FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Recommender Systems in the Domain of Early-stage Enterprise Investment

Evaluation in the context of recommendation systems

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Ing. Johannes Luef, BSc
Registration Number 0828182

Ing. Dipl.-Ing. Christian Ohrfandl, BSc
Registration Number 0926341

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Hannes Werthner
Assistance: Dimitris Sacharidis, MSc, PhD

Vienna, 31st January, 2019

Johannes Luef	Hannes Werthner
Christian Ohrfandl	Hannes Werthner

Declaration of Authorship

Ing. Johannes Luef, BSc
Stögersbach 27, 8241 Dechantskirchen

I hereby declare that I have written this thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 31st January, 2019

Johannes Luef

Declaration of Authorship

Ing. Dipl.-Ing. Christian Ohrfandl, BSc
Keißlergasse 18a/2/4, 1140 Wien

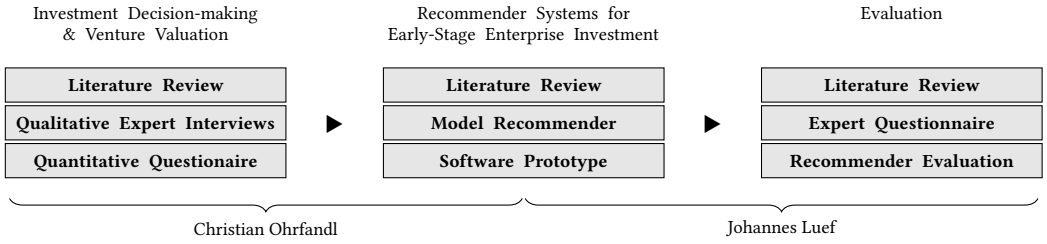
I hereby declare that I have written this thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 31st January, 2019

Christian Ohrfandl

Distinction of Joint Work

The present research is conducted as joint work between Johannes Luef and Christian Ohrfandl. Each author elaborates a specialization topic including individual research questions. Additionally, chapter 2 *Recommender Systems for Early-Stage Enterprise Investment* is jointly constructed by both authors. The results are composed as two separate theses, each assigned a different subtitle.



Acknowledgements

First of all, I would like to express my appreciation and gratitude to my advisor Professor Hannes Werthner, for supervising this master thesis and giving me the opportunity to pursue research in the domain of *Recommender Systems*.

Also, I want to thank Dimitris Sacharidis, who excellently assisted me through this journey with beneficial endless discussions about the domain of recommender systems and about the thesis itself.

Furthermore, it was a great pleasure for me to take this very journey with my co-author Christian Ohrfandl. I want to thank him for his brilliant cooperation on elaborating the shared chapter. Furthermore, I am pleased to have gained Christian as a really close friend.

I also feel immeasurable thankfulness for the endless support I receive from my family, not only during the process of writing my master thesis, but throughout my whole life in general. A special thanks goes to my father Johannes, guiding me on my way and my mother Aloisia, who consistently made my life easier in so many different ways. However, I am also grateful to the rest of my family, helping me to take this journey.

Furthermore I would like to thank all the people that participated in the expert questionnaire. Without their devotion to the demanding experiment, the research would not have been completed.

Last but not least, I want to thank my beloved girlfriend Maria Anna, for her unresting support and for being so incredibly patient and helpful, even during long nights of programming and writing on this thesis.

Abstract

The main objective of this thesis consists of the design of a recommender system, representing a novel method concerning the computational recommendation of early-stage enterprises to investors. In order to quantify decision rules the recommender system is based on, investors' requirements and behaviours need to be analysed utilizing qualitative- and quantitative research. Furthermore, demonstrating the behaviour of the proposed recommendation algorithms is a major task of this thesis. For this reason, a prototype of the recommender system is being crafted in software.

Based on the results of Christian Ohrfandl's[1] specialization topic, it can be concluded that the most important characteristics investors base their investment decisions on, are stated as the quality, size and composition of the management team, product- & public interest and the industry / market sector of an early-stage enterprise. Furthermore, the venture valuation methods most utilized by investors, most meaningful in terms of valuation quality in the context of early-stage enterprises and most beneficial when utilized in a recommendation system, are stated as the scorecard- and berkus methods. Finally, investors' requirements among the functionality of a recommender system in the domain of early-stage enterprise investment may be concluded as the construction of an investor profile.

The shared chapter 2 *Recommender Systems for Early-Stage Enterprise Investment* addresses the conceptualization of a recommendation system in the domain of early-stage enterprise investment based on the findings of co-author Christian Ohrfandl's specialization topic.The resulting recommender system includes various types of recommenders in a parallelized approach, that is, Collaborative filtering, content-based-, knowledge-based-, social- and hybrid recommendation algorithms. Additionally, the conceptual model of this recommender system has been implemented as a highly scalable, plugin-based software prototype that may be easily extended by different recommendation algorithms in future work.

The most important opportunity for future research is stated as qualitative- or quantitative evaluations of recommendation quality in terms of user satisfaction. These evaluations may answer the question, whether the implemented design decisions improve a user's utility when using the system. In fact, it is precisely this very evaluation that is being researched by co-author Johannes Luef in the course of the specialization topic chapter 3 *Evaluation*.

[1] Christian Ohrfandl, 2018.

Contents

CHAPTER 1

Introduction

The global financial crisis of October 2008 highly influenced the European economic market. According to the *Organisation for Economic Co-operation and Development (OECD)* and *Eurostat*, especially *new firm registrations* declined and *bankruptcies* increased in countries having a high level of financial development (such as Germany, France or non-OECD member economy USA) [Klapper and Love, 2011; OECD, 2016; Eurostat, 2016a]. In the last few years, economies of the EU-28 member states slightly recovered from the crisis and therefore, EU-wide enterprise entries rose by 6.8% in 2013 compared to 2012[1] [OECD, 2012; Eurostat, 2016b]. Especially the birthrate of enterprises in Austria rose by a mean of 8.1% during 2013 – 2015, indicating a total of 294.648 company births since the beginning of the financial crisis [WKO, 2016]. Klapper and Love [2011] argue that a positive and continuous birthrate of enterprises is the key factor for thriving innovation and is essential for the proceeding of the economic market's dynamics. Consequently, financing of new companies must be guaranteed in order to increase the birthrate of enterprises. Predominantly, funding of ventures is addressed by investors such as *Business Angels* or *Venture Capital Funds*, who provide capital particularly needed in the early stages of the company formation and beyond. However, as a result of the considerably large amount of enterprises entering the European market, potential investors face the problem of *information overload*. Due to its nature, information overload in the domain of venture valuation leads to the inapplicability of traditional *investment decision-making* criteria needed for managing an investor's investment portfolio. Therefore, the need for information filtering techniques based on computational *recommendation systems* emerges.

The main objective of this thesis consists of the design of a recommender system, representing a novel method concerning the computational recommendation of early-stage enterprises to investors. In order to quantify decision rules the recommender system is based on, investors' requirements and behaviours need to be analysed utilizing qualitative- and quantitative research. Furthermore, demonstrating the behaviour of the proposed recommendation algorithm is a

[1] 2012: excluding Greece; 2013: excluding Greece, Ireland and Poland

major task of this thesis. For this reason, a prototype of the recommender system is being crafted in software. According to Jannach et al. [2010, pp. 186, 187], the usability of the recommendation system's user interface plays a key role in terms of recommendation quality. Thus, a usability- and recommendation quality review of the prototype is being conducted in the course of empirical research. Finally, the following research questions will be answered:

(i) How can investment decision-making requirements and behaviours of investors be quantified for being used in a recommender system? [Christian Ohrfandl]

  (a) Which investment decision-making criteria are crucial to investors?

  (b) Which data needs to be provided by early-stage enterprises in order to be of interest to investors?

(ii) Which venture valuation methods best model the characteristics of early-stage enterprises? [Christian Ohrfandl]

(iii) How do the identified investment decision-making characteristics and venture valuation methods affect the model of a recommender system in the domain of early-stage enterprises? [Christian Ohrfandl]

(iv) Which recommendation algorithms and -techniques shall be considered in a computational recommendation system in the domain of early-stage enterprise investment, in order to guarantee highly personalized recommendations for investors? [Christian Ohrfandl, Johannes Luef]

(v) How can the *cold start problem* in the context of computational recommendation systems in the domain of early-stage enterprise investment, be addressed? [Christian Ohrfandl, Johannes Luef]

(vi) Which constraints does a software prototype of the computational recommendation system need to fulfil, in order to guarantee technical- and algorithmic feasibility? [Christian Ohrfandl, Johannes Luef]

(vii) How does the recommendation quality and the usefulness of the recommendation system affect user satisfaction? [Johannes Luef]

  (a) How may the recommender system be evaluated in terms of recommendation quality?

  (b) Which methodologies may be utilized to evaluate the recommendation system?

  (c) What implications do the results indicate?

  (d) How do the findings affect future research?

1.1 Expected Results

The main outcome of this thesis is stated as the construction of a usability improved prototype of a computational system capable of delivering highly personalized recommendations of early-stage enterprises to investors. In the course of this thesis, qualitative- and quantitative research is conducted in order to analyse investors' venture valuation criteria. Whereas the results of qualitative research (such as expert interviews or literature review) help identifying investors' investment decision-making criteria, the findings obtained by quantitative research (such as a questionnaire) quantify the gathered data and therefore highly affect the final design of the underlying recommendation system. Finally, modelling of the prototype's user interface design is needed to support the investor in exploring and filtering the entrepreneurial market. Thus, the recommendation quality of the algorithms will be evaluated in the course of qualitative research (such as expert interviews or offline evaluation testing respectively).

1.2 Methodology

The methodological approach consists of the following steps:

(i) Literature review and research on investors' investment decision-making criteria and -behaviours. To the authors' best knowledge, very few publications are available that discuss the issue of recommendations in the domain of venture valuation.

(ii) Research is conducted in a qualitative- and quantitative manner in order to gather significant data needed for modelling the recommender system.

(iii) Based on the data provided by the previous task, the purpose of this subtask is to specify recommendation algorithms that generate highly personalized recommendations of early-stage enterprises fitting investors' needs.

(iv) After successful specification and design of the recommendation system, a prototype is crafted in software.

(v) Finally, recommendation quality of the algorithms is reviewed in the course of empirical research.

1.3 State of the art

Due to its interdisciplinary nature, the *state of the art* of this thesis is considerably broad. Therefore, the remainder of this section is divided into three parts, each addressing research on the corresponding field of science, that is, *Venture Valuation*, *Recommender Systems* and *Usability Engineering*.

Venture Valuation is one of the key concepts of this thesis, as it provides necessary calculation models needed as input of investors' decision-making criteria. Although research on the valuation of ventures has become very popular in the last decade, little attention has been given to the field of innovative early-stage enterprises. One major characteristic of early-stage enterprises is the absence of profit combined with a rapid growth in revenue especially in the early stages of the venture [Rudolf and Witt, 2002, p. 259]. Unlike the traditional valuation models such as the *intrinsic value method*, this behaviour needs to be addressed by valuation models not solely relying on profit. Rudolf and Witt [2002, pp. 67, 81] argue that the *earnings value-* and *Discounted Cash Flow (DCF)* approaches extended by life cycle phases and phase models respectively, fit the initial characteristics of early-stage enterprises. Finally, a previous study conducted by Achleitner et al. [2004] indicates that 25% (50% respectively) of the interviewed investors utilize the earnings value- and DCF approaches for the valuation of ventures in the growth phase.

Recommender Systems have become an independent research area during the mid-1990s with roots in the fields of cognitive science, approximation theory, information retrieval, forecasting theory, management science and marketing [G. Adomavicius and Tuzhilin, 2005]. Nowadays, conferences and special interest groups such as *ACM Recommender Systems (RecSys)*, *ACM Special Interest Group on Information Retrieval (SIGIR)*, *User Modeling, Adaptation and Personalization (UMAP)* and *ACM Special Interest Group on Management Of Data (SIGMOD)* actively contribute to the field of recommender systems [Ricci et al., 2010]. However, to the authors' best knowledge, very few publications are available in the literature that discuss the issue of recommendations in the domain of venture valuation. Recently, Stone, Zhang, and Zhao [2013] and Zhao, Zhang, and Wang [2015] have proposed a new approach on addressing the problem of risk-hedged venture capital recommendation from a risk management perspective. The researchers proposed five algorithms analysing investors' investment behaviour and showed that the newly predicted investment opportunities compared to the *CrunchBase*[2] dataset lead to significant performance improvements in the context of recommendation quality. Nevertheless, a key limitation of this research may be seen in the fact that investors' human characteristics such as the level of risk aversion and personal interests on investment categories have not been taken into account. Hence, as indicated in the previously, focussed research on investors' needs is the most important part of this thesis and, consequently, a prerequisite to the specification of the recommender system.

Usability Engineering as a discipline of human computer interaction plays a vital role in the user interface design of the recommendation system's prototype. In recent years, innovative approaches to human interaction design have emerged in the form of *usability guidelines* supporting developers in the design of interfaces [Shneiderman and Plaisant, 2016, p. 75]. Companies dominating the IT market such as Microsoft[3] or Apple[4] provide their own usability

[2] Crunchbase is an internet platform offering the discovery of innovative companies and their staff: https://www.crunchbase.com/

[3] Usability guidelines Microsoft: https://developer.microsoft.com/en-us/windows/design

[4] Usability guidelines Apple: https://developer.apple.com/ios/human-interface-guidelines/

guidelines on how to interact with their systems. The main objective of these guidelines is stated as the standardization of task sequences that allow users to perform tasks in the same sequence and manner across similar conditions. Previous research has demonstrated that the consequent use of standardized task sequences reduces the user's workload [Shneiderman and Plaisant, 2016, p. 75]. However, information dashboard design has become popular in recent years as being a uniquely powerful tool for communicating important information. As reported by Few [2006, p. 97], the fundamental challenge of dashboard design involves information filtering and representational techniques. Nevertheless, other sophisticated approaches can be found in the literature. Shneiderman and Plaisant [2016, pp. 152-155] analyse and compare various aspects of expert reviews. The results obtained by Shneiderman and Plaisant suggest that expert reviews may occur early or late in the design phase. Furthermore, the authors claim that expert reviews are an effective way to improve the design quality of the user interface. Additionally, the authors' attention was not only focused on expert reviews but also on usability testing. The emergence of usability testing is an indicator of the profound shift in attention towards users' needs [Shneiderman and Plaisant, 2016, p. 156]. Finally, in order to guarantee a high quality interaction design, the aforementioned research concerning usability guidelines and dashboard design highly contributes to the outcome of this thesis.

1.4 Relevance to the Curriculum of Business Informatics

The study of business informatics focuses on the link between humans, organizations and information technology. In addition, *information processing* plays an important role in organizations and society. In the last few years there has been a growing interest in the subject of research-driven teaching of information- and communication systems in economics and society [TU Wien, 2013]. As a consequence, research on the analysis, modelling, design, implementation and evaluation of such systems has become very popular. In particular, the curriculum of the business informatics study highly correlates with the interdisciplinary characteristics of this thesis:

(i) *Innovation (188.915)*
The focus of the course Innovation lies on the evaluation of innovative projects utilizing case studies and business models. The implementation of a prototypical system based on business ideas is an assignment of the course.

(ii) *Business Intelligence (188.429)*
The course Business Intelligence plays a vital role in fundamental data mining technologies and their applications such as big data analysis. The educational objective of the course lies in the selection and application of appropriate data mining algorithms on a given dataset. The learning content is used to gain experience in different data mining techniques.

(iii) *Econometrics for Business Informatics (105.628)*
The focus of the course *Econometrics for Business Informatics* lies on the calculation of elementary econometric models and methods. In particular, the outcome of the recommen-

dation algorithm will employ econometric models in order to improve decision-making processes.

(iv) *Beyond the Desktop (183.639)*
The course *Beyond the Desktop* focuses on the understanding, interpretation and adaptation of human computer interaction methodologies. The graphical user interface of the prototype needs to support investors' needs.

(v) *Advanced Software Engineering (180.456)*
A solid understanding of software development and architectural design is an important requirement for the creation of the recommendation system's prototype.

CHAPTER 2

Recommender Systems for Early-Stage Enterprise Investment

2.1 Background

The very purpose of the current section 2.1 is defined as the elaboration of basic concepts of the *recommender systems* area of research. Furthermore, challenges and current research trends in the domain of computational recommendation systems are elaborated. Finally, a terminology of the basic concepts is being established by the definition of mathematical conventions and symbols, which is further utilized throughout the present work.

2.1.1 Recommender Systems Classification

The purpose of computational recommendation systems, which are also referred to as *recommender systems*, is the suggestion of *items* (certain topics of interest such as music, movies or news) the user might prefer over other items [Ricci et al., 2010]. These systems are usually implemented as algorithms in software. In order to improve accuracy of decision-making processes using recommendation algorithms, *user data* such as gender, age or interests is typically utilized and provided *directly* by the user or *indirectly* by analysis of user behaviour. In general, recommender systems help users evaluate *favourable* items in a large set of objects. In addition to adapting to the user's needs and consequently the enhancement of user satisfaction while interacting with the software, companies gain certain improvements such as the increase of sold items, user fidelity and -satisfaction, through the application of recommendation techniques [Ricci et al., 2010]. Due to its nature, the field of application concerning recommender systems is widespread, but usually addresses *Digital Media*, *E-Commerce* and *online services* such as travelling recommendations. Furthermore, the highly diverse field of application scenarios mostly relies on custom approaches to designing recommendation algorithms, leading to an even more complicated, extensively increasing field of study. As a consequence, rec-

ommender systems are classified into mainly three different categories, that is, *collaborative filtering*, *content-based recommendation* and *knowledge-based recommendation* [Jannach et al., 2010].

Due to the aspects of the present work, the previously mentioned recommendation categories as well as the specialization research areas of *social-* and *hybrid* recommendation systems are being investigated in the following subsections.

Collaborative Filtering Recommender Systems

The first of the three classification categories—*collaborative filtering*, which is also referred to as *collaborative recommendation*—exploits the overlapping interests between users, that is, the recommendation of items based on their shared set of interest [Jannach et al., 2010]. As an example, the purchasing history of user X and user Y at an online music store may be taken into account. It is assumed that the purchasing history of user X and Y is overlapping, implying a highly similar taste in music. If user X purchased a new song, a recommendation algorithm based on a collaborative filtering technique is very likely to propose the same song to user Y. In general, the main obstacle of this approach may be seen in the filtering of redundant items. Furthermore, *collaborative recommendation* addresses questions such as *How is similarity measured?*, *What about the recommendation of items that have never been sold yet?* and *What if the amount of ratings on an item is very low?* Due to its nature, the *Collaborative filtering* approach does not require any information about the *characteristics* or *attributes* of the item itself [Jannach et al., 2010]. This fact may be regarded a disadvantage, because recommendations might not only be based on overlapping user interests, but also on the content of items.

According to Aggarwal [2016, p. 29], neighbourhood-based collaborative filtering algorithms—which are also referred to as memory-based algorithms—were among the earliest algorithms developed in the context of collaborative filtering recommenders. These algorithms are based on the fact that similar users apply similar patterns of rating behaviour and, therefore, similar items receive similar ratings. The literature on collaborative filtering techniques mainly distinguishes these concepts into *User-based Neighbourhood Collaborative Filtering* and *Item-based Neighbourhood Collaborative Filtering*.

User-based Neighbourhood Collaborative Filtering While its calculation of recommendations is based on ratings provided by peer users similar to a target user, Jannach et al. [2010, p. 13] state that the main idea behind the user-based collaborative filtering approach is specified as follows: Given a ratings database and a target user, identify a set of peer users whose historic interests are similar to those of the target user and finally, predict items to the target user based on the identified neighbourhood users. Subsequently, the predicted score for the target user is computed by the peer users' item ratings.

Item-based Neighbourhood Collaborative Filtering In contrast to the user-based approach, item-based collaborative filtering calculates predictions utilizing the similarity between items instead of users [Jannach et al., 2010, p. 18]. In order to predict the rating of a target user for a target item, the target user's ratings for a set of peer items similar to the selected item

are considered. Finally, an item-based algorithm calculates a weighted average of all peer item ratings and predicts a rating for the target item to the target user.

Content-based Recommender Systems

Content-based recommendation—the second class of recommendation algorithms—mainly relies on information retrieved from the content of items and therefore, future recommendations are only based on content similarity between the *selected*- and the *investigated* item. Taking again the previous example of the shared purchasing history of user X's and Y's music into account, a content-based recommendation algorithm would recommend songs based on genre, band, year et cetera. The dependency on the purchasing history between user X and Y is—unlike the *collaborative filtering approach*—not taken into account. Additionally, information about the user's purchasing history and profile may automatically be retrieved (for instance via the purchasing behaviour) or manually set (such as preferences, interests and personal data about the user). According to Jannach et al. [2010], content-based recommendation algorithms have the advantages of direct recommendation once item information is available and the unnecessity of large user groups for an increase of recommendation accuracy, compared to other recommendation approaches. On the other hand, information on items mostly has to be provided manually, which may be expensive and prone to errors.

More formally, content-based recommender systems calculate (and recommend) a set of items that are most similar to a target user's already known items in terms of their *content*. [Jannach et al., 2010, p. 51] regard the following information the only necessity to this process:

(i) A description of the item characteristics

(ii) A user profile that describes (historic) interests of a user

However, both collaborative- and content-based recommendation techniques have the prerequisite of a purchasing history in common. Naturally, there are scenarios of unavailability of such purchasing histories, making collaborative- and content-based recommendation algorithms inapplicable.

Knowledge-based Recommender Systems

The present category of recommendation algorithms is referred to as *knowledge-based recommendation*, which—in contrast to the previously introduced recommendation approaches—mainly concentrates on mostly manually provided information on both the user and the object of interest. Therefore, this type of recommendation systems is not relying on the prevalence of a user's purchasing history [Jannach et al., 2010]. A car market place may be considered as an example. The average user buys a car just every couple of years, making it impossible for a recommendation algorithm to create a user profile—this user behaviour is also referred to as *one-time buyer*. In order to address this problem, knowledge-based recommendation algorithms consider characteristics of cars such as *age*, *driven distance*, *performance* or *brand* in combination with certain preferred features selected by the user (for instance via a virtual user interface).

Furthermore, user preferences may be weighted according to the relative importance to the user. Therefore, a so called *constraint-based recommender*—one of many different recommendation algorithms in the domain of knowledge-based recommender systems—is applied to match an aggregate of cars to the preferences of a user, resulting in a filtered set of cars most probably liked by the user. Due to this fact, it may be implied that the success of a knowledge-based recommender system relies on the interaction with the user. As a consequence, more elaborate approaches tend to implement an *interaction style* based on human conversation [Jannach et al., 2010].

Social Recommender Systems

Traditional recommender systems commonly ignore social connections between users [Massa and Avesani, 2004, p. 493]. In contrast, social recommender systems are based on social network structures such as trust and distrust. Recent research has shown that merging social network structures and recommender systems may improve the accuracy of recommendations and the user's experience [Aggarwal, 2016, p. 23]. Users who are socially connected are more likely to share the same- or similar interests. Subsequently, users may easily be influenced by *trusted* users, that is, there exists a high likelihood that a trusted person's recommendations are trusted as well [Victor et al., 2011, p. 49]. As a result, social trust relationships are highly correlated with similarity, that is, users usually have more trust in similar users. However, it is important to inform about the *asymmetry* of trust, leading to the fact that trust may be modelled as directed graph. An example may be seen in users that directly specify their trust or distrust relationships to other users. This obtained trust information is considered highly beneficial in the context of a user-based neighbourhood collaborative filtering algorithm, which computes more accurate recommendations by the sole utilization of trusted peer users [Victor et al., 2011, p. 52].

Hybrid Recommender Systems

Whereas the previously introduced recommendation techniques possess various advantages and disadvantages, hybrid recommenders leverage and combine multiple recommendation techniques in order to maximize recommendation performance in terms of quality (a user's utility) and outweigh the corresponding disadvantages (such as the coldstart / ramp-up problem) introduced by each of the utilized recommenders when applied separately. In particular, Burke states:

> “ …Hybrid recommender systems combine two or more recommendation techniques to gain better performance with fewer of the drawbacks of any individual one. Most commonly, collaborative filtering is combined with some other technique in an attempt to avoid the ramp-up problem. …
>
> *Burke [2002, p. 339]* ”

According to Jannach et al. [2010, pp. 128–142], hybrid recommendation systems are commonly divided into *monolithic-*, *parallelized* and *pipelined* hybridization designs. While a

monolithic hybridization design comprises a single recommender that integrates various recommendation algorithms by preprocessing- and combining different sources of knowledge, a parallelized hybridization design consists of multiple recommenders and a component that merges their results. In analogy to the latter, pipelined hybridization designs consist of multiple recommenders that are consecutively applied on a list of items. Independently, the introduced hybridization designs are further distinguished as follows: [Jannach et al., 2010, pp. 128–142; Burke, 2002, p. 340]

(i) Monolithic hybridization design
   - Feature combination
   - Feature augmentation

(ii) Parallelized hybridization design
   - Mixed
   - Weighted
   - Switching

(iii) Pipelined hybridization design
   - Cascade
   - Meta-level

In particular, *cascade hybrids* are considerably important in the course of the present work and are therefore described as follows: As stated earlier, being a member of the category of pipelined hybridization designs, the cascade hybrid consists of multiple recommenders that are consecutively applied on a list of items. This procedure sequentially refines the ranking of the list's items after each recommendation algorithm's application. However, only the initially applied recommender may alter the items included in the list of items, that is, the consecutively applied recommenders are prohibited from removing or adding items to the list. Ultimately, the last iteration calculates the final ranking of the item list.

2.1.2 Challenges and Current Trends in Research

Concluding major classification types of recommendation algorithms, a large field of study may be outlined. Be it *collaborative filtering*, *content-* or *knowledge-based recommendation*, *social-* or *hybrid recommender systems*: each approach has certain advantages and disadvantages depending on the goal about be achieved. However, the downside of the nearly endless range of applicable scenarios in the context of computational recommendation systems is the rise of a very complex problem domain.

The biggest problem of recommendation algorithms based on a ratings structure, arises from the estimation of ratings for items unknown to the user [G. Adomavicius and Tuzhilin, 2005]. In a more formal way, the goal of a recommendation algorithm is the maximization of a user's utility, that is, only items unknown to the user and exposing the highest probability of user satisfaction are recommended. In order to address this problem, algorithms usually rely on ratings for other items given by the same user. This fact also indicates the requirement of an initial subset of user ratings, before recommendations based on a ratings structure can be given. This scenario is also referred to as the *cold start* problem [Schein et al., 2002]. After

gaining an initial set of ratings, a user's utility is usually extrapolated utilizing methods of various domains such as machine learning, approximation theory or heuristics [G. Adomavicius and Tuzhilin, 2005]. Another problem deriving from a considerably small amount of quantified input data needed for extrapolation, may be seen in the user's trust into the Recommendation System [Ricci et al., 2010]. Wrong recommendations directly affect the trust a user has into the system, possibly resulting in the avoidance of the platform. Due to the high complexity of the mentioned problem domain, recommender systems have become an important research area, having roots in various fields of study.

According to G. Adomavicius and Tuzhilin [2005], recommender systems have become an independent research area during the mid-1990s with roots in the fields of cognitive science, approximation theory, information retrieval, forecasting theory, management science and marketing [G. Adomavicius and Tuzhilin, 2005]. Nowadays, conferences and special interest groups such as *ACM Recommender Systems (RecSys)* and *User Modeling, Adaptation and Personalization (UMAP)* were founded to thrive research in the area of recommendation algorithms [Ricci et al., 2010]. Considering the early age of start of research, *Recommender Systems* is a very young field of study that combines logics, human computer interaction (HCI), data mining and information retrieval. Due to its multidisciplinary nature, many challenges such as performance of algorithms, privacy & security issues and diversity of recommendations arise. Current emerging research topics can be seen in multidisciplinary areas, for instance, new visualization techniques, recommendations based on trust, personalization & search involving whole communities, social tagging systems (STS), recommendations based on different coherences and addressing security issues [Ricci et al., 2010].

2.1.3 Mathematical Conventions and Symbols

The first step of defining solutions (mathematical calculations) to the problem domain of the recommendation system lies in the definition of mathematical conventions and symbols. This subsection covers the basic symbols utilized throughout the present work. More advanced topics are elaborated in the corresponding sections 2.2 and 2.3.

In order to conform to the common recommender systems terminology, the present work will address *investors* as *users* and *ventures* as *items*. Let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ be defined as the set of users and $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\}$ as the set of items held by the platform. Without loss of generality, it shall be defined that the implementation of the present recommendation system is only permitted to recommend items *not yet known* to a certain user.

Similarity

In the concept of recommender systems, *similarity* of entities (user or item) plays an important role. Recommendations are created based on the assumption that similar users like similar items. The degree of distinction between the target- and an investigated (peer) *entity*—which is also referred to as *similarity*—, is dependent on the attributes of the entity (such as users or items) itself, that is, the *distinguishing features*, and the applied similarity measure [Huang,

2008, p. 51]. The result of a similarity measure is commonly represented as a numeric value, whereas larger values imply higher similarity and small values low similarity analogously.

As pointed out earlier, a similarity measure's main purpose is to find the degree of distinction between two entities. However, similarity measures are not universally usable. On the contrary, the selection of a similarity measure highly depends on the underlying type of dataset. Therefore, the following subsections elaborate on similarity measures utilized in the course of the present work.

Jaccard Similarity Coefficient Jaccard [1912, p. 39] defines the so called *coefficient of community* as the size of the intersection of two target sets A and B divided by the size of their union, more formally:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \;\Big|\; 0 \leq J(A,B) \leq 1 \tag{2.1}$$

$J(A,B)$ is a function $J : A, B \to \{x \in \mathbb{R}_{\geq 0} : x \leq 1\}$, whereas an output of 1 signals that two sets are similar and —analogously —0 specifies that two sets are dissimilar to each other.

The Cosine Similarity Coefficient is commonly utilized in the fields of information retrieval and text mining with the aim of comparing text documents represented as vectors of terms. The metric measures the similarity between two non-zero n-dimensional vectors based on the angle between them. The similarity between two items a and b—views of the corresponding vectors a and b—is formally defined as follows:

$$\cos(a,b) = \frac{\langle a,b \rangle}{|a| * |b|} \tag{2.2}$$

In the Euclidean space $\mathbb{R}^n$, the inner product $\langle a,b \rangle$ is given by the dot product. $\langle a,b \rangle$ is defined as $\sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$, where $|a|$ is stated as the Euclidean length of the vector, which is defined as the square root of the dot product of the vector with itself [Jannach et al., 2010, p. 19]. Furthermore, $\cos(a,b)$ is a function $\cos : a, b \to \{x \in \mathbb{R}_{\geq 0} : x \leq 1\}$, whereas —assuming the constraint of positive vector entries—1 states that the target- and investigated entities are *equal* and 0 expresses that both entities entirely *differ* from each other.

