Package recommendation to support a nutritionally healthier eating pattern

S.N. van Schaik

Utrecht University, the Netherlands

Abstract

Worldwide the number of people with overweight is increasing. One of the contributing factors is the overload of information. This is a problem which can be solved by incorporating a recommender system. In this research package recommendation is used to create personalized recipe week schedules to help users with a nutritionally healthier eating pattern. The contributions of this research are fourfold. (1) A systematic literature review has been conducted in the field of package recommendation. (2) A new form of package recommendation is proposed which makes use of multiple constraints and adjusted algorithms. (3) A system is build which could help users with a nutritionally healthier eating pattern. (4) A new data set is created which can be used for further research.

1 Introduction

The past years people asked me to help them with losing weight or a healthier eating pattern. While I have no degree in food, I have personal experience with food because of my sport routines. I used my knowledge to help people with their questions and made them food schedules. These schedules were personalised based on the taste preferences and nutritional needs of a person. The demand in my personal environment for these schedules increased, so I started to search for a better solution.

At first an exploratory research was conducted to explore the problem. Conversations with nutrition experts revealed some interesting facts. Dietitians manually create food schedules for their clients, meaning there is no (good) system to create personalised food schedules in a more efficient way. This gave me the idea to start this research and investigate how to build a system for producing personalized food schedules myself.

1.1 Problem statement

The need for systems to help people with a nutritionally healthy eating pattern is increasing. Worldwide the number of people with overweight is increasing [1].

Nowadays, 1 out of 3 adults worldwide is overweight and this number is even higher in the Netherlands with 1 out of 2 [2]. Being overweight is one of the main causes of nutrition-related noncommunicable diseases (NCD) such as diabetes and cardiovascular diseases, and increases the risk of several other diseases [1]. One of the main causes of being overweight is bad diets. Bad diets are thereby the second death cause worldwide, responsible for 18.8% of deaths [3]. There are many factors that impact people's eating patterns, including education [4], cost [5] and dish proportions. A change of eating patterns is needed, but the field of nutrition seems very complicated and the lack of supporting tools makes it hard for people to make the right decisions.

My hypothesis is that people do not know where to start with a healthier eating pattern because of this lack of knowledge and information overload. In the early 90's the IT domain started to research the information overload problem. This resulted in the field of recommender systems. Recommender systems suggest items to users that they probably like, filtering out all irrelevant items [6]. Different techniques have been used, such as collaborative filtering [7], content-based filtering [8] and hybrid filtering [9], which combines the former techniques. Techniques have become increasingly sophisticated, and the application domains have been extended.

Currently, recommendations are no longer only for single items, but also for sets of items, for instance, to plan a tourist route [10], decide what courses to follow [11], or, in this case, what combination of meals a person should eat to follow a nutritionally healthy eating pattern. This new field which recommends a set of items is called package- or bundle recommendation, from this point on called package recommendation.

Package recommendation is defined as the suggestion of a group of items that fit well together [12]. It could potentially help people with a healthier eating pattern. By making use of package recommendation eating schedules could be made based on personal taste and goals that people have, for instance, to lose weight or to cope with restrictions such as diabetes. To research the possibilities of package recommendation for healthy eating the following research questions have been made.

1.2 Research questions

To solve my main objective the following research question is composed.

RQ: *How can package recommendation be used to assist a nutritionally healthier eating pattern?*

This main research question will be answered by 3 sub questions, as specified below.

SQ1: *What is the current state of the art in the field of package recommendation?*

The first sub question is to create an understanding of the research performed and methods used until now. No systematic literature review has yet been conducted in this field. Figure 1 shows that the number of articles in this field is rapidly increasing. Therefor a timely systematic literature review is not an unnecessary luxury.

SQ2: *How can existing algorithms be applied and what adaptations are needed for food recommendation?*

Based on the outcomes of SQ1 different methods for package recommendation are considered and finally methods are picked which will be used to build a system which can recommend day- and week recipe schedules based on package recommendation.

SQ3: *How well does the system perform?*

Finally the system is evaluated to set a benchmark for new research.

1.3 Research methods

This thesis uses the following research methods.

Systematic literature review. To find the state-of-the-art in the field of package recommendation, answering SQ1, first a systematic literature review is conducted. As a good systematic literature review requires, multiple investigators conducted the review to ensure the research is done properly. The author of the thesis was helped by Agung Toto Wibowo and his first supervisor.

Evaluation. To evaluate the accuracy and diversity an evaluation of the system was conducted. This was done based on the packages created by the system, but also on individual algorithms of the system.

1.4 Contributions

The contributions of this research are fourfold. (1) A systematic literature review of the current state of the art and future research opportunities in package recommendation. (2) A new form of package creation is proposed which makes use of multiple constraints, adjusted algorithms and a hybrid approach. (3) A system is build which can be used to help people advise the right recipes for a nutritionally healthier eating pattern. (4) A new data set is created which can be used for further research to the healthy eating or package recommendation domains.

1.5 Thesis outline

In the following section, the research method for the systematic literature review is explained. It contains the research questions, search strategy, criteria and procedures to find, analyze and synthesize the data. Section 3 discusses the results of the review. The domains, package types, recommendation input types, package recommender systems (PRS) phases and techniques, and evaluation methods and metrics are explained. Section 4 introduces the guidelines provided by Foodfirst Network, the organization where this research was conducted. Section 5 describes the system and the methods used to construct recipe week schedules. In Section 6 the creation of the data set is explained. Section 7 explains the evaluations performed on the system. The last section contains the conclusions of this research and directions for future work.

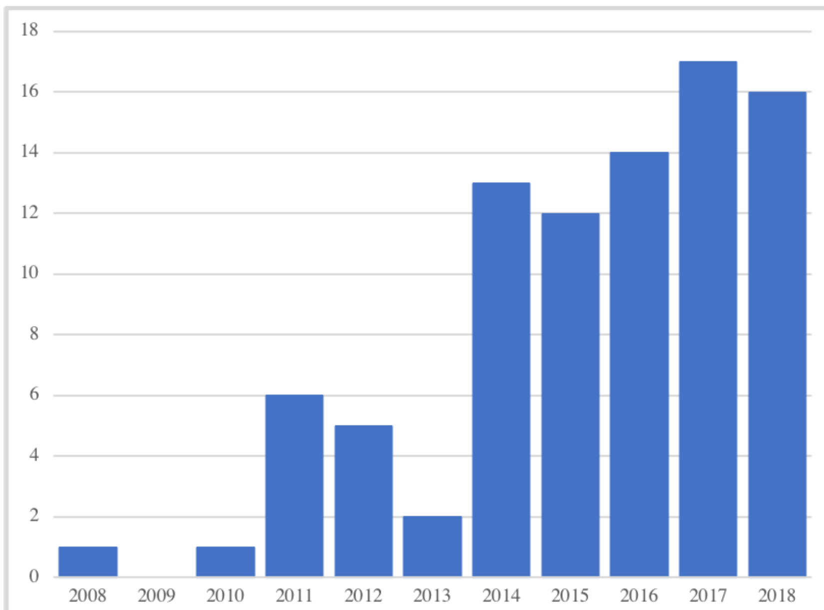

Figure 1: Number of PRS articles through the years in the literature review

Table 1: Search sources

Databases	ACM IEEE Xplore ScienceDirect Scopus Google Scholar
Searched items	Journal, conference and workshop papers
Search applied on	Article title, abstracts and keywords
Language	English
Search period	1999-2018

2 Data collection systematic literature review

2.1 Research questions

The main goal of this systematic literature review is to find the techniques used for package recommendations and how the performance of each technique is evaluated. To investigate this, we used the following research questions:

1. In what domains are package recommender systems applied?
2. What are the different techniques for package recommendation and how do they work?
3. What evaluation methods are used to evaluate the performance of the package recommendation methods?
4. What evaluation metrics are used when evaluating the performance of package recommendation methods?

2.2 Search strategy

To find all the relevant information about package recommendation, we defined a search strategy. At first relevant databases were selected to support a good search. The selected databases are listed in Table 1. The second step was to create a search string to search the databases.

To create a search string we started with basic terms such as "package recommendation" as keywords. By informal searches, we found more terms such as "bundle recommendation" and "clustering recommendation". We then used synonyms for these keywords to come up with the final search string.

"package recommendation" OR "package recommender systems" OR "package recommender system" OR "bundle recommendation" OR "recommending packages" OR "recommending package" OR "clustering recommendation" OR "clustering-based recommendation"

Table 2: Inclusion of papers

Database	Search string	1st inclusion	Snowballing		2nd inclusion	Total
			Backward	Forward		
ACM	24	2	0	0	0	2
IEEE Xplore	20	0	0	0	0	0
ScienceDirect	7	0	0	0	0	0
Scopus	90	28	0	0	0	20
General	0	0	412	400	74	74
Total	141	30	412	400	74	104

The search with the search string in the different databases is done by the first author. A list was made of all the articles and the double articles were removed. This resulted in 141 articles as can be seen in Table 2. With the inclusion criteria, from Section 2.3, the first inclusion round was performed. The first author reviewed all the articles based on title and abstract and marked all papers with the inclusion criteria it did or did not match. 30 of the 141 articles were included. During the inclusion, a new criterion was added called 'accessible'. Not every article could be accessed, so inaccessible articles were excluded from the research. The 30 included articles were used to perform the snowballing method. The snowballing method is a method to find new articles based on the current articles by using their references (backward snowballing) or cited by articles (forward snowballing). In this research both methods were applied. The backward snowballing method was applied 1 round, so the references of the 30 included articles were added to a list. The forward snowballing technique was applied until no new articles were found. After the backward- and forward snowballing methods were applied and the double articles were removed, 412 new articles remained from backward snowballing and 400 articles remained from forward snowballing. For the second inclusion round the methods from the first inclusion round were applied on the articles obtained by snowballing. This resulted in 74 included articles, which brings the total to 104 included articles for the review.

2.3 Inclusion and exclusion criteria

To select the studies suitable for this research, inclusion and exclusion criteria are defined. The inclusion criteria mentioned in the search strategy are defined below. The inclusion criteria are split into two categories. The first category are inclusion criteria that should all be met by an article.

IC1 Publication date of an article should be between 1999 and 2018.

IC2 Language should be English.

IC3 An article should be accessible.

IC4 The article should be peer-reviewed.

IC5 The article should be a journal, conference or workshop paper.

IC6 The focus of the article is on package recommendation as defined in the introduction. If this is retrievable from the title or abstract.

For the second category, every article should at least meet one of the criteria.

IC7 A method is proposed to solve a package recommendation problem.

IC8 An evaluation method is used to evaluate a package recommender system.

IC9 Evaluation metrics are used or compared for the evaluation of package recommender systems.

The exclusion criteria are used after the inclusion criteria are applied and have the function to exclude articles based on certain criteria. If one of these criteria is met, an article will be removed from the list.

EC1 The article is not about package recommendation as defined in the introduction.

EC2 An article is similar to another article and will be removed based on a quality assessment between the two articles.

The criteria were defined before the research took place to avoid bias. However, not all criteria could be foreseen, so IC3 was added during the research.

2.4 Data extraction and synthesis

The guidelines of Kitchenham and Charters [13] have been used for the data extraction. The data extraction is performed by the first and third authors. Both researchers had their own set of articles to review, but there was some overlap to check if data extraction was done in a similar way by both researchers. The disagreements or doubts about articles were discussed by the two researchers and solved by consensus. Checks were done on the data extracted to ensure both researchers agreed on the conclusions drawn. The researchers used an Excel spreadsheet to extract data, which was prepared before the research. The spreadsheet contained several columns, and some columns were added during the research because of new insights. For instance, for the package recommendation methods' column, phases were identified which resulted in new columns for each phase. Another example is the constraints which were added as a column. This resulted in the following data which has been extracted: review date, title, authors, year of publication, reference, database, package recommendation methods used per phase, constraints, domain, evaluation methods used, evaluation metrics used, research method, data set and size of the data set.

During the data extraction process the exclusion criteria were applied. Also there were a few cases where several articles were written about one research. Through the years the research was expanded, but the core remained the same.

Table 3: Article references for articles selected in the systematic literature review

Articles
P1 [14], P2 [15], P3 [16], P4 [17], P5 [18], P6 [19], P7 [20], P8 [21], P9 [22], P10 [23], P11 [24], P12 [25], P13 [26], P14 [27], P15 [28], P16 [29], P17 [30], P18 [31], P19 [32], P20 [33], P21 [34], P22 [35], P23 [36], P24 [37], P25 [38], P26 [39], P27 [40], P28 [41], P29 [42], P30 [43], P31 [44], P32 [45], P33 [46], P34 [47], P35 [48], P36 [49], P37 [50], P38 [51], P39 [52], P40 [53], P41 [54], P42 [55], P43 [56], P44 [57], P45 [58], P46 [59], P47 [60], P48 [61], P49 [62], P50 [63], P51 [64], P52 [65], P53 [66], P54 [67], P55 [68], P56 [69], P57 [70], P58 [71], P59 [72], P60 [73], P61 [74], P62 [75], P63 [76], P64 [77], P65 [78], P66 [79], P67 [80], P68 [81], P69 [82], P70 [83], P71 [84], P72 [85], P73 [86], P74 [87], P75 [88], P76 [89], P77 [90], P78 [91], P79 [92], P80 [93]

In these cases, the data from the articles was extracted, but were processed as if resulting from one article. Finally, some articles were excluded during the extraction process. A good example is P80 (table 3), which was excluded during the data synthesis process. During the analysis it became clear that no useful data could be extracted from that article. The exclusion and combining of articles during the data extraction and synthesis stage resulted in a total of 79 articles that have been used for this literature review. In Table 3 all the 79 included articles plus article P80 which was excluded during the data extraction, are mentioned. The P-numbers will be used during the research and behind each P-number is a number to indicate the place of the article in the reference list.

3 Results systematic literature review

3.1 Domains

As shown in Table 4, PRS have been used in several domains, though substantially more in the travel domain than other domains. As discussed in [94], recommendation domains have different characteristics.

