) performance, (5) quality of outcomes, (6) architecture, (7) interface & user experience and (8) privacy & security. Each dimension contains several criteria, that should be considered when the recommender system is designed. These dimensions can be seen as criteria, as they help in finding what algorithms are useful in what context.

3. How do organizations currently create (business) value from offering personalized recommendations?

Recommender systems are already becoming an integrated part of most e-commerce websites. Customer demands are increasing continuously, resulting in increased expectations of the services businesses can deliver. Organizations having recommender systems in place often offer one of the following five application models to create business value: (1) broad recommendation lists, (2) customer comments and ratings, (3) notification services, (4) product-associated services or (5) deep personalization. These application models aim to enrich the overall user experience. Thus, business value is created by offering a personalized, streamlined and enriched user experience. Thereby, the organizations might reap benefits because browsers are converted into buyers, cross-sell opportunities increase and intimate customer relationships are built.

4. What problems and challenges do startups specifically face in offering personalized recommendations to their customers?

Startups face challenges varying from finding sources of capital, coping with uncertainty to having the right knowledge and experience. While the potential value of personalized recommendations is often acknowledged, startups lack knowledge and experience in designing, operating and maintaining a recommender system within a reasonable amount of time with an acceptable amount of resources. For one, there are many different recommender system approaches and the number of available resources (literature and theories) is huge. Assumably, the process takes too long, costs too much and might result in a system that is not aligned with the business model. Thus, the main challenge for a startup is developing a recommender system that delivers value, fitting the MVP-cycle with a limited amount of resources.

Besides the main challenge, designing, operating and maintaining a recommender system introduces various other challenges. From coping with the cold-start problem to the portfolio effect and other use-case specific issues. The whole process is truly complex and needs to be carried out carefully.

5. What solutions can help address the problems and challenges that startups face?

Given the main challenge for startups, that is, developing a recommender system that delivers business value within a reasonable amount of time with limited knowledge and resources, this study suggests a framework that supports startups. The framework provides guidance from beginning to end, starting with defining business value and ending with monitoring and measuring the performance of the recommender system. During the whole process, startups are given tips, tools, and models that aim to make the design, operation and maintenance of a recommender system less cumbersome and more clear. Following the framework ensures the development of a recommender system that delivers business value. The framework has sorted out basic concepts and steps to follow, which saves time and thus resources for the particular startup.

Finally, the answers on the subquestions already answer the main question. Startups having difficulties with designing, operating and maintaining a recommender system, can use the proposed framework to save time and resources. The framework offers a business value driven approach, ensuring that the recommender system eventually is a meaningful software asset. The framework has yet to be further tested on multiple case studies. Analysis on the existing case study already proves that the framework is valuable to a certain extent.

Future research in the direction of a business value approach might go towards an approach which offers more detail and guidance in what algorithms could be used in different cases. The usefulness of the framework for non-startups could also be further researched, as this framework is likely to be valuable in other business cases. Lastly, the framework could be translated to an audit tool, to assess existing recommender systems on whether they truly add business value.

References

Adomavicius, Gediminas and Alexander Tuzhilin (2010). "Context-Aware Recommender Systems". In:

Arendsen, Martin (2010). *Succes met de requirements! : ontwikkeling, validatie en beheer van requirements voor informatiesystemen.* Den Haag: Academic Service. ISBN: 9789012582056.

Bao, Jie, Yu Zheng, and Mohamed F Mokbel (2012). "Location-based and Preference-Aware Recommendation Using Sparse Geo-Social Networking Data *". In:

Beel, Joeran and Stefan Langer (2015). "Research Paper Recommender System Evaluation : A Quantitative Literature Survey". In: 20.April 2013. DOI: `10.1007/s00799-015-0156-0.Research`.

Beel, Joeran, Corinna Breitinger, Stefan Langer, Andreas Lommatzsch, and Bela Gipp (2016). "Towards reproducibility in recommender-systems research". In: *User Modeling and User-Adapted Interaction* 26.1, pp. 69–101. ISSN: 15731391. DOI: `10.1007/s11257-016-9174-x`.