However, the cosine similarity coefficient has one major disadvantage, that is, the difference in rating scales between various users are not taken into account. As a consequence, the *adjusted cosine similarity* measure addresses this disadvantage by subtracting the corresponding mean from each co-rated pair [Sarwar et al., 2001, p. 288]. The output of the adjusted cosine similarity measure ranges in the interval of -1 to 1 in $\mathbb{R}$.

The Kendall Tau Distance $\mathcal{K}(\tau_1, \tau_2)$ is stated as commonly utilized *distance measure* for two total orders τ_1 and τ_2. This measure counts the number of *disagreements* between τ_1 and

τ_2 regarding pairs of candidates—which is also referred to as *discordant pairs* [F. Brandenburg, 2011, p. 3]. More formally, let the Kendall tau distance be defined as

$$\mathcal{K}(\tau_1, \tau_2) = \frac{\sum_{\{a,b\} \in P} \bar{\mathcal{K}}_{a,b}(\tau_1, \tau_2)}{n \cdot (n-1)/2} \tag{2.3}$$

where P is a set of unordered pairs of distinct elements in τ_1 and τ_2, $\bar{\mathcal{K}}_{a,b}(\tau_1, \tau_2) = 0$ if a and b are in the same order in τ_1 and τ_2 or $\bar{\mathcal{K}}_{a,b}(\tau_1, \tau_2) = 1$ if a and b are in the reverse order in τ_1 and τ_2 [Fagin, Kumar, and Sivakumar, 2003, p. 140]. Subsequently, $\mathcal{K}(\tau_1, \tau_2)$ equals to 0 if the two lists are exactly the same. Analogously, if one list is the reverse of the other, that is, both lists are of entirely opposite order, $\mathcal{K}(\tau_1, \tau_2)$ equals to $n(n-1)/2$ (where n is stated as the number of elements in the list).

Neighbourhood Formation

Another major design choice in the context of similarity measures that has a considerably large impact on recommendation quality and computational performance, lies in the *selection of the entity's neighbourhood*, that is, the (sub-)set of entities that are considered similar to a target entity. However, Jannach et al. [2010, pp. 17–18] argue that including the whole neighbourhood of entities affects the precision of recommendations due to the consideration of *false positives*, in other words, entities that are not significantly similar to the selected entity. Additionally, considering the whole entity neighbourhood could drastically increase calculation time (depending on the size of the neighbourhood).

The commonly accepted technique addressing the problems arising from a large neighbourhood lies in the reduction of its size either by definition of a *minimum similarity threshold* or by declaration of a maximum neighbourhood size k, that is, only including the k *nearest* neighbours in terms of similarity [Jannach et al., 2010, pp. 17–18]. However, limiting the size of the entity's neighbourhood introduces negative side effects such as finding the *fitting* value for k or the reduction of prediction coverage due to considerably high similarity thresholds. Therefore, J. Herlocker, Konstan, and Riedl [2002] conducted various experiments concerning the analysis of different weighting schemes and neighbourhood sizes. Based on the *MovieLens dataset*[1], J. Herlocker, Konstan, and Riedl argue that the entity's neighbourhood shall contain 20 to 50 entities in order to fit various real-world applications.

[1] MovieLens dataset: https://grouplens.org/datasets/movielens/

2.2 Methodology for Modelling Requirements

In the the course of the *methodology*, the reader is informed about the scientific approach utilized in the course of the present chapter, which is needed in order to gain knowledge for answering the research questions. The remainder of this section is organized as follows: Subsection 2.2.1 outlines the problem definition including the associated research questions, whereas subsections 2.2.2 to 2.2.6 describe different types of recommenders that are based on the research of the previous study by Christian Ohrfandl[2] and are about to be modelled throughout the upcoming results section. Finally, subsection 2.2.7 introduces the reader to the constraints of the recommendation system's software prototype.

2.2.1 Problem Definition & Research Questions

The research conducted in the present chapter is stated as the mathematical formulation of a recommendation system in the domain of early-stage enterprise investment, which is based on the qualitative- and quantitative research undertaken in the previous study by Christian Ohrfandl. In addition, the modelled recommendation system is being implemented as a software prototype. Finally, the following research questions will be answered:

(i) Which recommendation algorithms and -techniques shall be considered in a computational recommendation system in the domain of early-stage enterprise investment, in order to guarantee highly personalized recommendations for investors?

(ii) How can the *cold start problem* in the context of computational recommendation systems in the domain of early-stage enterprise investment, be addressed?

(iii) Which constraints does a software prototype of the computational recommendation system need to fulfil, in order to guarantee technical- and algorithmic feasibility?

With the aim of scientifically answering the mentioned research questions, the methodological approach addresses the *transition* from the venture valuation- and early-stage enterprise investment research conducted in the previous study by Christian Ohrfandl, to potentially fitting recommendation approaches (the reader is referred to Figure 2.1 for a detailed overview of the transitioning process). Therefore, the following subsections represent different types of recommenders, elaborate on the findings researched in the previous study by Christian Ohrfandl and discuss a possible conformity to certain recommendation algorithms. The mathematical modelling of these systems is conducted throughout section 2.3. Finally, section 2.4 discusses the results and provides answers to the research questions.

2.2.2 Collaborative Filtering

A very interesting and controversial finding in the previous study by Christian Ohrfandl may be seen in the fact that users incorporate the opinions of other users into their investment

[2] Christian Ohrfandl, 2018.

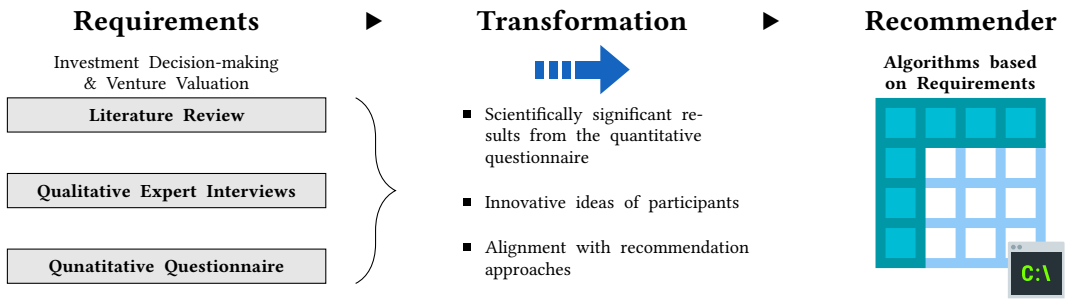


Figure 2.1: Transition process from venture valuation- and investment decision-making requirements to recommendation algorithms.

decision-making processes. In particular, the term *opinion* may be interpreted as a user's interaction upon items and expressed as *ratings* of various types (the reader is referred to the model subsection 2.3.1 for a detailed introduction to the present work's implementation of rating structures). This behaviour considerably relates to a commonly utilized recommendation approach—*Collaborative Filtering*.

Collaborative filtering techniques exploit the overlapping interests between users, that is, the recommendation of items based on their shared set of interests. As mentioned in the background section 2.1, these techniques base their recommendations on the fact that similar users apply a similar rating behaviour on items. Jannach et al. [2010, p. 3] state that due to its independence from an item's attributes, collaborative filtering approaches may be applied in situations where analysis of the content is difficult. In the context of the present present work, this fact is regarded an advantage, for it is the users' interests that might not entirely be based on the similarity of items' contents.

Due to the aforementioned aspects, the authors conclude that collaborative filtering techniques are beneficial for generating recommendations that improve user satisfaction in the context of early-stage enterprise investment. The reader is referred to subsection 2.3.2 for a detailed description of the present work's implementation of the user-based- and item-based neighbourhood collaborative filtering approaches.

2.2.3 Content-Based Recommendation

One crucial finding in the previous study by Christian Ohrfandl may be seen in the fact that a user's personal interests are considered highly important for the utilization in the domain of early-stage enterprise investment. A recommendation approach that highly correlates to this finding is defined as *Content-based Recommandation*, which utilizes the comparison of content among a user's historically interested items and new ones, that is, unknown items.

In contrast to the collaborative filtering approach, a content-based recommender has the advantage of not being reliant upon users' ratings data. As mentioned in the background section 2.1, a content-based recommendation system generates predictions based on an item's attributes combined with the target user's interests. The previous study findings by Christian

Ohrfandl show that a user's most important item attributes are defined as follows (the reader is referred to the model subsection 2.3.1 for a detailed introduction to the present work's implementation of the recommender's domain model):

- Product Interest
- Market Sector
- Life Cycle

In context to the previous studies research, the authors want to highlight the importance of content-based recommenders in the domain of early-stage enterprise investment. This technique provides solutions to problems such as *sparse ratings data* that especially apply to the domain of early-stage enterprise investment. Due to these aspects and the support by the scientific literature, the authors decide on implementing a content-based recommender in the course of the present work. The reader is referred to subsection 2.3.3 for a detailed representation of the content-based recommender's concept, model and mathematical foundations.

2.2.4 Knowledge-Based Recommendation

One of the main findings of the previous study by Christian Ohrfandl indicates that sorting/filtering of early-stage enterprises according to a user's personal preferences is considered highly important for the use in a recommender system. As discussed in the background subsection 2.1.1, knowledge-based recommender systems mainly base their recommendation algorithms on manually provided information on attributes of an item and the target user. Subsequently, these systems have the advantage of not being reliant upon a user's purchasing history and item ratings by other users. Furthermore, knowledge-based recommendation systems are commonly utilized if other recommendation techniques (such as collaborative filtering) are inapplicable. This is especially the case for considerably sparse ratings sets, as in the domain of early-stage enterprise investment.

In contrast to domains more accustomed to the use of recommender systems that are commonly based on large-scale ratings sets—such as digital media or e-commerce—the authors of the present chapter presume that ratings sets in the domain of early-stage enterprise investment are considerably sparse. This assertion is based on the results of the qualitative research conducted in the previous research by Christian Ohrfandl. In the course of expert interviews it has been shown that especially business angels hold/invest only in a handful of early-stage enterprises. This statement is also supported by the fact that the invested money per item is considerably higher when compared to other domains, leading to the phenomena of *one-time buyers*. Furthermore, knowledge-based recommender systems do not face the *cold start* problem, that is, a recommendation of early-stage enterprises may even be conducted in the early times of platform existence, due to the fact that these systems do not rely upon ratings of other users.

Due to the aforementioned reasons, the authors conclude that the expression of a user's personal preference through the utilization of a user's choices upon an early-stage enterprise's attributes, is seen as highly beneficial to the user's satisfaction. In particular, as the previous study has shown, these choices are based on an item's attributes most important to the user and are comprised of- but not limited to the following listing:

- Investment- and share ranges of items
- Venture valuations
- Date of creation
- Market sector
- Interest in an item based on *tags* describing an item's products
- Interest of potential future customers in the item
- Team
- Life cycle stage

Based on the introduced facts, the authors conclude that the present work highly benefits from knowledge-based recommendation techniques in terms of recommendation quality and user satisfaction. Subsequently, the authors decide that the present work shall implement a knowledge-based recommendation algorithm. The reader is referred to subsection 2.3.4 for a detailed representation of the knowledge-based recommender's concept, model and mathematical foundations.

2.2.5 Social Trust Recommendation

One key finding of the previous study by Christian Ohrfandl qualitative expert interviews may be seen in the fact that investors consider the opinion of other investors in the course of their investment decision-making processes. In particular, the participants mentioned that group investments directed by a lead investor and the general recommendation of early-stage enterprises by other investors may be seen as a common practice in the domain of early-stage enterprise investment. As a consequence, it may be concluded that investors express their connectedness towards other investors as *trust relationships*. Due to the compliance to these reasons, a recommendation system based on human trust, that is, a social recommendation system, may be utilized in the context of early-stage enterprise investment.

As mentioned in the present chapter's background section, social recommendation systems base their recommendation algorithms on explicitly provided trust between users. The *Social Recommendation* algorithm may be seen as a traditional *User-Based Neighbourhood Collaborative Filtering* algorithm but the similarity function is defined as trust relationship between users. On the basis of these implications, trust relationships are utilized in collaborative-filtering techniques to generate personalized recommendations. According to Jamali and Ester [2009, p. 397], trust among users plays an important role in social networks. Therefore, one kind of trust emerges from explicit trust between users, that is, trust among users is stated as directed graph between two users.

Due to the aforementioned aspects and the support by the scientific literature, the authors decide on implementing a social recommender in the course of the present work. The reader is referred to subsection 2.3.5 for a detailed description of the present work's implementation of the social recommender.

2.2.6 Hybrid Recommendation

A very important- and controversial finding of the previous studies qualitative- and quantitative research indicates that users actively consolidate other user's opinions on the investment in

certain items. However, the research by Christian Ohrfandl also indicates that users possess a solidified- and determined opinion on an item's set of preferable attributes as well, possibly restricting other users' influence. A self-contradictory situation arises that upon closer analysis motivates an innovative solution by the utilization of a hybrid recommender that refines a user's preferred list of items by the opinions of other users.

As introduced in the present chapter's background subsection 2.1.1, hybrid recommendation techniques utilize a combination of various other recommenders and calculate the final recommendations by the merge of these techniques according to different aspects. Subsequently, in combination with the facts determined from the paragraph above, that is, a refinement of an existing recommender's recommendations, the emerging constraints highly indicate the utilization of a *pipelined hybridization design*—in particular, a variation of a *cascading hybrid recommender*.

According to Jannach et al. [2010, pp. 138–139], cascading hybrid recommendation algorithms utilize a consecutive list of different recommenders, each refining the rank of the recommendations. Due to its design, only the initial recommender may define a list of items passed on to the other recommendation techniques that, subsequently, are not allowed to exclude existing- or include new items. Additionally, refinements themselves are considered to be of *evolutionary* type, that is, the ranking of the initial list of recommended items shall not be modified too deeply, but rather introduce modest changes. As a consequence, these constraints perfectly match to the previously defined achievable behaviour of a user's recommendation refinement by the utilization of other users' opinions.

Therefore, the authors propose the implementation of a cascading hybrid recommendation algorithm that mainly utilizes the knowledge-based recommender to define a user's list of recommended items and consecutively refines the list's ranking by the application of a user-based collaborative filtering technique, incorporating other users' item recommendations. The reader is referred to subsection 2.3.6 for a detailed representation of the hybrid recommender's concept, model and mathematical foundations.

2.2.7 Recommender System Prototype

One major part of the present research topic lies in the creation of a recommender system prototype in software that implements and comprises all mathematically modelled recommendation algorithms into one single platform. Due to the fact that the implementation of the prototype is affected considerably by the utilized type of recommendation algorithms, the reader is referred to subsection 2.3.7 for a detailed representation of the recommendation system's prototype.

2.3 Design of the Recommender System

In the course of the present section, the reader is informed about the present chapter's results—the implementation of various recommendation algorithms and the crafting of a software prototype, as defined by the methodology's transition from the previous study by Christian Ohrfandl. However, all recommenders share the same underlying (data)model of the present recommendation system and therefore, subsection 2.3.1 describes this model and the recommender system's general functionality in great detail. Independently, the remainder of this section is further organized as follows: Subsections 2.3.2 to 2.3.6 cover the results of one particular recommendation algorithm type each, that is, subsection 2.3.2 comprises *Collaborative Filtering*, subsection 2.3.3 outlines *Content-based Recommendation*, subsection 2.3.4 studies *Knowledge-based Recommendation*, subsection 2.3.5 defines *Social Recommendation* and, consequently, subsection 2.3.6 addresses *Hybrid Recommendation*. Finally, subsection 2.3.7 informs the reader about the constraints of the recommendation system's software prototype.

2.3.1 Model

With the aim of representing the core aspects of the present recommendation system, the remainder of this subsection is divided into four parts, that is, the recommender's domain model, item's attributes, user profile and, finally, user-item interactions. The reader is referred to Figure 2.2 to gain an overview of the system's core model.

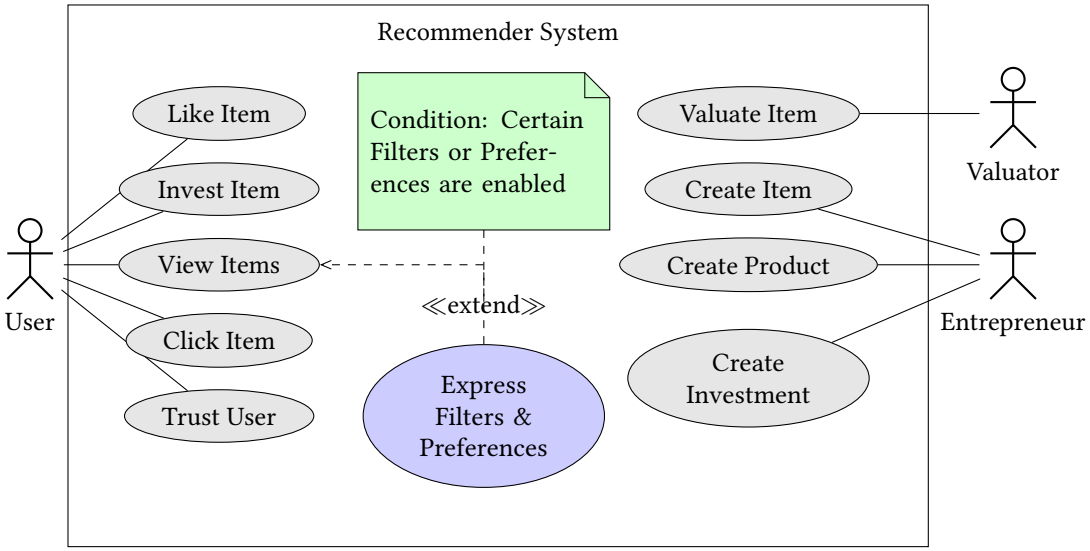


Figure 2.2: Conceptional illustration of the recommender system's model and -functionality on the item entity.

Entities of the System

The present subsection introduces the reader to the different domain model entities that build the basis for the recommendation algorithms and are therefore utilized throughout the present chapter.

Item (Venture) In order to conform to the common recommender systems terminology, the present work will address *ventures* as *items*. Analogously to the importance of the user entity (down below), an *Item* is considered a major part of the recommender system model. In particular, it is the items and their rankings that users are interested in.

User (Investor) In order to conform to the common recommender systems terminology, the present work will address *investors* as *users*. The *User* entity is regarded the major concept in the recommender system model and represents the main interface of interaction between the system and the real world—the investor. The main aim a user tries to achieve, that is, the semantic definition of a user's utility, is stated as gathering a ranked list of items that best fit the user's individual constraints induced by personally defined investment decision-making criteria. As a consequence, a user has two possibilities of interacting with the system:

- Personalized interaction: Participation in the system by providing information utilized by the recommender to generate personalized recommendations.
- Non-personalized interaction: Illustration of items without disclosing personal information. Therefore, the resulting item list does not induce any (personalized) ranking.

Valuator In contrast to the user- and entrepreneur entities, a *Valuator* is a special type of entity that possesses the knowledge to calculate a pre-money valuation of an item on the basis of the item's attributes. In particular, a pre-money valuation may be calculated by the utilization of the scorecard- or berkus methods and expressed as a certain amount of money. Due to the fact that valuators and subsequently, the whole valuation process, is conceptualized independently of the recommendation system, the present chapter will not give any further explanation on that matter.

Entrepreneur The creation of an item and the maintenance of its sub-entities is conducted by the *Entrepreneur*, an entity representing the owners of the item. If an item holds many owners, this set of entrepreneurs is considered a *Team*.

Item Content

Items need *distinguishing features*, for only then will recommenders be capable of calculating certain rankings. Therefore, the reader is referred to the following Table 2.1, which showcases a detailed representation of an item's attributes and the corresponding mathematical symbols that are further utilized in the course of the present chapter.

Table 2.1: Attributes of an item

Attribute	Symbol	Description
Name	–	The name of the item (specified by the entrepreneurs)
Description	–	Internal: The internal description of the item that is only visible to users (specified by the entrepreneurs).
		Public: Description viewable by the public (specified by the entrepreneurs).
City	–	The physical location of the item (specified by the entrepreneurs).
Product	$\mathcal{D}$	An item may possess a set of products or product ideas (provided by the entrepreneurs).
Investment	$\mathcal{D}.\mathcal{Z}$	Each of the item's products may hold a set of investment offerings—specified by the entrepreneurs—allowing users to invest.
Investment Amount	$\mathcal{Z}.Amount$	An investment has a certain investment amount, expressing the amount of money a user needs to invest in order to acquire a certain share of the item.
Investment Share	$\mathcal{Z}.Share$	The share of an item (in percent) a user acquires when investing in the corresponding product of the item.
Valuation	$\mathcal{D}.\mathcal{V}$	Each of the item's products may hold a set of pre-money valuations (specified by valuators).
Valuation Amount	$\mathcal{V}.Amount$	A valuation has a certain pre-money valuation amount, expressing the current worth of the item as amount of money.
Valuation Method	$\mathcal{V}.Method$	The valuation method utilized in the course of the valuation process (either the scorecard- or berkus method).
Date	$Date$	The item's date of creation (specified by the entrepreneurs).
Market Sector	$ms \in \mathcal{M}$	The market sector an item operates in (chosen by the entrepreneurs from a predefined set of market sectors $\mathcal{M}$).

To be continued…

...continued from previous page

Attribute	Symbol	Description
Product Interest	$\mathcal{D}.\mathcal{PI}$	Each of the item's products may hold a set of short product descriptions (provided by the entrepreneurs), depicting characteristics of the item in one-word phrases, which are also referred to as *tags*.
Public Interest	$\mathcal{P}$	The interest in the item expressed by the clicking behaviour of possible future customers. This attribute is calculated by the recommender system and influenced by external public viewers not authenticated to the recommendation system.
Team	$\mathcal{E}$	The item's team members represented as a set of entrepreneurs (specified by the entrepreneurs).
Life Cycle	$c \in \mathcal{C}$	The item's current life cycle stage (chosen by the entrepreneurs from a predefined set of life cycle stages $\mathcal{C}$).

User Profile

One of the major use cases of the user may be interpreted as the viewing of items. As already noted in the user's entity description, personalized- and non-personalized interaction with the system is mainly distinguished by the user's provided data on the affection towards certain attributes of items. If a user decides to offer the needed data, the recommendation system ranks the list of items according to the user's disclosed predilection among item attributes. In the course of the present chapter, this particular data is further categorized as *Filters* and *Preferences*. Finally, the user has the possibility to store these categorical settings in a *recommendation profile*, enabling the utilization of *hybrid* recommendation techniques. In particular, this profile holds certain user specified attributes depicted in Table 2.2. The reader is referred to the knowledge-based- and hybrid recommendation algorithms at subsections 2.3.4 and 2.3.6 for a detailed description, an elaboration on application scenarios, implementational specifics of Preferences & Filters and the recommendation profile.

In addition to the process of interacting with items, the user may also interact with other users of the recommendation system in the course of *fellowship*, that is, the explicit definition of trust towards other users. Trust relationships are defined as the crucial component of the social recommender and therefore, the reader is referred to subsection 2.3.5 for details on the implementation of trust in the domain of early-stage enterprise investment.

Table 2.2: Types of a user u's recommendation profile

Symbol	Type	Description
$\mathcal{L}^{\text{D}}$	Boolean	A boolean value enabling the Date preference.
$\mathcal{L}^{\text{HV}}$	Boolean	A boolean value enabling the High Valuation preference.
$\mathcal{L}^{\text{MS}}$	Boolean	A boolean value enabling the Market Sector preference.
$\mathcal{M}^u$	Set	A set of market sectors selected by user u.
$\mathcal{L}^{\text{ProdI}}$	Boolean	A boolean value enabling the Product Interest preference.
$\mathcal{PI}^u$	Set	A set of product interests, that is, characteristics of items' products specified in one-word phrases and also referred to as *tags*, specified by user u.
$\mathcal{L}^{\text{PubI}}$	Boolean	A boolean value enabling the Public Interest preference.
$\mathcal{L}^{\text{T}}$	Boolean	A boolean value enabling the Team preference.
$\mathcal{L}^{\text{LC}}$	Boolean	A boolean value enabling the Life Cycle preference.
$\mathcal{C}^u$	Set	A set of life cycle stages selected by user u.

User-Item Interactions

One of the key aspects of the recommendation system is stated as the modelling of interactions between the user- and the item entity. These interactions are further referred to as set of *input signals* utilizing the notation $\mathcal{R}$ and are divided into the following set of types: *Likes*, *Investments* and *Clicks*. The prediction of items to a target user is based on $\mathcal{R}$, which may also generically be defined as a set of *ratings* of user u on certain items. More formally, a rating is defined as function $r : \mathcal{U} \times \mathcal{I} \to \mathcal{D} \mid r \subset \mathcal{U} \times \mathcal{I}$ resulting in the form $r(u, i)$. The definition of $\mathcal{D}$ is dependent on the corresponding *input signals* and distinguished as follows:

(i) Likes (r^L): A user u has the possibility to *like* an item i, which expresses the preference of a user towards a certain item. The set of items liked by user u is defined as $\mathcal{I}^{L,u}$, where $\mathcal{I}^{L,u} \subset \mathcal{I} \ \wedge \ u \in \mathcal{U}$. The ratings structure $\mathcal{D}$ is defined as follows: $\mathcal{D} = \{0, 1\}$.

(ii) Investments (r^I): A user may *invest* once or several times in certain items depending on the availability of products and investment offerings. The set of items invested in by a certain user u is defined as $\mathcal{I}^{I,u}$, where $\mathcal{I}^{I,u} \subset \mathcal{I} \ \wedge \ u \in \mathcal{U}$. The user's utility towards an item is expressed as the number of investments for that very item. Therefore, the ratings structure $\mathcal{D}$ is defined as follows: $\mathcal{D} = \mathbb{N}_0$.

(iii) Clicks (r^C): A user u may *click* certain items, implicitly expressing interest in the corresponding item. The set of items clicked by a certain user u is defined as $\mathcal{I}^{C,u}$, where

$\mathcal{I}^{C,u} \subset \mathcal{I} \ \wedge \ u \in \mathcal{U}$. The ratings structure $\mathcal{D}$ is defined as follows: $\mathcal{D} = \mathbb{N}_0$.

The reader is referred to the collaborative filtering-, content-based- and social recommendation algorithms at subsections 2.3.2, 2.3.3 and 2.3.5 for detailed information on the application of input signals and ratings.

2.3.2 Collaborative Filtering

One of the most commonly utilized approaches in the domain of recommender systems is *collaborative filtering* [Jannach et al., 2010, p. 3]. A collaborative filtering recommendation algorithm utilizes a similarity measure to identify the most similar users or items based on a given ratings database. The reader is referred to the background subsection 2.1.3 for detailed information about the similarity measure and the ratings database. Afterwards, this approach generates predictions for items and, finally, recommends these items to the target user u. The collaborative filtering approaches utilized throughout the present work are divided into two types, that is *User-based-* and an *Item-based Collaborative Filtering*.

User-based Collaborative Filtering

Fundamental elements of a *User-based Neighbourhood Collaborative Filtering* approach are stated as the calculation of similarity between users and the neighbourhood selection process. Subsequently, the user-based similarity's distinguishing features are reflected by certain attributes of the user entity such as the user's *likes*, *investments* or *clicks* on certain items. The following approaches describe the calculation of similarity and the neighbourhood selection process in great detail.

Similarity functions A similarity function is a measurement that computes the similarity between two users, building the basis for the calculation of personalized recommendations. The similarity function's result is defined as real number of the interval $[0, 1]$, whereas 1 states that the target- and investigated entities are *equal* and 0 expresses that both entities entirely *differ* from each other. Each particular similarity function is described as follows:

The Likes similarity calculates the similarity $sim_L(u, u')$ between a set of items $\mathcal{I}^{L,u}$ liked by the target user u and the set of items $\mathcal{I}^{L,u'}$ liked by the peer user u'. The calculation of similarity between users is accomplished by the utilization of the *Jaccard similarity coefficient*. More formally, let $sim_L(u, u')$ be denoted as the *Likes* similarity function of user u and a peer user u', where

$$sim_L(u, u') = J\left(\mathcal{I}^{L,u}, \mathcal{I}^{L,u'}\right) = \frac{\left|\mathcal{I}^{L,u} \cap \mathcal{I}^{L,u'}\right|}{\left|\mathcal{I}^{L,u} \cup \mathcal{I}^{L,u'}\right|} \tag{2.4}$$

holds.

The Investments similarity calculates the similarity $sim_I(u, u')$ between a set of items $\mathcal{I}^{I,u}$ invested by the target user u and the set of items $\mathcal{I}^{I,u'}$ invested by the peer user u'. The calculation of similarity between users is accomplished by the utilization of the *Cosine similarity coefficient*, whereas the sets $\mathcal{I}^{I,u}$ and $\mathcal{I}^{I,u'}$ are represented as vectors depicting the number of investments of both users for the corresponding sets of items. More formally, let $sim_I(u, u')$ be denoted as the *Investments* similarity function of user u and a peer user u', where

$$sim_I(u, u') = \cos\left(\mathcal{I}^{I,u}, \mathcal{I}^{I,u'}\right) = \frac{\left\langle \mathcal{I}^{I,u}, \mathcal{I}^{I,u'} \right\rangle}{|\mathcal{I}^{I,u}| * |\mathcal{I}^{I,u'}|} \tag{2.5}$$

holds.

The Clicks similarity calculates the similarity $sim_C(u, u')$ between a set of items $\mathcal{I}^{C,u}$ clicked by the target user u and the set of items $\mathcal{I}^{C,u'}$ clicked by the peer user u'. The calculation of similarity between users is accomplished by the utilization of the *Jaccard similarity coefficient*. More formally, let $sim_C(u, u')$ be denoted as the *Clicks* similarity function of user u and a peer user u', where

$$sim_C(u, u') = J\left(\mathcal{I}^{C,u}, \mathcal{I}^{C,u'}\right) = \frac{\left|\mathcal{I}^{C,u} \cap \mathcal{I}^{C,u'}\right|}{|\mathcal{I}^{C,u} \cup \mathcal{I}^{C,u'}|} \tag{2.6}$$

holds.

Neighbourhood formation In the course of the User-based Collaborative Filtering approach, neighbours are peer users, whose historical interests are similar to those of the target user. More formally, a peer user u' is denoted as *neighbour* of user u, if u' is similar to u. Therefore, the neighbourhood function $N(u)$ reduces the set of peer users taken as input to the recommendation algorithm. According to Sarwar et al. [2001], the selection of an appropriate threshold plays a crucial role for the prediction quality of a neighbourhood collaborative filtering algorithm. Therefore, in order for a peer user to be included in the neighbourhood of the target user, the corresponding similarity function needs to reach or exceed a certain threshold. A detailed discussion about choosing an appropriate threshold can be found in the background subsection 2.1.3 or in Jannach et al. [2010, pp. 17–18] and Gedikli [2013, p. 11]. One finding of the previous study by Christian Ohrfandl indicates that ratings data in the domain of early-stage enterprise investment is considerably sparse. Due to this reason, the threshold of all subsequent neighbourhood functions is set to the real number 0.7. The mathematical notations of the neighbourhood functions are described as follows:

The Likes neighbourhood $N_L(u)$ function is denoted as neighbourhood of user u, where

$$N_L(u) = \{\ u' \in U \setminus u\ :\ sim_L(u, u') \geq 0.7\} \tag{2.7}$$

holds.