In economics, a distinction is made between experience goods (which consumers learn about through experience) and search good (for which direct experience is not needed) [95], and between sensory and non-sensory products [96]. Based on this, Tintarev and Masthoff [94] distinguish between items which are easy to evaluate objectively (such as light-bulbs and cameras) and those which require an experiential and subjective judgement (such as holidays and music). Almost all domains used for PRS are subjective ones.

In economics, cost is also seen as an important characteristic of a domain, with this not only including purchase price, but also the time and effort involved in the purchase/consumption and the psychological, physical, functional and social risk [97, 98, 96]. Tintarev and Masthoff [94] distinguish between high and low investment recommendation domains. Travel tends to be a high investment domain, so most papers are about package recommendation in high investment domains, though there is also work on lower investment domains (such as movies, books, cloths, and food). Additionally, what is a low investment domain when recommending a single item (such as movies and books) becomes a large investment when deciding on a set of movies to see or books to read, as this can still involve a large time investment to consume the package and more costs as paying for multiple items. In a higher investment domain, a user may take longer to decide, so it is likely that users will take longer to decide on packages than on single items, which increases the users' information needs and requirements for explanations in PRS.

Domain characteristics matter as they may impact the importance of different recommendation quality metrics. For example, a PRS can be incorrect in multiple ways: it can overestimate how much a user may like a particular package, or it can underestimate how much a user may like a package. The first can lead to recommendations the user may not end up liking. The second can lead to the user missing out on packages they may have liked. Research by [99] showed that in general users considered overestimation as less helpful than underestimation, but that overestimation was particularly problematic in high investment domains. They also found that users tended to be more forgiving about over- and underestimation in subjective domains. So, domain characteristics may impact the best method for combining individual items into packages.

Domain characteristics also influence the kind of constraints that are used in the package creation process. For example, in the travel domain, normally items are suggested which are geographically close to each other. For example, bundling "the Great Wall" in China with the "London Tower Bridge" in England would not be a great idea. But coordinating the "London Tower Bridge" with "Buckingham Pallace", "Big Ben" and "the British Museum" can be considered a good bundle since they are located in the same city. In the travel domain, the location characteristic prunes many items as package candidates.

3.2 Package Type

Table 5 shows the package types used in the systematic review articles. We distinguish between PRS that serve recommended packages as a sequence of items or as a complementary set of items. In a sequence of items, the items are presented in an ordered list. When the user consumes a recommended package, the user needs to follow the items in the order in which they are given in the list. For example, in the music domain a recommended playlist contains a sequence of songs. In the travel domain, a recommended sightseeing tour package contains a sequence of attractions. In contrast, in a complementary set of items, items are provided as an unordered list. For example, in the grocery domain, a PRS can bundle tea, coffee, and sugar in a package.

Combining the data in Tables 4 and 5, it is clear that in the systematic review articles sequences were only used in the travel and education domains. For some domains, sequences are not really an option. For example, it is hard to envisage

Table 4: Package recommendation domains in systematic review articles

Article nr	Domain	# articles
P22, P38-P77, P79	Travel	42
P10-P14	Education	5
P5-P7, P17	E-commerce	4
P20, P25, P27	Movies	3
P1, P24	Books	2
P2, P3	Clothing	2
P21, P23	Travel, Movies	2
P26, P28	Travel, Movies, Books	2
P36, P78	Task assignment	2
P34, P37	(Sport) team selection	2
P4	Cosmetics	1
P8	E-commerce, Supermarket	1
P9	E-commerce, Furniture	1
P15	Food	1
P16	Gaming	1
P29	Movies, Books, Electronics, Clothing	1
P31	Travel, Restaurants	1
P32	Search engine results	1
P33	Software doc. architecture	1
P35	Supermarket	1
P18, P19, P30	Unspecified	3

Table 5: Package types in systematic review articles

Article nr	**Package type**	**# articles**
P1-P10, P15-P37, P39, P44, P55, P56, P67, P75, P78	Complementary	40
P11, P13, P14, P38, P40-P43, P45-P54, P57-P66, P68-P74, P76, P77, P79	Sequence	38
P12	Unclear	1

a sequence of furniture items, cosmetics, or electronics. For some, it may be possible to do sequences, but it is rather far-fetched: for example, a sequence of cloths to wear over a week. For some, sequences are a possibility, such as movies to watch during a film festival, or books to read over a longer period that are sequentially thematically linked, but the commercial market may be small. In the education domain, there are two kinds of package recommendations: one that recommends a learning path (learning materials to consume in a certain sequence to reach a learning goal) and one that recommends a set of learning materials that are complementary to each other to reach a set of goals. In the travel domain, the vast majority of work is on sequences: typically points of interests are combined into an itinerary. The exceptions are cases where for example a package contains a flight and a hotel. We note that unexpectedly music playlists are not included here, whilst there has been research on playlist recommendation (e.g., [100, 101, 102]). In fact, music is absent as a domain for package recommendation in Table 4. This seems an artifact of the search terms that were used in the systematic review, with people in music recommendation having used the domain specific term "play list recommendation" (instead of using more generic terms such as package, bundle or cluster).

3.3 Package Consumer

The PRS can recommend a package to different types of package consumers. It the review, we found two types: individual users and a group of users. When the PRS recommends a package to a user, the PRS needs to consider the user's preferences. This situation is similar to that of traditional recommender systems, which need to gather and analyze a user's preferences from different inputs, for example, intrinsic and extrinsic input. When the PRS recommends a package to a group of users, it not only needs to consider each member's preferences but also needs to aggregate the members' preferences to provide a solution that is best of the group. In addition, there are situations where PRS are able to recommend to both types of users (individuals and groups). The articles of the systematic review were almost all about package recommendation to individuals, with the exception of P18, P23, P30, P43, and P48 which dealt with package recommendation to groups. The latter were all in the travel domain (or unspecified in terms of domain), which is remarkable as there seems an opportunity for package recommendation to groups also in other domains such as food.

3.4 Recommendation Input

As shown in Table 6, PRS differ in the input they use to produce recommendations. Just like traditional individual item recommender systems, PRS can use implicit and explicit input from users. Implicit input is provided unconsciously (for example, item clicks or time spent looking at an item), whilst explicit input is provided consciously (for example, in the form of ratings on a 1 to 5 scale). Contrary to traditional recommender systems which use input regarding individual items, PRS can also use input regarding item packages, such as package ratings or time spent looking at a package.

Table 6: Input types in systematic review articles

Article nr	**Input type**	**# articles**
P2, P24, P25, P27, P49, P50	Explicit	6
P7, P14, P16, P17, P29, P32, P35, P48, P65, P79	Implicit	10
P1, P3, P10, P12, P15, P18-23, P30, P38-40, P45, P47, P51-55, P74, P76	Explicit + Features	24
P4-6, P8, P9, P13, P26, P31, P34, P36, P41, P42, P44, P46, P57-64, P66-71, P73, P75	Implicit + Features	30
P28, P43, P56, P72, P77, P78	Explicit + Implicit + Features	6
P11, P37	Features only	2
P33	Unspecified	1

As seen in Table 6, both implicit and explicit input are used in PRS, with implicit input being used more often. Combining the data from Tables 4 and 6, this seems an effect of domain, with implicit feedback being used a lot in the travel domain (in 30 out of 47, i.e. 64% of cases). For example, a PRS in the travel domain can collect implicit input in the form of the previous journey of a user (containing a travel package or single destination) or the global positioning systems (GPS) coordinates of their current position. In some cases, such input helps the PRS to discard irrelevant items to be recommended. For example, in the travel domain when using GPS coordinates, the PRS discards attractions which are too far away from the user's current position. PRS focused solely on the movies domain all used explicit input (typically ratings from the MovieLens dataset). There were only 6 PRS articles that combined both implicit and explicit input, so more research could be done on this combination. There was

Table 7: Only individual vs also package input

Input type	Input data	Articles	#
Explicit	Individual	P1, P18-P24, P27, P39, P40, P45, P47, P49, P50, P52, P53, P55, P76	19
	Package	P2, P3, P10, P12, P15, P25, P54	7
	Unspecified	P30, P38, P51, P74	4
Implicit	Individual	P6, P26, P31, P32, P34, P36, P66	7
	Package	P4, P5, P7-P9, P13, P14, P16, P17, P29, P35, P41, P42, P44, P46, P48, P57-P65, P67-P71, P73, P75, P79	33
Explicit + Implicit	Individual	P28, P56, P77, P78	4
	Package	P43, P72	2
Features only		P11, P37	2
Unspecified		P33	1

no clear preference for input type in PRS for education, with both implicit and explicit input being used.

Another input type which is often used to recommend packages is the items' features. For example, in the travel domain, an attraction can have many features, such as a description, attraction type (e.g. museum), opening hours, season, cost, and location. Whilst this input type is also occasionally used in traditional recommender systems as a basis for content-based filtering, in PRS it plays an additional role, in that it forms the basis for constraints (e.g. rules on which colours can be combined in clothing outfits or which locations are good to combine in a tourist route). As seen in Table 6, features are used in most PRS articles surveyed (79% of articles that specified input type). This is even more pronounced in the travel domain (37 out of 42, i.e. 88%, of articles solely on travel). Combining the data from Tables 5 and 6, features are used more in PRS recommending packages as sequences than as complementary sets (84% compared to 73%).

Explicit and implicit data can concern individual items or item packages. Table 7 shows which articles used input data only for individual items, and which also used data for packages. For implicit data, we assumed that travel (point of interest) check-in data provides package data (as one can see sequences in check-ins) and that items bought together also can be regarded as a package. Only some articles (27%) that used only explicit data used package data compared to most articles (83%) that used only implicit data.

3.5 PRS Phases and Techniques

Researchers have solved the PRS problem using different techniques. In general, they used phases to recommend packages. We have identified three phases and classified articles by the techniques they use in each phase:

- The *model learning* phase where the PRS learns several aspects required to produce recommendations such as a user's preferences.
- The *package creation* phase where the PRS mixes and matches items into packages and collects these packages into a package candidates list.
- The *package selection* phase where the packages to be recommended are selected from the package candidate list. In this phase, the researcher evaluates the package candidates using techniques such as top-N.

Even though we classified the PRS articles based on the three phases, some articles only used one or two. For example, some researchers used pre-defined packages or random coordination for the second phase, and focused on understanding the user's preferences and also selecting the best packages to recommend to a user. Tables 8-10 show which techniques were used for each phase in the survey articles.

3.5.1 Model Learning Phase

In the model learning phase, the PRS uses the input as described in Section 3.4 to obtain knowledge about the user's preferences and also item characteristics. As shown in Table 8, several techniques are used, in particular clustering, collaborative filtering (CF), user preference modelling, item relationship modelling, and topic modelling.

The algorithm used is also influenced by the PRS input. PRS which only use explicit input such as ratings, tend to use CF methods such as matrix factorization (MF), item-based CF, memory based CF and so on. For example, Wibowo et al. [15, 16], used user-item-rating and user-package-rating matrices of clothes as input and used MF to obtain users' and items' latent factors. Combined with an aggregation function (such as minimum, maximum, or harmonic mean), they then used these latent factors to approximate the package rating.

PRS that used unstructured data on items or packages, such as text, often used topic modelling techniques such as Latent Dirichlet Allocation (LDA). LDA parses the text descriptions and automatically extracts specific topics which relate to items or packages. For example, Xiong et al. [88] used travel website information as input and used LDA-based topic analysis to automatically extract the topics. They then matched the extracted topics with the users' interests. Zhang et al.[66] used LDA to classify the POI categories from content descriptions of each POI. They then determined whether a proposed route (containing a sequence of POIs) is feasible in the sense of containing at least a certain number of POI categories.

Table 8: PRS Phase 1: Model Learning

Technique		**Articles**	#
Clustering		P4, P13, P30, P35, P43, P71, P79	7
Collaborative Filtering	Matrix factorization (incl. BPR)	P2, P3, P8, P14, P16, P25, P27, P46, P47, P53, P61	11
	Item-based CF	P1, P8, P19, P20, P23, P39	6
	Feature-centric CF	P47	1
	Memory-based CF	P21	1
	Hybrids (e.g. using Gausians)	P54	1
	Other	P28, P58, P59	3
	Unspecified	P10, P49, P57, P67, P70	5
Other User Preference Modeling	Check-ins	P42, P43, P68, P69	4
	Correlated cross-occurrence	P49	1
	Multi-attribute utility theory	P49	1
	Content-based filtering	P34, P60	2
	Clustering optimisation diversity	P18	1
Item Relationship Modeling	Markov Chain, Probability model	P7, P22, P29	3
	Pattern mining (e.g. Apriori)	P43, P48	2
	Ontology	P38, P77	2
Topic Modeling	LDA	P09, P53, P65, P75	4
	Gibbs sampling	P65, P67	2
	Bernoulli	P65	1
	Bayesian	P44, P67	2
	Restricted Bolzmann machine	P15	1
	TF-IDF	P60	1
	Word embeddings (Course2Vec, CBOW)	P14, P61	2
	Unspecified	P64, P73	2
No model learning		P6, P12, P17, P32, P37, P40, P45, P50-P52, P63, P76	12
Unspecified		P5, P11, P24, P26, P31, P33, P36, P41, P55, P56, P62, P66, P74, P78	14

Some PRS used item relationship modelling (such as Markov chain, probability model, Apriori and ontology) to model the relation between items. For example, Yu et al.[42] used a Markov chain-based approach to model the relations of particular products with regards to the users' sequential behaviour. Mikhailov et al.[90] used an ontology to model a.o. the similarity between attractions. With the available survey data no real relationship between the input type and item relationship modelling can be found.

Some PRS used domain specific methods, for example, some travel PRS used check-in data to deduct user preferences. The more a POI is visited by a certain user, the more preferred a certain POI is according to this method. Then the categories of the most visited POIs are determined to calculate the preferences of a user for the categories in the system.

In some cases clustering was used to learn the user model. Clustering is used to find similar groups of items or similar groups of users. For instance, in P4 clusters of items are made to determine the preferences of a user for a certain group of items. Clustering users is used to find users with similar preferences or qualities. For instance in P13, students with similar results were clustered to find what courses fit best with a certain group of students. Clustering is not only applied for individual recommendations, but also for group recommendations. For example, in P30 user clustering is used to determine the preferences of a group of users for a group PRS. Except for P30 all other articles that used clustering made use of implicit input data.