Belkin, Nicholas J and W Bruce Croft (1992). "Information filtering and information retrieval: Two sides of the same coin?" In: *Communications of the ACM.*

Bobadilla, J, F Ortega, A Hernando, and A Gutiérrez (2013). "Recommender systems survey". In: *Knowledge-Based Systems* 46, pp. 109–132. DOI: `10.1016/j.knosys.2013.03.012`.

Burke, Robin (2002). "Hybrid Recommender Systems: Survey and Experiments". In: *User Modeling and User-Adapted Interaction* 12.4, pp. 331–370.

Burke, Robin and Maryam Ramezani (2011). "Matching Recommendation Technologies and Domains". In: *Recommender Systems Handbook.* Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor. Boston, MA: Springer US, pp. 367–386. ISBN: 978-0-387-85820-3. DOI: `10.1007/978-0-387-85820-3_11`.

Chen, Pei-Yu, WuShin-yi, and Jungsun Yoon (2004). "The Impact of Online Recommendations and Consumer Feedback on Sales". In: *ICIS 2004 PROCEEDINGS* 2004.

Choudary, Sangeet Paul, Marshall W Van Alstyne, and Geoffrey G Parker (2016). *Platform Revolution: How Networked Markets Are Transforming the Economy–And How to Make Them Work for You.* 1st. W. W. Norton & Company. ISBN: 0393249131, 9780393249132.

Consortium for IT Software Quality. *Code Quality Standards.* URL: `http://it-cisq.org/standards/`.

Cremonesi, Paolo, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos, and Roberto Turrin (2011). "LNCS 6948 - Looking for Good Recommendations: A Comparative Evaluation of Recommender Systems". In: *LNCS* 6948, pp. 152–168.

Edmunds, Angela and Anne Morris (2000). "The problem of information overload in business organisations: a review of the literature". In: *International Journal of Information Management* 20.1, pp. 17–28. ISSN: 02684012. DOI: 10.1016/S0268-4012(99)00051-1.

Evers, Natasha (2003). "The Process and Problems of Business Start-Ups". In: *The ITB Journal The ITB Journal ITB Journal Issue Number* 4.7.

Falk, Kim (2017). *Practical Recommender Systems*. ISBN: 9781617292705.

Francesco, Ricci, Rokach Lior, Shapira Bracha, and Kantor Paul B. (2011). *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Boston, MA: Springer US, pp. 0–845. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3. arXiv: arXiv:1011.1669v3.

Garfinkel, Robert, Ram Gopal, Bhavik Pathak, Rajkumar Venkatesan, and Fang Yin (2006). "Empirical Analysis of the Business Value of Recommender Systems". In:

Ge, Mouzhi, Carla Delgado-Battenfeld, and Dietmar Jannach (2010). "Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity". In:

Gorakala, Suresh (2016). *Building recommendation engines: understand your data and user preferences to make intelligent, accurate, and profitable decisions*. ISBN: 9781785883538.

Gunawardana, Asela, Guy Shani, and Shanigu@bgu Ac Il (2009). "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks". In: *Journal of Machine Learning Research* 10, pp. 2935–2962.

He, Jianming and Wesley W Chu (2010). "A Social Network-Based Recommender System (SNRS)". In: *Data Mining for Social Network Data*. Ed. by Nasrullah Memon, Jennifer Jie Xu, David L Hicks, and Hsinchun Chen. Boston, MA: Springer US, pp. 47–74. ISBN: 978-1-4419-6287-4. DOI: 10.1007/978-1-4419-6287-4_4.

Herlocker, Jonathan L, Joseph A Konstan, and John Riedl (2000). "Explaining Collaborative Filtering Recommendations". In:

Hevner, A R, S T March, J Park, and S Ram (2004). "Design science in information systems research". In: *MIS Quaterly* 28.1, pp. 75–105. DOI: 10.2307/25148869.

Hou, Lei, Kecheng Liu, Jianguo Liu, and Runtong Zhang (2017). "Solving the stabilityaccuracydiversity dilemma of recommender systems". In: *Physica A: Statistical Mechanics and its Applications* 468, pp. 415–424. ISSN: 03784371. DOI: 10.1016/j.physa.2016.10.083.