The Investments neighbourhood $N_I(u)$ function is denoted as neighbourhood of user u, where

$$N_I(u) = \{\ u' \in U \setminus u\ :\ sim_I(u, u') \geq 0.7\} \tag{2.8}$$

holds.

The Clicks neighbourhood $N_C(u)$ function is denoted as neighbourhood of user u, where

$$N_C(u) = \{\ u' \in U \setminus u\ :\ sim_C(u, u') \geq 0.7\} \tag{2.9}$$

holds.

Recommendation Algorithms Typically, the *prediction* or *score* of items to a user is formally expressed as a *utility function* of the form $S : Users \times Items \rightarrow \mathbb{R}_{\geq 0} \mid R \subset Users \times Items$ [Gediminas Adomavicius, Manouselis, and Kwon, 2011, pp. 769–770]. Subsequently, the adaptation to the present work is stated as $S : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}_{\geq 0} \mid S \subset \mathcal{U} \times \mathcal{I}$. Without loss of generality, it shall be defined that a user u is capable of knowing whether a certain item recommendation is appropriate and therefore, user u knows *if* a certain item i belongs to the personal set of likeable items. Subsequently, the main task of the recommendation algorithm is to construct a ranked list, such that the utility function $S(u, i)$ is maximized.

The final ordered item recommendation set comprises a certain amount of items that constitute a high probability of being liked by the target user. The ranking set is based on each item's utility calculated by the utility function S in a descending order. The prediction algorithms utilized by the present work are elaborated based on mathematical formulations as follows:

The Likes algorithm (UB^L) calculates the utility of a target item based on a weighted factor that depends on the similarity of the peer users and their rankings of the target item. Finally, item recommendations are calculated based on the liked items of the target user's neighbourhood and weighted by the similarities and item occurrences of the corresponding peer user. More formally, the recommender is defined as follows:

$$S^{UB^L}(u, i) = \frac{\sum\limits_{u' \in N(u)} sim(u, u') \cdot r^L(u', i)}{|N(u)|} \tag{2.10}$$

where u is defined as the target user, u' is stated as the investigated (peer) user, i is specified as the target item, $sim(u, u') \in \{sim_L(u, u'), sim_I(u, u'), sim_C(u, u')\}$ is stated as similarity function and $N(u) \in \{N_L(u), N_I(u), N_C(u)\}$ is defined as the corresponding user neighbourhood of user u.

The Investments algorithm (UB^I) calculates the utility of a target item based on a weighted factor that depends on the similarity of the peer users and their rankings of the target item. Finally, item recommendations are calculated based on the invested items of the

target user's neighbourhood and weighted by the similarities and item investments of the corresponding peer user. More formally, the recommender is defined as follows:

$$S^{UB^I}(u,i) = \frac{\sum\limits_{u' \in N(u)} sim(u,u') \cdot r^I(u',i)}{\sum\limits_{u' \in N(u)} sim(u,u')} \tag{2.11}$$

where u is defined as the target user, u' is stated as the investigated (peer) user, i is specified as the target item, $sim(u,u') \in \{sim_L(u,u'), sim_I(u,u'), sim_C(u,u')\}$ is stated as the similarity function and $N(u) \in \{N_L(u), N_I(u), N_C(u)\}$ is defined as the corresponding user neighbourhood of user u.

The Clicks algorithm (UB^C) calculates the utility of a target item based on a weighted factor that depends on the similarity of the peer users and their rankings of the target item. Finally, item recommendations are calculated based on the clicked items of the target user's neighbourhood and weighted by the similarities and item clicks of the corresponding peer user. More formally, the recommender is defined as follows:

$$S^{UB^C}(u,i) = \frac{\sum\limits_{u' \in N(u)} sim(u,u') \cdot r^C(u',i)}{\sum\limits_{u' \in N(u)} sim(u,u')} \tag{2.12}$$

where u is defined as the target user, u' is stated as the investigated (peer) user, i is specified as the target item, $sim(u,u') \in \{sim_L(u,u'), sim_I(u,u'), sim_C(u,u')\}$ is stated as the similarity function and $N(u) \in \{N_L(u), N_I(u), N_C(u)\}$ is defined as the corresponding user neighbourhood of user u.

Item-based Collaborative Filtering

Analogously to user-based neighbourhood collaborative filtering, the calculation of similarity between items and the neighbourhood selection process are regarded fundamental procedures of the *Item-based Neighbourhood Collaborative Filtering* approach. Subsequently, the item-based similarity's distinguishing features are reflected by certain attributes of the user entity, such as the user's *likes*, *investments* or *clicks* on certain items. The following approaches describe the calculation of similarity and the neighbourhood selection process in great detail.

Similarity functions A similarity function is a measurement that computes the similarity between two items, building the basis for the calculation of personalized recommendations. The similarity function's result is defined as real number of the interval $[0, 1]$, whereas 1 states that the target- and investigated entities are *equal* and 0 expresses that both entities entirely *differ* from each other. Each particular similarity function is described as follows:

The Likes similarity calculates the similarity $sim_L(i, i')$ between a set of users $\mathcal{U}^{L,i}$ who liked the target item i and the set of users $\mathcal{U}^{L,i'}$ who liked the peer item i'. The calculation of similarity between items is accomplished by the utilization of the *cosine similarity coefficient*. More formally, let $sim_L(i, i')$ be denoted as the *Likes similarity* function of item i and a peer item i', where

$$sim_L(i, i') = \cos\left(\mathcal{U}^{L,i}, \mathcal{U}^{L,i'}\right) = \frac{\left\langle \mathcal{U}^{L,i}, \mathcal{U}^{L,i'} \right\rangle}{|\mathcal{U}^{L,i}| * |\mathcal{U}^{L,i'}|} \tag{2.13}$$

holds.

The Investments similarity calculates the similarity $sim_I(i, i')$ between a set of users $\mathcal{U}^{I,i}$ who invested in the target item i and the set of users $\mathcal{U}^{I,i'}$ who invested in the peer item i'. The calculation of similarity between items is accomplished by the utilization of the *cosine similarity coefficient*. More formally, let $sim_I(i, i')$ be denoted as the *Investments similarity* function of item i and a peer item i', where

$$sim_I(i, i') = \cos\left(\mathcal{U}^{I,i}, \mathcal{U}^{I,i'}\right) = \frac{\left\langle \mathcal{U}^{I,i}, \mathcal{U}^{I,i'} \right\rangle}{|\mathcal{U}^{I,i}| * |\mathcal{U}^{I,i'}|} \tag{2.14}$$

holds.

The Clicks similarity calculates the similarity $sim_C(i, i')$ between a set of users $\mathcal{U}^{I,i}$ who clicked the target item i and the set of users $\mathcal{U}^{I,i'}$ who clicked the peer item i'. The calculation of similarity between items is accomplished by the utilization of the *cosine similarity coefficient*. More formally, let $sim_C(i, i')$ be denoted as the *Clicks similarity* function of item i and a peer item i', where

$$sim_C(i, i') = \cos\left(\mathcal{U}^{C,i}, \mathcal{U}^{C,i'}\right) = \frac{\left\langle \mathcal{U}^{C,i}, \mathcal{U}^{C,i'} \right\rangle}{|\mathcal{U}^{C,i}| * |\mathcal{U}^{C,i'}|} \tag{2.15}$$

holds.

Neighbourhood formation In the course of the Item-based Collaborative Filtering approach, neighbours are the transpose of the user-based neighbours. More formally, a peer item i' is denoted as *neighbour* of item i, if i' is similar to i. The neighbourhood function $N(i)$ reduces the set of peer items taken as input to the recommendation algorithm. According to Sarwar et al. [2001], the selection of an appropriate threshold plays a crucial role for the prediction quality of a neighbourhood collaborative filtering algorithm. Therefore, in order for a peer item to be included in the neighbourhood of the target item, the corresponding similarity function needs to reach or exceed a certain threshold. A detailed discussion about choosing an appropriate threshold can be found in the background subsection 2.1.3 or in Jannach et al. [2010, pp. 17–18] and Gedikli [2013, p. 11]. One finding of the previous study by Christian Ohrfandl indicates that ratings data in the domain of early-stage enterprise investment is considerably sparse. Due to this reason, the threshold of all subsequent neighbourhood functions is set to the real number 0.7. The mathematical notations of the neighbourhood functions are described as follows:

The Likes neighbourhood $N_L(i)$ function is denoted as neighbourhood of item i, where

$$N_L(i) = \{\ i' \ \in \ I \setminus i \ : \ sim_L(i, i') \geq 0.7\} \tag{2.16}$$

holds.

The Investments neighbourhood $N_I(i)$ function is denoted as neighbourhood of item i, where

$$N_I(i) = \{\ i' \ \in \ I \setminus i \ : \ sim_I(i, i') \geq 0.7\} \tag{2.17}$$

holds.

The Clicks neighbourhood $N_C(i)$ function is denoted as neighbourhood of item i, where

$$N_C(i) = \{\ i' \ \in \ I \setminus i \ : \ sim_C(i, i') \geq 0.7\} \tag{2.18}$$

holds.

Recommendation Algorithms The main task of the recommendation algorithm is stated as the construction of a ranked list, such that the utility function $S(u, i)$ is maximized. The final ordered item recommendation set comprises a certain amount of items that constitute a high probability of being liked by the target user. The ranking set is based on each item's utility calculated by the utility function S and sorted in a descending order. The prediction algorithms utilized by the present work are elaborated based on mathematical formulations as follows:

The Likes algorithm (IB^L) calculates the utility of a target item based on a weighted average factor that depends on the similarity and the ranking of the corresponding peer item. In the course of the item-based approach, peer items are considered for similarity calculation rather than peer users. Finally, item recommendations are calculated based on the liked items of the target item's neighbourhood and weighted by the similarity and the item likings of the corresponding peer item. More formally, the recommender is defined as follows:

$$S^{IB^L}(u, i) = \frac{\sum\limits_{i' \in N(i)} sim(i, i') \cdot r^L(u, i')}{|N(i)|} \tag{2.19}$$

where u is defined as the target user, i is specified as the investigated item, i' is stated as the investigated peer item, $sim(i, i') \in \{sim_L(i, i'), sim_I(i, i'), sim_C(i, i')\}$ is specified as the similarity function and $N(i) \in \{N_L(i), N_I(i), N_C(i)\}$ is defined as the corresponding item neighbourhood of the investigated item i.

The Investments algorithm (IB^I) calculates the utility of a target item based on a weighted average factor that depends on the similarity and the ranking of the corresponding peer item. In the course of the item-based approach, peer items are considered for similarity calculation rather than peer users. Finally, item recommendations are calculated based on the

invested items of the target item's neighbourhood and weighted by the similarity and the item investments of the corresponding peer item. More formally, the recommender is defined as follows:

$$S^{IB^I}(u,i) = \frac{\sum_{i' \in N(i)} sim(i,i') \cdot r^I(u,i')}{|N(i)|} \tag{2.20}$$

where u is defined as the target user, i is specified as the investigated item, i' is stated as the investigated peer item, $sim(i,i') \in \{sim_L(i,i'), sim_I(i,i'), sim_C(i,i')\}$ is specified as the similarity function and $N(i) \in \{N_L(i), N_I(i), N_C(i)\}$ is defined as the corresponding item neighbourhood of the investigated item i.

The Clicks algorithm (IB^C) calculates the utility of a target item based on a weighted average factor that depends on the similarity and the ranking of the corresponding peer item. In the course of the item-based approach, peer items are considered for similarity calculation rather than peer users. Finally, item recommendations are calculated based on the clicked items of the target item's neighbourhood and weighted by the similarity and the item clicks of the corresponding peer item. More formally, the recommender is defined as follows:

$$S^{IB^C}(u,i) = \frac{\sum_{i' \in N(i)} sim(i,i') \cdot r^C(u,i')}{|N(i)|} \tag{2.21}$$

where u is defined as the target user, i is specified as the investigated item, i' is stated as the investigated peer item, $sim(i,i') \in \{sim_L(i,i'), sim_I(i,i'), sim_C(i,i')\}$ is specified as the similarity function and $N(i) \in \{N_L(i), N_I(i), N_C(i)\}$ is defined as the corresponding item neighbourhood of the investigated item i.

2.3.3 Content-based Recommendation

Content-based Recommendation—which is also referred to as *Content-based Filtering*—is solely based on the user's personal interest in an item. Therefore, its major concept is defined as content comparison between historically interested items and new ones, that is, unknown items. Subsequently, those not yet known items maximizing the target user's utility are recommended. The basis for determining similarity in the context of content-based filtering is stated as content comparison between certain attributes of the user's historically interested items and the corresponding attributes of peer items not yet known to the user (the reader is referred to subsection 2.3.1 for a detailed description of an item's model). The following approaches describe the calculation process of the content-based algorithm in detail.

User Profile

In the content-based filtering approach the *User Profile* is represented as a virtual item including all *Product Interest*, *Market Sector* and *Life Cycle* attributes merged among all of user u's historic item interests. The reader is referred to Table 2.2 for a detailed description of the attributes.

Product Interest The present content-based filtering implementation considers an item's product tags in order to compute similarity between items. A set of an item's products' tags $u.\mathcal{PI}$ of the liked items of the target user u is stated by the following equation:

$$u.\mathcal{PI} = \bigcup_{i \in I^{L,u}} \{(\forall d \in i.\mathcal{D})\,[d.\mathcal{PI}]\} \tag{2.22}$$

where u is defined as the target user, i is specified as the historically liked item of the target user u, $\mathcal{PI}$ is defined as the set of product tags and $\mathcal{D}$ is stated as a product.

Market Sector The present content-based filtering implementation considers an item's market sectors in order to compute similarity between items. A set of market sectors $u.\mathcal{M}$ of liked items of the target user u is stated by the following equation:

$$u.\mathcal{M} = \bigcup_{i \in I^{L,u}} \{(\forall ms \in i.\mathcal{M})\,[ms]\} \tag{2.23}$$

where u is defined as the target user, i is specified as the historically liked item of the target user u and $\mathcal{M}$ is defined as the set of market sectors.

Life Cycle The present content-based filtering implementation considers an item's life cycle in order to compute similarity between items. A set of life cycles $u.\mathcal{C}$ of liked items of the target user u is stated by the following equation:

$$u.\mathcal{C} = \bigcup_{i \in I^{L,u}} \{(\forall c \in i.\mathcal{C})\,[c]\} \tag{2.24}$$

where u is defined as the target user, i is specified as the historically liked item of the target user u, $\mathcal{C}$ is defined as the set of life cycles.

Recommendation Algorithm

The main task of the recommendation algorithm is stated as the calculation of a ranked list, such that the utility function $S^{CB}(u, i)$ is maximized. The final ordered item recommendation set comprises a certain amount of items that constitute a high probability of being liked by the target user. The ranking of this set is based on each item's utility calculated by the utility function S^{CB} and sorted in descending order. The whole utility function S^{CB}'s calculation is based on the computation of the previously introduced attributes and, subsequently, the corresponding similarity functions. Therefore, the *similarity functions* and, ultimately, the recommendation algorithm are described in great detail as follows:

Product Interest similarity The Product Interest similarity function calculates the similarity between items based on the *Jaccard Similarity Coefficient*. More formally, let $sim_{\mathcal{PI}}(u.\mathcal{PI}, i'.\mathcal{D}.\mathcal{PI})$

be denoted as the *Product Interest similarity* function utilizing product tags of items historically liked by user u and the peer item i''s product tags $i'.\mathcal{D}.\mathcal{PI}$, where

$$sim_{\mathcal{PI}}\left(u.\mathcal{PI}, i'.\mathcal{D}.\mathcal{PI}\right) = J\left(u.\mathcal{PI}, i'.\mathcal{D}.\mathcal{PI}\right) = \frac{|u.\mathcal{PI} \cap i'.\mathcal{D}.\mathcal{PI}|}{|u.\mathcal{PI} \cup i'.\mathcal{D}.\mathcal{PI}|} \tag{2.25}$$

holds.

Market sector similarity The Market sector similarity function calculates the similarity between items based on the *Jaccard Similarity Coefficient*. More formally, let $sim_{\mathcal{M}}(u.\mathcal{M}, i'.\mathcal{M})$ be denoted as the *Market Sector similarity* function utilizing market sectors of items historically liked by user u and the peer item i''s market sector $i'.\mathcal{M}$, where

$$sim_{\mathcal{M}}\left(u.\mathcal{M}, i'.\mathcal{M}\right) = J\left(u.\mathcal{M}, i'.\mathcal{M}\right) = \frac{|u.\mathcal{M} \cap i'.\mathcal{M}|}{|u.\mathcal{M} \cup i'.\mathcal{M}|} \tag{2.26}$$

holds.

Life cycle similarity The Life cycle similarity function calculates the similarity between items based on the *Jaccard Similarity Coefficient*. More formally, let $sim_{\mathcal{C}}(u.\mathcal{C}, i'.\mathcal{C})$ be denoted as the *Life Cycle similarity* function utilizing distinct life cycles of items historically liked by user u and the peer item i''s life cycle $i'.\mathcal{C}$, where

$$sim_{\mathcal{C}}\left(u.\mathcal{C}, i'.\mathcal{C}\right) = J\left(u.\mathcal{C}, i'.\mathcal{C}\right) = \frac{|u.\mathcal{C} \cap i'.\mathcal{C}|}{|u.\mathcal{C} \cup i'.\mathcal{C}|} \tag{2.27}$$

holds.

Finally, the recommendation algorithm calculates item i''s utility to the target user u based on the average score of all individual similarities each weighted by an individual importance factor w. This factor is derived from the findings of the previous study by Christian Ohrfandl stating that the market sector is more important than the life cycle and, in turn, the life cycle is more important than product tags. Finally, the recommender is defined as follows:

$$\begin{aligned} S^{CB}(u, i') = &\frac{sim_{\mathcal{PI}}(u.\mathcal{PI}, i'.\mathcal{D}.\mathcal{PI}) \cdot w_{\mathcal{PI}}}{3} \\ &+ \frac{sim_{\mathcal{M}}(u.\mathcal{M}, i'.\mathcal{M}) \cdot w_{\mathcal{M}}}{3} \\ &+ \frac{sim_{\mathcal{C}}(u.\mathcal{C}, i'.\mathcal{C}) \cdot w_{\mathcal{C}}}{3} \end{aligned} \tag{2.28}$$

where u is defined as the target user, i' is specified as the investigated item, $sim_{\mathcal{PI}}(u.\mathcal{PI}, i'.\mathcal{D}.\mathcal{PI})$, $sim_{\mathcal{M}}(u.\mathcal{M}, i'.\mathcal{M})$, $sim_{\mathcal{C}}(u.\mathcal{C}, i'.\mathcal{C})$ are stated as individual similarities and $w_{\mathcal{PI}} = 0.2$, $w_{\mathcal{M}} = 0.5$, $w_{\mathcal{C}} = 0.3$ are defined as statically set weighting factors summing up to the integer 1.

2.3.4 Knowledge-based Recommendation

The major concept of the knowledge-based recommender in the context of the present work is based on an item's *properties* and the sets of *filters* Φ and *preferences* Π. Generally speaking, each filter or preference matches certain filter- or preference *attributes* provided by the target user to the corresponding properties of each item (the reader is referred to subsection 2.3.1 for a detailed representation of an item's model). Whereas a filter excludes items from the set of recommended items if the constraints of the user defined filter parameters are not fulfilled, a preference may generally be seen as an option to sort the set of recommended items based on the individual preference settings defined by the target user. It is important to note that filters are considered *hard constraints* that need to be fulfilled entirely by an item in order to be included in the list of recommended items. In contrast, each preference setting is weighted, that is, the user has the possibility to set an importance parameter on each selected preference individually. Finally, according to the user's selected and specified filters or preferences—which may also be considered *decision rules*—the knowledge-based recommender computes and ranks items that are most probably being liked by the user.

Filters

The first major aspect of the knowledge-based recommender is stated as the set of *filters* Φ that —in contrast to preferences—are regarded *hard constraints*, having the need of being fulfilled by an item in order to even be included in the recommendation list. However, filters are also based on certain user specified attributes $\mathcal{L}^{\Phi}$. From a process-oriented perspective, the application of filters directly influences the base set of items $\mathcal{I}$ that may be further utilized for recommendation in the course of the selection of preferences. Therefore, the formal representation of the output of a filter $\phi \in \Phi$ is defined as the set $\mathcal{I}^{\phi} \subset \mathcal{I}$ and calculated as follows:

$$\mathcal{I}^{\phi} = \phi(\mathcal{I}, \mathcal{L}^{\phi}) \tag{2.29}$$

where ϕ is defined as function $\phi : \mathcal{I}, \mathcal{L}^{\phi} \rightarrow \mathcal{I}^{\phi}$ and $\mathcal{L}^{\phi} \subset \mathcal{L}^{\Phi}$ is stated as filter ϕ's set of attributes specified by the user.

One major design criterion of the knowledge-based recommender is stated as user u's freedom of choice in the context of the selection of preferences and filters. As a consequence, filters need to be user selectable on an individual basis. However, this aspect also implies that multiple filters may be selected by user u at the same time. In order to address this situation, the filter component of the knowledge-based recommender merges the output of each user selected filter by *intersection*, more formally:

$$\mathcal{I}^{\Phi^u} = \bigcap_{\phi \in \Phi^u} \phi(\mathcal{I}, \mathcal{L}^{\phi}) \tag{2.30}$$

where $\Phi^u \subset \Phi$ is defined as the set of filters selected by user u and $\mathcal{I}^{\Phi^u}$ is stated as the intersected output of all filter functions that depend on $\phi \in \Phi^u$, $\mathcal{I}$ and $\mathcal{L}^{\phi}$.

On the basis of Notation (2.29), the algorithms of each filter ϕ and the utilized set of user defined attributes $\mathcal{L}^{\phi}$ are depicted in detail in the following listing (the reader is referred to subsection 2.3.1 for information on the utilized attributes):

(i) Investment Range (ϕ_{IR}): The input to this filter is a user defined *investment amount range* defined by a minimum/maximum interval. The filter itself includes a certain item i in the result set, if there exists at least one investment for at least one of item i's products that has an investment amount lying within a user defined investment amount range. More formally, ϕ_{IR} is defined as follows:

$$\begin{aligned}\mathcal{I}^{\text{IR}} &= \phi_{\text{IR}}\left(\mathcal{I}, \mathcal{L}^{\text{IR}}\right) \\ &= \{i \in \mathcal{I} \,:\, (\exists d \in i.\mathcal{D})\,(\exists z \in d.\mathcal{Z})[\, MinAmount \leq z.Amount \leq MaxAmount]\,\}\end{aligned} \tag{2.31}$$

where $\mathcal{I}^{\text{IR}} \subset \mathcal{I}$ is considered the set of items calculated by the ϕ_{IR} function, $i.\mathcal{D}$ conforms to item i's set of products whereas $d \in \mathcal{D}$ is stated as one particular product of item i. Furthermore, $d.\mathcal{Z}$ is the set of product d's investment offerings whereas $z \in d.\mathcal{Z}$ is stated as one particular investment of product d. Additionally, $z.Amount$ is defined as a particular investment money offering's amount of money specified by item i. Finally, $MinAmount$ to $MaxAmount$ is stated as user u's specified investment amount range of the interval $[a, b] = \{a \in \mathbb{R}_{\geq 0}, b \in \mathbb{R}_{\geq 0} : a \leq b\}$ and is therefore considered the present filter's set of attributes $\mathcal{L}^{\text{IR}} = \{MinAmount, MaxAmount\}$.

(ii) Share Range (ϕ_{SR}): In analogy to the previous filter function, the input to the Share Range filter is a user defined *investment share range* defined by a minimum/maximum interval. The filter itself includes a certain item i in the result set, if there exists at least one investment for at least one of item i's products that has an investment share lying within a user defined investment share range. More formally, ϕ_{SR} is defined as follows:

$$\begin{aligned}\mathcal{I}^{\text{SR}} &= \phi_{\text{SR}}\left(\mathcal{I}, \mathcal{L}^{\text{SR}}\right) \\ &= \{i \in \mathcal{I} \,:\, (\exists d \in i.\mathcal{D})\,(\exists z \in d.\mathcal{Z})[\, MinShare \leq z.Share \leq MaxShare]\,\}\end{aligned} \tag{2.32}$$

where $\mathcal{I}^{\text{SR}} \subset \mathcal{I}$ is considered the set of items calculated by the ϕ_{SR} function. Furthermore, $z.Share$ is defined as a particular investment share offering's amount specified by item i. Finally, $MinShare$ to $MaxShare$ is stated as user u's specified investment share range of the interval $[a, b] = \{a \in \mathbb{Z}_{\geq 0}, b \in \mathbb{Z}_{\geq 0} : 0 \leq a \leq b \leq 100\}$ (in percent) and is therefore considered the present filter's set of attributes $\mathcal{L}^{\text{SR}} = \{MinShare, MaxShare\}$.

(iii) Valuation Range (ϕ_{VR}): Analogously to the previous filter functions, the input to the Valuation Range filter is a user defined *pre-money valuation amount range* defined by a minimum/maximum interval. The filter itself includes a certain item i in the result set, if there exists a valuation for at least one of item i's products that has a pre-money valuation amount lying within a user specified pre-money valuation range. More formally, ϕ_{VR} is defined as follows:

$$\begin{aligned}\mathcal{I}^{\text{VR}} &= \phi_{\text{VR}}\left(\mathcal{I}, \mathcal{L}^{\text{VR}}\right) \\ &= \{i \in \mathcal{I} \,:\, (\exists d \in i.\mathcal{D})\,(\exists v \in d.\mathcal{V})[\, MinVal \leq v.Amount \leq MaxVal]\,\}\end{aligned} \tag{2.33}$$

where $\mathcal{I}^{\text{VR}} \subset \mathcal{I}$ is considered the set of items calculated by the ϕ_{VR} function, $i.\mathcal{D}$ conforms to item i's set of products whereas $d \in \mathcal{D}$ is stated as one particular product of item i. Furthermore, $d.\mathcal{V}$ is the set of product d's valuations whereas $v \in d.\mathcal{V}$ is stated as one particular valuation of product d. Additionally, $v.Amount$ is defined as a particular pre-money valuation's amount held by the system. Finally, $MinVal$ to $MaxVal$ is stated as user u's specified pre-money valuation amount range of the interval $[a, b] = \{a \in \mathbb{R}_{\geq 0}, b \in \mathbb{R}_{\geq 0} : a \leq b\}$ and is therefore considered the present filter's set of attributes $\mathcal{L}^{\text{VR}} = \{MinVal, MaxVal\}$.

(iv) Valuation Method (ϕ_{VM}): In contrast to the previous filter functions, the input to the Valuation Method filter is a user defined set of *valuation methods* that are matched against existing valuations for products of item i. The filter itself includes a certain item i in the result set, if there exists a valuation for at least one of item i's products that was rated utilizing a pre-money valuation method being an element of a user specified set of valuation methods. More formally, ϕ_{VM} is defined as follows:

$$\begin{aligned} \mathcal{I}^{\text{VM}} &= \phi_{\text{VM}}\left(\mathcal{I}, \mathcal{L}^{\text{VM}}\right) \\ &= \{i \in \mathcal{I} \,:\, (\exists d \in i.\mathcal{D})\,(\exists v \in d.\mathcal{V})[\, v.Method \in \mathcal{VM}]\,\} \end{aligned} \tag{2.34}$$

where $\mathcal{I}^{\text{VM}} \subset \mathcal{I}$ is considered the set of items calculated by the ϕ_{VM} function. Furthermore, $v.Method$ is considered pre-money valuation v's valuation method. Finally, $\mathcal{VM}$ is the set of valuation methods specified by user u that complies to the variations depicted by the *power set* $\wp(\{scorecard, berkus\})$ and is therefore considered the present filter's set of attributes $\mathcal{L}^{\text{VM}} = \mathcal{VM}$.

Preferences

The second major aspect of the knowledge-based recommender's recommendation algorithm is the modelling of preferences that induce a ranking of items. The knowledge-based recommender enables the user to select a set of various preferences and assign each of which an individual *importance* rating—a weight—that expresses the user's utility towards a certain preference. Finally, the aggregation of each user specified preference's item rankings is accomplished by the implementation of a *Weighted Borda count*—a variation of a vote-counting scheme in the area of *collective decision-making*.

The Preference Function The first step of the knowledge-based recommender in the context of preferences is stated as the calculation of ranked item lists corresponding to the user specified preferences and their attributes. The semantics of the rank itself are defined as follows: Let preference π induce a *strict weak order* $<_\pi$ over a set of items. The authors denote that an item i is preferred over—that is, *of higher utility* than—another item j in the context of preference π as $i <_\pi j$. Subsequently, in the case of indifference among items with respect to preference π, that is, two items are incomparable, the following notation is denoted: $i \sim_\pi j$. Due to illustration purposes, the combination of the mentioned notations is denoted as $i \lesssim_\pi j$ stating that item i

is either preferred over- or indifferent from item j. This particular ordering scheme enables the assignment of ranks—consecutive natural numbers starting from 1—to items.

User u's individual rank for item i utilizing preference π among a certain set of items $\mathcal{I}$ is defined as follows:

$$\rho_{i,\pi,\mathcal{I}} = \pi(i, \mathcal{L}^{\pi}, \mathcal{I}) \tag{2.35}$$

where $\rho_{i,\pi,\mathcal{I}}$ is specified as the individual rank of user u's preference π for item i in the domain of $\mathcal{I}$, π is defined as the function $\pi : i, \mathcal{L}^{\pi}, \mathcal{I} \rightarrow \{n \in \mathbb{Z}_{>0} : 0 < n \leq |\mathcal{I}|\}$ resulting in the form $\pi(i, \mathcal{L}^{\pi}, \mathcal{I})$ and $\mathcal{L}^{\pi}$ is stated as user defined set of attribute values corresponding to preference π's domain of attributes.

In order to calculate the rank of a set of items $\mathcal{I}$ in the course of a certain preference π, a new data structure needs to be introduced with the purpose of expressing the rank $\rho_{i,\pi,\mathcal{I}}$ of an item i among a set of other items $\mathcal{I} \setminus i$. Let $\mathcal{I}^{\pi}$ be denoted as a set of ordered pairs $(i, \rho_{i,\pi,\mathcal{I}})$, more formally: $\mathcal{I}^{\pi} = \{(i, \rho_{i,\pi,\mathcal{I}}) : i \in \mathcal{I}, \rho_{i,\pi,\mathcal{I}} \in \{n \in \mathbb{Z}_{>0} : 0 < n \leq |\mathcal{I}|\}\}$.