Several PRS did not use model learning, for example, because their package construction did not require user preferences, but only used known item features and constraints. For example, in P37 the goal is to recommend a team consisting of complementary team members. This is based on members' skills and social fit in a team, so model learning of preferences is not needed. Model learning is also not needed when users explicitly enters their preferences (e.g. as a search query) and only that information is used to make recommendations, so no history or data from other users is used.

3.5.2 Package Creation Phase

As mentioned above, the package creation phase is used to generate a package candidates' list. In this process, items are combined with other items into a package.

As can be seen in Table 9, many researchers have regarded this as a knapsack problem, where the solution is a combinatorial optimization in which a collection of items is selected which maximizes the value or minimizes the cost, whilst remaining within certain constraints. For example in the travel domain, the knapsack problem can be defined as how to include as many POIs as possible in an itinerary, whilst remaining within the user's budget (money and/or time). Several knapsack algorithms were implemented, using for example dynamic programming, search algorithms (e.g., greedy search), and evolutionary algorithms (such as Ant Colony Optimization [75])[1].

[1] Ant Colony Optimization iteratively randomized the items involved in a package and eval-

Some researchers have regarded package creation as a clustering problem, where the package combination is obtained from a common value in a group of data. For example, in the travel domain, POIs can be clustered based on their attributes (such as geographical location, POIs type and so on). In the survey articles, several clustering methods have been applied, but two methods are used the most: nearest neighbours and fuzzy clustering. Nearest neighbours (k-NN) is a clustering method which finds the k most similar items based on a target item. Sometimes k-NN is used by other methods. For instance, the papers which make use of the BOBO algorithm use k-NN to create packages around BOBO's pivots (target items). The other commonly used method fuzzy clustering is very similar to k-means. Just like in k-means, k clusters of data points are created with fuzzy clustering. However, where a data point with k-means can only belong to one cluster, in fuzzy clustering a data point can belong to multiple clusters. In the survey articles, clustering is used equally often with implicit and explicit data. The input data is in 75% individual item data, and also in 75% of the articles clustering is used to produce complementary packages (rather than sequences). So, there has been most focus on clustering when producing complementary packages based on individual item data.

When a user likes two items individually, this does not necessarily mean that the user would like the combination of the two. For example, somebody may like a red pair of trousers and an orange shirt, but may not want to wear them together. Therefore, when combining items into packages, most papers used constraints. For example, in the clothing domain, constraints have specified which colours, patterns, and formality to combine. Constraints can be manually constructed or learned. In the travel domain, a cost function such as travel time and distance is often used as a basis of constraints (e.g., not to select two items that are too far from each other).

Some papers used predefined packages, or assigned items randomly to packages to create a package list. Others created all possible packages, based on a package model, which specifies the frequency of item types in a package or other constraints. For example, in a clothes PRS, if a package can contain a shirt and a pair of trousers, they would generate all possible combinations of a shirt and pair of trousers. Similarly, in an educational PRS, all possible course sequences can be generated, taking course pre-requisites, maximum number of courses to take in each term, and which course runs in which semester into account (P13).

3.5.3 Package Selection Phase

Package selection is the last phase of package recommendation. In this phase, the PRS selects a number of packages from a package candidates' list obtained in the previous phase. As can be seen in Table 10, the most common approach for recommending packages is top-N, whereby the N best packages (as estimated by the PRS) are recommended to the user in a ranked list. Top-1, a sub case of Top-N, where only the best package is recommended is also very popular,

uated its estimated value (such as travel distance), whilst improving the solutions randomly in several iterations

Table 9: PRS Phase 2: Package Creation

Technique		Articles	#
Knapsack	Greedy (e.g. Ford-Fulkerson)	P9, P10, P11, P19, P21, P28, P31, P32, P33, P39, P40, P48, P52, P54, P63, P70, P76	16
	Random walk	P5, P12	2
	Heuristic	P31, P37, P45, P47, P79	5
	Branch and Bound	P7, P58	2
	Brute force (e.g. Breadth first, Depth first)	P33, P43, P47, P53, P54, P66	6
	Dynamic programming (e.g. Dijkstra shortest path, Floyd–Warshall)	P41, P42, P50-P52, P58, P55	7
	Integer linear programming	P10, P20, P24, P61	3
	Ant colony algorithm	P62	1
	Recurrent Neural Network	P68, P69	2
	Zero-suppressed Decision Diagram (ZDD)	P17	1
Clustering	K-means	P27, P73	2
	Nearest neighbours (incl. case-based reasoning)	P1, P19, P31, P32, P38, P39, P73	6
	Hierarchical	P71	1
	Fuzzy	P26, P36, P73, P78	4
Expanding from pivot items (e.g. Bundles One-By-One)		P4, P31, P32, P39	4
Similarity	Jaccard	P32, P56	2
	Association rules	P35	1
	Apriori	P59	1
Predefined		P25, P44, P75	3
Random selection		P2, P3	2
All possible		P6, P8, P13, P15, P18, P22, P23, P65	4
Constraints		P1, P3, P5, P6, P9-P11, P13, P17-P23, P26, P28, P30-P43, P45, P47, P48, P50-P55, P58-P62, P67-P79	58
Unspecified		P14, P16, P29, P46, P49, P57, P64	7

Table 10: PRS Phase 3: Package Selection

Technique	**Articles**	**#**
Top-N	P1, P3, P5, P6, P8, P9, P11, P12, P14, P15, P18-P23, P26, P28, P30-P32, P34-P36, P38, P39, P42-P44, P48, P49, P54, P56, P58-P60, P64-P69, P71, P72, P75	45
Top-1	P2, P4, P7, P10, P13, P16, P17, P25, P33, P40, P41, P45, P47, P50, P51-P53, P55, P61-P63, P73, P76, P77, P79	25
Multiple packages unranked	P24, P37, P70, P74, P78	5
Unspecified	P27, P29, P46, P57	4

and has been used more often than in individual item recommendation. In an individual item recommender, a popular alternative to Top-N is to show all items (not ordered) with a rating system (e.g. stars) to indicate the recommender's estimated user preferences for each item. This is not really used in PRS, simply because the number of packages tends to be far too large. Some PRS show multiple packages that have been deemed suitable without providing a ranking. It is likely that the number of packages shown to the user depends also on the complexity of packages. For example, when a package contains many items, it is likely that the number of packages shown is smaller. We did not find a domain effect yet: for example, the percentage of travel PRS that used Top-1 is 34% and very similar to the 32% of all PRS that used Top-1[2]. More investigation is needed into the effect of package complexity and domain on the way recommendations are and should be presented.

3.6 Evaluation

Table 11 shows how PRS have been evaluated[3].

Computational evaluations The vast majority of articles surveyed did not evaluate the PRS through a user study, but instead used an off-line computational evaluation method. Most (64%) papers that used a computational evaluation method measured accuracy. This is in line with many studies of traditional recommender systems, where the emphasis has been on prediction accuracy and top-N recommendation accuracy.

[2]Whilst many travel PRS produce more complicated packages containing multiple points of interests, some travel PRS just combine flights and hotels, so the domain categories we used is not necessarily a reflection of package complexity

[3]Some articles used more than one form of evaluation, so the number of evaluation forms can be higher than the total number of articles

Accuracy. Prediction accuracy normally measures the extent to which the predicted item ratings correspond with the actual item ratings[4] [103]. Standard measures include Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE), whilst one paper (P8) also used the less often used Weighted Average Percentage Error (WAPE). Less frequently, prediction accuracy is based on the accuracy of the ranking, mostly measured by using the correlation between the predicted ranked list and the actual ranked list, for example using Kendall's rank correlation coefficient (Kendalls' tau) or Fagin's intersection metric (P22).

Prediction accuracy treats errors in predictions for good and bad items equally, whilst recommender systems tend to only show a limited number of items to a user with good rating predictions. Therefore, recommender systems are also often evaluated on the relevance of the recommendations in a ranking situation using a top-N recommendation task [104][5]. Standard measures for this are precision@N and recall@N, measures that combine precision and recall such as Area Under Curve (AUC) and F_1, and measures that reflect whether the user selects/consumes the package recommended. Similarly to normal recommender systems, the latter can be measured through the hit-rate, for example measuring the proportion of recommended packages that are clicked on. However, whilst in a normal recommender system, a recommendation is either used or not, in a PRS the situation is more complex. Users could for example use part of a recommendation. Therefore, several surveyed papers used the Jaccard similarity metric, which measures the similarity between two sets (the set of items in the recommended package compared to the set of items actually selected). When the package recommended is a sequence, one can also look at the longest common subsequence, as the Jaccard similarity metric does not take order into account.

Additionally, some accuracy measures of relevance take the ranking positions into account such as normalized Discounted Cumulative Gain (nDCG), Average Precision (AP), Mean Average Precision (MAP), and Weighted Average Precision (WAP). Table 11 shows that accuracy measures based on relevance are most popular in PRS evaluations.

Many papers used a combination of different accuracy metrics. For example, papers that measured the accuracy of predicted item rankings, often also measured the relevance of the recommendations using nDCG. Similarly, papers that measured the relevance of recommendations using nDCG also reported Precision@N (with the exception of P21).

Measuring accuracy requires a gold standard: actual ratings[6] (and/or rankings) for recommended packages. As was shown in Table 7, many PRS did not use package ratings. In such cases, there is no real gold standard. Indeed accuracy was measured more often in papers that used package ratings: 71% of

[4]Where actual item ratings are not necessarily given explicitly, but can be inferred through implicit input.

[5]Metrics such as Fagin's intersection metric are in between these two categories: they consider rankings and can be applied to top N

[6]Explicitly or implicitly acquired.

papers that used package ratings compared to 40% of papers that used only individual ratings. Remarkably, this still means that 12 papers measured accuracy despite only having individual item data. In such cases, typically a package was deemed as good as the average rating of the package items. Another issue with the computational measurement of accuracy is that from the 42 articles that used package ratings, 33 articles only used implicit package ratings. Implicit data often struggles to make a distinction between items the user has seen and disliked, and items the user has never seen (the latter will be very frequent in PRS).

More recently, researchers have been arguing against only using accuracy measures of recommender systems, and advocating to use also measures such as coverage, confidence, diversity, novelty, serendipity, utility, and scalability [105].

Scalability is the extent to which the PRS can deal with larger data sets (in terms of processing power required and speed). In the papers reviewed, after accuracy, scalability was evaluated the most, with scalability being measured in 42% of the papers which had computational evaluations. A high proportion (75%) of papers that evaluated scalability did not evaluate accuracy.

Utility is the extent to which a recommendation is useful, so its value. Two kinds of value can be distinguished: value to the package consumer and value to the package provider. In the surveyed articles, the ways to calculate package value were often domain specific. For example, a book PRS used revenue gain to measure value for the package provider. A travel PRS used travel time as one way to measure value for the package consumer (with a higher travel time meaning lower value). An educational PRS used grade point average and graduation time, which can measure value for both package provider and consumer. Utility was measured in 27% of the papers which had computational evaluations. In about half these papers, it replaced the accuracy measurement.

Coverage is the percentage of users for which the system can provide recommendations and/or the portion of items that can be recommended. Standard measures include the Gini index and Shannon entropy. Sometimes the performance of the system can be measured also specifically for new ('cold') items or users (who have fewer than a certain number of ratings). Coverage was measured in only 12% of the papers which had computational evaluations.

Diversity is traditionally the extent to which recommendations are dissimilar from each other. In PRS, two types of diversity are used, intra-package diversity and inter-package diversity. Intra-package diversity is the diversity of items within a package. Inter-package diversity is the diversity between recommended packages. Out of 8 articles, only P31 used inter-package diversity. Intra-package diversity is calculated as 1 minus the average similarity between any two items in the package, which is typically calculated based on the item features. Diversity was measured in only 12% of the papers which had computational evaluations; all but one of these used features.

Table 11: Evaluation methods and metrics in systematic review articles

Evaluation	Metrics		Articles	#
Computation	Accuracy: Rating	RMSE, MSE, MAE, WAPE	P2, P8, P15, P25, P27, P47, P53, P57	8
	Accuracy: Ranking	Kendall's Tau, Fagin's intersection metric, Degree of agreement	P22, P44, P67, P73	4
	Accuracy: Relevance	Precision@N	P1, P4, P7, P8, P12, P20, P27, P28, P29, P30, P32, P39, P46, P59, P61, P65, P67, P68, P69, P71	20
		Recall@N	P3, P4, P8, P14, P18, P27, P29, P30, P46, P61, P65, P67, P71	13
		AUC	P16, P29	2
		F_1	P1, P27, P39, P61, P68, P69, P71	7
		Hit-rate, Jaccard similarity, Longest common subsequence	P9, P42, P56, P63, P65, P70, P72	7
	Accuracy: Relevance + Ranking	nDCG	P4, P21, P22, P28, P32, P44, P59, P65, P68, P69, P73	11
		AP, MAP, WAP	P10, P12, P18, P22, P30, P60, P64	7
	Scalability	Execution time, Processing time	P5, P6, P10, P11, P17, P20, P23, P24, P26, P28, P31, P33, P34, P40, P41, P43, P45, P47, P48, P53, P54, P58, P59, P62, P63, P76, P78, P79	28
		Memory storage	P59	1
	Utility	Package value/cost, Order size	P7, P8, P13, P20, P21, P23, P24, P31, P43, P45, P47, P53, P62, P68, P69, P71, P72, P79	18
	Coverage		P5, P15, P16, P24, P31, P37, P41, P56	8
	Diversity	Inter-, Intra package diversity	P1, P28, P31, P32, P39, P68, P69, P70	8
	Cohesion		P23, P32, P37	3
	Perplexity		P67, P75	2
	Novelty		P44	1
User study	Satisfaction	Perceived usefulness, usability	P3, P23, P40, P43-P45, P48, P49, P50, P51, P63, P66, P67, P76	14
		Retention	P78	1
	Performance	Throughput, Execution time	P36, P78	2
	Accuracy: Relevance + Ranking	MRR	P49	1
		Acceptance rate, Click-to-open, Conversion-to-open	P7, P9, P36, P78	4
Expert study			P33, P52, P55, P76	4
None			P19, P35, P38, P74, P77	5

Cohesion is the extent to which items within a package belong together in terms of similarity. This is typically the opposite of intra-package diversity. Only 3 articles explicitly considered cohesion. The trade off between cohesion and diversity seems domain dependent. In some domains (e.g. team recommendation) cohesion may well be more important, whilst in other domains diversity may be more important.