Hristakeva, Maya and Kris Jack (2015). *The Components of a Recommender System*. URL: https://buildingrecommenders.wordpress.com/2015/11/10/the-components-of-a-recommender-system/.

Hughes, Bob (2009). *Software project management*. London: McGraw-Hill Higher Education. ISBN: 9780077122799.

Kruchten, P (2002). "Tutorial: introduction to the Rational Unified Process/sup /spl reg//". In: *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, pp. 703–.

Kunegis, Jérôme, Alan Said, and Winfried Umbrath (2009). "A Recommender System for Semantic Networks". In:

Lu, Jie, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang (2015). "Recommender system application developments: A survey". In: *Decision Support Systems* 74, pp. 12–32. DOI: 10.1016/j.dss.2015.03.008.

Ozok, A. Ant, Quyin Fan, and Anthony F. Norcio (2010). "Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: results from a college student population". In: *Behaviour & Information Technology* 29.1, pp. 57–83. ISSN: 0144-929X. DOI: 10.1080/01449290903004012.

Panniello, Umberto, Alexander Tuzhilin, Michele Gorgoglione, U Panniello, M Gorgoglione, and A Tuzhilin (2014). "Comparing context-aware recommender systems in terms of accuracy and diversity 1 Motivation and introduction". In: 24, pp. 35–65. DOI: 10.1007/s11257-012-9135-y.

Park, Deuk Hee, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim (2012). "A literature review and classification of recommender systems research". In: *Expert Systems with Applications* 39.11, pp. 10059–10072. ISSN: 09574174. DOI: 10.1016/j.eswa.2012.02.038.

Porter, Michael E (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: Free Press.

Radhakrishnan, Abirami, Xingxing Zu, and Varun Grover (2008). "A process-oriented perspective on differential business value creation by information technology: An empirical investigation ". In: *Omega* 36, pp. 1105–1125. DOI: 10.1016/j.omega.2006.06.003.

Ries, Eric (2011). *The lean startup : how today's entrepreneurs use continuous innovation to create radically successful businesses.* ISBN: 9780670921607.

Ross, Jeanne W, Peter Weill, and David C Robertson (2006). *Enterprise architecture as strategy : creating a foundation for business execution.* Cambridge, MA: Harvard Business School Press. ISBN: 1-591-39839-8=978-1-591-39839-4.

Safran, Mejdl and Dunren Che (2017). "Real-time recommendation algorithms for crowdsourcing systems". In: DOI: 10.1016/j.aci.2016.01.001.

Schafer, J Ben, Joseph A Konstan, and John Riedl (2001). "E-Commerce Recommendation Applications". In: *Data Mining and Knowledge Discovery* 5, pp. 115–153.

Schumpeter, J.A. (1934). "The Theory of Economic Development: An Inquiry into Profits, Capital, Credit, Interest and the Business Cycle". In: 3.2, pp. 2068–317.

Storey, D J (1985). "THE PROBLEMS FACING NEW FIRMS [1]". In: *Journal of Management Studies* 22.3, pp. 327–345. ISSN: 1467-6486. DOI: 10.1111/j.1467-6486.1985.tb00079.x.

Wax, Dustin M. (2008). *7 Essential Guidelines For Functional Design.* URL: https://www.smashingmagazine.com/2008/08/7-essential-guidelines-for-functional-design/ (visited on 07/01/2017).

Yera, Raciel, Jorge Castro, and Luis Martínez (2016). "A fuzzy model for managing natural noise in recommender systems". In: *Applied Soft Computing Journal* 40, pp. 187–198. ISSN: 15684946. DOI: 10.1016/j.asoc.2015.10.060.

Zheng, Yu, Lizhu Zhang, Xing Xie, and Wei-Ying Ma (2009). "Mining interesting locations and travel sequences from GPS trajectories". In: *Proceedings of the 18th international conference on World wide web - WWW '09.* New York, New York, USA: ACM Press, p. 791. ISBN: 9781605584874. DOI: 10.1145/1526709.1526816.