Due to the fact that—from a process-oriented point of view—a preference π is interpreted as a function calculating a ranked list of items, each preference $\pi \in \Pi$ may utilize individual logic depending on the corresponding input factors (attributes) in order to calculate the rank among each item $i \in \mathcal{I}$. Therefore, the algorithms of each preference $\pi \in \Pi$ and, subsequently, the utilized set of attributes $\mathcal{L}^{\pi}$ are described in detail in the following listing (the reader is referred to subsection 2.3.1 for information on the utilized attributes):

(i) Date (π_{D}): The Date preference ranks item i in context to the domain of all items $\mathcal{I} \setminus i$ based on i's date of creation—that is, $i.Date$—whereas more recent dates are ranked better than their older counterparts. Due to the fact that the Date preference does not rely upon any user specified input, this preference's user specified set of attributes is defined as $\mathcal{L}^{\mathrm{D}} = \emptyset$.

(ii) High Valuation (π_{HV}): The basis for the High Valuation preference is stated as the average pre-money valuations for item i in the context of the domain of all items $\mathcal{I} \setminus i$. A higher average valuation is ranked better than its lower counterparts. Analogously to the previous Date preference, the present High Valuation preference does not rely upon any user specified input. Therefore, this preference's user specified set of attributes is defined as $\mathcal{L}^{\mathrm{HV}} = \emptyset$.

(iii) Market Sector (π_{MS}): The input to this preference is a user selected subset of a predefined set of market sectors—that is, $\mathcal{L}^{\mathrm{MS}} = \mathcal{M}^{u} \subset \mathcal{M}$—that are matched to item i's defined market sector. Matching itself is accomplished by the calculation of the Jaccard correlation coefficient between the set of selected market sectors and item i's actual market sector. In the context of the domain of all items $\mathcal{I} \setminus i$, item i is ranked better, if its calculated Jaccard correlation coefficient is higher than the ones of the other items or lower analogously.

(iv) Product Interest (π_{PRODI}): The input to this preference is a user defined set of *tags*—short one-word phrases describing the item—and therefore, $\mathcal{L}^{\mathrm{PRODI}} = \mathcal{PI}^{u}$. Matching itself is

accomplished by the calculation of the Jaccard correlation coefficient between the set of user defined tags $\mathcal{PI}^u$ and the set of tags of each of item i's products. Subsequently, the highest Jaccard correlation coefficient among item i's products is selected. In the context of the domain of all items $\mathcal{I} \setminus i$, item i is ranked better, if the selected Jaccard correlation coefficient is higher than the ones of the other items or lower analogously.

(v) Public Interest (π_{PUBI}): The core aspect of the Public Interest preference is a set of public interest ratings for item i, that is, $\mathcal{P}^i$. A rating $p \in \mathcal{P}^i$ itself is of the form $p = \{x \in \mathbb{Z}_{>0} : x < 6\}$. Calculation of the rank is conducted by building the average among all ratings p of item i and put in the context to the results of the set of items $\mathcal{I} \setminus i$. A higher average rating is ranked better than its lower counterparts. Due to the fact that the Public Interest preference does not rely upon any user specified input, this preference's user specified set of attributes is defined as $\mathcal{L}^{\text{PUBI}} = \emptyset$.

(vi) Team (π_{T}): Based on the the empiric results and literature review of the previous study by Christian Ohrfandl, the team of an item is considered more successful, if it consists of at least two persons. Building upon the scorecard method, teams fulfilling this constraint are awarded the integer number 1, 0 otherwise. In the context of the set of items $\mathcal{I} \setminus i$, item i is ranked better, if its team size—that is, $|\mathcal{E}^i|$ where $\mathcal{E}^i$ is defined as the set of entrepreneurs forming item i's team —is greater than 1. If so, item i's *boolean* team value is set to *true*, that is, item i is ranked better than other items whose boolean team value is set to *false* (or lower analogously). Due to the fact that the Team preference does not rely upon any user specified input, this preference's user specified set of attributes is defined as $\mathcal{L}^{\text{T}} = \emptyset$.

(vii) Life Cycle (π_{LC}): The input to this preference is a user selected subset of a predefined set of an item's life cycle stages—that is, $\mathcal{L}^{\text{LC}} = \mathcal{C}^u \subset \mathcal{C}$—that are matched against item i's defined life cycle. Matching itself is accomplished by the calculation of the Jaccard correlation coefficient between the set of user specified life cycle stages and item i's actual life cycle stage. In the context of the domain of all items $\mathcal{I} \setminus i$, item i is ranked better, if its calculated Jaccard correlation coefficient is higher than the ones of the other items or lower analogously.

Implementation of the Borda Count In contrast to other collective decision-making methods (such as the plurality rule), the Borda count does not only take the preferred alternative among each agent into account, but rather utilizes all alternatives under the constraint that each agent needs to rank *all* alternatives from most- to least preferred. Each alternative on an agent's ranked list is assigned an integer number, that is, the most preferred alternative is assigned the highest integer n (commonly, the number of alternatives n of the set of alternatives A is either defined as $n = |A|$ or $n = |A| - 1$). Finally, each alternative's merged score is calculated by summation of each individual alternative's scores among all agents. The result is a ranked list of alternatives, exposing preferences among all agents. [Garcia-Lapresta and Martinez-Panero, 2002, p. 167] However, this basic approach requires equality among the importance of agents, thus making this approach only partly applicable to the characteristics of the present work.

In the context of the previously introduced agents, each preference may be seen as an agent and therefore, the amount of agents in the system is equal to the amount of preferences selected by the user. However, the aforementioned original version of the Borda count is not applicable to the present work, due to its constraint on the equality among agents. This implication arises from the fact that each preference's importance may not be equal among other preferences, that is, a user may consider some preferences more important than other preferences. Therefore, the authors propose a *Weighted Borda count* that utilizes all user selected preferences' individually calculated ranked item lists as input and ranks/merges the items based on the selected preferences' weights. The following paragraphs explain this recommendation procedure in great detail.

Once each item $i \in \mathcal{I}^{\pi}$ is ranked on the basis of a certain preference $\pi \in \Pi$, the next step of the knowledge-based recommender is stated as the process of assigning scores to each item i according to the Borda count.

As already introduced, the Borda count assigns the highest positive Integer, that is, $|\mathcal{I}^{\pi}|$, to item $i \in \mathcal{I}^{\pi}$ with the highest rank ρ_{max}. In general, this means that the Borda score of item i is the number of all items (including item i) that are not preferred over i. Furthermore, as introduced previously, there exists the possibility of indifferent items $i \sim_{\pi} j$—which is also referred to as *ties*—that need to be addressed by the knowledge-based recommender. Therefore, the following equation shows the calculation of a Borda score for an item i among preference π in the domain of a set of items $\mathcal{I}^{\pi}$:

$$B_{i,\pi,\mathcal{I}^{\pi}} = |\{j \in \mathcal{I}^{\pi} : i \lesssim_{\pi} j\}| \tag{2.36}$$

Subsequently, the Borda count needs to assign scores to each item in $\mathcal{I}^{\pi}$ and store the *item / value* tuples as ordered pairs of the form

$$\mathcal{I}_B^{\pi} = \{(i, B_{i,\pi,\mathcal{I}^{\pi}}) : i \in \mathcal{I}^{\pi}, B_{i,\pi,\mathcal{I}^{\pi}} \in \{x \in \mathbb{Z}_{>0} : x \leq |\mathcal{I}^{\pi}|\}\} \tag{2.37}$$

where $\mathcal{I}_B^{\pi}$ is stated as the set of ordered *item / borda score* pairs and $B_{i,\pi,\mathcal{I}^{\pi}}$ is stated as particular Borda score for item i among preference π in the domain of $\mathcal{I}^{\pi}$.

In the case of ties, the knowledge-based recommender distributes the same Borda score $B_{i,\pi,\mathcal{I}^{\pi}}$ among all indifferent items of a certain rank. However, as Equation (2.36) specifies, the amount of indifferent items sharing one specific or *tied* rank decreases the available Borda scores. More generically, the procedure of addressing ties in the course of the Borda count is exemplified as follows:

$$B_{x+1} = B_x - \left|\mathcal{I}^{B_x}\right| \tag{2.38}$$

where B_{x+1} is specified as the next (decreasing) Borda score, B_x is stated as the current Borda score and $\mathcal{I}^{B_x} \subset \mathcal{I}_B^{\pi}$ is defined as the set of items sharing the current Borda score B_x. The reader is referred to Figure 2.3 for an exemplifying illustration on the calculation of Borda scores and the processing of ties in the context of interval B_n (the highest Borda score $|\mathcal{I}_B^{\pi}| = 255$) and B_1 (the lowest Borda score equal to 1).

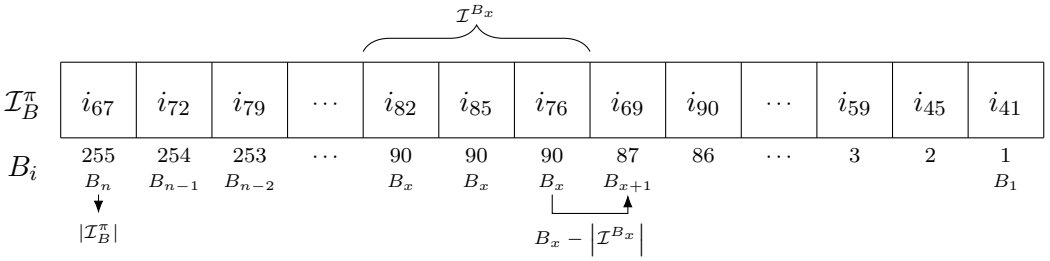


Figure 2.3: Example on the processing of tied Borda scores in the context of the knowledge-based recommender's preference calculation.

Weighted- and Merged Borda Count The final step of knowledge-based recommender in the context of preferences is stated as the calculation of a ranked list of items merged among all user specified preference lists maximizing the user's utility. A user u's selected preferences $\Pi^u \subset \Pi$ are weighted according to a user specified *importance* weighting factor w_π that expresses user u's utility towards a certain preference $\pi \in \Pi^u$. The weights of all user specified weights w_π among $\pi \in \Pi^u$ are summed up to the number 1, that is, $\sum_{\pi \in \Pi^u} w_\pi = 1$. Ultimately, let the merged score of an item $i \in \mathcal{I}$ among the user specified set of preferences Π^u for the domain of items $\mathcal{I}$ be defined as follows:

$$S^{KB}_{i,\Pi,\mathcal{I}^u} = \sum_{\pi \in \Pi^u} B_{i,\pi,\mathcal{I}} \cdot w_\pi \tag{2.39}$$

where $S^{KB}_{i,\Pi^u,\mathcal{I}} \in \mathbb{R}_{>0}$.

The semantics of an item i's score are defined as follows: Let the merged rank be stated as *strict weak order* $<_\Pi$ among the set of preferences Π, then the ranking between two items i and j is defined as follows:

$$i <_\pi j \Leftrightarrow S^{KB}_{i,\Pi,\mathcal{I}} > S^{KB}_{j,\Pi,\mathcal{I}} \tag{2.40}$$

that is, item i expresses a higher utility than item j, if the merged Borda score $S^{KB}_{i,\Pi,\mathcal{I}}$ is higher than $S^{KB}_{j,\Pi,\mathcal{I}}$.

Based on the fact that the knowledge-based recommender does not solely address a single item but is rather applied on a set of items $\mathcal{I}$, the output of the present recommender is stated as a ranked set of items. Therefore, a new data structure needs to be introduced that expresses the rank of an item among a set of other items. More formally, let $\mathcal{I}^{\Pi^u}$ be denoted as a set of ordered pairs $\left(i, S^{KB}_{i,\Pi^u,\mathcal{I}}\right)$, where i is stated as item $i \in \mathcal{I}$ and $S^{KB}_{i,\Pi^u,\mathcal{I}}$ is defined as item i's merged Borda score among all user specified preferences Π^u in the domain of a set of items $\mathcal{I}$. The mathematical notation is specified as follows:

$$\mathcal{I}^{\Pi^u} = \left\{\left(i, S^{KB}_{i,\Pi^u,\mathcal{I}}\right) : i \in \mathcal{I}, S^{KB}_{i,\Pi^u,\mathcal{I}} \in \mathbb{R}_{>0}\right\} \tag{2.41}$$

whereas—due to illustration purposes—the knowledge-based recommender sorts the calculated set of ordered pairs $\mathcal{I}^{\Pi^u}$ according to $S^{KB}_{i,\Pi^u,\mathcal{I}}$ in descending order, that is, items of the highest utility to the user are shown first.

Sequential Recommendation Process

Finally, it shall be emphasized that in the course of the present work's implementation of the knowledge-based recommender, filters and preferences may be applied independently of each other. However, the following listing of constraints need to be considered:

(i) No filters or preferences selected by the user: In this case, the *whole* set of items $\mathcal{I}$ is returned by the knowledge-based recommender. However, it shall be noted that this particular set of items does not possess a specific order, that is, the logical order specified by the underlying database is considered. As a consequence, the returned set of items is specified as $\mathcal{I}$.

(ii) Only filters selected by the user: This particular case specifies that the *filtered* set of items $\mathcal{I}^{\Phi^u}$ —which depends on a user specified set of filters Φ^u and their corresponding attributes—is returned by the knowledge-based recommender. However, analogously to the case of no selected filters or preferences, this particular set of items does not impose a specific order, that is, the logical order specified by the underlying database is considered. Subsequently, the returned set of items is specified as $\mathcal{I}^{\Phi^u}$.

(iii) Only preferences selected by the user: In contrast to the previous case, the *whole* set of items $\mathcal{I}$ is returned by the knowledge-based recommender. However, this particular set of items is ranked according to the user specified preferences Π^u and their corresponding attributes. Therefore, the returned ranked set of items is specified as $\mathcal{I}^{\Pi^u}$.

(iv) Filters and preferences selected by the user: This specific case utilizes the whole functionality of the knowledge-based recommender. First, the user specified set of filters Φ^u is applied on the whole set of items $\mathcal{I}$ in the course of the present recommender's filtering algorithms. Afterwards, the set of items $\mathcal{I}^{\Phi^u}$ calculated in the previous step is taken as input to the knowledge-based recommender's preference calculation functions that rank this particular set of items according to the user specified preferences Π^u and their corresponding attributes. Subsequently, based on Equation (2.41), let the returned ranked set of items $\mathcal{I}^{\Phi^u\Pi^u}$ in the context of a user specified combination of filters and

preferences be mathematically denoted as:

$$\begin{aligned}\mathcal{I}^{\Phi^u \Pi^u} &= \bigcup_{i \in \mathcal{I}^{\Phi^u}} \left(i, S^{KB}_{i,\Pi^u,\mathcal{I}^{\Phi^u}} \right) \\ &= \bigcup_{i \in \bigcap_{\phi \in \Phi^u} \phi(\mathcal{I},\mathcal{L}^\phi)} \left(i, \sum_{\pi \in \Pi^u} B_{\pi,i,\mathcal{I}^{\Phi^u}} \cdot w_\pi \right) \\ &= \bigcup_{i \in \bigcap_{\phi \in \Phi^u} \phi(\mathcal{I},\mathcal{L}^\phi)} \left(i, \sum_{\pi \in \Pi^u} \pi(i, \mathcal{L}^\pi, \mathcal{I}^{\Phi^u}) \cdot w_\pi \right)\end{aligned} \tag{2.42}$$

Explanatory example

As of the complexity of the knowledge-based recommender, the following explanatory example visualizes the filters' and preferences' affect on a set of items. Table 2.3 shows this example's set of items:

Table 2.3: Attributes of example items

Attribute	Item a	Item b	Item c	Item d
Products	$\{p_{a_1}, p_{a_2}\}$	$\{p_{b_1}\}$	$\{p_{c_1}, p_{c_2}, p_{c_3}\}$	$\emptyset$
Investments	$\{z_{p_{a_1}}\}$	$\{z_{p_{b_1}}\}$	$\{z_{p_{c_1}}, z_{p_{c_3}}\}$	$\emptyset$
Investment Amount	$z_{p_{a_1}}$: €20000	$z_{p_{b_1}}$: €5000	$z_{p_{c_1}}$: €60000	$\emptyset$
			$z_{p_{c_3}}$: €50000	
Product Interest	$\{Smartphone, IT\}$	$\{Biology\}$	$\{IT\}$	$\{Food\}$
Date of creation	06.10.2016	24.11.2017	04.04.2015	27.12.2016

Additionally, Table 2.4 shows the user selected filter- and preference settings:

Table 2.4: User selected filter- and preference settings

Filter / Preference	Symbol	$\mathcal{L}^\phi$ / $\mathcal{L}^\pi$	Importance
Investment Range	ϕ_{IR}	$MinAmount = €5000$	-
		$MaxAmount = €55000$	
Product Interest	π_{ProdI}	$\{IT, Biology, Smartphone\}$	0.8
Date	π_{D}	$\emptyset$	0.3

Due to the fact that one filter and two preferences are selected, the knowledge-based

recommender first applies the filters and utilizes the calculated set of items as basis for the ranking according to the preferences. The Investment Amount filter validates whether a certain item fulfils the constraint that there exists at least one investment for at least one product having an investment amount ranging within the interval $MinAmount$ to $MaxAmount$. This constraint holds for the following set of items: $\{a, b, c\}$.

The next step of the knowledge-based recommender is stated as the ranking of the filtered items according to each of the user specified preferences and calculate Borda scores. Finally, the merged Borda score is computed. Table 2.5 illustrates this process:

Table 2.5: Ranking, Borda score and final merged Borda score ranking

Item	π_{PRODI} Ranking		π_{D} Ranking		**Merged Borda score**
	$\rho_{\pi_{\text{PRODI}},i,\mathcal{I}^{\Phi^u}}$	$B_{\pi_{\text{PRODI}},i,\mathcal{I}^{\Phi^u}}$	$\rho_{\pi_{\text{D}},i,\mathcal{I}^{\Phi^u}}$	$B_{\pi_{\text{D}},i,\mathcal{I}^{\Phi^u}}$	$S^{KB}_{i,\Pi^u,\mathcal{I}}$
a	1	3	2	2	$2.\overline{72}$
b	2	2	1	3	$2.\overline{27}$
c	2	2	3	1	$1.\overline{72}$

As may be implied by Table 2.5, the final merged Borda scores for items a, b and c are calculated according to Equation (2.42) as follows:

$$S^{KB}_{a,\Pi^u,\mathcal{I}} = \frac{B_{\pi_{\text{PRODI}},a,\mathcal{I}^{\Phi^u}} \cdot w_{\pi_{\text{PRODI}}} + B_{\pi_{\text{D}},a,\mathcal{I}^{\Phi^u}} \cdot w_{\pi_{\text{D}}}}{w_{\pi_{\text{PRODI}}} + w_{\pi_{\text{D}}}} = \frac{3 \cdot 0.8 + 2 \cdot 0.3}{0.8 + 0.3} = 2.\overline{72}$$

$$S^{KB}_{b,\Pi^u,\mathcal{I}} = \frac{B_{\pi_{\text{PRODI}},b,\mathcal{I}^{\Phi^u}} \cdot w_{\pi_{\text{PRODI}}} + B_{\pi_{\text{D}},b,\mathcal{I}^{\Phi^u}} \cdot w_{\pi_{\text{D}}}}{w_{\pi_{\text{PRODI}}} + w_{\pi_{\text{D}}}} = \frac{2 \cdot 0.8 + 3 \cdot 0.3}{0.8 + 0.3} = 2.\overline{27}$$

$$S^{KB}_{c,\Pi^u,\mathcal{I}} = \frac{B_{\pi_{\text{PRODI}},c,\mathcal{I}^{\Phi^u}} \cdot w_{\pi_{\text{PRODI}}} + B_{\pi_{\text{D}},c,\mathcal{I}^{\Phi^u}} \cdot w_{\pi_{\text{D}}}}{w_{\pi_{\text{PRODI}}} + w_{\pi_{\text{D}}}} = \frac{2 \cdot 0.8 + 1 \cdot 0.3}{0.8 + 0.3} = 1.\overline{72}$$

2.3.5 Social Trust Recommendation

The basic idea of the social recommender associated with the present work is based on the set of trust relationships among users. A user may *follow* another user, that is, the expression of a unidirectional trust relation between the users. As a consequence, information about the fellowship of users highly affects the neighbourhood selection process of the present recommendation algorithm. The social recommender's recommendation algorithm first finds users similar to a target user based on the distribution of the target user's trust. Subsequently, the algorithm calculates the utility of a target item based on the trust relationships between the target user, the peer users and the peer users' rankings of the target item. In fact, the trust relationship modelled in the course of the social recommender may not be regarded a similarity measure but rather be defined as a special kind of neighbourhood selection, that is, a user $u' \neq u$

is included in user u's neighbourhood, if u expresses trust to u'. More formally, let $N_T(u)$ be donated as neighbourhood of user u, where

$$N_T(u) = \{u' \in U \setminus u : u' \in \mathcal{T}^u\} \tag{2.43}$$

and $\mathcal{T}^u \subset \mathcal{U} \setminus u$ is defined as the set of users trusted by user u.

The mathematical notations of the corresponding recommendation algorithms are illustrated as follows:

Trust Likes algorithm (SR^L)

Item recommendations are calculated based on the liked items of the target user's neighbourhood and weighted by item likes of the corresponding peer users. More formally, the social recommender SR^L is specified as follows:

$$S(u,i) = \frac{\sum\limits_{u' \in N_T(u)} r^L(u',i)}{|N_T(u)|} \tag{2.44}$$

where u is defined as the target user, u' is stated as the investigated (peer) user, i is specified as the target item and $N_T(u)$ is defined as the user neighbourhood of user u.

Trust Investments algorithm (SR^I)

Item recommendations are calculated based on the invested items of the target user's neighbourhood and weighted by item investments of the corresponding peer user. More formally, the social recommender SR^I is specified as follows:

$$S(u,i) = \frac{\sum\limits_{u' \in N_T(u)} r^I(u',i)}{|N_T(u)|} \tag{2.45}$$

where u is defined as the target user, u' is stated as the investigated (peer) user, i is specified as the target item and $N_T(u)$ is defined as the user neighbourhood of user u.

Trust Clicks algorithm (SR^C)

Item recommendations are calculated based on the clicked items of the target user's neighbourhood and weighted by item clicks of the corresponding peer user. More formally, the social recommender SR^C is specified as follows:

$$S(u,i) = \frac{\sum\limits_{u' \in N_T(u)} r^C(u',i)}{|N_T(u)|} \tag{2.46}$$

where u is defined as the target user, u' is stated as the investigated (peer) user, i is specified as the target item and $N_T(u)$ is defined as the user neighbourhood of user u.

2.3.6 Hybrid Recommendation

The general concept of the hybrid recommender is stated as a *pipelined cascading* approach based on the target user's item recommendations calculated by the knowledge-based recommender and refined by the knowledge-based item recommendations of all other users in the neighbourhood of the target user. First, the hybrid recommender calculates the similarity between the target user and each of the other users in the neighbourhood on the basis of the item recommendation lists returned by the knowledge-based recommender. The utilized similarity function is based on the *Kendall tau distance*. Based on the calculated similarity scores, the target user's neighbourhood is formed utilizing a certain threshold. Finally, the hybrid recommender calculates the merged score of an item on the basis of its individual *user scores* assigned by the knowledge-based recommender, weighted by the particular user's similarity score and summed up over all users of the neighbourhood. Subsequently, this process is applied to all items of the platform leading to a ranked item recommendation list.

The following approaches introduce the reader to the hybrid recommender's similarity function, the formation of the target user's neighbourhood and, ultimately, the calculation of item recommendations. Due to the fact that some of the following definitions are derived from- and based on the research conducted in the course of the knowledge-based recommender, the reader is referred to subsection 2.3.4 for a detailed elaboration on that matter.

Similarity Function

A similarity function is a measurement that values the similarity between two objects as real number of the interval $[0, 1]$, whereas 1 indicates that the objects of interest are considered to be the same and, analogously, 0 indicates that these objects completely differ from each other. In the context of the hybrid recommender, the objects of interest are defined as the target user u on the one side and a particular peer user $u' \in \mathcal{U} \setminus u$ on the other side. With the aim of calculating the similarity between these two users, the present recommender bases its comparisons on the knowledge-based recommender's final lists of ranked items for each of the users u and u'—that is, $\mathcal{I}^{\Pi^u}$ and $\mathcal{I}^{\Pi^{u'}}$. Therefore, the *distinguishing feature* between the two users is stated as the *internal item rankings* of these two sets. However, until now, the knowledge-based recommender is only applicable on the currently authenticated user of the platform, that is, no user specified data on certain recommendation settings is stored in memory. This circumstance changes in the course of the hybrid recommender.

In order to calculate the similarity between a target user u and another user u' based on their recommendation settings utilized in the course of the knowledge-based recommender—that is, *Filters* and *Preferences*—the hybrid recommender utilizes a *user-based recommendation profile* that stores these settings in memory (the reader is referred to the present chapter's model subsection 2.3.1 for a detailed representation of the profile's attributes). However, due to the pipelined cascading hybrid's requirement on a *finalized* set of items to elaborate upon, that is, in the course of the hybrid recommender, no items may be removed or added between each calculation iteration, only the Preference settings are stored in each user's recommendation profile. As a consequence, the set of items needed for the calculation of user similarity and the

computation of recommendations, is defined as the whole set of items held by the platform—that is, $\mathcal{I}$. In order to calculate the similarity between two users based on the mentioned constraints, a similarity function capable of identifying and measuring the differences between the rankings of two item lists is needed.

One possible candidate fitting to the present modelling constraints is stated as the *Kendall tau distance* (the reader is referred to the background subsection 2.1.3 for a detailed representation and the calculation of the Kendall tau distance), which calculates the distance between two ranked lists based on their number of discordant pairs, that is, the number of pairs having a different order among the lists. However, the original implementation does not consider the existence of ties, that is, items of the same rank between each of the lists. In order to solve this issue, the present implementation of the Kendall tau distance does not solely take the rank of an item in each list into account, but rather utilizes the score $S^{KB}_{i,\Pi^u,\mathcal{I}}$ calculated by the knowledge-based recommender for item $i \in \mathcal{I}$ in the context of user u's predefined recommendation settings. The plausibility of this approach is backed by the final score calculations of the knowledge-based recommender, which rely on weighting factors that sum up to the integer 1. This design decision leads to the fact that the calculated ranks increase or decrease *linearly* and share a maximum rank defined as $|\mathcal{I}|$. Therefore, these lists of ranked items are considered comparable as long as both of them contain the same items. Subsequently, a tie between the knowledge-based recommender's scores $S^{KB}_{i,\Pi^u,\mathcal{I}}$ and $S^{KB}_{i,\Pi^{u'},\mathcal{I}}$ of users u and u' for item $i \in \mathcal{I}$ emerges, if $S^{KB}_{i,\Pi^u,\mathcal{I}} = S^{KB}_{i,\Pi^{u'},\mathcal{I}}$ holds. However, the scenario of ties needs to be considered in the calculation of the similarity measure as well, in order to be addressed properly.

Kendall tau is defined as distance between two ranked lists or rankings. A ranking for a user u is represented as ρ^u, that is, each item i possesses an assigned rank ρ^u_i, whereas a high rank expresses that item i is of high utility in the context of the target user u (low utility analogously). Consequently, a discordant pair is defined as follows: Let (i, j) be denoted as discordant pair between two lists ρ^u and $\rho^{u'}$ of users u and u', if one of the following conditions holds:

$$\begin{aligned}
&\rho^u_i < \rho^u_j \wedge \rho^{u'}_i > \rho^{u'}_j \\
&\rho^u_i > \rho^u_j \wedge \rho^{u'}_i < \rho^{u'}_j \\
&\rho^u_i = \rho^u_j \wedge \rho^{u'}_i \neq \rho^{u'}_j \\
&\rho^u_i \neq \rho^u_j \wedge \rho^{u'}_i = \rho^{u'}_j
\end{aligned} \tag{2.47}$$

Subsequently, the calculation of the Kendall tau distance is conducted by computing the number of discordant item pairs between two lists, normalized by the total number of list items $|\rho^u| = |\rho^{u'}|$, more formally:

$$\mathcal{K}(\rho^u, \rho^{u'}) = \frac{n_d}{n \cdot (n-1)/2} \tag{2.48}$$

where ρ^u and $\rho^{u'}$ are stated as user u's- and user u''s list of ranked items calculated by the knowledge-base recommender utilizing the users' individual Preference settings, n_d is defined

as the number of discordant pairs between ρ^u and $\rho^{u'}$ and $n = |\rho^u| = |\rho^{u'}| = |\mathcal{I}|$ is specified as the size of each list.

Finally, in order to utilize the Kendall tau distance as similarity measure between two users u and u', $\mathcal{K}(\rho^u, \rho^{u'})$ needs to be subtracted from 1, more formally: art

$$k_{u,u'} = sim_H(u, u') = 1 - \mathcal{K}(\rho^u, \rho^{u'}) = 1 - \frac{n_d}{n \cdot (n-1)/2} \tag{2.49}$$

As of the complexity of the presented similarity measure, the following explanatory example visualizes the calculation of the similarity between two users u and u' utilizing the Kendall tau similarity measure. Table 2.6 shows this example's set of items combined with the user-based rank calculated by the knowledge-based recommender.

Table 2.6: User rankings per item

Rankings per user	Item a	Item b	Item c	Item d
ρ^u	4	3	3	1
$\rho^{u'}$	2	3	4	1

The first step of calculating the similarity between user u and u' is stated as determining the discordant pairs between the users' lists of rankings. The reader is referred to Table 2.7 for an illustration on this calculation process.

Table 2.7: Calculation of discordant pairs

Pair	ρ_i^u vs. ρ_j^u	$\rho_i^{u'}$ vs. $\rho_j^{u'}$	Count (n_d)
(a, b)	$4 > 3$	$2 < 3$	$\times$
(a, c)	$4 > 3$	$2 < 4$	$\times$
(a, d)	$4 > 1$	$2 > 1$	
(b, c)	$3 = 3$	$3 < 4$	$\times$
(b, d)	$3 > 1$	$3 > 1$	
(c, d)	$3 > 1$	$4 > 1$	

Finally, the calculation of the similarity score $k_{u,u'}$between user u and u' is conducted by the utilization of $\mathcal{K}(\rho^u, \rho^{u'})$ and accomplished as follows:

$$k_{u,u'} = sim_H(u, u') = 1 - \mathcal{K}(\rho^u, \rho^{u'}) = 1 - \frac{n_d}{n \cdot (n-1)/2} = 1 - \frac{3}{4 \cdot (4-1)/2} = 0.5$$

Neighbourhood Formation

With the aim of improving recommendation quality and reducing calculational overhead, the hybrid recommender implements a neighbourhood function $N_H(u)$ that reduces the set of peer users taken as input to the recommendation algorithm. The neighbourhood function itself is based on a threshold that a peer user's similarity score needs to reach or surpass in order for the peer user to be included into the neighbourhood.