Confidence is the system's trust in its own predictions. We did not find any papers explicitly mentioning confidence, but did find two papers using the related concept Perplexity, which is a measure of uncertainty.

Novelty is the extent to which users were unaware of recommended items[7]. This was only evaluated in one paper (P44). To be able to measure novelty, one needs to know which packages a user will consume in future, so typically a dataset which includes time, so that recommendations are based on data up till a certain point of time[8]. Novelty can then be measured by looking whether a recommended item was already known to the user (so consumed earlier) or not. P44 used an implicit dataset for this.

In principal, it is possible to evaluate aspects such as accuracy, coverage, diversity and scalability also during a user study, in which study participants interact with a PRS. However, most papers surveyed used these metrics in an off-line setting, mostly using an existing data set as a basis for both the system and the evaluation (mostly using n-fold cross-validation, whereby a part of the data is used to inform the system and a part to evaluate it, and this is repeated n times)[9][10]. This is why most papers using these metrics are presented under the computation evaluation method in the table.

User studies and expert evaluations Only 23% of articles contained a user study: 13 articles combined a computational evaluation with a user study, and 5 contained only a user study. As shown in Table 11, user studies mainly focused on user satisfaction with the package recommendations (including perceived usefulness), though there was also some work on accuracy and performance. Only 3 papers contained an expert evaluation; one of these combined an expert evaluation with a user study. The expert evaluation all focused on satisfaction.

Satisfaction is how pleased users (or experts) are with the PRS and its recommendations. Satisfaction was normally measured through surveys which ask

[7] Another measurement is serendipity: the extent to which successful recommendations are *surprising* to the user (e.g. a recommendation of a new book by their favourite author may be novel, but not surprising). Serendipity was not measured for PRS at all, neither in the computational evaluations nor user studies

[8] Splitting the data in the earlier time period into a training and test set as usual, whilst adding the later time period to the test set

[9] In some cases (e.g. P18), a user study is done to create a new dataset, but this user study is not used to directly measure the performance of the system, but rather to construct a dataset that is used again in an off-line setting

[10] In some cases (e.g. P25) one dataset is used to produce the PRS and another to evaluate the PRS

participants' opinions about recommendations (perceived usefulness, so related to the utility measurements in the computational evaluations). Sometimes the usability of the PRS was measured (e.g. P51, P66, P76). One paper (P78) considered user retention: so how long they would keep using the PRS. Another measure that has been mentioned for recommender systems is the users' **trust** in the system [105]. This can be measured by considering how many recommendations are followed or by asking users whether they find the recommendations reasonable. It is often hard to measure trust independently from user satisfaction, and as can be seen, the measurements taken in PRS for satisfaction are implicitly measuring trust as well.

Performance. To measure performance, the throughput or execution time was used. For instance, throughput was measured by counting the number of completed tasks per minute (P36, P78). Execution time is quite similar, but is focused on *one* task and how long it takes for that specific task to be selected and completed (P36).

Accuracy measurements in the user studies were all related to the relevance of recommendations, and either used the mean reciprocal ranking (MRR; a variant of MAP that was used in the computational evaluations), or the rate at which users accepted recommendations, clicked on them, or consumed them (so, similar to the hit-rate in the computational evaluations, but now based on the data of users who had actually used the PRS).

3.7 Conclusions

This section presented the results of a systematic review of PRS, which after applying inclusion and exclusion criteria looked at 79 articles in detail. This area of recommender systems' research is still relatively immature. We note the following challenges for PRS which require future work.

Need for more open data sets that contain package ratings. Many recommender systems' data sets have been released, but so far most are for the recommendation of individual items. Some researchers have creatively created package data sets from available data sets using several assumptions. For example, P39 created travel packages for users by using individual POI ratings. By combining these POI ratings and taking the POI popularity and intra package diversity into account a score for each package was calculated. The problem is that combining several items a user likes does not mean that the combination of those items is also appreciated by the user. Combining two pieces of clothes that somebody likes does not have to mean that they would like the combination of those items as an outfit.

A few researchers have produced their own package data sets through explicit user ratings in studies, but these data sets are still very limited in size and only available in some domains. For example, Wibowo et al. [15, 16] collected package recommendations in the clothes domain by asking participants to rate a "top" of clothes (such as a shirt) and a "bottom" of clothes (such as pants) individually and as a package. For another example, Sharma et al., P25 [38] collected ratings for sets of movies from active users of Movielens on movies each user had rated individually in the past. Researchers have often resorted to the use of implicit package data, but whilst this gives some insights in what users tend to consume together, and may therefore like, this is not necessarily indicative of the best possible combinations (as users may just not have been aware of other options) and does often make it hard to distinguish between packages which the user dislikes and packages which the users has just never noticed. Package data sets are not just important as a basis for recommendations, but are vital to get a reliable measurement of accuracy. Using the average of individual item ratings to produce a gold standard for the package rating is in many cases not right. For example, if a user adores a red pair of trousers and adores an orange shirt, this does not necessarily make this a great outfit. Similarly, a user may really like the British museum and the Victoria and Albert museum, but may be unlikely to combine them in a one day outing. Whilst other measures (such as cohesion and diversity) may contribute to the estimation of the goodness of a package recommendation, without actual user package ratings (or other ways of gather user opinions on packages), it is hard to reliably measure accuracy, or even to investigate the impact of diversity and cohesion on (perceived) accuracy. Therefore, for the field of PRS to progress, the creation of large open data sets that contain both individual and package ratings is crucial. Given the reliance of most PRS systems on features for the construction of packages, such data sets also need to contain item features.

Need for more sophisticated ways of dealing with data sparsity and package cold starts. Package rating matrices are even sparser than individual item rating matrices. There has been some initial research on ways to reduce matrix sparsity[106]. Additionally, whilst individual item recommender systems often suffer from user cold start problems (difficulty to recommend to new users who have not rated anything yet) and item cold start problems (difficulty to recommend new items which have not been rated by anybody yet), PRS have a package cold start problem: the difficulty of recommending a new package. Each new item that is added could potentially lead to a very large number of new packages that contain that item. More research is needed on how to deal with data sparsity in the package rating matrix and how to deal with the package cold start problem. Given it is unlikely that this problem can be fully solved for large scale real world systems, there is a need for research on accurate estimations of package ratings based on individual item ratings, (sparse) ratings of other packages, and item features. This research will need to be done in multiple domains, as this is certainly to a large extent domain dependent[11]. This research will require user studies to validate the estimation formulas.

Need for more efficient algorithms. Package recommendation is computationally complex, with many approaches to model learning (e.g. of package

[11] Domain types may be distinguished, which may share certain parts of this accuracy estimation, for example the importance of package cohesion

preferences) and package creation NP-hard. It is therefore important to produce more efficient algorithms, which make optimal use of heuristics (such as constraints) that are based on evidence-based insights on which item combinations go well together in a certain recommendation domain.

Need for more sophisticated metrics. Most evaluations used the same metrics as used for individual item recommender systems. There is a need for metrics specifically developed for PRS. For example, most papers used traditional accuracy metrics, whilst in a PRS, particularly when package size increases, users may decide to consume part of a package. This means that it is no longer solely a question of comparing packages consumed (as whole entities) to packages recommended, but that the content (the items in the set) of what is actually consumed needs to be compared against what has been recommended, and that in case of the package being a sequence, also the order of consumption of individual items needs to be considered. In such cases, traditional accuracy metrics (such as Precision@N which was most popular in the papers surveyed) no longer suffice. Only a few papers used Jaccard similarity and longest common sub-sequence to perform such a comparison that takes package content into account. Even those metrics will need more work, and will need adjusting to fully capture the complexity of PRS evaluation. For example, where in the sequence the longest common sub-sequence is (for example, at the start or the end) may matter for users' perceptions of whether the recommendation was followed and useful. If the longest common sub-sequence in a travel package was at the start, perhaps a user really liked the package, but ran out of time (or got lost). Similarly, more work is needed on diversity and cohesion metrics. Additionally, user studies could benefit from a reliable scale to measure appreciation with packages, so that users' opinions on package details (such as the start, finish, cohesion, diversity, serendipity) can be measured. Existing questionnaires are mainly focused on individual item recommendations, and to the best of our knowledge there is no validated scale for PRS.

Need for more comprehensive evaluations and user studies. Whilst most papers contained an evaluation, predominantly these were computational evaluations and evaluations of accuracy. Clearly, there is more to the goodness of a recommendation than accuracy, just as has been argued for single item recommender systems as discussed above. Consuming a package requires more investment by the user (in time and/or money) than consuming an individual item. Recommending packages is also more complicated than recommending individual items. Whilst most of the current studies use computational evaluations, this is not enough to understand this complex problem. Users need to be more involved in the evaluations by doing user studies. This could help to better understand how package recommendation works and what a good package is.

Need for more domains to be studied. The focus of package recommendation till this point is mainly on the travel domain (with each other domain only studied in a couple of papers). However, there are many other domains that could be broader studied. This could result in a deeper and broader understanding of the package recommendation field.

This review also underpins the decisions for the recipe week schedule recommendation system discussed in the remainder of this thesis. These will be decisions discussed in detail in section 5.

4 Food guidelines

To help people with a healthier lifestyle, there have to be guidelines to determine what is healthy and what not. Such guidelines can be used to determine what kind of food should be eaten and in which quantities. Because this research is focused on the IT part of the solution and I have only amateur knowledge about food, the choice was made to use already existing guidelines. In this section the guidelines used from Foodfirst Network will be explained.

4.1 Foodfirst Network

Foodfirst Network is company which has the goal to help people with a healthier lifestyle by giving advice on food, exercise, relaxation and sleep. This is done through their website where recipes, exercises and articles can be found by members of the platform. All recipes, exercises and articles are made based on guidelines.

The reason to use the guidelines of Foodfirst Network is because the expertise underpinning those guidelines. Experts from different kind of fields create guidelines for Foodfirst Network, based on scientific research. Furthermore, the company has the same goal as this thesis, namely to help people with an healthier lifestyle. Finally, Foodfirst Network is willing to cooperate in this research by exchanging knowledge and giving access to there customer database to create a data set.

4.2 Guidelines

The Foodfirst Network guidelines are all written down in a document called the "FFN bijbel" [12]. Foodfirst Network makes use of different types of guidelines. There are guidelines about single recipes, day schedules, week schedules and guidelines per profile. For this research only the guidelines for the different profiles will be used for the system. The individual recipes all meet the single recipes guidelines. For example, every lunch recipe should at least contain 100g of vegetables. The day- and week schedule guidelines are very general and can not be applied for users with certain restrictions such as allergies or a vegetarian diet. For example, the week guidelines state that a recipe week schedule should contain 1-2 times red meat. However, for a user which is vegetarian this is

[12] In English: FFN Bible

not appropriate. Additionally it is assumed that the day- and week schedule guidelines in general will be met by creating recipe week schedules.

The profile guidelines are split into two different types: (1) goal- and (2) characteristics guidelines. Goal guidelines are guidelines which help users to achieve certain goals, such as losing weight. Characteristics guidelines are guidelines for factors which have to be taken into account, such as allergies, pregnancy, etc. Users are able to combine certain goals and characteristics. However, there are some impossible combinations which will be mentioned later.

Below the different guidelines will be specified. This will include the guidelines for day- and week schedules, for later use in the evaluation. Guidelines that are not mentioned, because they are not relevant for the research, can be found in Appendix 1 where the "FFN bijbel" document is located.

4.2.1 Goal guidelines

Below four different goals are listed. There are more goals in the "FFN Bijbel, but these are the ones that impact the recipe recommendation. For instance, losing weight has impact on the recipes that are appropriate to recommend. Other goals that have not been chosen for this research are goals that do not impact recipe recommendation according to the guidelines used by Foodfirst Network. For instance, the goal to "sleep better". Foodfirst Network has other content to help users with these goals. The numbers behind the goals correspond to the numbers in Appendix 1.

- Lose weight (A1)
- Stay healthy (A2)
- Improved bowel movements (A7)
- Gain more muscle mass (A8)

Each goal has its own guidelines. These guidelines are made specific for the whole day and for each meal of the day.

Lose weight (A1)

- **Day:** $\leq$ 1700 kcal
- **Breakfast:** $\leq$ 400 kcal
- **Lunch:** $\leq$ 650 kcal
- **Dinner:** $\leq$ 650 kcal

Stay healthy (A2)
Stay healthy is the default goal. According to Foodfirst Network, this means that a user does want to remain at the same weight and just eat without any

restrictions. So all recipes on the platform are available [13].

Improved bowel movements (A7)

- **Day:** $\geq$ 24g fiber
- **Breakfast:** $\geq$ 6g fiber
- **Lunch:** $\geq$ 8g fiber
- **Dinner:** $\geq$ 10g fiber

Gain more muscle mass(A8)

- **Day:** $\geq$ 65g protein
- **Breakfast:** $\geq$ 15g protein
- **Lunch:** $\geq$ 25g protein
- **Dinner:** $\geq$ 25g protein

4.2.2 Characteristics guidelines

Below the characteristics guidelines can be found. There are 2 types of guidelines: (1) numeric guidelines and (2) Boolean guidelines. Numeric guidelines are guidelines which contain a number. For instance, all the goal guidelines are numeric guidelines. Boolean guidelines are guidelines which can only be yes or no. For instance, somebody has an allergy or not. Some of the guidelines that are mentioned in the "FFN bijbel" are not yet fully specified, for instance pregnancy. For this reason those are not taken into account for this research.

Diabetes type 2 (B3)

- **Day:** $\leq$ 75g carbohydrates
- **Breakfast:** $\leq$ 25g carbohydrates
- **Lunch:** $\leq$ 25g carbohydrates
- **Dinner:** $\leq$ 25g carbohydrates

Cardiovascular disease (B4)

- **Day:** $\leq$ 5g salt and $\geq$ 3000mg potassium
- **Breakfast:** $\leq$ 1g salt and $\geq$ 750mg potassium
- **Lunch:** $\leq$ 2g salt and $\geq$ 1000mg potassium

[13] This means if a user would only eat recipes of Foodfirst Network. It could be argued if no guidelines are sensible, because it could indicate that a user could eat as many calories as he or she wants, which is not the case.

- **Dinner:** $\leq$ 2g salt and $\geq$ 1250mg potassium

High blood pressure (B5)

- **Day:** $\leq$ 5g salt and $\geq$ 3000mg potassium
- **Breakfast:** $\leq$ 5g salt and $\geq$ 3000mg potassium
- **Lunch:** $\leq$ 5g salt and $\geq$ 3000mg potassium
- **Dinner:** $\leq$ 5g salt and $\geq$ 3000mg potassium

Malnutrition (B7)

- **Day:** $\geq$ 65g protein and $\geq$ 2000 kcal
- **Breakfast:** $\geq$ 15g protein and $\geq$ 500 kcal
- **Lunch:** $\geq$ 25g protein and $\geq$ 750 kcal
- **Dinner:** $\geq$ 25g protein and $\geq$ 750 kcal

I am a fanatic athlete (C1)

- **Day:** $\geq$ 65g protein and $\geq$ 2000 kcal
- **Breakfast:** $\geq$ 15g protein and $\geq$ 500 kcal
- **Lunch:** $\geq$ 25g protein and $\geq$ 750 kcal
- **Dinner:** $\geq$ 25g protein and $\geq$ 750 kcal

Surgery (C5)

- **Day:** $\geq$ 65g protein and $\geq$ 2000 kcal
- **Breakfast:** $\geq$ 15g protein and $\geq$ 500 kcal
- **Lunch:** $\geq$ 25g protein and $\geq$ 750 kcal
- **Dinner:** $\geq$ 25g protein and $\geq$ 750 kcal

Boolean guidelines:

Vegetarian (B1) [14]
All recipes without meat or fish.

Allergy (B2)
Below all allergies that are specified within the recipes of Foodfirst Network are mentioned.

- Cow's milk

[14] Vegan is not specified within Foodfirst Network. Still a decision has to be made whether Foodfirst Network will incorporate vegan guidelines or not.

- Gluten
- Raw tomato, paprika, carrot and red pepper
- Stone- and seed fruits
- Peanuts
- Nuts
- Sesame seed
- Soy
- Celery
- Egg
- Crustaceans
- Coriander

Little time to cook (C6)
All recipes $\leq$ 20 minutes preparation time.

4.2.3 Guideline combinations and impossible combinations

As mentioned above, guidelines can be combined. However, there are also some guidelines which can not be combined. Losing weight (A1) and malnutrition (B7) are impossible to combine from a technical point of view, but also from an medical point of view. Similarly for losing weight (A1) and surgery (C5) in the Foodfirst Network system.

The combination diabetes type II (B3) with improved bowel movements (A7) is impossible because few recipes meet the guidelines. More work will be required to provided suitable recipes for these combinations.

- A1 + B7
- A1 + C5
- B3 + A7

5 Personalized recipe package recommender system

In this section the system to recommend personalized food packages is explained. As described in Section 3.5 a package recommender system consist of three phases: model learning, package creation and package selection. The system will be explained according to these phases. In each phase the possibilities and choices will be explained, as well as the algorithms used. Figure 2 provides an overview of the system. Each of the blocks will be explained in subsections.

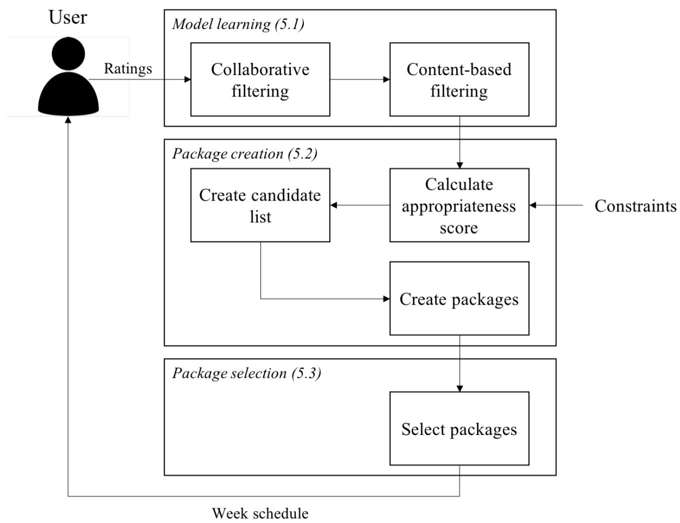

Figure 2: System architecture

5.1 Model learning

As described in Section 3.5.1, not all package recommender systems make use of a model learning phase. In some cases the preferences of users are not necessary to make recommendations. The research in this thesis does take the preferences of users into account, thus it makes use of model learning. To determine how the preferences should be elicited, a number of decisions had to be made. First, what type of input data should be used. The elicitation of a user's preferences can be done with implicit- or explicit input data. Both come with advantages and disadvantages.

Implicit vs explicit input data. Implicit input lacks negative feedback. It is hard to determine which recipes a user did not like. For instance, a user ignored an item because he or she did not like it, or because he or she did not know it was there [107]. Another disadvantage is that implicit input data is inherently noisy. The preferences of users can only be guessed, because the data only indicates how long or often somebody has visited an item [107]. The number of times a user visited an item does not necessarily indicate a user has a higher preference for an item, where a higher rating in explicit input data does indicate the preference. Because of these arguments, implicit input data inferences are often regarded as less accurate than explicit input data [107, 108, 109]. However, explicit data could be biased by users responding in a socially desirable way [110], or by maintaining a self-image for others [111]. Additionally, explicit feedback requires effort from users in contrast to implicit input data [108]. Hence, according to Zhao et al. there has been a transition in the use of input data from explicit to implicit [109].

Based on the different arguments, there is no method that should be preferred above the other. In this situation the factors of Foodfirst Network should also be taken into consideration. Within their platform there already is data on how many times a recipe has been visited and by who. However, their content is organized by presenting new recipes at the top and the old ones at the bottom, a lot of recipes have few views. This makes this data not appropriate for this research. Hence, explicit input data is more accurate and easier to infer. For this reason this research will make use of explicit input data.

Single user vs group recommendation. Another important decisions that had to be made was if the system should recommend to a single user or a group of users. Both options could be argued for. A lot of people have dinner together, which would be an argument to use group recommendation. However, this form of recommendation is more complex and also it makes it harder to implement user's goals. Therefore this research will make use of single user recommendation. The group recommendation could be an option in a follow up research.

Below different methods to determine a user's preferences will be shortly elaborated. Again, decisions were made to select methods for this research. Additionally, the working of these chosen methods is explained.

5.1.1 Collaborative filtering

As individual ratings (explicit) are used as input for this research, it is convenient to take a look at the articles which have this same input type. Table 6 shows that P2, P24, P25, P27, P49 and P50 use explicit input. Analyzing these articles for their methods used for model learning, shows that 3 use matrix factorization, 1 uses the multi-attribute utility theory, for 1 it is not clear what method is used, and 1 does no apply model learning at all. So, matrix factorization is used most often in this situation. Hence, this research will apply matrix factorization in combination with gradient descent to learn the user model [15].

Matrix factorization. Matrix factorization is a method which can be used to predict the null values within a matrix [112]. In this case to predict ratings for recipes that have not yet been rated by users. A small example of what a rating matrix will look like is shown in Table 12.

The main principle of matrix factorization is that it splits the rating matrix into two matrices called P and Q. By factorizing the values of matrices P and Q, an approximation of the input matrix can be created. P and Q are created by the system and look like the matrices in Tables 13 and 14. The matrices contain latent features, in this example called F1 and F2. These latent features are the weights which are used to calculate the ratings and find dependencies in the matrix. In this case there are 2 latent features, but there can be an unlimited number of latent features. The more latent features, the more accurate, but

[15] Gradient descent is most commonly used in combination with matrix factorization.

Table 12: User-item rating matrix

	Item A	Item B	Item C	Item D	Item E
User 1	4				1
User 2		3	3		1
User 3	4		3	5	

Table 13: Matrix P

	F1	F2
User 1	0.71	1.26
User 2	0.72	0.61
User 3	0.57	1.47

also the slower the system. This is a trade off that has to be made.

(1) The system starts by enter random values for latent features in the matrices P and Q. In this research this will be values between 0 and 1. (2) Then the system will use the dot product to calculate the first predictions. The formula for the dot product is the following:

$$(F1_P \times F1_Q) + (F2_P \times F2_Q) = Prediction$$

If the given examples are used and the prediction for User 1 and Item A has to be calculated this is done as follows.

$$(0.71 \times 0.44) + (1.26 \times 0.47) = 0.90$$

As can be seen, the 0.90 is not even close to the actual rating User 1 gave to Item A. This means the latent features have to be adjusted to get closer to the actual rating, also known as reducing the error. First the system has to know what the actual error is. This is calculated as follows.

Table 14: Matrix Q

	Item A	Item B	Item C	Item D	Item E
F1	0.44	0.99	0.59	0.01	0.45
F2	0.47	0.57	0.11	0.04	0.38

$$(Observation - Prediction)^2 = Squared\ Error$$

If the squared error is applied to the example, the following happens.

$$(4 - 0.90)^2 = 9.61$$

The squared error for Item A for User 1 is 9,61. If the squared error is calculated for all observations, so not the null values, this results in the Sum of Squared Errors (SSE). The SSE indicates how much the total error of the system is. The goal is to minimize the SSE to get the most accurate predictions.

$$\sum(Observation - Prediction)^2 = SSE$$

Gradient descent. To decrease the SSE, the system has to adjust the latent features F1 and F2. This is done by an algorithm called gradient descent [113]. This algorithm calculates by what margin, and in what direction the features have to be changed. The algorithm can be seen below.

$$New\ F_P = F_P + 2 \times \alpha \times (observation - prediction) \times F_Q$$

$$New\ F_Q = F_Q + 2 \times \alpha \times (observation - prediction) \times F_P$$

The F_P and F_Q are the latent features and α is the learning rate. The learning rate is the size of the steps the system takes. If the steps are too large, the system will be less accurate, but if the steps are too small, the system will need a lot of time to learn. An example of the learning rate is 0.01. If the gradient descent algorithm is applied on the example this will look as follows.

$$\begin{aligned}
New\ F1_P &= 0.71 + 2 \times 0.01 \times (4 - 0.90) \times 0.44 = 0.74\\
New\ F1_Q &= 0.44 + 2 \times 0.01 \times (4 - 0.90) \times 0.71 = 0.48\\
New\ F2_P &= 1.26 + 2 \times 0.01 \times (4 - 0.90) \times 0.47 = 1.29\\
New\ F2_Q &= 0.47 + 2 \times 0.01 \times (4 - 0.90) \times 1.26 = 0.55
\end{aligned}$$

With this new latent feature values, new predictions and a new error can be calculated.

$$(0.74 \times 0.48) + (1.29 \times 0.55) = 1.06\ (4 - 1.06)^2 = 8.64$$

The new prediction is 1.06 and the error has dropped from 8.61 to 8.64. This process should be done for all observations in the matrix to complete 1 iteration. The reduction of the SSE is an iterative process until the reduction of the SSE is below a certain number per iteration or after an x number of iterations has been done. When this process is done the system can calculate the predictions for the null values of items that have some ratings.

5.1.2 Content-based filtering

Within Foodfirst Network new recipes and new users are added regularly. To overcome the cold-start problem a content-based filtering method has been selected to complement the collaborative filtering. Therefore the features of items, so in this case the ingredients of recipes, should be extracted. Content-based filtering methods calculate the similarity or correlation between items based on item features. There are several methods to calculate the similarity between items. For instance, the Jaccard measure, which is used often in the papers of the systematic review. Some other methods which can be used are Pearson correlation, Dice coefficient and cosine similarity.

However, before this similarity measure is calculated, it is important to know if all features of an item are equally important. In the case of recipes, this is not the case. For instance, most recipes contain olive oil. This is not normally an important ingredient for people who may care more about ingredients such as chicken. However, if ingredients are used as equally important in the similarity calculations, an ingredient like olive oil would have much impact on the rating prediction the system would give a recipe. An ingredient such as chicken should make a lot more difference. To determine the importance weights could be given to ingredients. While there are several methods to do this, most researchers make use of the vector space model for content-based filtering. In particular the TF-IDF method is popular [114]. This method was introduced to analyze text in documents. The TF (Term Frequency) part counts the number of times a certain word occurs in the text in relation to the total number of words in that text. Than the IDF (Inverse Document Frequency) determines how many documents contain a certain word in relation to the total number of documents in the set. It basically tries to determine how important a certain term is to give it a weight.

Another method that could be used to determine the weights of ingredients is the Winnow algorithm [115]. Hereby all ingredients start with a certain value/begin weight, for instance 1, and a threshold for instance 5. For every recipe which is rated it determined if the rating is positive or negative. If a user is positive about a certain recipe, and the sum of all weights of the recipe's ingredients is equal or lower than the threshold, than those weights are multiplied by 2. If a user is negative about a certain recipe and the sum of the recipe's ingredients weights is higher than the threshold, than those weights should be divided by 2.

The above mentioned method is basic and effective. However, there is an issue. The output of the collaborative filtering method is a predicted rating for each recipe. With this method it is harder to make rating predictions, because the weights are not between 0 and 1.