Nevertheless, Jannach et al. [2010, pp. 17–18] state that choosing the appropriate threshold is a non-trivial task. If the threshold is set too high, the coverage is reduced considerably. In contrast, if the threshold is chosen too low, the size of the neighbourhood is only partially reduced (the reader is referred to the background subsection 2.1.3 for a detailed discussion on neighbourhood formation). However, previous research by Christian Ohrfandl indicates that the future ratings sets and the amount of users held by the platform is stated as being considerably low. Due to these reasons, the threshold is set to the real number 0.7. More formally, the neighbourhood of the hybrid recommender's recommendation algorithm is formed as follows:

$$N_H(u) = \{u' \in U \setminus u : sim_H(u, u') \geq 0.7\} \tag{2.50}$$

Hybrid Recommendation Algorithm (S^H)

The hybrid recommender's final step is stated as the calculation of a ranked list of items merged among the peer users' recommendation settings maximizing the target user's utility. In order to calculate the hybrid recommender's final rank for an item i in the context of the target user u, each peer user $u' \in N_H(u)$'s knowledge-based recommender's score $S^{KB}_{i,\Pi^{u'},\mathcal{I}}$ is weighted by u''s similarity score $k_{u,u'}$ and averaged on the basis of $|N_H(u)|$. Ultimately, let the hybrid recommender's score of an item $i \in \mathcal{I}$ be defined as follows:

$$S^H_{i,u,N_H(u),\mathcal{I}} = \sum_{u' \in N_H(u)} \frac{k_{u,u'} \cdot S^{KB}_{i,\Pi^{u'},\mathcal{I}}}{|N_H(u)|} \tag{2.51}$$

where $S^H_{i,u,N_H(u),\mathcal{I}} \in \mathbb{R}_{>0}$.

The semantics of an item i's score are defined as follows: Let the hybrid recommender's rank be stated as *strict weak order* $<_{S^H}$ among the set of items $\mathcal{I}$, then the ranking between two items i and j is defined as follows:

$$i <_{S^H} j \Leftrightarrow S^H_{i,u,N_H(u),\mathcal{I}} > S^H_{j,u,N_H(u),\mathcal{I}} \tag{2.52}$$

that is, item i expresses a higher utility than item j, if the hybrid recommender's score $S^H_{i,u,N_H(u)}$ is higher than $S^H_{j,u,N_H(u)}$.

Based on the fact that the hybrid recommender does not solely address a single item but is rather applied on a set of items $\mathcal{I}$, the output of the present recommender is stated as a ranked set of items. Therefore, a new data structure needs to be introduced that expresses the rank of an item among a set of other items. More formally, let $\mathcal{H}^{u,N_H(u),\mathcal{I}}$ be denoted as a set of

ordered pairs $(i, S^H_{i,u,N_H(u),\mathcal{I}})$, where i is stated as item $i \in \mathcal{I}$ and $S^H_{i,u,N_H(u),\mathcal{I}}$ is defined as item i's hybrid recommender's score among user u and its neighbourhood $N_H(u)$ in the domain of all items $\mathcal{I}$. Finally, the mathematical notation is specified as follows:

$$\mathcal{H}^{u,N_H(u),\mathcal{I}} = \left\{ \left(i, S^H_{i,u,N_H(u),\mathcal{I}} \right) : i \in \mathcal{I}, S^H_{i,u,N_H(u),\mathcal{I}} \in \mathbb{R}_{>0} \right\} \tag{2.53}$$

whereas—due to illustration purposes—the hybrid recommender sorts the calculated set of ordered pairs $\mathcal{H}^{u,N_H(u),\mathcal{I}}$ according to $S^H_{i,u,N_H(u),\mathcal{I}}$ in descending order, that is, items of the highest utility to the target user u are shown first.

2.3.7 Recommender System Prototype

The main purpose of the proposed recommendation system's software prototype is to provide a personalized recommendation experience to the user of the system. Therefore, all mathematically designed recommendation algorithms are implemented in a *parallelized* approach, that is, each of these recommenders are separately applicable by the user. It has been decided that no constraints on whether one particular recommendation algorithm shall be preferred over another or whether various algorithms shall be combined in order to enhance the likelihood of improving the user's satisfaction, need to be fulfilled. These considerations are being elaborated in the recommender system's evaluation specialization topic, which is not part of the present work. Figure 2.4 illustrates the parallelized recommendation architecture of the recommender system prototype.

One major design choice of implementing the recommendation system's software prototype is stated as the adaptation of a *plugin-based software architecture* that allows for dynamic extensions of the recommendation system (such as inclusion of new recommendation algorithms). Therefore and due to performance aspects in terms of computational calculation, the prototype's *recommendation engine* and the *user interface* are separated into two components implemented as independent pieces of software. While the recommendation engine is developed utilizing the *Java*[3] programming language, *Angular*[4] is chosen for crafting the web-based user interface. The communication between both components is accomplished by the utilization of RESTful[5] WEB services.

Finally, backed by the research conducted in the course of the previous study by Christian Ohrfandl and nowadays' zeitgeist, one of the core principles utilized for developing the web-based user interface is stated as *responsive design*, that is, a user interface that actively adapts to the platform it is consumed on (such as mobile devices or desktop computers). The reader is referred to Figures 2.5 and 2.6 for a visual representation of a user's desktop- and an item's mobile view.

[3] Java programming language: https://java.com/de/download/

[4] Angular web application platform: https://angular.io/

[5] Z. Shelby [2012]. *Constrained RESTful Environments (CoRE) Link Format.* RFC 6690. RFC Editor, pp. 1–22.

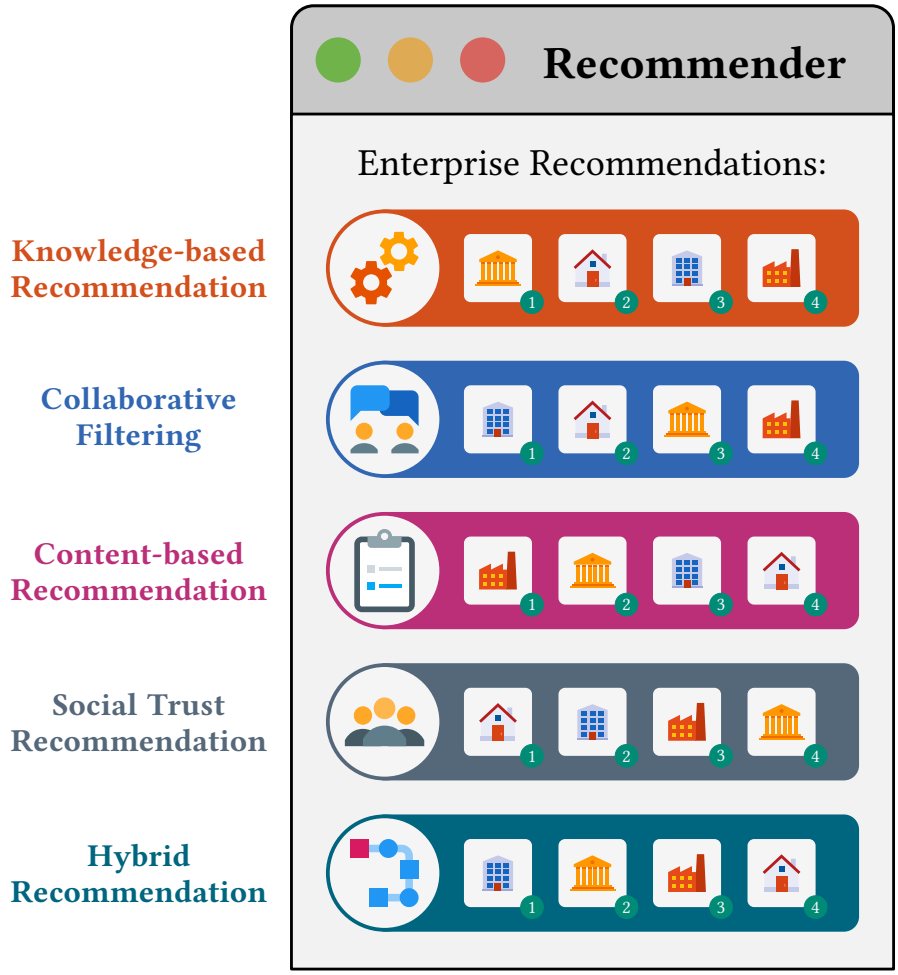


Figure 2.4: General architecture of the recommender system prototype based on parallelized types of recommenders.

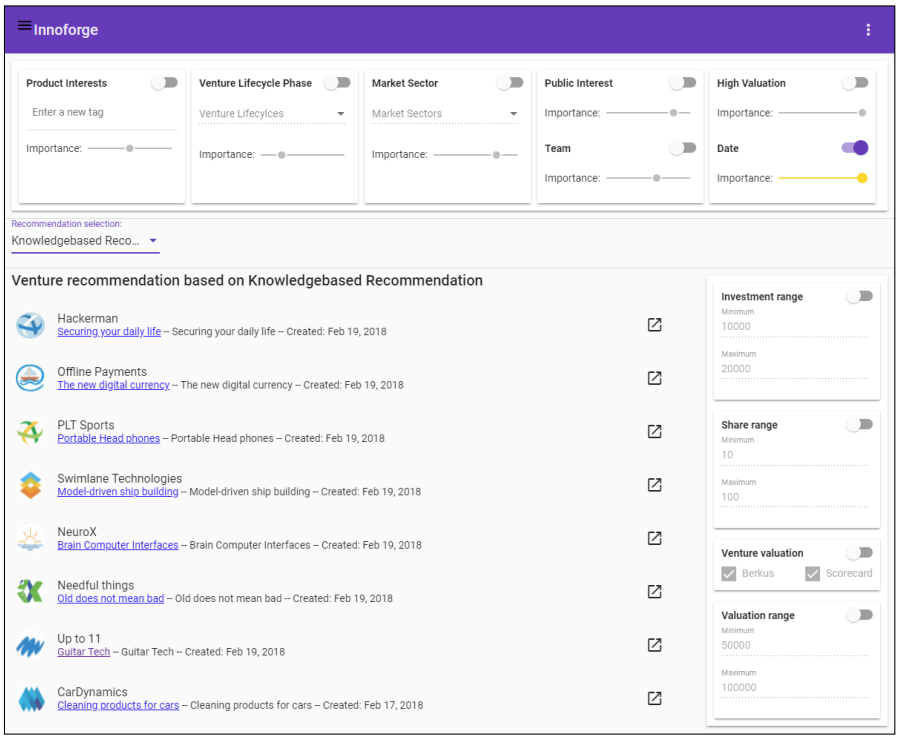


Figure 2.5: The user view of the recommender system prototype (Preferences & Filters).

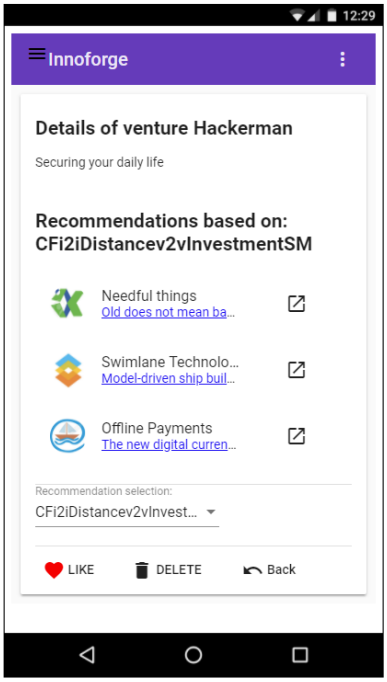


Figure 2.6: Responsive illustration of the recommender system's prototype's item view.

2.4 Answers to the Research Questions

The objective of this section lies in the discussion of the present chapter's research questions (the reader is referred to subsection 2.2.1 of the methodology section) on the basis of the obtained results.

Which recommendation algorithms and -techniques shall be considered in a computational recommendation system in the domain of early-stage enterprise investment, in order to guarantee highly personalized recommendations for investors?

Research conducted in the course of the previous study by Christian Ohrfandl indicates many different use cases a potential recommender system in the mentioned domain needs to address. The crucial aspects of these findings include but are not limited to personalized recommendation of items in the course of other users' rating behaviour, viewing the most interesting items based on a user's personal preferences, exploring items that are similar to items the user interacted with in the past and expressing trust among other users. Independently, all of these use cases share the concept of personalization in its own individual way. With the aim of addressing these personalization concepts, the present work's recommender system unites multiple types of recommendation algorithms that, ultimately, address all of the mentioned use cases. These algorithms include the following recommendation techniques: Collabortive filtering, content-based-, knowledge-based, social- and hybrid recommendation.

How can the cold start problem in the context of computational recommendation systems in the domain of early-stage enterprise investment, be addressed?

The cold start problem in the domain of recommendation systems arises if there is not enough or no ratings data for items available. Subsequently, recommendation architectures that build their recommendations on ratings data are not applicable any longer (such as collaborative filtering techniques). Especially the domain of early-stage enterprise investment faces this circumstance, for it is this domain's users who share the characteristics of *one time buyers*, that is, users who do not intend to buy items on a frequent basis. In order to overcome this issue, the present work implements a knowledge-based recommender that does not rely on ratings data and therefore enables the user to utilize the recommendation system even if there is not sufficient ratings data available.

Which constraints does a software prototype of the computational recommendation system need to fulfil, in order to guarantee technical- and algorithmic feasibility

Monolithic sofware architectures generally face the problem of hardly scalable resources due to the chosen system architecture. Nowadays' recommender systems show that their domains of application generate considerably large amounts of data that needs to be processed in a

considerably short amount of time. Due to these reasons, the software architecture of the present work's software prototype was designed highly scalable on the basis of a micro services architecture. One of the constraints of this approach is stated as the split of the recommendation engine and the user interface enabling independent scaling of each of these components if needed. Independently of the recommender system's performance in terms of computational calculation speed, today's zeitgeist shows that a revolution in human computer interfaces changed the way users interact with the digital world. However, as revolutionary as these changes might be, in a time that is mainly characterized by the efficiency of processes, it is these very changes in human computer interaction—the children of this revolution—users are not capable of reviving from: Mobile devices. In order to address this circumstance, the user interface of the recommendation system is developed utilizing responsive design functionality that actively adapts to the platform it is consumed on (such as mobile devices or desktop computers).

Conclusion

The purpose of the present chapter *Recommender Systems for Early-Stage Enterprise Investment* was to conceptualize a recommender system in the domain of early-stage enterprise investment based on the findings of co-author Christian Ohrfandl's specialization topic. The crucial aspects of these findings include but are not limited to personalized recommendation of items in the course of other users' rating behaviour, viewing the most interesting items based on a user's personal recommendation settings, exploring items that are similar to items the user interacted with in the past and expressing trust among other users. Based on these constraints, a recommender system was modelled that unites the following set of recommenders: *Collaborative filtering*, *content-based-*, *knowledge-based-*, *social-* and *hybrid* recommendation algorithms. Finally, the conceptual model of this very recommender system has been implemented as a highly scalable, plugin-based software prototype that might be easily extended by different recommendation algorithms in the course of future research.

To the authors' best knowledge, very few publications are available in the literature that combine the domain of early-stage enterprise investment with the domain of recommender systems. Therefore, this thesis's present chapter may not only be seen as modest contribution to the scientific research domain of recommender systems, but also be valued as novel approach to objectively transform investors' rules of thumb or gut feelings into transparent decision-making processes utilized in the course of a recommendation system. In particular, the utilized comprehensive approach, that is, the inclusion of a wide range of recommenders with the aim of maximizing the user's utility independently of the use case, enables the collection of various amounts of meta data that may be taken as basis for future research. Nevertheless, there are certain factors limiting the present research.

A limitation of the present work may be seen in the fact that the design choices the present recommender is implemented upon were not evaluated in the course of the present chapter. Therefore, the authors are not able to make any significant statements on the verification of the overall recommendation quality in terms of user satisfaction. Additionally, the selected and implemented types of recommendation algorithms are based on- and therefore also limited to the research findings of the previous study by Christian Ohrfandl. However, the outcomes of the present work generate a wide range of future research possibilities.

One of the most interesting opportunities for future research arises from the data obtained after users' frequent use of the platform. The gathered data may be further analysed in order to gain insights on potential relationships among quantifiable user behaviour or may even lead to the finding of generally valid success factors of early-stage enterprises. Independently, additional research may address the mentioned limitations of the present work. In particular, qualitative- or quantitative evaluations of recommendation quality in terms of user satisfaction may answer the question, whether the implemented design decisions improve a user's utility when using the system. In fact, it is precisely this very evaluation that is researched by co-author Johannes Luef in the course of his specialization topic *Evaluation*.

CHAPTER 3

Evaluation

3.1 Background

The following chapter 3 presents the evaluation of the recommender systems designed in the previous chapter 2. First, the concepts of evaluating recommender systems are introduced, following an excursus on information retrieval. Furthermore, the evaluation metrics, conventions, and notation are presented. Finally, the finding of the evaluation are presented and commented.

3.1.1 Challenges and Current Trends in Research

In chapter 2 a variety of different recommendation algorithms were introduced to the reader. However, different personalised recommendation lists alone might be of limited value for the user, when the user has to decide between different alternatives of the generated recommendation lists. In other words, presenting the distinct recommendation lists will make it impossible for the users to determine whether the user trust without inspecting all of them in detail. In a more formal way, the goal of a recommendation algorithm is the maximisation of a user's utility. With this in mind, it is relevant to be able to measure recommendation algorithm against each other in order to objectively estimate their recommendation quality and user satisfaction. Concluding major evaluation methods of recommendation algorithms, a large field of study may be outlined. According to Gediminas Adomavicius, Manouselis, and Kwon [2011], there exist three different types of evaluation: offline, user studies and online experiments; each approach has certain advantages and disadvantages, depending on the goal to be achieved. However, the drawback of the variety of applicable scenarios in the context of evaluating recommender systems is the rise of a very complex problem domain. Further, Jannach et al. [2010] argue that the widespread use of recommender systems makes it crucial to develop methods to realistically and accurately assess their true performance and effect on users. A good example may be seen in web search applications, where approaches of *Information Retrieval* are applied. The user requires the relevant items in the first page or within the first 10 results. In the domain early-stage enterprise investment, there is a similar behaviour, with higher relevant ranked

items being more interesting to the target user domain. The present work addresses ranking accuracy measures to examine the true performance of the parallelised algorithms to users.

3.1.2 Excursus Information Retrieval (IR)

The present subsection aims at a literature review concerning the evaluation of recommender systems. According to Ceri [2013], *Information retrieval* is a discipline that deals with the representation, storage, organization and access to information items. The goal of information retrieval is to obtain information that might be useful or relevant to the user. Information retrieval is strongly connected to the machine learning field, which grew out of traditional statistics and artificial intelligence communities [Edgar and Manz, 2017, p. 154]. However, a subfield of machine learning applies computational algorithms to turn empirical data into usable classification models. These so - called *Classification* models, split the data into classes (e.g. relevant and non-relevant items) for the user. The evaluation of recommender systems shares a number of similarities with that of classification models. The reader is referred to the following paragraph for more details. Practical machine learning applications such as in information retrieval and recommender systems solve ranking problems, which generate and return a ranked list of items to evaluate.

The following subsections address the principles of information retrieval in great detail and describe the classic metrics of information retrieval (such as precision and recall).

Classification models

Classification is the process of predicting the class of given data points. In classification, inputs are divided into two or more classes and the learner, also called classifier. The classifier produces a model that assigns unseen inputs to one or more of these classes. The data used to learn these kind of classes is referred to as the *training set* [Aggarwal, 2015, p. 18]. The learned model may then be used to determine the estimated class labels for items, where the class is missing.

Training and Test Set

After a model is generated, it needs to be evaluated. A clear separation of training and testing data sets need to be done and is defined as follows:

- **Training set** ($\mathcal{R}^{Train}$) – a subset of the data to build and train a model
- **Test set** ($\mathcal{R}^{Test}$) – a subset of the data to test the trained model and test the accuracy of the final model

More formally, if $\mathcal{R}$ defined as a set of *ratings* of user u on certain items, see subsection 2.3.1 for a detailed description, then a small subset $\mathcal{R}^{Test} \subset \mathcal{R}$ is used for testing and the set ($\mathcal{R}^{Train}$) $\mathcal{R} - \mathcal{R}^{Test}$ is used for training the model. Otherwise, when applying the training set for the evaluation as well, a high risk of being biased would arise [Jannach et al., 2010, p. 177]. In

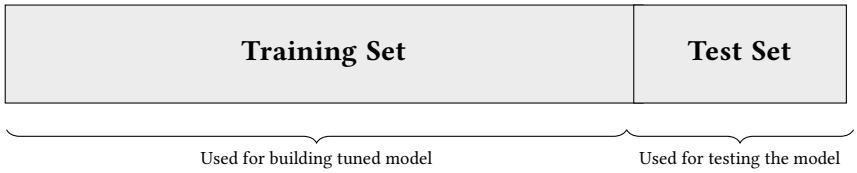


Figure 3.1: Division of Training and Test Data

addition, a high accuracy might indicate that test data has leaked into the training set [Ricci et al., 2010, p. 237].

In particular, data partitioning is a necessary step in machine learning. The basic idea behind, is to separate the available historical data into a training set and a testing set. The available data is not pre-partitioned into training and test data sets. Therefore, it is important to be able to divide the data into these partitions. Ricci et al. [2010] argue that most of the available division methods are *hold-out* and *cross-validation*.

Hold out The basic idea behind the hold-out method is simply to pick two independent datasets d_0 and d_1. These datasets are called training- and test set, respectively. In other words, a portion of the ratings are hidden, and the remaining ratings are used to train the model. A common split when using the hold-out method is using 80% of the ratings matrix for training and the remaining 20% of the ratings for testing. The overall accuracy results in the predicting accuracy of the hidden ratings matrix. An advantage of such an approach is not being vulnerable to overfitting of the specific data set because the ratings used for evaluation are hidden during training. However, such an approach has major disadvantages. First of all, it underestimates the true accuracy because only a portion of the whole ratings are used in the training process [Aggarwal, 2016, p. 238]. Moreover, a further disadvantage can be seen in the case where the hold-out entries (d_0) have a higher average rating than the whole ratings matrix. In that case, the hold-in (d_1) as well as the hold-out entries have a lower average rating than in the ratings matrix, which results in a pessimistic bias in the evaluation process. Consequently, the present work will not cover the *hold-out* method.

Cross validation A widely-used and popular data segmenting method is *cross-validation* [Aggarwal, 2016, p. 238], in which the data is divided into k sets of equal size, called *folds*. Basically, $|R|/k$ is the size of each fold in the rating matrix $\mathcal{R}$. In a further step, one fold is used for testing, and the $k-1$ remaining folds are used for training. This process is repeated k times by using each of the folds as test set. In practice, the value of k is a fixed integer with common values between the interval of 5 to 10. The accuracy metric at each process is then evaluated over a hidden fold during each training process. The overall accuracy is the average of all individual accuracy results. One special kind of cross validation is the *leave-one-out method*, in which k is equal to the total number of entries in the data set. This method has an advantage only in situations, in which the data is sparse. The whole rating matrix $\mathcal{R}$ will be used once for learning the model [Jannach et al., 2010, p. 178].

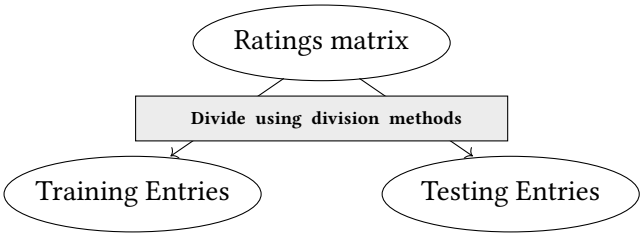


Figure 3.2: Division methods [Aggarwal, 2016, p. 239]

True vs. False, Positive vs. Negative

In a binary classification test, where there are two categories, positive and negative, the measurement of performance is done via the true positive rate (sensitivity) and the true negative rate (specificity). A true positive is an outcome where the model correctly predicts the positive class. Similarly, a true negative is an outcome where the model correctly predicts the negative class. In contrast, a false positive is an outcome where the model incorrectly predicts the positive class. In addition, a false negative is an outcome where the model incorrectly predicts the negative class. [Jannach et al., 2010, p. 171] In the terminology true or false refers to the assigned classification being correct or incorrect, while positive or negative refers to assignment to the positive or the negative category. As stated above, four classification outcomes are achievable, which can be displayed in following confusion matrix:

	Relevant	Non relevant
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

Table 3.1: Precision and Recall confusion matrix

Accuracy Metrics of classification in Information Retrieval

In particular, *Precision* and *Recall* are main evaluation criteria [Ceri, 2013, p. 3] in the course of *Information retrieval* and are therefore introduced to the reader. Also Jannach et al. [2010, p. 180] argue that *Precision* and *Recall* are the best-known classification metrics to identify the n most relevant items for a target user. These results can be evaluated properly in terms of Precision and Recall [Ceri Stefano, 2013, p. 7]. Precision (P) and Recall (R) are obtained from the confusion matrix shown in Table 3.1, were the items are separated into two classes, relevant or non-relevant.

Precision (P) is the fraction of retrieved documents that are relevant [Manning, Raghavan, and Schütze, 2009, p. 155]. More formally,

$$Precision(P) = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} = P(relevant|retrieved) \qquad (3.1)$$

Recall (R) on the other hand, is the fraction of relevant documents that are retrieved [Manning, Raghavan, and Schütze, 2009, p. 155] and is defined as follows:

$$Recall(R) = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = R(retrieved|relevant) \tag{3.2}$$

Recall indicates how many of the actual positives can be found.

3.1.3 Evaluating a Recommender

The relevance of an item to a user is a key factor determining recommendation quality. In the last few years, there has been a growing interest in the field of recommender systems, usually focusing on the design of new algorithms for recommendations. A recommender system typically provides the target user with a list of recommended items that the user might prefer over other items [Ricci et al., 2010, p. 265]. The problem, which occurs for the developer of an application, is to find the most appropriate algorithm for the field of application. Therefore, most recommenders have been evaluated and ranked on their prediction strength, which describes the aptitude to accurately predict the users' preferred items. However, such decisions are based on accuracy metrics, comparing the accuracy of different algorithms. An evaluation method that is widespread in computer sciences and particularly in sub fields such as machine learning, information retrieval and recommender systems is *off-line evaluation* with historical user-interaction data [Jannach et al., 2010, p. 169]. *Off-line methods* are the most common methods for evaluating recommender systems from a research and practice perspective [Aggarwal, 2016, p. 265].

The following subsection will focus on Off-line evaluation for a recommender system and therefore this method was introduced to the reader. Further, existing metrics, as well the advantages and disadvantages in the context of the present work will be highlighted.

Off-line evaluation In Off-line evaluations, historical data, such as user ratings is used. Previous research has demonstrated that off-line methods are a well-known technique for testing recommendation algorithms, because standardised evaluation measures (such as predictive accuracy) have been developed [Ricci et al., 2010, p. 266]. Off-line evaluation on historical data provide a time- and cost- efficient benchmark on recommendation algorithms. A significant off-line evaluation additionally provide a good approximation of the utility function to be optimised by the deployed system [Steck, 2013, p. 213]. Furthermore, if the data set has been collected, it can be used to compare different algorithms, across research agreed standardised benchmark set-ups [Said and Bellog, 2014, p. 129]. More formally, the algorithm is used to predict a certain value for an item from a data set and the results are analysed applying one or more of the accuracy metrics. Therefore, the need for a data set for evaluation arises. According to J. L. Herlocker et al. [2004, p. 13], the literature on evaluating recommender distinguishes between *synthesized* and *natural data sets*. The researcher is confronted with the choice of an existing (natural) data set that might only partial apply the properties of the target domain, or synthesising a data set specifically to apply to those properties. In contrast to a natural

data set, a synthesised data set has the advantage of exploring more techniques for modelling user interests and generating synthetic data from those models. The results offered by J.L. Herlocker in J. L. Herlocker et al. [2004, p. 13] suggest that synthesised data sets could lead to the development of more accurate recommender algorithms. A major drawback of synthesised data sets is that data might fit one of the algorithms better than other. Therefore, the need for clearly defined theoretical properties is high [J. L. Herlocker et al., 2004, p. 13]. The focus of recent research has been on the properties a dataset should have in order to best model the task for which the recommender is being evaluated. As reported by J. L. Herlocker et al. [2004, p. 14], domain features reflect the nature of the content being recommended rather than any particular system. According to J. L. Herlocker et al. [2004, p. 14] *domain features* of interest include the following:

- the content topic being recommended / rated and the associated context in which rating / recommendation takes place
- the user tasks are supported by the recommender
- the novelty- and the quality need
- the cost/benefit ratio of false/true positives/negatives
- the granularity of true user preferences

Therefore, it is significant that the tasks, the algorithm is designed to support are similar to the tasks supported by the system from which the data is collected. Otherwise, evaluating an algorithm on significantly different domain features could result in an unexpected outcome [J. L. Herlocker et al., 2004, p. 15].

The main purpose of the off-line evaluation in relation to the present work is to filter out an appropriate recommendation algorithm and therefore, a variety of accuracy metrics were introduced to the reader in detail.

Accuracy Metrics Accuracy is the most discussed measurement in the domain of recommender systems literature. In Recommender Systems, a basic assumption is that a system that provides more accurate predictions will be preferred by the target user [Ricci et al., 2010, p. 281]. According to J. L. Herlocker et al. [2004, p. 20], an *Accuracy Metric* measures how close a recommender system's predicted ranking of items for a target user differs from the user's true ranking of relevance. The literature on evaluating recommender systems distinguishes between predictive accuracy, decision-based accuracy and rank-based accuracy metrics [Celma, 2010, p. 110].

In particular, as mentioned in subsection 3.1.2, *Ranking-based Accuracy Metrics* are considerably important in the course of the present work and are therefore described in the following paragraphs in great detail.

Ranking-based Metrics

In the earlier paragraphs, *Precision*, subsection 3.1.2 and *Recall*, subsection 3.1.2 are introduced as set-based measures. These measures are computed using unordered sets of items and are not applicable to solving ranking-based problems [Ceri, 2013, p. 7]. Furthermore, these measures have to be extended to evaluate the ranked recommendation item lists. In a ranking-based context, appropriate sets of recommended items are naturally given by top-*k* recommendations, where results are sorted by relevance and only a fraction of the items are returned to the user. This approach determines the order of the predicted items for a target user and compares this ranking with a reference ranking (ground truth) by the target user. Based on the introduced facts and the support of the literature, the author concludes that the present work will introduce the following highly beneficial ranking-based metrics for evaluation. Before we present different ranking-based metrics, we briefly recap various technologies already defined in subsection 2.3.1. More formally, in a recommender system, let a target user $u \in \mathcal{U}$, set of all items $\mathcal{I}$, ordered set of top-k recommended items $I_k(u) \subset \mathcal{I}$, and set of relevant items $\mathcal{I}_u^+$. In the context of *Ranking-based Metrics*, relevant response means that binary labels are assigned to items, *relevant* or *non-relevant* for the target user. These relevant items are considered as truth observations of all items. In ranking-based metrics, higher values indicate better accuracy. The following metrics are suited for the present work and are therefore introduced to the reader.