Because TF-IDF is used in most cases and the Winnow algorithm is harder to implement, the TF-IDF algorithm is chosen as algorithm for this research. In the case of recipes it can be used to determine the importance of ingredients in a recipe. For the TF part we will use the amount of every ingredient that is added to a recipe rather than frequency per se. So for instance we count the weight of every ingredient and take that number as input. However, different kinds of weighting terms are used to indicate the amount of an ingredient that should be added to a recipe. Examples can be a thee spoon of Kurkuma, 400g tomatoes, 30ml olive oil, 3 branches of coriander etc. So to be able to use this method, the weight of each ingredient must be unified first. This is done by transforming all ingredients to weights in gram. All ingredients that were listed in ml were transformed 1:1 to grams. So 100ml became 100 gram and 1 liter became 1000 gram. Ingredients that were listed as pieces, so 1 apple, 4 branches of mint etc., were all searched for weight on Google to get an indication. When there was a range of the weight, for instance an apple is between 125g and 175g, the average was taken. The list of transformed ingredients is located in Appendix 2. One of the downsides is that the weight of the ingredients is not exactly right in this way. However, accuracy is not that important. It is more about an indication of the weight to determine the ingredient's importance. Another downside is that the intensity of the taste of ingredients is not taken into account. For instance, a gram of mint has more impact on the taste than a gram of tomato. This can be refined in future.

When all ingredients are unified in gram, the TF part can be calculated with the following formula. I_W is the weight of an ingredient in gram and $\sum I$ is the sum of the weight of all ingredients in a recipe.

$$TF = \frac{I_W}{\sum I}$$

The IDF part calculates how important an ingredient is in relation to all other ingredients. This is done based on the amount of recipes it occurs in. The less recipes it occurs in, the more impact it has on the rating of a recipe. For instance, olive oil is used very often in recipes, so probably does not have a big impact on a recipe, while chicken does not occur that often and is thus more important. Below is the formula for the IDF part, were R is the total number of recipes in a set and n is the number of recipes an ingredient occurs in.

$$IDF = log(\frac{R}{n})$$

TF-IDF is calculated by taking the product of TF and IDF. This score is calculated for all ingredients in a recipe.

$$\textit{TF-IDF} = TF \times IDF$$

Cosine similarity

When the weights of the ingredients are calculated, we need to calculate the similarity between recipes. Cosine similarity is the method which is most commonly used in combination with TF-IDF. For this reason cosine similarity is the similarity method for this research. The formula for cosine similarity is described below.

$$Cos = \frac{R_1 \cdot R_2}{||R_1|| \times ||R_2||}$$

The upper part of the formula calculates the dot product between two recipes. So the TF-IDF score of recipe 1 ingredient A times the TF-IDF score of recipe 2 ingredient A. This is done for all ingredients of the compared recipes.

$$R_1 \cdot R_2 = (R_{1A} \times R_{2A}) + (R_{1B} \times R_{2B}) + ...$$

The bottom part of the formula calculates the squared product of each ingredient. This is done by calculating the sum of each squared ingredient and than taking the root of the sum.

$$||R_1|| = \sqrt{({R_{1A}}^2 + {R_{1B}}^2 + ...)}$$

Rating prediction

When the similarities of the recipes are calculated these measures can be used to predict the remaining null values. For instance, if a new recipe is added this recipe has not yet been rated. Collaborative filtering will thus not work in this case. The similarity of recipes which have been rated are used to calculate a prediction. The idea is to predict the ratings based on the ratings given to the most similar recipes. In this case the 3 most similar recipes that already have been rated or have a rating prediction are used to predict a rating for not yet predicted recipes. The following formula should be used with the 3 most similar recipes. R_n is the rating of a recipe and $simR_n$ is the similarity of a recipe.

$$Rating\ prediction = \frac{(R_1 \times simR_1) + (R_2 \times simR_2) + (R_3 \times simR_3)}{simR_1 + simR_2 + simR_3}$$

With this formula the recipe that is most similar is more important than recipes that are less similar. This means that the rating of the most similar recipe has more effect on the prediction than less similar recipes. For instance, a rating for a recipe has to be predicted and the 3 most similar recipes have a similarity of 0.725, 0.790 and 0.921 and ratings of 3, 5 and 5. The prediction is calculated as follows.

$$Rating\ prediction = \frac{(3 \times 0.725) + (5 \times 0.790) + (5 \times 0.921)}{0.725 + 0.790 + 0.921} = 4.40$$

The rating prediction for this recipe is 4.40. This can be done for all the remaining recipes to create a full rating matrix. When the rating matrix is complete, the system will perform the next step.

5.2 Package creation

Methods. As can be seen in Section 3.5.2, four types of techniques have been identified for the creation of packages during the systematic literature review: knapsack, clustering, expanding from pivot items, and similarity measures. The requirements for this research is that the food packages are diverse and meet the requirements for preferences and constraints. One of the results of the systematic literature review was that most package recommendation research make use of a knapsack solution. A knapsack solution fits well with the requirements of the system, but it has a problem regarding this research. It focuses on maximizing the package score. In most cases a user would like the system to present the optimal solution. However, the optimal solution means the system would recommend the same package every day. In travel this is ideal, because somebody plans a holiday maybe once or twice a year. For a recommendation which has to be done every day, this is not a good solution. Clustering is used to group certain items based on features. This can be used to create clusters based on items of one group, or selecting an item from each cluster to cover diversity. This diversity could be useful for this system. However, items do not have to be very different in every package, but packages must not be the same every day or every week. Expanding from pivot items selects a pivot item and constructs packages around a pivot. This could be a good method for the creation of packages. In the following section a combination of the knapsack problem and expanding from pivot items is used.

To create packages, first a candidate list of items should be realized as mentioned in Section 3.5.2. A candidate list of items could contain all items in a set, but in many cases it is a selection of items which meet certain criteria. For this research the user preferences of the model learning phase, and the user-specific constraints from Section 4.2 are used to create a candidate list. The candidate list must consist of recipes that users would like to eat and that meet the constraints.

Constraints. What stands out is that all articles, except one, of the systematic review articles make use of constraints. These articles use constraints to filter out packages or items that exceed budget. For instance available time has been used as a constraint. A package was constructed based on an amount of time.

This research makes use of two types of constraints: numeric- and Boolean constraints. These types are elaborated in Section 4.2. The numeric constraints are soft constraints while the Boolean constraints are hard constraints. There is a small margin by which the numeric constraints are allowed to be exceeded. Because food is consumed every day, there is some margin for error. For instance, if a user would consume a little too much carbohydrates now, but saves some carbohydrates the next day, on average the food consumption is good. For Boolean constraints there is no margin for error. For example, if somebody is allergic the ingredients should not be consumed.

Appropriateness score. To determine a candidate list, recipes should be

Table 15: Rating to appropriateness score conversion

Rating	**Appropriateness score**
5	1.0
4	0.8
3	0.55
2	0.3
1	0.1

assessed. The assessment should be based on the combination of the preferences and constraints. However, the format of the preferences and constraints should be changed. Therefore, an appropriateness score was introduced. To recommend a recipe, a method was developed to calculate an appropriateness score based on the user preferences and user-specific constraints. The appropriateness score is a score between 0 and 1. To calculate this score the user ratings, numeric constraints and Boolean constraints are transformed into a score between 0 and 1.

The first step is to determine which constraints belong to a user. If no constraints apply, the appropriateness score is the score of the transformed rating. This will be explained below.

The second step is to transform the ratings, numeric constraints, and Boolean constraints into scores between 0 and 1. Each of these types are calculated in a different way.

- For the ratings, a transformation table is constructed, see Table 15.
- **Boolean constraints.** These are either true or false; true corresponds to a score of 1 and false to a score of 0.
- **Numeric constraints.** If the value of a recipe is within the numeric constraints, the value of the numeric constraint is 1. If the value of a recipe exceeds the numeric constraint, the score for the constraint should be calculated. The more the constraint is exceeded, the lower the score is. There are two formulas to calculate the score, one for constraints where the value has to be greater or equal to the constraint and one for constraints where the value has to be lesser or equal to the constraint. The formula below is for values that have to be greater or equal to a constraint. Hereby V_R is the value of a recipe and V_C is the constraint value.

$$Score = 1 - (((\frac{V_R}{V_C}) - 1) \times 2.5)$$

The next formula is for values that have to be lesser or equal to a constraint.

$$Score = 1 - ((1 - (\frac{V_R}{V_C})) \times 2.5)$$

In practice this works as follows. If an user has a maximum intake of 650 calories a day and a meal has 690 calories, the calculation works as follows.

$$Score = 1 - (((\frac{690}{650}) - 1) \times 2.5) = 0.85$$

If all scores for the ratings and constraints are calculated, the appropriateness score can be determined. The rating of a recipe and all constraints that have been calculated in the steps before will be used in the following formula. NC are the numeric constraints, BC are the Boolean constraints, n is the number of numeric constraints, and *score rating* is the appropriateness score of the rating. This will result in an appropriateness score between 0 and 1.

$$Overall\ Score = (\frac{NC_1 + NC_2 + ... + NC_n}{n}) \times (BC_1 \times BC_2 \times ... \times BC_n) \times\ score\ rating$$

When the appropriateness scores of all the recipes are known a candidate list can be constructed. To create food schedules, only recipes should be selected that fit the preferences and constraints of a user. To meet these requirements only recipes with an appropriateness score of ≥ 0.8 are added to the candidate list.

BOBO algorithm. As mentioned in the beginning of this section the BOBO algorithm is chosen as method to create packages. The method was proposed by Amer et al. to create bundles of items (P31). The algorithm can be seen in Algorithm 1. The idea is that at each step an item is chosen as pivot. Then a valid bundle is built around the pivot by selecting the closest items to the pivot. The picking of items is done by the pick_bundle algorithm which can be seen in Algorithm 2. The bundles are created by selecting items from a list of items and one item can only be in one bundle. Constraints are taken into account, because the algorithm makes use of a knapsack solution by greedily picking items.

Algorithm 1: BOBO (P31)

Input: I, α, f, β, minimum bundle score μ, and number of bundles c
Output: a set of c valid candidate bundles

```
Cand ← ∅;
Pivots ← I;
while Pivots ≠ ∅ and Cand < c do
    w ← pick an element from Pivots;
    Pivots ← Pivots \{w};
    S ← pick_bundle(w, I, α, f, β);
    if score(S) ≥ μ then
        Pivots ← Pivots \S;
        Cand ← Cand ∪ {S};
    end
    return Cand
end
```

Algorithm 2: pick_bundle (P31)

Input: pivot w, set of items I, parameters α, f, β

```
s ← {w}; covered ← {w.C}; active ← I \{w}; finish ← false;
while not finish do
    i ← argmax_[i∈active] s(i,w);
    if i.α ∉ covered then
        if f(s ∪ {i} ≤ β) then
            s ← s + i; covered ← covered ∪ {i.α};
        else
            finish ← true;
        end
        active ← active \{i};
    end
end
```

For the creation of week schedules a simplified version of the BOBO- and pick_bundle algorithm is made. This algorithm can be seen in Algorithm 3. Because the created candidate list is already checked for constraints, this is left out in the new algorithm. Additionally, there are three lists items are picked from instead of one list: breakfast-, lunch-, and dinner recipes. For each package one item is randomly picked from one bundle, while the BOBO algorithm greedily picks items until the value is maximized (see Algorithm 4).

Algorithm 3: Day schedule

Input: set of 21 breakfast recipes b, set of 21 lunch recipes l, set of 21 dinner recipes d
Output: a set of 21 valid day schedule packages
$Cand \leftarrow \emptyset$;
while $b \neq \emptyset$ ***and*** $Cand < 21$ **do**
    $w \leftarrow$ pick an element from b;
    $b \leftarrow b \setminus \{w\}$;
    $S \leftarrow$ pick_bundle(w, l, d);
    $Cand \leftarrow Cand \cup \{S\}$;
    return *Cand*
end

Algorithm 4: Day schedule pick_bundle

Input: w, set of lunch recipes l, set of dinner recipes d
$x \leftarrow$ pick element from l;
$l \leftarrow l \setminus \{x\}$;
$y \leftarrow$ pick element from d;
$d \leftarrow d \setminus \{y\}$;
$S \leftarrow$ w + x + y;

To create week schedules Algorithms 3 and 4 have been slightly modified. Algorithm 5 is a modified version of algorithm 3. Instead of three lists with items there is one list with candidate day schedules. The algorithm will continue until three candidate week schedules are made instead of the 21 day schedules of Algorithm 3. Algorithm 6 is more similar to the original pick_bundle algorithm than Algorithm 4 because it iterates to pick items.

Algorithm 5: Week schedule

Input: set of 21 day schedules *day*
Output: a set of 3 valid week schedule packages
$Cand \leftarrow \emptyset$;
while $day \neq \emptyset$ ***and*** $Cand < 3$ **do**
    $w \leftarrow$ pick an element from day;
    $day \leftarrow day \setminus \{w\}$;
    $S \leftarrow$ pick_bundle(w, day;
    $Cand \leftarrow Cand \cup \{S\}$;
    return *Cand*
end

Algorithm 6: Week schedule pick_bundle

Input: w, set of day recipes day
$i \leftarrow 1$;
while $i \leq 7$ **do**
  $x \leftarrow$ pick element from day;
  $i \leftarrow i + 1$;
  $S \leftarrow w + x$;
end
return S;

5.3 Package selection

In Section 3.5.3 it is explained that nearly all research focuses on top-N or top-1 recommendation. So, the packages with the most value to the user are recommended. However, in this research it is not about the most value, but about good packages. As already mentioned this has to do with the fact that packages are recommended on a regular basis instead of once in a while like is the case for most research. From the algorithms in Section 5.2 three candidate week schedules remain. One of those is randomly picked and recommended to the user.

6 Data set creation

For the system to work and to evaluate it, we require a data set. To start with the creation of the data set, first a subset for all recipes was made. The total number of recipes is 642. The goal was to get a minimum of 20 ratings per recipe and that each participant rates 30 recipes on average. To start of a goal of 100 participants was made. This would result in a total of 3000 ratings. With a minimum of 20 ratings per recipe this results in a subset of 150 recipes out of the 642 recipes. 50 breakfast, 50 lunch, and 50 diner recipes are selected. The selection of these recipes was done randomly.