Precision @ *k* is defined as

$$\mathrm{P@}k = \frac{|\mathcal{I}_u^+ \cap I_k(u)|}{|I_k(u)|} \tag{3.3}$$

where, $|\mathcal{I}_u^+ \cap I_k(u)|$ is the number of relevant items among the top-*k* items, $|I_k(u)|$ is the number of top-*k* recommended items and k is defined by the researcher [Ricci et al., 2010, p. 284]. Precision @ *k* on available data is useful for relative comparisons, as to determine the best among competing models. The lack of relevant items are a main explanation for a low precision value.

Average Precision @ *k* is the average of the precision value obtained for a set of top-*k* items existing after each relevant item is retrieved and is defined as

$$\mathrm{AveP@}k = \frac{1}{|\mathcal{I}_u^+ \cap I_k(u)|} \sum_{n=1}^{k} \mathrm{P@}n \cdot \mathrm{rel}_n \tag{3.4}$$

where, $|\mathcal{I}_u^+ \cap I_k(u)|$ is the number of relevant items among the top-k items, $\mathrm{P@}n$ is the value given by Equation (3.3), rel_n is $\begin{cases} 1 & (i_n \in \mathcal{I}_u^+) \\ 0 & (\text{otherwise}) \end{cases}$ and i_n is the item returned at the *n*-th position/rank . In other words, rel_n equals 1 if the item at rank *n* is a relevant item, 0 otherwise.

Mean Average Precision @ k in contrast to Precision @ k, MAP averages the values given by Equation (3.4) across all users in the dataset. According to Aggarwal [2016, p. 246], it is defined as

$$\mathrm{MAP@}k = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{|\mathcal{U}|} (\mathrm{AveP@}k)_u \tag{3.5}$$

where, k is defined by the researcher, $\mathcal{U}$ is the set of users and $(\mathrm{AveP@}k)_u$ given by Equation (3.4) for a specific user $u \in \mathcal{U}$.

Normalized Discounted Cumulative Gain @ k is the final approach measuring ranking accuracy in the context of the present work. The nDCG computes a score for $I(u)$, which places emphasis on higher ranked relevant items. nDCG allows specification of an expected ranking within $\mathcal{I}_u^+$, that is, the accuracy metric can incorporate a relevance factor, which suggest likely the k-th item is to rank at the top of a recommendation list and it directly corresponds to an expected ranking of the relevant items [Steck, 2013, p. 218]. According to Aggarwal [2016, p. 245], the Normalised Discounted Cumulative Gain is defined as

$$\mathrm{nDCG@}k = \frac{\mathrm{DCG}_k}{\mathrm{IDCG}_k} = \frac{\sum_{n=1}^{|\mathcal{I}|} D_k(n) \left[i_n \in \mathcal{I}_u^+ \right]}{\sum_{n=1}^{|\mathcal{I}|} D_k(n)} \tag{3.6}$$

where DCG_k indicates how well $I(u)$ corresponds to the relevant items and $IDCG_k$ is the best DCG_k that $I(u)$ exactly matches to $\mathcal{I}_u^+$, $D_k(n)$ is the discount factor and rel_n is the ground-truth relevance factor of item i_n, which is computed as a heuristic function of the ratings.

$$D_k(n) = \begin{cases} (2^{\mathrm{rel}_n} - 1) / \log_2(n+1) & (1 \leq n \leq k) \\ 0 & (n > k) \end{cases} \tag{3.7}$$

$$\mathrm{rel}_n = \begin{cases} 1 & (i_n \in \mathcal{I}_u^+) \\ 0 & (\mathrm{otherwise}) \end{cases} \tag{3.8}$$

Finally, the following table gives a brief summary of used mathematical symbols in this section to the reader.

Table 3.2: Mathematical symbols

Symbol	Definition
$\mathcal{R}^{Train}$	a subset of the data to build and train a model

To be continued...

...continued from previous page

Symbol	Definition
$\mathcal{R}^{Test}$	a subset of the data to test the trained model and test the accuracy of the final model
$\mathcal{R}$	a set of *ratings* of user u
u	a target user
$\mathcal{I}$	a set of all items
$\mathcal{U}$	a set of all users
$I(u)$	a set of items of user u
$I_k(u)$	a ordered set of top-k recommended items
$\mathcal{I}_u^+$	a set of relevant items
$D_k(n)$	is the discount factor in the context of nDCG
rel_n	is the ground-truth relevance factor of item i_n

3.2 Methodology for Evaluating the Algorithms

The following section of the present work introduces the reader to the *Methodology for Evaluating the Algorithms*, in order to gain knowledge to answer the research questions. Since the theoretical foundations are already defined in the background section 3.1, this section starts with a short introduction of the *Problem Definition & Research Questions* in subsection 3.2.1, followed by the *Design of the Empirical Validation* in subsection 3.2.2, where the *Expert Questionnaire* for generating an evaluation dataset and the *Participants* of the study are described. In its following thematic block, subsection 3.2.2 deals with *Data Collection* and *Dataset Generation* for evaluating the recommendation algorithm, derived from the questionnaire. This is a necessary step to be able to evaluate the parallelised recommender algorithms. Finally, subsection 3.2.3 describes the *Evaluation Procedure* of the present specialisation topic in detail.

3.2.1 Problem Definition & Research Questions

In Chapter 2, a variety of recommendation algorithms, elaborated by the authors were introduced to the reader. These recommendation algorithms were implemented in a newly-created software prototype. However, different instantiations of the software prototype results in different recommendation lists. In order to objectively and definitely characterise the recommendation quality of an algorithm, a standardised off-line evaluation method with a historical data set for analysing is a prerequisite. As elaborated in the specialisation topic *Investment Decision-making & Venture Valuation*[1], no public data set is available for this domain. Additionally, public datasets such as *MovieLens*[2] are not applicable to the domain of early-stage enterprise investments. As a consequence, the present work can not rely on publicly available datasets and therefore, a dataset has to be created for evaluating and the need for evaluating the recommendation algorithms emerges. The main focus of the present specialisation topic lies in the search for an evaluation method for the implemented recommender algorithms in Chapter 2 and the utilisation of the evaluation method in terms of recommendation quality for the user in the domain of early-stage enterprise investment. In particular, the following research questions will be addressed:

(i) How does the recommendation quality and the usefulness of the recommendation system affect user satisfaction?

   (a) How may the recommender system be evaluated in terms of recommendation quality?

   (b) Which methodologies may be utilized to evaluate the recommendation system?

   (c) What implications do the results indicate?

   (d) How do the findings affect future research?

[1] Christian Ohrfandl, 2018.

[2] MovieLens - https://movielens.org/

3.2.2 Design of the Empirical Validation

In this section, the experiment conducted for evaluating the recommendation algorithm is introduced to the reader. The purpose of the *empirical validation* lies in the collection of valid expert data on investment decision-making criteria to generate a data set for evaluating the invented algorithms in Chapter 2. The scientific instruments utilized for evaluation are mainly based on off-line experiments. According to Jannach et al. [2010, p. 178], an off-line experiment is performed by using a historical dataset to simulate the behaviour of users who interact with the recommendation system. Furthermore, off-line experiments have the advantage, of no user interaction with real users being required and the comparison of a wide range of candidate algorithms at low costs is possible [Jannach et al., 2010, p. 179]. In particular, different algorithm instantiations are measured to their performance according to other algorithms. In other words and with the support of the literature, the present work measures how accurately a specific algorithm can predict the actual ranking that users have previously assigned, compared to others.

Data Collection & Data Set Generation

An important motivation of this present work is the desire to collect data about *Investment Decision-making* processes that can be evaluated by several algorithms. The data serves as a fuel for the evaluation task and the collected data has the opportunity to be perfectly matched with the properties of the target domain. Without correct and adequate data, the evaluation task becomes obsolete and the recommender cannot produce any reasonable recommendation lists. As already mentioned in the previous Chapter 2, a natural data set in the domain *Investment Decision-making & Venture Valuation*[3] is not publicly available and therefore a synthesised data set has to be created. In order to evaluate algorithms off-line, it is necessary to simulate the behaviour of users that interact with the system and the user uses the recommendations [Jannach et al., 2010, p. 179]. This process is typically done by capturing historical user data. Thereafter, a certain amount of this information was hidden in order to simulate the rating of how a specific user will rate an item. As the main purpose of the off-line evaluation is to filter algorithms, the data used for the off-line evaluation process should closely match the data model requirements from the results in subsection 2.3.1.

The following *Expert Questionnaire* is designed to aggregate all necessary data to match the data model requirements of the software prototype, to evaluate the five main recommenders with all different versions described in Chapter 2, subsection 2.3.1.

Characteristics of the Expert Questionnaire

The expert questionnaire is based on the results in Chapter 2. The main goal is the collection of valid expert data to generate a synthesised data set, which can be incorporated into the software prototype. The gathered data is regarded as the basis for evaluating the algorithms. The expert questionnaire contains twelve questions and includes different types of questions:

[3] Christian Ohrfandl, 2018.

- Open questions
- Closed questions
  - Ordinal
  - Categorical

and is divided into the following sections

- Characteristics of the Investor
  - Question 1 to 9 addresses the profile of an investor (User Profile), see subsection 3.3.1
- Investment decision on early-stage enterprises
  - Question 10 address the ground truth rankings from investors, see subsection 3.3.1
- Investor fellowship
  - Question 11 to 12 addresses the social trust relationships among investors (Social network), see subsection 3.3.1

The questionnaire includes questions about the participants' personal *preferences* relating to an early-stage enterprise and their *Investment decision criterion* relating to such enterprises, given a fact-sheet of ten pre-defined enterprises. The questionnaire was designed to need a maximum participation time of 15 minutes. The lessons from the research of the co-author Christian Ohrfandl's specialisation on *Investment Decision-making & Venture Valuation*, teach us that experts in the domain of early-stage enterprises have practically no time and get bored, if the questionnaire took too long. It is important to ask neutral questions, which do not stress the participant. Participants may also furnish fictitious answers, as for instance when they perceive the answer as private. Indeed, there is quite a lot amount of research in the area of questionnaire design, but it is not germane to the present work. However, fundamental design principles have to be incorporated in the questionnaire. According to Atteslander [2010, p. 295], a pre-test was conducted in order to ensure that participants interpret and answer questions in the way relevant to the expected results of the present work. Such pre-test assists to determine whether participants understand the questions, and whether they can perform that the tasks or have the information that the questions require. On the basis of the results of the pre-test, additional information was added to the ranking question (Q10) and the questions wording were rephrased for a clear understanding. The following paragraph describes the elaborated questions.

Questions Based on evaluation of the requirements of the software prototype (subsection 2.3.1), twelve questions were constructed for being utilised in the expert questionnaire. The original questionnaire is in German and the following questions are translated there from.

Table 3.3: Question types in the expert questionnaire

Idx	Type	Details
Q1	Closed	Likert type (Ordinal), Scale 1–5
Q2	Closed	Likert type (Ordinal), Scale 1–5
Q3	Closed	Multiple choice (Categorical), Multiple answers possible
Q4	Closed	Likert type (Ordinal), Scale 1–5
Q5	Closed	Multiple choice (Categorical), Multiple answers possible
Q6	Closed	Likert type (Ordinal), Scale 1–5
Q7	Closed	Likert type (Ordinal), Scale 1–5
Q8	Closed	Likert type (Ordinal), Scale 1–5
Q9	Closed	Likert type (Ordinal), Scale 1–5
Q10	Closed	Rating scale (Ordinal), Scale 1–5
Q11	Closed	Multiple choice (Categorical), Multiple answers possible
Q12	Open	Text-input

The questionnaire includes questions of *Likert* type, which call for responses on the Likert scale. Likert scales are widely used to measure attitudes and opinions with a greater degree of nuance than a dichotomous question (such as "yes" or "no" question). By definition [Robertson, 2012, p. 6], the Likert scale is a five- or seven-point scale that offers a range of answer options — from one extreme attitude to another, like “strongly agree” to “strongly disagree”. Typically, they include a moderate or neutral midpoint. The present work’s default answer set utilised in combination with the Likert scale is depicted in the following listing:

1 … Unimportant
2 … Rather unimportant
3 … Neutral
4 … Rather important
5 … Important

The questions of the expert questionnaire addressed in the present work are the following:

Question 1: A recommendation based on the innovations/products of early-stage enterprises is important to me. The main objective of this question is to gain the importance of recommending innovations/products of an early-stage enterprise to a user. According to the

results of Chapter 2, recommending early-stage enterprises based on innovations/products is part of the user preference profile.

Question 2: A recommendation based on the current stage (life cycle) of the early-stage enterprise is important to me. The main goal of this question is to gain the importance of recommending early-stage enterprises based on a life cycle (stage) to a user. According to the results of Chapter 2, recommending enterprises based on the life cycle (stage) is part of the user preference profile.

Question 3: Which life cycles of an early-stage enterprise are of interest? The main purpose of this question is to elicit information regarding interest in the life cycles of early-stage enterprises to a user. As noted in the investment decision-making & venture valuation specialisation topic by Christian Ohrfandl, this question considers the most important life cycles (stages) and the ones from the fact sheet; the reader is referred to the appendix chapter A for a complete listing. According to the results of Chapter 2, recommending enterprises based on the life cycle is part of the user preference profile. The question is a multiple choice type question, with an answer set divided as follows:

1 … Pre-seed stage
2 … Seed stage
3 … Start-up stage
4 … Expansion stage
5 … Later stages

Question 4: A recommendation based on the market sector is important to me. The main objective of this question is to learn the importance of recommending early-stage enterprises based on the market sector to a user. Due to the results of Chapter 2, recommending enterprises based on the market sector is part of the user preference profile. The question is a Likert type question that shares the same rating scale introduced to the reader in subsection 3.2.2.

Question 5: Which market sectors would you be interested in? The goal of this question is to ascertain the market sector(s) of early-stage enterprises of interest to a user. As an outcome in the investment decision-making & venture valuation specialisation topic by Christian Ohrfandl, recommending enterprises based on a market sector is important. According to the results of Chapter 2, recommending enterprises based on the market sector is part of the user preference profile. The question is a multiple choice type question with an answer set as follows:

1 … Health
2 … Travel
3 … Hardware
4 … Software
5 … Automotive
6 … Mobile application
7 … Cloud computing
8 … Big data
9 … Artificial intelligence
10 … Education
11 … Games
12 … Financing
13 … Social media
14 … e-Commerce
15 … Block chain
16 … Smart City
17 … Productivity
18 … Tourism

19 … Services	23 … Navigation	27 … Smart Life
20 … Advertising	24 … Sport	28 … Shopping
21 … Medicine	25 … Fitness	
22 … Pharmacy	26 … Human resources	

The market sectors were derived from early-stage enterprises mentioned in the fact sheet; the reader is referred to Question 10 subsection 3.2.2. The participant has the choice of extending the given answer set with remarks relating to market sectors.

Question 6: An already public acceptance / interest of the product of the early stage enterprise is important to me? The main purpose of this question is to gain the importance of recommending early-stage enterprises based on the public interest to a user. According to the results of Chapter 2, recommending enterprises based on the public acceptance is part of the user preference profile.

Question 7: The management team of an early stage enterprise is important to me. The main objective of this question is to gain the importance of recommending early-stage enterprises based on an available management team (more than one founder) to a user. According to the results of Chapter 2, recommending enterprises based on a management team is part of the user preference profile.

Question 8: An already existing valuation is important to me. The main objective of this question is to gain the importance of recommending early-stage enterprises based on a valuation available to a user. According to the results of Chapter 2, recommending enterprises based on available valuations is part of the user preference profile.

Question 9: An early-stage enterprise established earlier is more important to me than later ones. The main purpose of this question is to ascertain the importance of recommending early-stage enterprises based on the founding date of an enterprise to a user. According to the results of Chapter 2, recommending enterprises based on the founding date is part of the user preference profile.

Question 10: Please select 5 out of 10 early-stage enterprises from the fact sheet and rank them in the order in which you would invest (Investment preference)? The objective of this question is to gain ground truth rankings from investors for evaluating the algorithms invented in Chapter 2. The question is a rank order type question having a rating scale from 1 to 5. The question includes a fact sheet of ten pre-defined early-stage enterprises. The reader is referred to the appendix chapter A for a complete listing of the pre-defined ten early-stage enterprises. The aim of the early-stage enterprises selection process is to choose the most suitable enterprises, who can meet the requirements for evaluating the algorithms. The main requirements for such enterprise are:

- a known innovation/product
- a known life cycle
- a known market sector
- information about the founder or management team
- founding date

The final selection of enterprises is a balanced variety of different products, life cycles, market sectors, known or unknown valuations, sole founder or management team and the founding date. The reader is referred to the appendix chapter A for a complete listing of the pre-defined ten early-stage enterprises. Only on the basis of these enterprises will the participants be able to answer this question.

Question 11: Would you take into account the opinions of well-known investors / business angels in the start-up scene in your decision-making process or would you follow them? (Multiple answers possible) The main goal of this question is to ascertain the trust relationships among participants. As an outcome in the investment decision-making & venture valuation specialisation topic by Christian Ohrfandl, recommending enterprises based on the decision of other investors is important. According to the results of Chapter 2, recommending enterprises based on the decision of other investors is part of the user preference profile. The question is a multiple choice question with an answer set indicating the following well-known investors in the DACH[4] countries :

1 … Herbert Gartner	6 … Rudolf Dömötör	11 … Brigitte Ederer
2 … Hansi Hansmann	7 … Martin Bittner	12 … Selma Prodanovic
3 … Tanja Sternbauer	8 … Florian Kandler	13 … Hermann Hauser
4 … Markus Raunig	9 … Claus Raidl	
5 … Herwig Springer	10 … Hilde Umdasch	14 … Michael Altrichter

The names of the listed well-known investors are from festivals such as *Pioneers*[5] or *4GAMECHANGERS*[6], events such as *Founder & Investor talks* at TU Vienna i2c[7], the *AWS Greenworth Batch*[8] or from information in the public domain. Further, the participant had the choice of extending the given answer set with other well-known investors.

[4] Germany (D), Austria (A), Switzerland (CH)
[5] Pioneers Festival - https://pioneers.io/
[6] 4GameChangers Festival - https://4gamechangers.io/
[7] Innovation Incubation Center (i2c) – TU Wien - https://i2c.ec.tuwien.ac.at/
[8] Austria Wirtschaftsservice - https://www.aws.at/

Question 12: To derive the significance of the individual algorithms, we require a large number of expert opinions. We would be very happy if you could recommend other people whom you think can provide inputs to our questionnaire (please provide contact details such as name, email address, phone number or website) The main objective of this question is to gain more participants and to draw broader trust relationships among investors. The question type is an open question.

Participants & Domain

The participants for the expert questionnaire shall be experts in the field of early-stage enterprise investment. This criterion about the characteristics of experts especially applies to investors such as business angels, but also includes experts such as researchers working for incubators, accelerators, aid money agencies specialising in the field of early-stage enterprises or universities. These are the same interest groups as mentioned in the previous study by Christian Ohrfandl on *Investment Decision-making & Venture Valuation.*

3.2.3 Evaluation Procedure

The *Evaluation Procedure* introduces the reader in detail to how the data is collected, how it is transformed into the prototype, which assumptions are derived and how the recommendations are made. The experiment is divided into four phases, as described by the following passages:

Phase 1 - Expert questionnaire

In the first phase of the evaluation procedure, the task for the participants was to fill out the online expert questionnaire capturing their individual preferences, investment preference about ten predefined early-stage enterprises and their possible fellowship with well-known investors and others. The participants of the expert questionnaire shall be experts in the field of early-stage enterprise investment and therefore, an invitation to answer the questionnaire accessible online is sent to the corresponding participants via e-mail. The e-mail includes an introduction about the problem domain of the present work and a request for participation. The online questionnaire framework used is GoogleForms[9] and the participant had the choice to take part anonymously or by indicating name/e-mail address. In the questionnaire, the questions needs to be asked without bias. The framework provides randomisation tools to present each question and each option for answering in different orders. In addition, the ranking question uses this randomisation technique to list each pre-defined early-stage enterprise in a different order. The participants are able to send the questionnaire to potential additional participants they know. After a participant completes the online questionnaire, the results are stored for later evaluation phases.

[9] GoogleForms - https://www.google.com/forms/about/

Phase 2 - Dataset Preparation and Dataset Generation

The purpose of the present phase *Dataset Preparation & Generation* is to transform the collected valid expert data into a dataset. The raw data collected by the expert questionnaire has to be transformed into a readable form according to the software prototype requirements defined in subsection 2.3.1. Therefore, a special transformation pattern based on the existing data model of the software prototype is designed. The pattern includes transformation rules for:

- User preferences
- Venture likes
- Investor follows

The data collection instrument, the expert questionnaire, captured a rich user profile of the participants and the questionnaire answers of a specific participant. The answers needs to be translated into a user profile according the data model of the software prototype. Questions one to nine concentrate on the preferences for a user and eleven to twelve on the fellowship with the other users. In particular, for each questionnaire item, a user indicated the importance on a five points Likert scale, ranging from "Unimportant" to "Important". The contribution of each item to the score ranges from 1 to 5, which does not fit the requirements of the prototype. The importance weight factor is by definition ranging from 0% to 100%. Therefore, a transformation rule is created and divides the scale into five sections. The following multicolumn lists the new associated values:

0% …	Unimportant	50% …	Neutral	100% …	Important
25% …	Rather unimportant	75% …	Rather important		

A user u has the possibility to *like* an item i in the software prototype, which expresses the preference of a user towards a certain item. Every ranked early-stage enterprise from the questionnaire of a specific participant is handled as a liked item and is mapped into the prototypes' data model. The reader is referred to the definition of a *User profile* and *User-Item Interactions* in subsection 2.3.1 for a complete insight.

Due to the fact that preferences can be activated and deactivated in the software prototype, the "Unimportant" or 0% value is handled as a special case. The value also deactivates the specific preference in the prototype and does not consider it in the recommendation calculation process.

The investor fellowship of the present work is based on the set of trust relationships among users. A participant of the expert questionnaire express their connectedness towards investors as *trust relationships* in Question 11. Every trust relationship from the questionnaire of a specific participant is mapped into a relationship into the prototypes' data model. The reader is referred to the definition of a *trust relationship* in subsection 2.3.1.

After a successful transformation, the dataset is ready for further utilisation in the evaluation process.

Phase 3 - Recommendations

After the historical data set is available, the *Recommendation phase* follows. The aim of this phase is to generate recommendation lists for evaluating the algorithms. Therefore, the data set is split into two parts. One part is a randomly selected share of known ratings, the training set. The training set is used as input to train the algorithm and build the model. For the remaining share of data, the test set is required as ground truth to evaluate the model's quality. The model allows the software prototype to compute recommendations at runtime. The model building and recommendation steps are repeated several times, to ensure reliability of measurements of the random splits. Finally, the recommendation lists were stored for later analysis.

Phase 4 - Evaluation of the Algorithms

The final phase in the evaluation procedure, ranking-based metrics mentioned in the subsection 3.1.3 aim to capture the quality of a particular ranking, based on the ground truth ranking that a user has indicated in the expert questionnaire. In order to provide a comparison of the algorithms, an analysis of the recommendation list results is performed. Figure 3.3 visualises the evaluation principle.

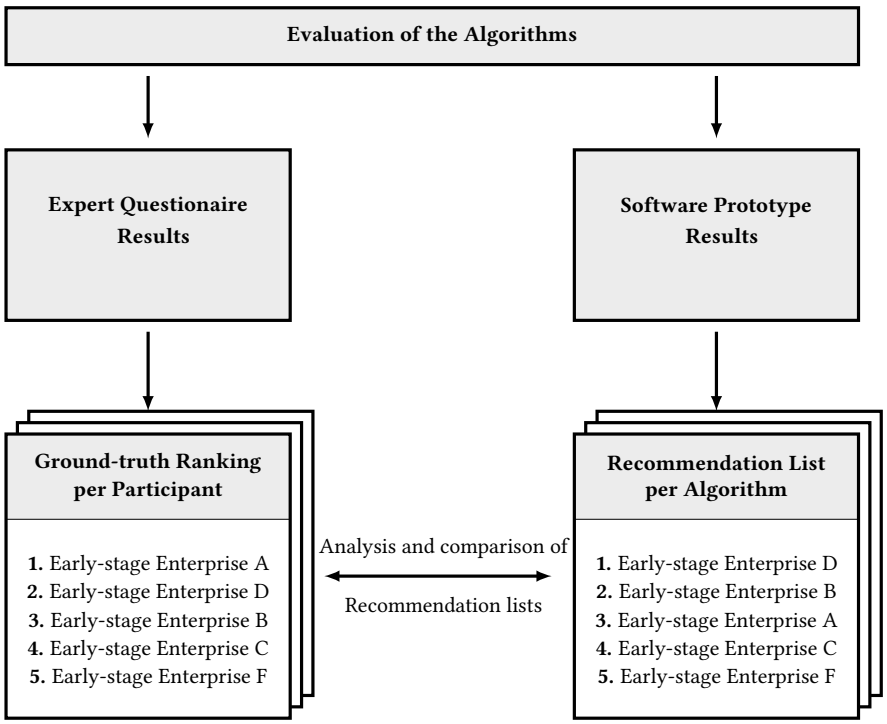


Figure 3.3: Evaluation Procedure

3.3 Evaluation Results

In this section of the present work, the reader is informed about the results of the experiment and evaluation of the algorithms implemented in Chapter 2. The experiment and the evaluation follow the scientific process specified by the methodology, section 3.2 and describes them in phases. The structure of this chapter starts with the subsection *Expert Questionnaire Results*, followed by subsections on *Data Preparation* and *Dataset Generation*. Further, in subsection *Recommendation* the recommendation lists for evaluation are created and evaluated in subsection *Evaluation of the Algorithms*. Finally, subsection *Limitation of Evaluation* point outs the limitations of the present experiment.

Pedhazur and Schmelkin define an *Experiment* thus:

> “ … An experiment is a study in which at least one variable is manipulated and units are randomly assigned to different levels or categories of manipulated variable(s). …
>
> *Pedhazur and Pedhazur Schmelkin [1991, p. 251]* ”

According to the definition, the present work's experiment follows the analogy, where the "units" are the participants of the expert questionnaire, "manipulated variable" is the type of recommender and the "categories of manipulated variables" are the different types of recommender algorithm.

The overall objective of the present work was to mainly provide an evaluation principle for early-stage enterprise investment in the domain of computational recommendation systems. Every different recommender algorithm from Chapter 2 is evaluated with regard to the measures, defined in subsection 3.1.3. Results of the experiments condensed in Table 3.14 are discussed below.

3.3.1 Phase 1 - Expert questionnaire results

One of the first steps is gathering valid data from experts for evaluation. The participants, experts in the domain of early-stage enterprise investments should behave as they would in a real-world environment. Due to the complexity of various recommender engines in the software prototype combined with time consuming tasks, none of the contacted investors was cooperative in supporting the evaluation process and spent their time on-line on the software prototype. Therefore, a modest and less time consuming method is designed to motivate the contacted investors. The newly introduced approach is a simulation of the software prototype through a questionnaire. The answers are collected as input data for a data set capable for evaluation. This data collection instrument captures a rich user profile of the participants. The questionnaire comprises 12 questions separated into three sections:

(a) ***Preferences* of the *User Profile***
An important characteristic of the software prototype is to state *Preferences* on early-

stage enterprises by a user. Recommender, such as the *Knowledge-based Recommender*, subsection 2.3.4 and *Hybrid Recommender*, subsection 2.3.6, include the preferences in their recommendation calculation process. As a consequence, questions about *Preference* information need to be utilised in order to evaluate the mentioned recommender.

(b) ***Ranking* of the ten pre-defined early-stage enterprises**
A Recommender, such as the *User-based Collaborative Filtering*, subsection 2.3.2, *Item-based Collaborative Filtering*, subsection 2.3.2 and *Content-based Recommendation*, subsection 2.3.3 includes historical user-interactions in their recommendation calculation process. The historical user-interactions are simulated through a ranking order question. The challenge for the participant was to rank 5 out of 10 pre-defined early-stage enterprises in the expert questionnaire. Hence, the ranking order question has the power to collect historical user-interactions and this data is needed in order to evaluate the mentioned recommender.

(c) ***Fellowship* to other well-known people in the domain of early-stage enterprise investments**
A major implementation detail of the software prototype for the user is to state their trust among other users. Therefore, questions concerning the possible fellowship with other well-known investors are included in the expert questionnaire. The questions needs to be utilised in order to evaluate the *Social Trust Recommender*, subsection 2.3.5.

The expert questionnaire was designed in March 2018 and the first invitations were sent to the participants in mid April 2018. The participants, from Austria and Germany, were recruited through a mailing list, originating from networking in the domain of early-stage enterprise investment. Due to a considerably low percentage of participants at that time, a friendly reminder was sent to the participants in mid May 2018. Reasons for this unsatisfactory response rate may be seen in the fact that the contacted participants, in the domain of early-stage enterprise investments, had not had the time to do so. Lessons learned in previous research, indicate that, motivating people in the domain of early-stage enterprise investments without any incentives, such as money, is a hard task. As a consequence and as a simplification, the questionnaire was made accessible at an online setting and the participating time was achievable within a fifteen minute time frame. At the end of June 2018, the response rate of participants reached ~51,5%, that is, 35 out of 68 persons. The participants' responses were carefully checked for their quality. For example, participants who give the same scales for all questions or conflicting answers were checked. Fortunately, no outlier were found; only one participant did not complete the ranking question. The sample size of 35 participants is sufficient for continuing with the evaluation process.

Descriptive statistics

Descriptive statistics quantitatively describes and summarizes features of the collected questionnaire information. Descriptive statistics are divided into measures of central tendency and measures of variability. Measures of central tendency include the mean, median, and mode,

while measures of variability include the standard deviation, variance, the minimum and maximum variables. Due to the fact that the most present questionnaire questions is based on an ordinal scale, it is not possible to calculate a degree of difference on a relative basis. Descriptive statistics supports to better understand what the present data is trying to report, which results in a reasonable understanding of the overall recommender evaluation. In the course of this passage, the collected data of the present questions is analysed and evaluated according to descriptive statistics methods.

Question 1: A recommendation based on the innovations/products of early-stage enterprises is important to me. The main objective of this question is to gain the importance of recommending innovations/products of an early-stage enterprise to a user. According to the results of Chapter 2, recommending early-stage enterprises based on innovations/products is part of the user preference profile. This is a Likert type question that shares the same rating scale as mentioned in subsection 3.2.2. The reader is referred to Table 3.4 for summarized descriptive statistics for Question 1, where 1 means important and 5 means unimportant to the participant.

Table 3.4: Question 1 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q1	35	1.857	0.879	1	1	2	2	5

Furthermore, the following Figure 3.4 visualizes a plot of the participants' responses and the included histogram visualizes the missing - and completed answers of Question 1.

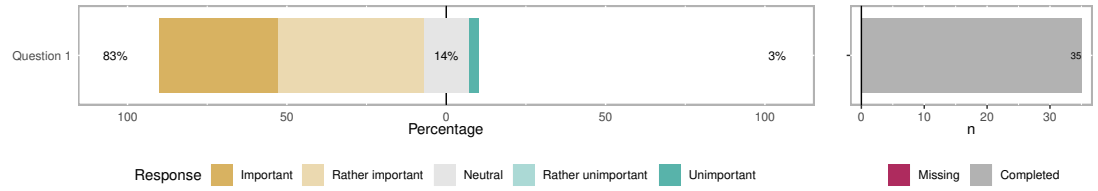


Figure 3.4: Question 1: Likert-Plot

In summary, the observation of 35 participants shows a mean of 1.86 by a standard deviation of 0.88, which indicate that recommendations based on the innovations/products is rather important to the participants.

Question 2: A recommendation based on the current stage (life cycle) of the early-stage enterprise is important to me. The main goal of this question is to gain the importance of recommending early-stage enterprises based on a life cycle (stage) to a user. According to the results of Chapter 2, recommending enterprises based on the life cycle (stage) is part of

the user preference profile. This too a Likert type question that shares the same rating scale, introduced in subsection 3.2.2. The reader is referred to Table 3.5 for summarized descriptive statistics for Question 2, where 1 means important and 5 means unimportant to the participant.

Table 3.5: Question 2 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q2	35	1.743	0.980	1	1	1	2	5

Furthermore, the following Figure 3.5 visualizes a plot of the participants' responses and the included histogram visualizes the missing - and completed answers of Question 2.

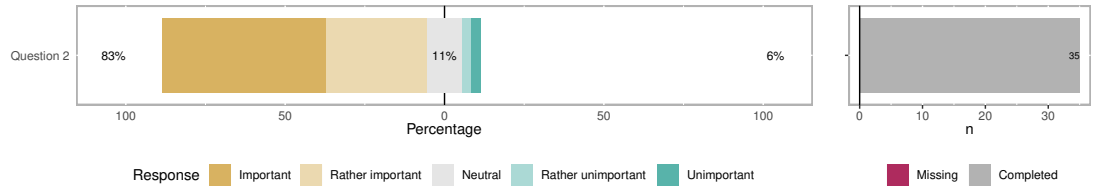


Figure 3.5: Question 2: Likert-Plot

The majority of those who responded to this question felt that a recommendation based on the life cycle is important, visualised in Figure 3.5. In summary, the mean score for question 2 is 1.74, which indicates an importance on a life cycle of a early-stage enterprise by the participants.

Question 3: Which life cycles of an early-stage enterprise are of interest? The main purpose of this question is to elicit information regarding interest in the life cycles of early-stage enterprises to a user. As noted in the investment decision-making & venture valuation specialisation topic by Christian Ohrfandl, this question considers the most important life cycles (stages) and the ones from the fact sheet; the reader is referred to the appendix chapter A for a complete listing. According to the results of Chapter 2, recommending enterprises based on the life cycle is part of the user preference profile. The reader is referred to Figure 3.6 for the frequency distribution of the early-stage investment phases for Question 3, where 1 means important and 5 means unimportant to the participant.

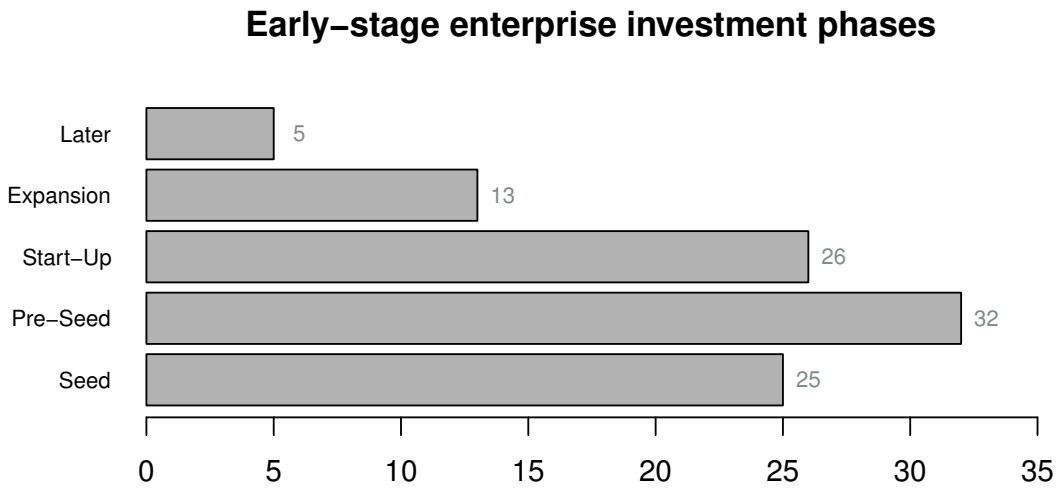


Figure 3.6: Question 3: Frequency

In summary, the observation of 35 participants shows that earlier early-stage investment phases (life cylce), such as "Start-up" or "Pre-Seed"-phase are more interesting to the participants than later phases.

Question 4: A recommendation based on the market sector is important to me. The main objective of this question is to learn the importance of recommending early-stage enterprises based on the market sector to a user. Due to the results of Chapter 2, recommending enterprises based on the market sector is part of the user preference profile. The question is a Likert type question that shares the same rating scale introduced in subsection 3.2.2. The reader is referred to Table 3.6 for summarized descriptive statistics for Question 4, where 1 means important and 5 means unimportant to the participant.

Table 3.6: Question 4 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q4	35	2.200	0.964	1	2	2	3	4

Furthermore, the following Figure 3.7 visualizes a plot of the participants' responses and the included histogram visualizes the missing - and completed answers of Question 4.

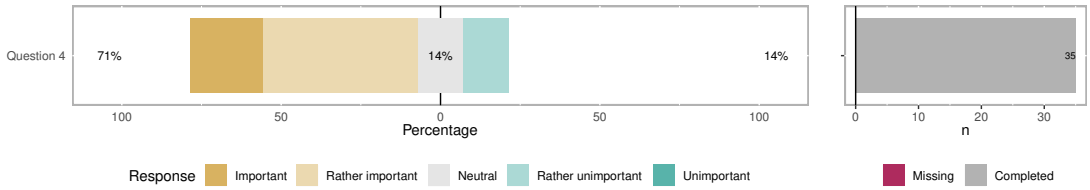


Figure 3.7: Question 4: Likert-Plot

In Figure 3.7, there is a clear trend to the importance of a market sector to the participants, where an early-stage enterprise is operated in. The mean score for question 4 is 2.2.

Question 5: Which market sectors would you be interested in? The goal of this question is to ascertain the market sector(s) of early-stage enterprises of interest to a user. As an outcome in the investment decision-making & venture valuation specialisation topic by Christian Ohrfandl, recommending enterprises based on a market sector is important. According to the results of Chapter 2, recommending enterprises based on the market sector is part of the user preference profile. Figure 3.8 shows the frequency distribution of the market sectors for Question 5.

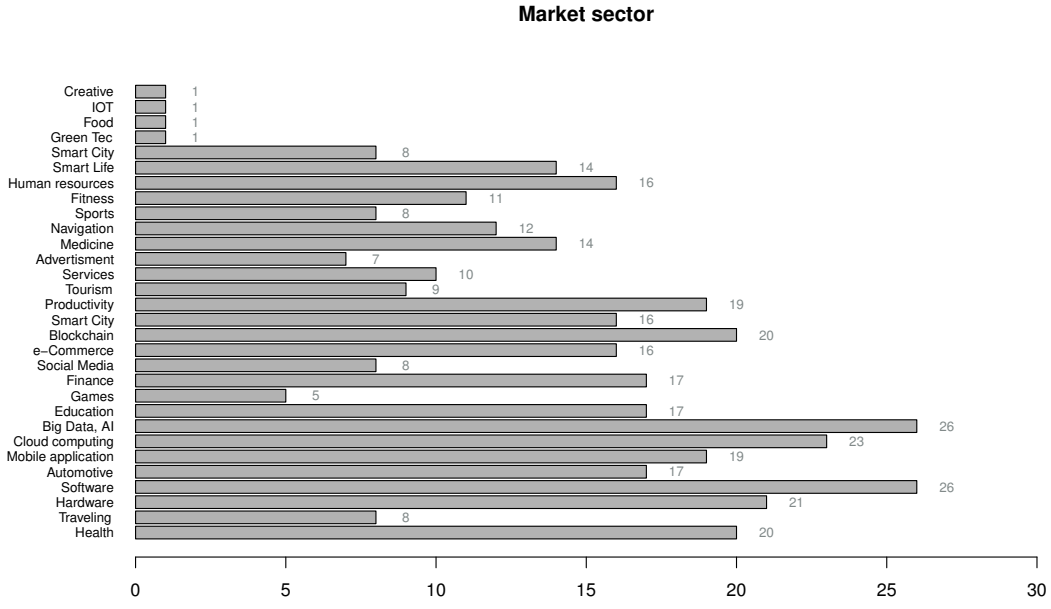


Figure 3.8: Question 5: Frequency

Question 6: An already public acceptance / interest of the product of the early stage enterprise is important to me? The main purpose of this question is to gain the importance of recommending early-stage enterprises based on the public interest to a user. According to the results of Chapter 2, recommending enterprises based on the public acceptance is part of the user preference profile. The question type is a Likert type question that shares the same rating scale, introduced to the reader in subsection 3.2.2. Table 3.7 presents summarized descriptive statistics for Question 6, where 1 means important and 5 means unimportant to the participant.

Table 3.7: Question 6 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q6	35	3.886	0.963	2	3	4	5	5

Furthermore, the following Figure 3.9 visualizes a plot of the participants' responses and the included histogram visualizes the missing- and completed answers of Question 6.

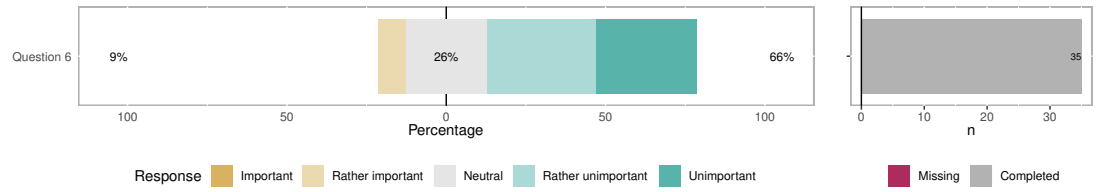


Figure 3.9: Question 6: Likert-Plot

The majority of those who responded to this question felt that the level of awareness of an early-stage enterprise is rather unimportant, the reader is referred to figure Figure 3.9. The mean score is 3.89 and confirms this statement.

Question 7: The management team of an early stage enterprise is important to me. The main objective of this question is to gain the importance of recommending early-stage enterprises based on an available management team (more than one founder) to a user. According to the results of Chapter 2, recommending enterprises based on a management team is part of the user preference profile. The question type is a Likert type question that shares the same rating scale, introduced to the reader in subsection 3.2.2. The reader is referred to Table 3.8 for summarized descriptive statistics for Question 7, where 1 means important and 5 means unimportant to the participant.

Table 3.8: Question 7 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q7	35	1.714	0.926	1	1	2	2	5

Furthermore, the following Figure 3.10 visualizes a plot of the participants' responses and the included histogram visualizes the missing- and completed answers of Question 7.

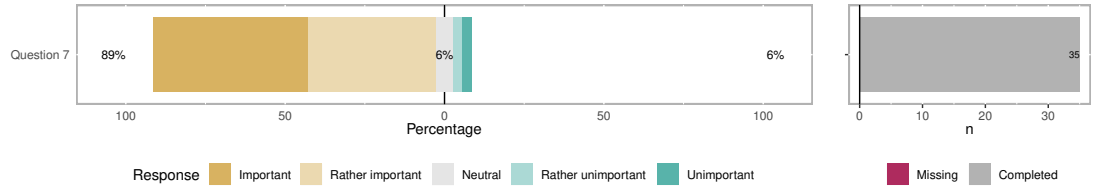


Figure 3.10: Question 7: Likert-Plot

In response to Question 7, the majority of those surveyed indicate that a management team in the context of an early-stage enterprise is important. The mean score for question 7 is 1.71 and confirms this statement.

Question 8: An already existing valuation is important to me. The main objective of this question is to gain the importance of recommending early-stage enterprises based on a valuation available to a user. According to the results of Chapter 2, recommending enterprises based on available valuations is part of the user preference profile. The question is a Likert type question that shares the same rating scale, introduced to the reader in subsection 3.2.2. The reader is referred to Table 3.9 for summarized descriptive statistics for Question 8, where 1 means important and 5 means unimportant to the participant.

Table 3.9: Question 8 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q8	35	3.914	0.951	2	3	4	5	5

Furthermore, the following Figure 3.11 visualizes a plot of the participants' responses and the included histogram visualizes the missing- and completed answers of Question 8.

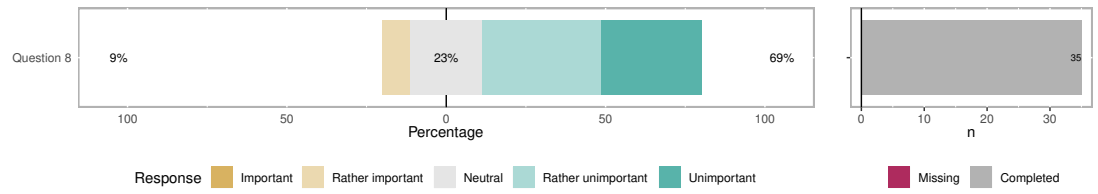


Figure 3.11: Question 8: Likert-Plot

Question 9: An early-stage enterprise established earlier is more important to me than later ones. The main purpose of this question is to ascertain the importance of recommending early-stage enterprises based on the founding date of an enterprise to a user. According to the results of Chapter 2, recommending enterprises based on the founding date is part of the user preference profile. The question is a Likert type question that shares the same rating scale, introduced to the reader in subsection 3.2.2. The reader is referred to Table 3.10 for summarized descriptive statistics for Question 9, where 1 means important and 5 means unimportant to the participant.

Table 3.10: Question 9 – Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q9	35	2.686	1.132	1	2	3	3	5

Question 9 indicate with a mean of 2.67, that the founding date of an early-stage enterprise has no impact on their selection by an investor. Furthermore, the following Figure 3.12 visualizes a plot of the participants' responses and the included histogram visualizes the missing- and completed answers of Question 9.

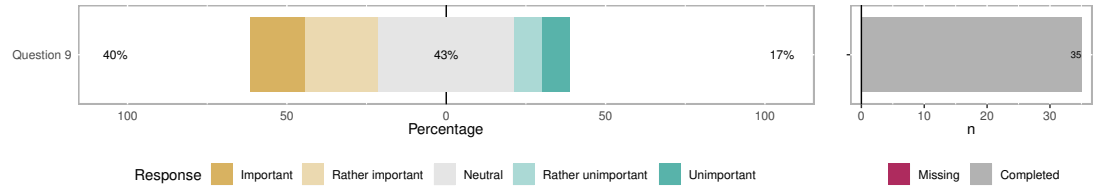


Figure 3.12: Question 9: Likert-Plot

Furthermore, the following Table 3.11 summaries the Likert-typed scale questions of the 35 participants about their user profile, where 1 means important and 5 means unimportant to the participant.

Table 3.11: Summarised descriptive statistics of Likert-typed scale questions

Question	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Q1	35	1.857	0.879	1	1	2	2	5
Q2	35	1.743	0.980	1	1	1	2	5
Q4	35	2.200	0.964	1	2	2	3	4
Q6	35	3.886	0.963	2	3	4	5	5
Q7	35	1.714	0.926	1	1	2	2	5
Q8	35	3.914	0.951	2	3	4	5	5
Q9	35	2.686	1.132	1	2	3	3	5

Question 10: Please select 5 out of 10 early-stage enterprises from the fact sheet and rank them in the order in which you would invest? (Investment preference) The main objective of this question is to gain ground truth rankings from investors for evaluating the algorithms implemented in Chapter 2. The question is a rank order type question having a rating scale from 1 to 5. The question includes a fact sheet of ten pre-defined early-stage enterprises. The total number of participants, which ranked 5 out of 10 enterprises according their investment preference is 34. Only on the basis of these enterprises will the participants be able to answer this question. The reader is referred to the appendix chapter A for a complete listing of the pre-defined ten early-stage enterprises. Furthermore, the reader is referred to Table 3.12 for summarized descriptive statistics for Question 10 grouped by the underlying early-stage enterprise. The following table lists the total number of received ratings per early-stage enterprise, stated by N and the distribution of the ratings. The highest rating is given by 1 and the lowest rating by 5. The mean represents the central rating value of the received ratings grouped by the early-stage enterprise from the fact sheet. A mean tending to 1 indicates ratings in the top ranks, a mean tending to 5 indicates ratings in the low ranks respectively.

The favoured early-stage enterprises represent by Table 3.12 as well by Figure 3.13, which gained the most ratings in the top ranks are "Mimo", with a mean of 2.55 and "TourRadar", with a mean of 2.34. The standard deviations of both enterprises are (1.6) and (1.4) respectively. The most remarkable result to emerge from the data is that the early-stage enterprise "Mimo" scored higher top-ranks than every other enterprise. Mimo is an online platform that teaches programming skills and is an award winning early-stage enterprise in Austria. However, no substantial differences between the ratings in the high vs. in the low ranks of popular and non-popular early-stage enterprises given by the participants were found. In contrast to the popular enterprises in public, "Healcloud" as a less popular early-stage enterprise in the business-to-business domain, gained also ratings in the high ranks from the participants. Further analysis on the data found out, that there were no significant differences in the rating behaviour between early-stage enterprises with an existing valuation to that one, where no valuation exist.

Furthermore, the following Figure 3.13 visualizes the overall frequency distribution of the participants' responses grouped by the early-stage enterprises of Question 10.

Table 3.12: Question 10 – Descriptive Statistics

Enterprise	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Parkbob	22	3.091	1.342	1	2.2	3	4	5
Primed Mind	6	3.333	1.506	1	2.5	4	4	5
hiMoment	15	2.867	1.125	1	2	3	3.5	5
Toby	6	3.833	1.602	1	3.250	4.5	5	5
bikemap	18	2.944	1.211	1	2	3	4	5
PreScreen	22	3.364	1.293	1	2	4	4	5
Healcloud	16	2.812	1.601	1	1	2.5	4	5
Mimo	22	2.545	1.595	1	1	2	3.8	5
TourRadar	17	2.235	1.348	1	1	2	3	5
HelperLine	11	4.091	1.136	2	3	5	5	5

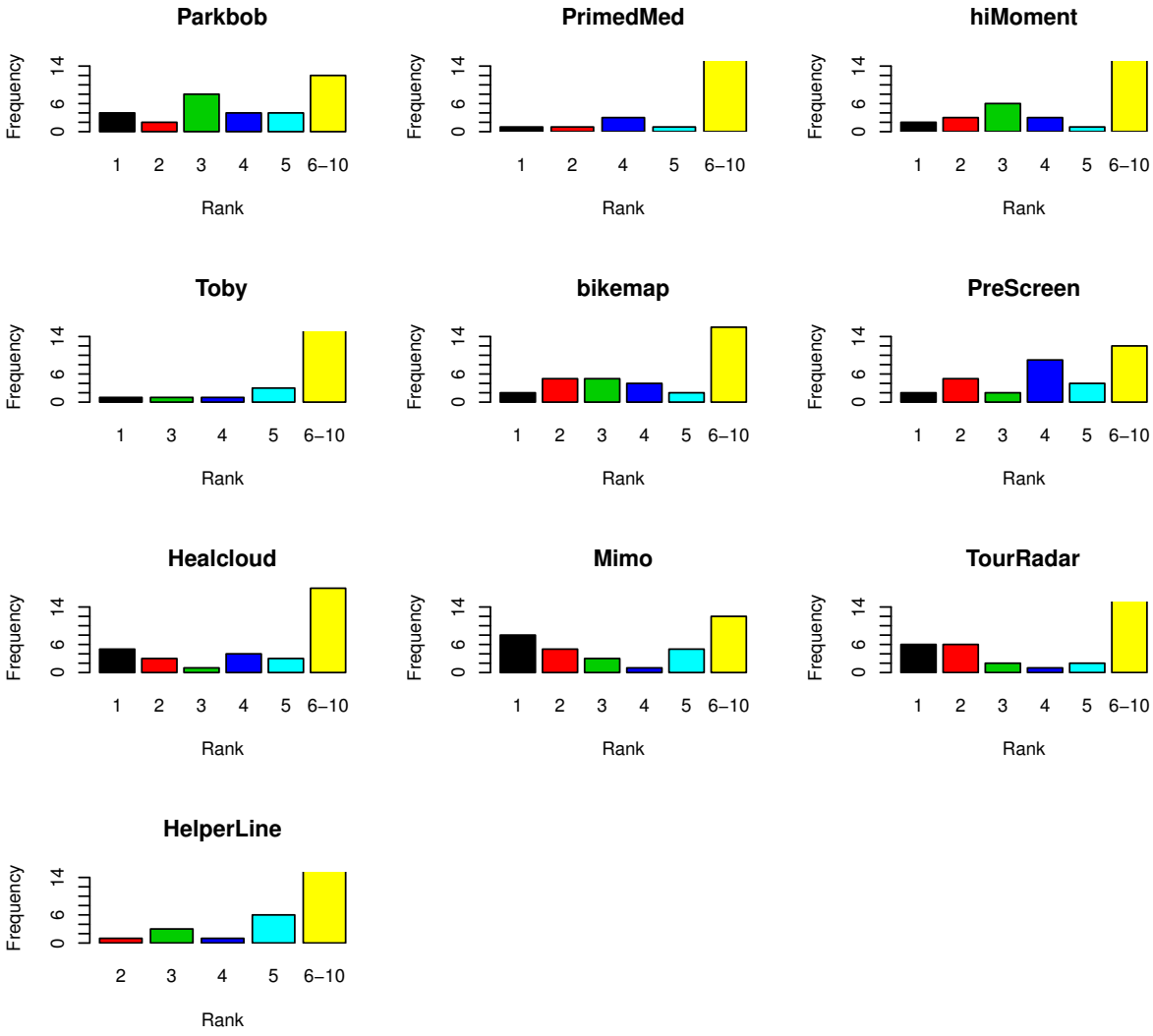


Figure 3.13: Question 10: Early-stage Enterprise Rating Frequency

Question 11: Would you take into account the opinions of well-known investors / business angels in the start-up scene in your decision-making process or would you follow them? (Multiple answers possible) The main goal of this question is to ascertain the trust relationships among participants. As an outcome in the investment decision-making & venture valuation specialisation topic by Christian Ohrfandl, recommending enterprises based on the decision of other investors is important. According to the results of Chapter 2, recommending enterprises based on the decision of other investors is part of the user preference profile. The question is a multiple choice question with an answer set indicating the following well-known investors in the DACH[10] countries. The reader is referred to Figure 3.14 for the frequency distribution of trust relationships for Question 11. Hermann Hauser, Hansi Hansmann and Herbert Gartner are established investors and have good relations within the community that people trust them.

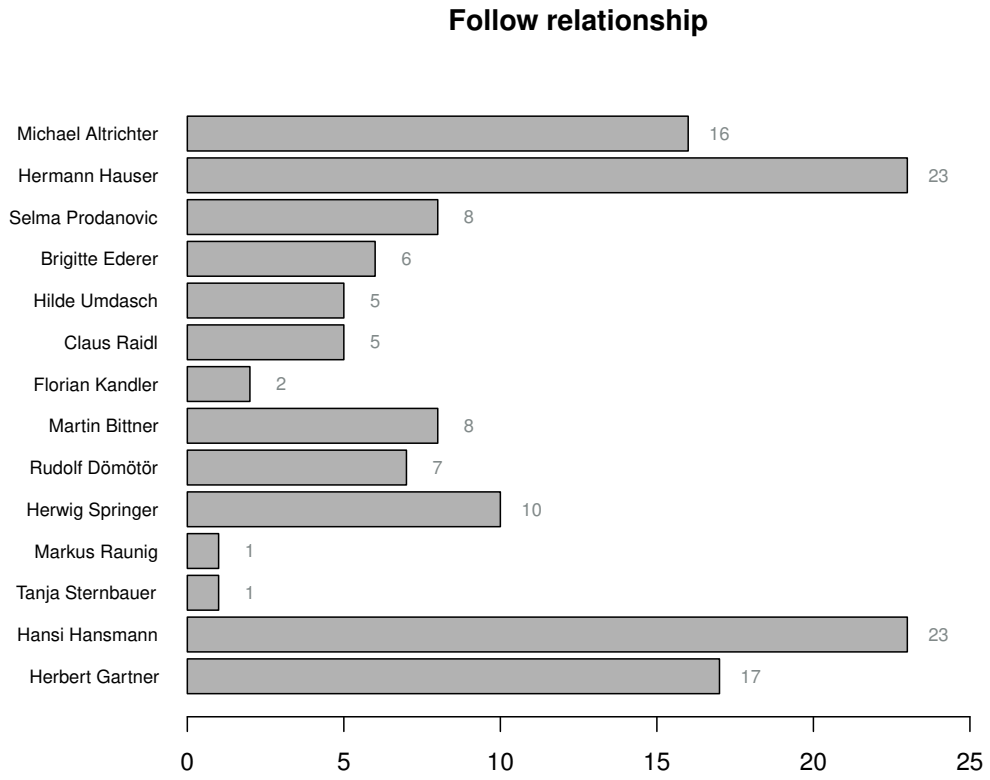


Figure 3.14: Question 11: Frequency

[10]Germany (D), Austria (A), Switzerland (CH)

One major key finding of the previous studies qualitative expert interviews may be seen in the fact that investors consider the opinion of other investors in the course of their investment decision-making processes. In particular, the participants mentioned that group investments directed by a lead investor and the general recommendation of early-stage enterprises by other investors may be seen as a common practice in the domain of early-stage enterprise investment. As a consequence, it may be concluded that investors express their connectedness towards other investors as *trust relationships*. According to Jamali and Ester [2009, p. 397], trust among users plays an important role in social networks. Therefore, one kind of trust emerges from explicit trust between users, that is, trust among users is stated as directed edge between two users. In the present paragraph, the collected data of question 11 is analysed and visualized as a social network graph, see Figure 3.15. The Social network graph identifies the investors whom the participants follow. A participant from the contact list is a person, who submitted the expert questionnaire. An investor is a person specified in question 11. The investors from questions 11 were also invited to submit the questionnaire. A green coloured circle in Figure 3.15 visualizes a participant and specified investor from question 11, who has participated the questionnaire. In the graph the yellow coloured circle visualizes a participant from the expert contact list, who submitted the questionnaire. A red coloured one a specified investor from the investor list in question 11, who did not submit the questionnaire. The size of the circle represents the number of trust relationships. The fact that only a small number of investors gain the most trust relationships indicate that group investments directed by a lead investor in the domain of early-stage enterprise investment is correct.

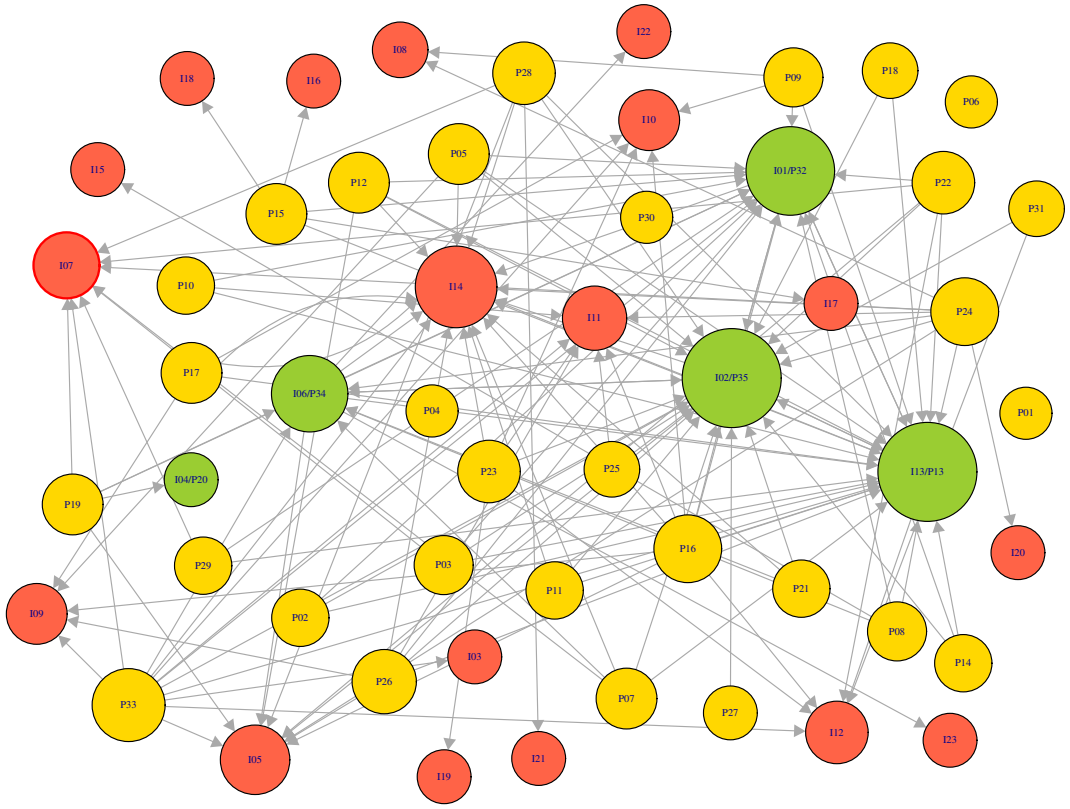


Figure 3.15: Social network graph

Table 3.13: Terminology - Social network graph (Figure 3.15)

RED	: A specified investor from the investor contact list in question 11, who did not participate the questionnaire.
YELLOW	: A participant from the investor contact list, who take the questionnaire.
GREEN	: A specified investor from the investor contact list in question 11, who is also a participant and take the questionnaire.

Question 12: To derive the significance of the individual algorithms, we require a large number of expert opinions. We would be very happy if you could recommend other people whom you think can provide inputs to our questionnaire (please provide contact details such as name, email address, phone number or website) The main objective of this question is to gain more participants and to draw broader trust relationships among investors. The question type is an open question. Question 12 to be complete, can not be analysed via descriptive statistic methods.

3.3.2 Phase 2 - Data Preparation and Dataset Generation

The *Data Preparation and Data set Generation* step produces out of the expert questionnaire an analysable and ready to model data set. It contains the tasks and transformation rules described in detail by the *Methodology*, in subsection 3.2.3.

Data splitting

The following *Data splitting* step in the experiment concentrates on partitioning the data set into a training- and a test set. As shown in the *Expert Questionnaire* section, subsection 3.3.1 the collected data of 35 participants has a manageable size. Except of one, all other participants completed the ranking question. On the basis of the ranking question, ranging from 1 to 5, the experiment splits the data into five folds. Due to the aforementioned aspects and the support by the scientific literature, the experiment of the present work utilizes as data splitting method the special kind of *Cross Validation*, the *Leave-one-out* method. Due the nature of the *Leave-one-out* method, every rated item is used as test set, while the remaining four rated items are used as training set to train a specific recommendation algorithm. The reader is referred to Phase 1 - Expert questionnaire results for details on how the ratings are given by the participant.