To measure the food preferences of people in the best way, the idea was to give people a diverse list of recipes to find their taste along the whole food preference spectrum. There were three options to select recipes. The first was to find literature about food preferences and based on that select recipes that cover the total food preference spectrum. Initially this was a logical choice, but the food preferences research had 4 main categories: salt, sour, sweet and bitter. For single ingredients it is easy to determine if it is salty or sour, but for recipes this is different. Several of these categories are mixed and this results into a very difficult process to determine if a recipe is sweet, sour or something in between. A second option was to get the total list of ingredients used in the recipes of our data set and try to creating sets of recipes that cover as much of these ingredients as possible, while have diverse ingredients between recipes. This idea was actually quite good, but the problem was that it would involve a lot of manual labour. Because of time constraints this was considered as not the best option. The last option was to randomly select 150 recipes from the total set. And this option was chosen.

Now an example is given of the randomization process. Each category, breakfast, lunch, and diner, has its own randomization process. For instance, a list of all diner recipes was made and this list was used as input for the list randomizer on Random.org. In Figure 3, part 1 can be seen that the list of diner recipes is entered into the application. When the randomize button is pressed the order of the list is randomized. The results of this process can be seen in Figure 4.

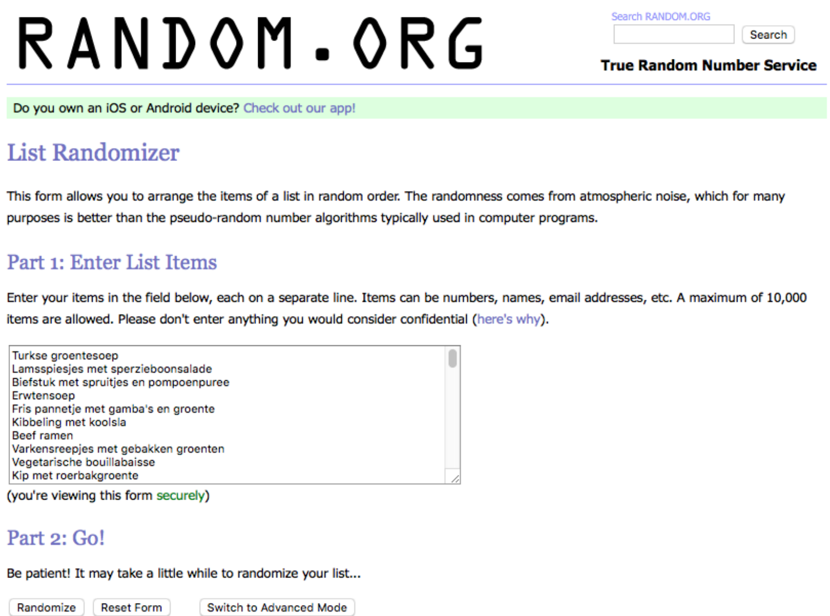

Figure 3: Random recipe selection with Random.org part 1

To collect data a questionnaire was used to capture user's ratings of recipes. The participants got to see a title, photo, and the ingredients of the recipe and had to rate the recipe between 1-5 based on this information. To predict unknown ratings in a rating matrix with machine learning, participants should rate partially overlapping recipes and partially different recipes. This means that participants their recipe ratings partially overlap. An application which could randomly create questionnaires based on a list of questions to meet the requirements was searched for, but with no success. Due to the lack of this sort of application, 10 questionnaires were made which all have overlap with some of the other questionnaires. A scheme was made to determine which recipes were selected for each questionnaire. In Table 16 the articles selected for each questionnaire are listed. These numbers are based on the 50 breakfast, lunch, and diner recipes that have been randomly selected. To select recipes for each

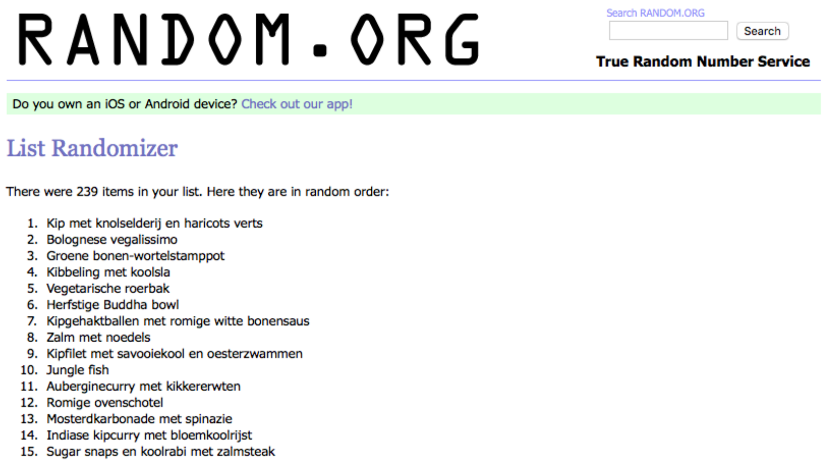

Figure 4: Random recipe selection with Random.org part 2

questionnaire the same list randomizer of Random.org was used.

Because the more ratings the better, each participant got the possibility to rate a maximum of 5 questionnaires. For each questionnaire a path is made with follow up questionnaires. After a participant had finished a questionnaire he or she was asked to fill in another questionnaire. To prevent double ratings for recipes, already asked recipes were removed. All 10 questionnaire paths are showed in Table 17.

The questionnaire was opened online on the 7th of August and closed on the 24th of September. At first personal connections were used to spread the questionnaire. Because there is no good or bad answer or any other form of way to influence the results in favor of the research, this is no problem. These connections also spread the questionnaire with their connections. The second group were members of the Foodfirst Network platform. In total 55 members of the Foodfirst Network platform and 84 connections and their relations participated. This total number of 149 participants resulted in 10.544 ratings. 3.645 breakfast, 3.560 lunch, and 3.339 diner ratings. Every recipe is at least rated 49 times and the maximum is 98 ratings for a recipe. This exceeded the goals we had set for the data set.

7 Evaluation

In this section the evaluation of the system mentioned in Section 5 is described. Two important kinds of evaluations were identified in the systematic literature review: computational evaluation and user study & expert evaluation. In most research computational evaluation is used. However, user studies could be very important as well. Due to time constraints, this research only used computational evaluation.

Table 16: Recipe selection for questionnaires

Questionnaire	Breakfast	Lunch	Diner
1	1-10	46-5	16-25
2	6-15	6-15	31-40
3	11-20	11-20	41-50
4	16-25	26-35	11-20
5	21-30	21-30	36-45
6	26-35	31-40	21-30
7	31-40	16-25	1-10
8	36-45	1-10	6-15
9	41-50	36-45	26-35
10	46-5	41-50	46-5

Table 17: Questionnaire paths

Q1	Q2	Q3	Q4	Q5
1	3	5	7	9
2	4	6	7	10
3	1	6	8	9
4	2	7	9	10
5	2	3	9	10
6	1	3	5	8
7	1	2	4	8
8	3	5	6	10
9	1	4	5	7
10	2	4	6	8

7.1 Experimental settings

There are many different kinds of computational evaluation methods. For this research the system is evaluated on accuracy and diversity. Accuracy is chosen to measure the performance of this newly proposed system. Accuracy is important for all recommender systems to know if the predictions made are trustworthy. Diversity is evaluated because the system is built to recommend to users on a daily basis. This means diversity is important, because users would not like to eat the same recipes over and over again. Additionally, with the diversity measure the assumptions made about the day- and week schedule guidelines in Section 4.2 can be examined.

Accuracy. The results of the systematic literature review in Section 3 show many types of accuracy metrics. One of the most used types of methods are the rating accuracy metrics. These metrics measure the extent to which a prediction corresponds to an actual rating. The downside of these methods is that high and low rated items are treated equally, while users tend to be interested in high rated items. For this reason, Precision@N, Recall@N, and F_1 have been chosen to evaluate the accuracy [116]. These are all metrics that evaluate the top-N rated items. Precision@N is the ratio between the relevant recipes among the retrieved recipes. Recall@N determines the sensitivity of a system. The ratio between the total number of relevant items and the amount of relevant items that were retrieved. Relevant recipes for this research are recipes with a rating of 3,5 or higher, because these are the recipes users prefer to eat. The F_1-measure is the harmonic mean of the combination of precision and recall. Below the formulas are provided. True positives (TP) are recipes that are predicted as relevant and also are relevant. False positives (FP) are recipes that are predicted as relevant, but are not relevant. False negatives (FN) are recipes that have been predicted as not relevant, but are relevant.

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

K-fold cross validation. To evaluate the performance of the system, the accuracy metrics were combined with the k-fold cross validation method. K-fold cross validation is a method to determine how a system will react to unseen data. This method is used when the data available is limited, which is the case for this research. With k-fold cross validation, the data set is split into several equally sized parts called folds. Next, a part of the data is used to train the model and one fold is used to validate the model. All folds have to be used as validation fold once, so the process iterates as often as the number of folds chosen. In this research the data will be split up into four different folds. This means the above described process will be iterated four times. Each time the values for precision, recall and F_1 will be determined. The average of these values over the iterations will be the precision, recall and F_1 values for this research.

The accuracy metrics and k-fold cross validation were only applied to the matrix factorization part of the system. There is not enough data to evaluate the content-based filtering part and there were no package ratings collected to evaluate the packages. These are limitations in this research and are mentioned as future work.

Diversity. For diversity only two methods were identified during the systematic literature review: intra- and inter package diversity. With inter package diversity the diversity between packages is measured, while intra package diversity measures the diversity between items in a package. Both types of diversity were used to conduct the evaluation.

The measurement of diversity is a hard problem to solve for packages. In the systematic review some papers mention methods to solve this problem. Because diversity is the opposite of similarity and similarity metrics have already been used in this research, the already used methods will be used to determine intra- and inter package diversity. This means that the closer the similarity is to 0, the more diverse the packages are. The similarity of recipes in a day package, so a breakfast, lunch and dinner recipe, is calculated by taking the mean of the similarities. The total intra package similarity is calculated by taking the mean of the sum of the similarity of day packages.

For inter package diversity the cosine similarity between two packages is calculated. This is basically defined by the number of equal recipes that have been used in two compared packages. The formula for this calculation is mentioned in Section 5.1.2 [16].

Personas. To evaluate the system, week recipe schedules have been made for different personas. These personas have different constraints. For the personas, three users have been randomly selected from the data set to evaluate the personas based on ratings given [17]. The selected users are user 28, 34 and 71. All users will be used to evaluate each persona. The users are used to take preferences into account during the evaluation.

According to the design of the system, for each user candidate lists, day packages, and week packages were created. In total for each persona three week packages per user were created and compared on diversity. If there are less than 21 valid recipes to select for the candidate list, the total number of valid recipes is selected. When the week schedules are constructed, and the number of valid

[16] Subsequently using cosine similarity is an oversimplification of reality. Recipes which have similar ingredients, but are not the same recipe are now seen as not similar at all. Method from intra package diversity would have been better

[17] Users can be chosen randomly. However, it could be that a user who rated recipes and has a nuts allergy gave low ratings to recipes with nuts. However, it could also be that a user did not like nuts.

recipes is between 21 and 14 recipes, two week schedules will be made. If the number is between 7 and 14, one week schedule will be made. If the candidate list has less than 7 recipes, the week schedule will be made with multiple of the same recipes.

Persona 1: The first persona is someone without any constraints or goals. This means, the system could recommend any recipe to a user, within the user's preferences, according to the constraints of Foodfirst Network.

Persona 2: The second persona is a person who has a gluten allergy (B2). This means the system should take the Boolean constraint into account. This is one of the allergies that affects the number of appropriate recipes most.

Persona 3: The third persona is a person who has many constraints. The person has diabetes type II (B3) and suffers from high blood pressure (B5). He is advised to lose weight (A1) and he is also a vegetarian (B1). This is probably a rare scenario because of the many constraints, but it is meant to ultimately test the possibility to recommend and evaluate the diversity of recipe week schedules.

For the evaluation, the system is evaluated by programming parts of the system in Visual Basic for Applications (VBA) inside Excel. Due to time constraints no fully working prototype was constructed; Foodfirst Network's development team is in the process using my design specification for an implementation. For matrix factorization 5 latent features and 100 iterations were used as settings, and for gradient descent $\alpha = 0.001$ was applied.

7.2 Results

Accuracy. For breakfast, lunch and dinner, seperate calculations were made. Table 18 shows the results of the precision, recall and F_1 metrics for all recipes. Where precision has similar values for breakfast, lunch and dinner recipes, recall has more varied values. If the number of observations are considered, these accuracy results are quite good. A lot of recipes have not been rated, so the content-based filtering method was applied. While the content-based method is not evaluated, the feeling is that this method is less accurate than the collaborative filtering method, due to the fact that some recipes have no recipes that are very similar. This would impact the prediction. However, this should be evaluated in future research.

Diversity. Tables 19, 20 and 21 show the number of candidate recipes for each of the personas in relation to the randomly selected users. For persona 1 only User 34 could not select enough valid candidate dinner recipes. This means only two complete week schedules could be made. As Tables 20 and 21 show, the number of valid candidate recipes decreases when the number of constraints gets higher.

The results of the intra package diversity can be seen in Table 22. The

Table 18: Accuracy metrics, average over 4 iterations

Metric	**Breakfast**	**Lunch**	**Dinner**
Precision	0.723	0.755	0.741
Recall	0.636	0.721	0.683
F_1	0.676	0.738	0.710

Table 19: Persona 1 candidate recipes, based on the preference data of Users 28, 34 and 71

User	**Breakfast**	**Lunch**	**Dinner**
28	141	189	156
34	18	102	20
71	37	39	206

similarities show that packages of the system are very diverse. For User 34, the diversity is a little less than for the other users. This could be due to the fact that User 34 had the least candidate recipes overall. However, User 71 has less candidate items for Persona 3 than User 34, but more diverse packages. It could also be that the preferences of User 34 are less diverse than of the other users.

Table 23 shows the results of the inter package diversity. The results indicate that the more constraints are applied, the less diverse packages become. This is due to the decrease of the number of candidate packages. The average results show that for the Personas 1 and 2 packages are diverse. For Persona 3 the average shows the diversity is much worse.

8 Conclusions and future work

In this research several contributions have been made. At first a systematic literature review has been conducted in the domain of package recommendation. As second a new data set has been made which can be used in new researches.