3.3.3 Phase 3 - Recommendations

After the utilization of the *Data splitting method* and the availability of training and test splits, the prototype recommender engine produce recommendation lists based on the introduced recommendation algorithm in Chapter 2 and the lists were stored for evaluating. In the course of the present work, the following recommendation algorithm shall be evaluated:

(a) User-based Collaborative Filtering (UB-CF), in subsection 2.3.2

(b) Item-based Collaborative Filtering (IB-CF), in subsection 2.3.2

(c) Content-based Recommendation (CB), in subsection 2.3.3

(d) Knowledge-based Recommendation (KBR), in subsection 2.3.4

(e) Social Trust Recommendation (SR), in subsection 2.3.5

(f) Hybrid Recommendation (HR), in subsection 2.3.6

3.3.4 Phase 4 - Evaluation of the Algorithms

In this section of the present work, the reader is informed about the ranking evaluation procedure. As mentioned in the background subsection 3.1.3, ranking-based metrics are for assessing recommendation accuracy and the usefulness of recommendations to a target user. Ranking itself is a beneficial method, when the purpose of the recommendation is, for each user, to choose a tiny number of items from the entire list of all available items. Before the experimental results are presented to the reader, a brief overview of the evaluation procedure is given.

The experiment starts with a shuffling of the ground truth ranking stated by the investors. The utilisation of randomisation methods avoids selection biases when selecting subgroups from the data set. In the following, the data set is split into training- and test sets according to the *Leave-one-out method* and the recommendation lists were generated by participants and by every recommendation engine. The last step is repeated for every fold. As for the execution of the accuracy measurement, a total of 1050 personalized recommendation lists were produced by the software prototype. The personalized recommendation lists results out of 35 participants times the six recommendation algorithm times the five folds of the leave-one-out method. The final phase in the evaluation procedure, the ranking-based metrics compare the quality of a particular generated ranking, based on the ground truth ranking that a participant has expressed in the expert questionnaire. In order to provide a comparison of the algorithms, the following metrics are utilised:

- **Precision @ *k* (P@k)**, where the researcher sets k ranging from 1 to 3, covering the most relevant items. Precision @ k calculates the number of relevant items among the top-k items.
- **Mean Average Precision @ *k* (MAP@k)**, where the researcher sets k ranging from 1 to 3, covering the most relevant items. MAP @ k takes also the ranking within the top-k items into account, and not only the number of relevant items.
- **Normalized Discounted Cumulative Gain @ *k* (nDCG@k)**, where the researcher sets k ranging from 1 to 3, covering the most relevant items. nDCG @ k takes also the ranking within the top-k items into account and places emphasis on higher ranked relevant items.

3.3.5 Discussion of Results

A basic assumption in a recommender system is that a system that provides more accurate predictions, will be preferred by a user. As a consequence, the present work's evaluation strategy is to concentrate on relevant items, ranked in the top positions (e.g. Top-1 to Top-3). As per results obtained indicated in Table 3.14, the most important observation in evaluating recommender systems in the domain of early-stage enterprise investment, lies in the fact that state-of-the-art techniques, such as *Collaborative Filtering* performs better than other approaches in the experiment of ranking in the top positions. In particular, the *Social Trust Recommendation algorithm* is more accurate then every other algorithm. A possible explanation for this behaviour lies in fact that the *Social Trust Recommendation algorithm* is a special case of *User-based Collaborative Filtering* with a different neighbourhood selection process. Further, the findings in the *Investment Decision-making & Venture Valuation* specialisation topic in a previous study by Christian Ohrfandl indicate that fellowship with others plays a significant role in the early-stage enterprise investment process. Thus, the *User-based Collaborative Filtering* approach, as well the *Social Trust Recommendation algorithm*, predict the majority of relevant items in the top-k positions, compared to the rest of the implemented algorithm. The example experiment showed that simple algorithms seem very effective compared to ones that are more complex.

Additionally, approaches such as *Knowledge-based Recommendation*, *Hybrid Recommendations*, *Content-based Recommendations*, etc. are less precise and less useful for a user. A further interesting observation, which is not part of the present research, is that from the performance perspective, the *Hybrid Recommendation* approach takes much more computational power than every other implemented recommendation algorithm.

Table 3.14 reports the overall test results of the evaluated recommendation algorithms regarding various accuracy measures. All the mentioned measurements apply to each participant, and the average of all 35 participants is reported. The metrics determine the order of the predicted items for a participant, and compare this with the order of the ground truth ranking. In general, higher values of the ranking measures indicate better accuracy. The accuracy measures refer only to the top 1 to 3 items of the ranked list of items, this is denoted by '@[1-3]'. In section 3.4 a complete list of all abbreviations is referred. For visualisation purposes, the following Table 3.14 reported the accuracy results of the evaluated recommender listed by every recommendation algorithm. The highest values for each metric are highlighted in Table 3.14. The *Social Trust Recommendation algorithm* performs best in relation to other algorithm due the fact of a different neighbourhood selection process. In the process only trusted users and their rated early-stage enterprises are included in the calculation process.

Table 3.14: Overall Evaluation Results

Recommender	P@1	P@2	P@3	MAP@1	MAP@2	MAP@3	nDCG@1	nDCG@2	nDCG@3
UB-CF	0,2176	0,2529	0,2235	0,2176	0,3618	0,4167	0,2176	0,2450	0,2261
IB-CF	0,1471	0,1500	0,1490	0,1471	0,2235	0,2725	0,1471	0,1493	0,1488
CB	0,1235	0,1500	0,1216	0,1235	0,2118	0,2333	0,1235	0,1440	0,1254
KBR	0,1529	0,1176	0,1176	0,1529	0,1941	0,2333	0,1529	0,1256	0,1238
SR	**0,2571**	**0,2607**	**0,2238**	**0,2571**	**0,3893**	**0,4393**	**0,2571**	**0,2599**	**0,2341**
HR	0,1235	0,1088	0,1078	0,1235	0,1706	0,2059	0,1235	0,1122	0,1107

Table 3.15 reports the *Spearman rank correlation* that measures the strength of association between two recommendation lists and the direction of the relationship. In terms of the strength of relationship, the values of the correlation coefficient varies between +1 and -1. A value of +1 indicates a perfect degree of association between the two lists and a value of -1 implies that the lists are the reverse of each other. Consequently, -1 indicates that the lists are more dissimilar. As the correlation coefficient factor approximates to 0, there is no correlation between the two recommendation lists. A moderate association (0.4439) indicate only the *Knowledge-based Recommendation* approach with the *Hybrid Recommendation* one. The *Hybrid Recommendation* algorithm is based on the *Knowledge-based Recommendation* algorithm, as an explanation for this moderate association.

Table 3.15: Spearman's rank correlation coefficient matrix

Recommender	UB-CF	IB-CF	CB	SR	KBR	HR
UB-CF	**1,0**	0,1736	-0,2048	0,0886	0,0569	0,0345
IB-CF	0,1736	**1,0**	-0,0312	0,0715	0,0856	0,1773
CB	-0,2048	-0,0312	**1,0**	0,0841	0,2521	0,1540
SR	0,0886	0,0715	0,0841	**1,0**	0,0709	0,1264
KBR	0,0569	0,0856	0,2521	0,0709	**1,0**	0,4439
HR	0,0345	0,1773	0,1540	0,1264	0,4439	**1,0**

3.3.6 Limitation of the Experiment

Off-line evaluation methods are the most established technique in scientific research for evaluating recommendation algorithms [Jannach et al., 2010, p. 177]. According to Jannach et al. [2010], more than 50 % of the researchers in the last decade have selected Off-line evaluation of recommender systems and a variety of generally accepted evaluation measures were developed. A major drawback of off-line evaluation is that it does not measure the current state of the user for reacting to the recommender system in the future. Therefore, further usage of the software prototype might evolve the data over time and the actual recommendation lists may not reflect the recommendation lists in the future. In addition, a key limitation may be seen in the regional limitation of the expert questionnaire. Further, not all of the implemented similarity measures in connection with a recommendation algorithm could be tested, due the lack of data. The click rate could not measured through the expert questionnaire; so for, the similarity measure based on clicks, the reader is referred to subsection 2.3.2 is excluded for evaluating in the present work. In addition, for investment similarity, the reader is referred to subsection 2.3.2, which is excluded for evaluating. Reasons justifying this exclusion may be seen in the fact that an investor regards this information as private and is not willing to share it. Nevertheless, since these similarity measures use the same recommendation algorithm, it may be concluded that they will perform in an analogous way. Last but not least the missing support by the experts may be seen as a limitation of the present experiment. There was no further interest on the present research and on the software prototype itself. The experiment depends on having access to experts, organizations, or further data to be able to generate a valid expert data set. The risk of being biased is being quite high in the domain of early-stage enterprise investment, due the fact that a expert may begin to sympathise with an enterprise, which results in biased data. The possibility that a expert already invested in one of the enterprises in the fact-sheet can not be ruled out.

3.4 Answers to the Research Questions

This section discusses the present Chapter's research questions. The following questions are answered on the basis of the results obtained in section 3.3.

How does the recommendation quality and the usefulness of the recommendation system affect user satisfaction?

The question outlines the basic principle of how the recommendation quality and the usefulness to a user can be measured. Further, this question contains two sub-questions about the utilised methodology, the principle of evaluation and the effect on future research. In these fast-paced times, information power is of great importance, especially in the domain of early-stage enterprise investment. As described in detail in the introduction (Chapter 1), investors in that domain are confronted with the problem of *information overload*. Therefore, the need for information filtering techniques based on computational recommendation systems emerges to satisfy the user. Subsequently, a basic assumption in recommender system is that a system that provides more accurate predictions will be preferable to the user. In a domain with quick investment decisions on the agenda, persons do not have the time to rummage through long lists. The literature suggests that ranking-based evaluation best fits in the measurement of the usefulness of a recommender system. Ranking-based evaluations provide a more realistic perspective of the true usefulness of the recommender system because, in general, the user only views the top-k items rather than all the items. This phenomena can also be seen in the application of web search engines, where higher placed search results can be seen than later on results. Finally, the reader is referred to the following sub-questions of the current research question for detailed answers of the utilised scientific approach.

How may the recommender system be evaluated in terms of recommendation quality?

Ranking-based metrics as a part of *Off-line Experiments* have the capability for evaluating recommender systems. Ranking-based metrics compare the order of the predicted recommendation list for a given user with a ground truth ranking stated by a real user. The utility calculated during this comparison indicates the recommendation quality/usefulness for the user.

Which methodologies may be utilized to evaluate the recommendation system?

The scientific literature suggests a variety of evaluation methods and provides different standardised benchmark frameworks. Due to the fact of a variety of applicable recommendation algorithm in the domain of early-stage enterprise investment, researched in chapter 2. These recommendation algorithms include the following techniques: Collaborative filtering, Content-based, Knowledge-based, Social- and Hybrid recommendation. A modest and less time consuming method can be found in an *Off-line Experiment* with historical user-interactions and is applied

by the present work. An Off-line Experiment has the advantage of comparing all these candidate algorithms at low cost, because no further user-interaction is required.

Which implications do the results indicate?

As simple as they are, the results indicate that simple algorithms, such as *User-based Collaborative Filtering* or *Social Trust Recommendations* seem very effective, compared to the ones that are more complex. For a detailed representation of the results, see section 3.3.

How do the findings affect future research?

A crucial aspect of this assumption about recommendation quality is that it can only be verified by explicitly asking the user about the recommended items. This is the starting point for further research, with a small set of candidate algorithms in the domain of early-stage enterprise investment. The candidate algorithm can be tested by more costly scientific instruments, such as user studies or on-line experiments.

Conclusion

In this section, the contribution of the present work is summarised and further challenges, which could be addressed in future research, are introduced to the reader. The objective of the present Chapter is to outline a procedure to evaluate the modelled recommendation algorithm in Chapter 2, based on the outcome of co-author Christian Ohrfandl's specialisation topic *Investment Decision-making & Venture Valuation.* A further goal of the current Chapter is to filter out inappropriate recommendation techniques in the domain of early-stage enterprise investment. The literature suggests a variety of evaluation methods and provides different standardised benchmark frameworks but there are no de facto rules or standards how to evaluate a recommendation algorithm in the domain of early-stage enterprise investment. In other recommender system evaluation experiments, where the researchers train their recommendation algorithm on public available data set, such as MovieLens data set, in the domain of early-stage enterprise investment, is no public dataset available. Furthermore, very few publications evaluating recommender systems are available in the literature that can be utilised for evaluation in the domain of early-stage enterprise investment. Therefore, the researched scientific instrument can be valued as a novel approach for evaluating recommender systems in the domain of early-stage enterprise investment. To sum up, the main contribution of the present work are:

- Identified venture valuation methods
- Identified investment decision-making criteria
- Determined requirements of a recommender system
- Construction of a recommender system software prototype
- Evaluation of the parallelised recommendation algorithms

However, there are limitations to the present work. One is the low number of participants of answering the expert questionnaire, which leads to data possibly insufficient to conduct a meaningful experiment and to make significant statements on the evaluated recommendation technique. Regional limitation can be seen as a further limitation of the present work.

Finally, other types of research can be conducted making use of on-line experiments or user studies in further research. On-line experiments or user studies are more costly compared to off-line experiments. In off-line experiments, a variety of recommendation techniques can be tested at a low cost, whereas in on-line experiments/user studies a relatively small set of candidate recommendation techniques are tested. Further research can address the *User-based Collaborative Filtering* as well the *Social Trust* recommendation technique for a broader analysis.

Another opportunity is to form an own early-stage enterprise with the obtained results of the present work and generate business value.

List of Abbreviations

AveP@k Average Precision @ *k* 63

CB Content-based Recommendation . . . . . xiii, 32, 91, 93, 94

HR Hybrid Recommendation . . . . . xiii, 46, 91, 93, 94

IB-CF Item-based Collaborative Filtering 29, 91, 93, 94

KBR Knowledge-based Recommendation . . . . . xiii, 35, 91, 93, 94

MAP@1 Mean Average Precision @ *1* 93

MAP@2 Mean Average Precision @ *2* 93

MAP@3 Mean Average Precision @ *3* 93

MAP@k Mean Average Precision @ *k* 64, 92

nDCG@1 Normalized Discounted Cumulative Gain @ *1* 93

nDCG@2 Normalized Discounted Cumulative Gain @ *2* 93

nDCG@3 Normalized Discounted Cumulative Gain @ *3* 93

nDCG@k Normalized Discounted Cumulative Gain @ *k* 64, 92

P Precision 60

P@1 Precision @ *1* 93

P@2 Precision @ *2* 93

P@3 Precision @ *3* 93

P@k Precision @ *k* 63, 64, 92

R Recall 60, 61

SR Social Trust Recommendation . . . . . xiii, 44, 91, 93, 94

UB-CF User-based Collaborative Filtering 26, 91, 93, 94

List of Figures

List of Tables

Bibliography

Primary Sources

Achleitner, Ann-Kristin et al. (2004). "Venture Capital/Private Equity-Studie 2004: Company (E)valuation und EVCA Valuation Guidelines : Bestandsaufnahme der Unternehmensbewertungpraxis von Beteiligungskapitalgesellschaften". In: *Finanz-Betrieb : FB ; Zeitschrift für Unternehmensfinanzierung und Finanzmanagement, 2004, Vol.6(10), pp. 701-709*. ISSN: 14378981.

Adomavicius, G. and A. Tuzhilin (2005). "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". In: *IEEE Transactions on Knowledge and Data Engineering* 17.6, pp. 734–749. ISSN: 1041-4347.

Adomavicius, Gediminas, Nikos Manouselis, and YoungOk Kwon (2011). "Multi-Criteria Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci et al. Boston, MA: Springer US, pp. 769–803. ISBN: 978-0-387-85820-3.

Aggarwal, Charu C. (2015). *Data Mining*. Cham : Springer International Publishing : Imprint: Springer, ISBN: 3-319-14142-2.

— (2016). *Recommender Systems: The Textbook*. 1st. New York, NY, USA: Springer International Publishing. ISBN: 9783319296579.

Atteslander, Peter (2010). *Methoden der empirischen Sozialforschung*. 11., neu bearb. u. erw. Aufl. ESV basics. Berlin: Schmidt. ISBN: 9783503097401.

Burke, Robin (2002). "Hybrid Recommender Systems: Survey and Experiments". In: *User Modeling and User-Adapted Interaction* 12.4, pp. 331–370. ISSN: 1573-1391.

Celma, Òscar (2010). "Music Recommendation". In: *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-13287-2.

Ceri Stefano Bozzon Alessandro, Brambilla Marco (2013). *Web Information Retrieval*. Data-Centric Systems and Applications. Berlin, Heidelberg : Springer Berlin Heidelberg : Imprint: Springer, ISBN: 3-642-39314-4.

Ceri Bozzon, Brambilla (2013). *Recommender Systems: An Introduction*. Berlin, B, Germany: Springer. ISBN: 978-3-642-39313-6.

Christian Ohrfandl, Johannes Luef (2018). "Recommender Systems in the Domain of Early-stage Enterprise Investment : Investment Decision-making & Venture Valuation".

Edgar, Thomas W. and David O. Manz (2017). "Chapter 6 - Machine Learning". In: *Research Methods for Cyber Security*. Ed. by Thomas W. Edgar and David O. Manz. Syngress, pp. 153–173. ISBN: 978-0-12-805349-2.

Eurostat (2016a). *Business demography by size class (from 2004 onwards, NACE Rev. 2)*. As consulted online on 24 July 2016. URL: http://appsso.eurostat.ec.europa.eu/nui/submitViewTableAction.do.

— (2016b). *Business demography statistics*. As consulted online on 24 July 2016. URL: http://ec.europa.eu/eurostat/statistics-explained/index.php/Business_demography_statistics.

F. Brandenburg A. Gleissner, A. Hofmeier (2011). "Comparing and Aggregating Partial Orders with Kendall Tau Distances". In: MIP-1102.

Fagin, Ronald, Ravi Kumar, and D. Sivakumar (2003). "Comparing Top k Lists". In: *SIAM Journal on Discrete Mathematics* 17.1, pp. 134–160.

Few, Stephen (2006). *Information dashboard design : the effective visual communication of data*. 1st. North Sebastopol, USA: Sebastopol, Calif, O'Reilly. ISBN: 978-0-596-10016-2.

Garcia-Lapresta, Jose Luis and Miguel Martinez-Panero (2002). "Borda Count versus Approval Voting: A Fuzzy Approach". In: *Public Choice* 112.1-2, pp. 167–184.

Gedikli, Fatih (2013). *Recommender Systems and the Social Web: Leveraging Tagging Data for Recommender Systems*. Springer Vieweg. ISBN: 9783658019471.

Herlocker, Jon, Joseph A. Konstan, and John Riedl (2002). "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms". In: *Information Retrieval* 5.4, pp. 287–310. ISSN: 1573-7659.

Herlocker, Jonathan L. et al. (2004). "Evaluating Collaborative Filtering Recommender Systems". In: *ACM Trans. Inf. Syst.* 22.1, pp. 5–53. ISSN: 1046-8188.

Huang, Anna (2008). *Similarity Measures for Text Document Clustering*. As consulted online on 05 August 2017. Christchurch, New Zealand.

Jaccard, Paul (1912). "THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1". In: *New Phytologist* 11.2, pp. 37–50. ISSN: 1469-8137.

Jamali, Mohsen and Martin Ester (2009). "TrustWalker: A Random Walk Model for Combining Trust-based and Item-based Recommendation". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: ACM, pp. 397–406. ISBN: 978-1-60558-495-9.

Jannach, Dietmar et al. (2010). *Recommender Systems: An Introduction*. 1st. New York, NY, USA: Cambridge University Press. ISBN: 9780521493369.

Klapper, Leora and Inessa Love (2011). "The impact of the financial crisis on new firm registration". In: *Economics Letters* 113.1. As consulted online on 24 July 2016, pp. 1–4. ISSN: 0165-1765.

Manning, Christopher D., Prabhakara Raghavan, and Hinrich Schütze (2009). *Introduction to information retrieval*. 1. publ. Cambridge [u.a.]: Cambridge Univ. Press. ISBN: 9780521865715.

Massa, Paolo and Paolo Avesani (2004). "Trust-Aware Collaborative Filtering for Recommender Systems". In: *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004, Agia Napa, Cyprus, October 25-29, 2004. Proceedings, Part I*. Ed. by Robert Meersman and Zahir Tari. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 492–508. ISBN: 978-3-540-30468-5.

OECD (2012). *Business start-up rates diverging across OECD economies*. Ed. by Organisation for Economic Co-operation, Development (OECD) – Media Relations, and Statistics Directorate. As consulted online on 24 July 2016. URL: http://www.oecd.org/std/business-stats/businessstart-upratesdivergingacrossoecdeconomies.htm.

— (2016). *Timely Indicators of Entrepreneurship (ISIC4) – Number of enterprise entries (Q1-2007 - Q1-2016*. Ed. by Organisation for Economic Co-operation and Development (OECD) – OECD.Stat. As consulted online on 24 July 2016. URL: http://stats.oecd.org/Index.aspx?DataSetCode=TIMELY_BDS_ISIC4#.

Pedhazur, Elazar J. and Liora Pedhazur Schmelkin (1991). *Measurement, design, and analysis*. Hillsdale, NJ [u.a.]: Erlbaum. ISBN: 0805810633.

Ricci, Francesco et al. (2010). *Recommender Systems Handbook*. 1st. New York, NY, USA: Springer-Verlag New York, Inc. ISBN: 9780387858197.

Robertson, Judy (2012). "Likert-type Scales, Statistical Methods, and Effect Sizes". In: *Commun. ACM* 55.5, pp. 6–7. ISSN: 0001-0782.

Rudolf, Markus and Peter Witt (2002). *Bewertung von Wachstumsunternehmen: traditionelle und innovative Methoden im Vergleich*. Springer-Verlag.

Said, Alan and Alejandro Bellog (2014). "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks". In: *Proceedings of the 8th ACM Conference on Recommender Systems*. RecSys '14. Foster City, Silicon Valley, California, USA: ACM, pp. 129–136. ISBN: 978-1-4503-2668-1.

Sarwar, Badrul et al. (2001). "Item-based Collaborative Filtering Recommendation Algorithms". In: *Proceedings of the 10th International Conference on World Wide Web*. WWW '01. Hong Kong, Hong Kong: ACM, pp. 285–295. ISBN: 1-58113-348-0.

Schein, Andrew I. et al. (2002). "Methods and Metrics for Cold-start Recommendations". In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '02. Tampere, Finland: ACM, pp. 253–260. ISBN: 1-58113-561-0.

Shelby, Z. (2012). *Constrained RESTful Environments (CoRE) Link Format*. RFC 6690. RFC Editor, pp. 1–22.

Shneiderman, Ben and Catherine Plaisant (2016). *Designing the user interface : strategies for effective human-computer interaction*. 6th. Boston , USA: Upper Saddle River, NJ, Addison-Wesley. ISBN: 978-0134380384.

Steck, Harald (2013). "Evaluation of Recommendations: Rating-prediction and Ranking". In: *Proceedings of the 7th ACM Conference on Recommender Systems*. RecSys '13. Hong Kong, China: ACM, pp. 213–220. ISBN: 978-1-4503-2409-0.

Stone, Thomas, Weinan Zhang, and Xiaoxue Zhao (2013). "An Empirical Study of Top-n Recommendation for Venture Finance". In: *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, pp. 1865–1868. ISBN: 978-1-4503-2263-8.

Victor, P. et al. (2011). "Trust- and Distrust-Based Recommendations for Controversial Reviews". In: *IEEE Intelligent Systems* 26.1, pp. 48–55. ISSN: 1541-1672.

WKO (2016). *Unternehmensneugründungen 1993 - 2015 – Endgültige Ergebnisse*. Ed. by Austrian Economic Chamber – Statistik Österreich – Stabsabteilung Statistik. As consulted online on 24 July 2016. URL: http://wko.at/statistik/Extranet/Neugr/ng2015e-gesamt.pdf.

Zhao, Xiaoxue, Weinan Zhang, and Jun Wang (2015). "Risk-Hedged Venture Capital Investment Recommendation". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM.

Secondary Sources

TU Wien (2013). *Studienplan (Curriculum) für das Masterstudium Business Informatics an der Technischen Universität Wien*. Studienplan 2011, in der Fassung vom Juni 2013, gültig ab 1. Oktober 2013. As consulted online on 12 August 2016. Vienna University of Technology, pp. 1–93.

Appendices

APPENDIX A

Fact sheet

The following tables in the appendix introduces the reader to the characteristics of the ten pre-defined early-stage enterprises. On basis of the attached fact sheet, the participants were able to answer question number ten, subsection 3.2.2 of the expert questionnaire. The original fact sheet was written in German language and was delivered to the participant on an A4 page, the following is a translated version of it.

Table A.1: Fact sheet - Parkbob

Parkbob – Find free parking spots - The number of available real-time spots increases daily. Avoid parking fines Information about parking rules Alerts in payment zones, resident zones, etc. We use bicycles and we take the underground. But we are also driving cars. We know the problems associated with parking from our own experience. That's why we've joined forces, and developed our innovative parking app - helping everyone save time, money and nerves!	
Life cycle:	Seed
Product(USP):	Traffic management/Traffic routing(GIS)
Market sector:	Smart City
Public interest:	Yes
Valuation:	Unknown
Management team:	No (1 Founder)
Founded:	2015
Further details:	http://www.parkbob.com/

Table A.2: Fact sheet - Primed Mind

Primed Mind – Mindset coaching combines the elements of meditation, hypnotherapy, and life coaching. By that, we create a structure that is aimed to help you achieve your goals. Life is about balance. Often, the most productive thing you can do is relax. Meditation and mindfulness techniques will allow you to reduce anxiety and stress and you will learn from your past and live in the present. With these techniques it will help you stay in the present and you will reach a state of clarity more easily. In order to be the best, you must feel your best! Hypnotherapy techniques are designed to release you from your past and break down subconscious barriers that are holding you back from the life you want to live. Our visualization techniques will help you to reduce anxiety in different settings, such as high pressure situations, public speaking, and social interactions. After completing our sessions, you will feel refocused and you will not let your past stand in your way.	
Life cycle:	Seed
Product(USP):	Mobile Mindset-Coaching App
Market sector:	Smart Life
Public interest:	Yes
Valuation:	Unknown
Management team:	Yes (2 Founders)
Founded:	2017
Further details:	https://www.primedmind.com/

Table A.3: Fact sheet - hiMoment

hiMoment – Your daily happiness boost at your fingertips. hiMoment is built on a powerful principle: Use past and present moments to build your future happiness. Our algorithmic micro-coaching will help you focus on the great moments in your life and build a habit for happiness. Get your daily happiness exercises, teach yourself mindfulness and increase your mood and health.	
Life cycle:	Seed
Product(USP):	Mood barometer
Market sector:	Health
Public interest:	Yes
Valuation:	Yes, EUR 1.3 M (10/2017)
Management team:	Yes (2 Founders)
Founded:	2017
Further details:	https://himoment.com/

Table A.4: Fact sheet - Toby

Toby – A good shopping list learns about you and makes your life easier. With Toby you can create multiple coloured lists and share them with your friends. Toby can also send you location based reminders for things you urgently need to buy. Toby is your smart shopping assistant and is super easy to use. We have bundled all the most popular features of shopping lists in just one app.	
Life cycle:	Start-up
Product(USP):	Shopping Assistant
Market sector:	Shopping
Public interest:	Yes
Valuation:	Yes, EUR 300 T (07/2016)
Management team:	Yes (2 Founders)
Founded:	2016
Further details:	http://tobyapp.com/

Table A.5: Fact sheet - Bikemap

Bikemap – You love to discover new bike routes and great cycling regions? No matter if you're riding a mountainbike, crossbike, e-bike or road bike - the free Bikemap cycling route planner helps you discover the latest and greatest cycling paths.	
Life cycle:	Start-up
Product(USP):	Cycling routes management
Market sector:	Tourism
Public interest:	Yes
Valuation:	Unknown
Management team:	Yes (2 Founders)
Founded:	2007
Further details:	https://www.bikemap.net/

Table A.6: Fact sheet - PreScreen

PreScreen – Prescreen is a cloud-based applicant tracking system that enables you to publish job vacancies online and offline and give candidates the opportunity to apply for jobs directly. Prescreen gathers all applications collectively, analyses the data and helps you with your evaluation. This makes time and labour-intensive processes more efficient.	
Life cycle:	Start-up
Product(USP):	Candidate management
Market sector:	e-Recruiting/ Human resources City
Public interest:	No
Valuation:	Unknown
Management team:	Yes (7 Founders)
Founded:	2013
Further details:	https://prescreen.io/

Table A.7: Fact sheet - Healcloud

Healcloud – At Healcloud, we are working to transform the quality of care and healthcare data. Our patent pending solutions empower patients, liberate medical practitioners, and leverage big data to deliver actionable insights that help drive better health outcomes, supporting clinical trials and life sciences research. Join us in our social mission today.	
Life cycle:	Seed
Product(USP):	e-Health management
Market sector:	Health
Public interest:	No, B2B
Valuation:	Yes, EUR 5M (03/2016)
Management team:	Yes (2 Founders)
Founded:	2015
Further details:	http://healcloud.com/de/home-de/

Table A.8: Fact sheet - Mimo

Mimo – Learn to code, build apps, and much more. Mimo is an app that teaches computer science in a fun and interactive way. Join more than 2 million learners: learn to code, make apps/games/websites, automate your life, advance your career, and much more – no matter how much experience and time you have! Mimo creates a personalized curriculum of fun and effective exercises, projects, and challenges that fits into your daily routine and keeps you motivated.	
Life cycle:	Start-up
Product(USP):	Programming study platform
Market sector:	Education
Public interest:	Yes
Valuation:	Unknown
Management team:	Yes (4 Founders)
Founded:	2016
Further details:	https://getmimo.com/

Table A.9: Fact sheet - Tourradar

TourRadar – to be the online marketplace for touring. TourRadar provides travellers with everything they need to plan and book their next great escape in one place. From thoughtfully crafted tours and their itineraries, videos and photos, to how-to guides and travel experts available 24-hours a day, 7-days a week.	
Life cycle:	Start-up
Product(USP):	Trip planning
Market sector:	e-Commerce
Public interest:	Yes
Valuation:	Unknown
Management team:	No (1 Founder)
Founded:	2010
Further details:	https://www.tourradar.com/

Table A.10: Fact sheet - Helferline

Helferline – The computer does not do what it should, the Internet is not working, the printer is spinning and the smart phone is on strike? Everyone knows these problems. All this is now a case for HELFERLINE: The Founders have a network of qualified technicians built to help quickly and cheaply on the ground; booked directly comfortable via hotline.	
Life cycle:	Seed
Product(USP):	Brokerage platform
Market sector:	Services
Public interest:	Yes
Valuation:	Unknown
Management team:	Yes (3 Founders)
Founded:	2016
Further details:	https://www.helferline.at/