Table 20: Persona 2 candidate recipes, based on the preference data of Users 28, 34 and 7

User	**Breakfast**	**Lunch**	**Dinner**
28	72	121	126
34	5	59	17
71	22	25	166

Table 21: Persona 3 candidate recipes, based on the preference data of Users 28, 34 and 7

User	**Breakfast**	**Lunch**	**Dinner**
28	51	50	12
34	8	20	0
71	11	9	7

Table 22: Intra package similarity

User	**Persona 1**	**Persona 2**	**Persona 3**
28	0.015	0.015	0.026
34	0.024	0.042	-
71	0.018	0.025	0.018
Average	0.019	0.027	0.022

Table 23: Inter package similarity

User	**Persona 1**	**Persona 2**	**Persona 3**
28	0.016	0.064	0.238
34	0.206	0.524	-
71	0.016	0.158	0.826
Average	0.079	0.249	0.532

And at last a new package recommendation system in the food domain has been defined. This research describes what methods could be used and how existing and new methods are used to recommend recipe week schedules. However, this research can be extended in many different ways.

User study. In this research only computational methods have been used to evaluate the system. While these are good methods, the best way to evaluate a user's opinion is by involving users in the evaluation. This could be done by conducting a user study to evaluate the system.

Inter package diversity. The inter package diversity in this research was measured by applying cosine similarity. This caused an oversimplification of reality. For future research other diversity measured should be used to get a better indication of the diversity. One of the methods could be by calculating the mean of the similarity between several recipes as has been done for the intra package diversity.

Evaluate all other scenarios with personas. Evaluate more possible scenarios. In this research only two personas were created were the system had constraints to take into account. Different scenarios should be tested to find weak spots of the system.

Guidelines. During this research no day- and week schedule guidelines have been taken into consideration. Day guidelines are already considered by the recipe developers of Foodfirst Network, but this is not the case for week schedule guidelines. It was assumed that those were met if packages were created because of the diversity of the recipes. However, in future work packages should be evaluated based on these recipe week schedule guidelines which can be found in the "FFN Bijbel".

Algorithm biases. In this research biases in the algorithms are not used. For instance, in collaborative filtering biases can be used which take the rating behaviour of users into consideration. Some users tend to give higher ratings than others. This could be future work to improve the accuracy of the algorithms.

Ingredient intensity. In this research the taste intensity of ingredients per gram is not taken into consideration. For instance a gram of mint has a more intense taste than a gram of tomato. This taste intensity has an effect on the importance of an ingredient in a recipe. It should be researched how this intensity can be used in the TF-IDF algorithm or in an other method.

Package ratings. The assumption in this research is that there is no clear difference between the package ratings and combining the single item ratings. In most cases this could be the case, but in some occasions it will probably differ. For instance, most people would not like three times a meal with fish on the same day. Another example could be, if the three recommended recipes require a lot of different ingredients, where more overlapping ingredients would be more appropriate in terms of lower costs and less waste of food.

Recipe adjustments. Adjust the amount of a recipe to make it meet the constraints. In this case recipes would not per definition be inappropriate if the standard nutritional values do not meet the constraints. The adjustment could be several sizes of a recipe, or a percentage of recipes. The challenge is that recipes often take the size of products in the supermarket into consideration. The recipe adjustments should not decrease the convenience for users.

Group recommendation. The current system focuses on single users. But as mentioned in Section 5.1 users do often eat in company. For instance, with the household. In this case it would be more convenient to recommend recipe week schedules for a group of users.

Extending personalization. For this research guidelines provided by Foodfirst Network were used. These guidelines do not take a lot of personal factors not into consideration. For instance, length, age, gender and weight of a user, while these factors impact the nutritional needs of a user. In future work the possibilities to further personalize the system should be explored and researched.

Performance evaluation. Performance is important for most systems. In most cases it is an trade off between factors such as accuracy and performance. The time it takes to make recommendations should be taken into consideration in future research.

Input data. In this research explicit input data has been used. However, there are also arguments to choose for implicit feedback. For instance, the trend has shifted from explicit to implicit input data. Thereby it would reduce the effort for users of the system. Also other explicit input data could be used. For instance a like or dislike.

Recommendation explanation. An extension of the current system could be to explain why recommendations were made by the system. For instance, because a user liked recipe 1, recipe 10 is recommended. Or because the goal of a user is to lose weight, recipe 5 is recommended.

9 Challenges

During this thesis I had to take several obstacles.

Combining thesis and internship. Combining my work for Foodfirst Network and my thesis was sometimes difficult. Not in the case that there were conflicts about the content, but both tasks cost a lot of time which made it sometimes hard to combine. If too many effort was used for one task, the other was suffering.

Keep focus. From my personal experience it is sometimes hard for me to keep focus. My head is always full of ideas, which can make me a little chaotic.

Additionally my thoughts are moving faster than I can speak or write. So sometimes I forgot why I made a certain decision and this made it hard to memorize the way of thinking.

Expertise. At my internship company there was nobody with any knowledge about recommender systems or machine learning. This made it extra difficult, because I had to figure out everything myself. Additionally, it was harder to ask if I was doing the right thing, because the only person who could verify my work was my first supervisor. However, the up side is that I was able to learn very much by making mistakes.

References

[1] World Healthcare Organization. *Obesity and overweight.* Tech. rep. http://www.who.int/news-room/fact-sheets/detail/obesity-and- overweight. World Healthcare Organization, Nov. 2018.

[2] Volksgezondheidenzorg.info. *Overgewicht: huidige situatie.* URL: `https://www.volksgezondheidenzorg.info/onderwerp/overgewicht/cijfers-context/huidige-situatie#node-overgewicht-volwassenen`.

[3] Emmanuela Gakidou et al. "Global, regional, and national comparative risk assessment of 84 behavioural, environmental and occupational, and metabolic risks or clusters of risks, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016". In: *The Lancet* 390.10100 (2017), pp. 1345–1422.

[4] Tomoko Shimakawa et al. "Dietary intake patterns and sociodemographic factors in the Atherosclerosis Risk in Communities Study". In: *Preventive medicine* 23.6 (1994), pp. 769–780.

[5] Anju Aggarwal et al. "Does diet cost mediate the relation between socioeconomic position and diet quality?" In: *European journal of clinical nutrition* 65.9 (2011), p. 1059.

[6] Paul Resnick and Hal R Varian. "Recommender systems". In: *Communications of the ACM* 40.3 (1997), pp. 56–59.

[7] Paul Resnick et al. "GroupLens: an open architecture for collaborative filtering of netnews". In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work.* ACM. 1994, pp. 175–186.

[8] Bruce Krulwich and Chad Burkey. "Learning user information interests through extraction of semantically significant phrases". In: *Proceedings of the AAAI spring symposium on machine learning in information access.* Menlo Park: AAAI Press. 1996, pp. 100–112.

[9] Marko Balabanovic and Yoav Shoham. "Fab: content-based, collaborative recommendation". In: *Communications of the ACM* 40.3 (1997), pp. 66–73.

[10] Gunjan Kumar, Houssem Jerbi, and Michael P O'Mahony. "Towards the Recommendation of Personalised Activity Sequences in the Tourism Domain". In: *RecTour 2017 2nd Workshop on Recommenders in Tourism. Como, Italy, 27 August 2017*. ACM. 2017.

[11] Huang Yueh-Min et al. "A Markov-based recommendation model for exploring the transfer of learning on the web". In: *Journal of Educational Technology & Society* 12.2 (2009), p. 144.

[12] Francesco Ricci, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook". In: *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[13] Barbara Kitchenham and Stuart Charters. "Guidelines for performing systematic literature reviews in software engineering". In: (2007).

[14] Jing-jie ZHU et al. "A Package Recommendation Model Based on Credit and Time". In: *DEStech Transactions on Computer Science and Engineering* wcne (2017).

[15] Agung Toto Wibowo et al. "Matrix Factorization for Package Recommendations". In: *Proceedings of the RecSys 2017 Workshop on Recommendation in Complex Scenarios (ComplexRec 2017)*. CEUR-WS. 2017.

[16] Agung Toto Wibowo et al. "Incorporating Constraints into Matrix Factorization for Clothes Package Recommendation". In: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. ACM. 2018, pp. 111–119.

[17] Guannan Liu et al. "Modeling buying motives for personalized product bundle recommendation". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.3 (2017), p. 28.

[18] Senjuti Basu Roy et al. "Constructing and exploring composite items". In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM. 2010, pp. 843–854.

[19] Matthew Sessoms and Kemafor Anyanwu. "SkyPackage: From finding items to finding a skyline of packages on the semantic web". In: *Joint International Semantic Technology Conference*. Springer. 2012, pp. 49–64.

[20] Tao Zhu et al. "Bundle recommendation in ecommerce". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM. 2014, pp. 657–666.

[21] Moran Beladev, Lior Rokach, and Bracha Shapira. "Recommender systems for product bundling". In: *Knowledge-Based Systems* 111 (2016), pp. 193–206.

[22] Murium Iqbal, Adair Kovac, and Kamelia Aryafar. "A Multimodal Recommender System for Large-scale Assortment Generation in E-commerce". In: *arXiv preprint arXiv:1806.11226* (2018).

[23] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. "Recommendation systems with complex constraints: A course recommendation perspective". In: *ACM Transactions on Information Systems (TOIS)* 29.4 (2011), p. 20.

[24] Zhan Li, Olga Papaemmanouil, and Georgia Koutrika. "CourseNavigator: interactive learning path exploration". In: *Proceedings of the Third International Workshop on Exploratory Search in Databases and the Web*. ACM. 2016, pp. 6–11.

[25] Nan Li et al. "Enter a job, get course recommendations". In: *iConference 2017 Proceedings Vol. 2* (2017).

[26] Jie Xu, Tianwei Xing, and Mihaela Van Der Schaar. "Personalized course sequence recommendations". In: *IEEE Transactions on Signal Processing* 64.20 (2016), pp. 5340–5352.

[27] Karypis Morsy. "Learning Course Sequencing for Course Sequence Recommendation". In: (2018).

[28] Yanhong Pan, Yanfei Zhang, and Rong Zhang. "Combo-Recommendation Based on Potential Relevance of Items". In: *Asia-Pacific Web Conference*. Springer. 2016, pp. 505–517.

[29] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. "Generating and personalizing bundle recommendations on Steam". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 2017, pp. 1073–1076.

[30] Yasuyuki Shirai et al. "Incremental set recommendation based on class differences". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2012, pp. 183–194.

[31] Hanan Mengash and Alexander Brodsky. "GCAR: A Group Composite Alternatives Recommender Based on Multi-Criteria Optimization and Voting". In: *2014 47th Hawaii International Conference on System Sciences*. IEEE. 2014, pp. 1113–1121.

[32] Lakshmanan Khabbaz Xie. "Efficient Algorithms for Recommending Top-k Items and Packages". In: (2011).

[33] Panagiotis Kouris, Iraklis Varlamis, and Georgios Alexandridis. "A package recommendation framework based on collaborative filtering and preference score maximization". In: *International Conference on Engineering Applications of Neural Networks*. Springer. 2017, pp. 477–489.

[34] Min Xie, Laks VS Lakshmanan, and Peter T Wood. "Composite recommendations: from items to packages". In: *Frontiers of computer science* 6.3 (2012), pp. 264–277.

[35] Roberto Interdonato et al. "A versatile graph-based approach to package recommendation". In: (2013), pp. 857–864.

[36] Shuyao Qi et al. "Recommending packages with validity constraints to groups of users". In: *Knowledge and Information Systems* 54.2 (2018), pp. 345–374.

[37] Loc Do, Hady W Lauw, and Ke Wang. "Mining revenue-maximizing bundling configuration". In: *Proceedings of the VLDB Endowment* 8.5 (2015), pp. 593–604.

[38] Mohit Sharma, F Maxwell Harper, and George Karypis. "Learning from Sets of Items in Recommender Systems". In: *arXiv preprint arXiv:1904.12643* (2019).

[39] Vincent Leroy et al. "Building representative composite items". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM. 2015, pp. 1421–1430.

[40] Remigio Hurtado Ortiz, Rodolfo Bojorque Chasi, and César Inga Chalco. "Clustering-Based Recommender System: Bundle Recommendation Using Matrix Factorization to Single User and User Communities". In: *International Conference on Applied Human Factors and Ergonomics*. Springer. 2018, pp. 330–338.

[41] Panagiotis Kouris et al. "A versatile package recommendation framework aiming at preference score maximization". In: *Evolving Systems* (2018), pp. 1–19.

[42] Wenli Yu et al. "ProductRec: Product Bundle Recommendation Based on User's Sequential Patterns in Social Networking Service Environment". In: *2017 IEEE International Conference on Web Services (ICWS)*. IEEE. 2017, pp. 301–308.

[43] Hanan Mengash and Alexander Brodsky. "Tailoring Group Package Recommendations to Large Heterogeneous Groups Based on Multi-Criteria Optimization". In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE. 2016, pp. 1537–1546.

[44] Sihem Amer-Yahia et al. "Composite retrieval of diverse and complementary bundles". In: *IEEE Transactions on Knowledge and Data Engineering* 26.11 (2014), pp. 2662–2675.

[45] Horatiu Bota et al. "Composite retrieval of heterogeneous web search". In: *Proceedings of the 23rd international conference on World wide web*. ACM. 2014, pp. 119–130.

[46] Christian Villavicencio, Silvia Schiaffino, and J Andrés Dıaz Pace. "Solving Package Recommendation Problems with Item Relations and Variable Size". In: *Argentine Symposium on Artificial Intelligence (ASAI 2015)-JAIIO 44 (Rosario, 2015)*. 2015.

[47] Min Xie, Laks VS Lakshmanan, and Peter T Wood. "Generating top-k packages via preference elicitation". In: *Proceedings of the VLDB Endowment* 7.14 (2014), pp. 1941–1952.

[48] Yan Fang et al. "Customized Bundle Recommendation by Association Rules of Product Categories for Online Supermarkets". In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC).* IEEE. 2018, pp. 472–475.

[49] Sihem Amer-Yahia et al. "Task composition in crowdsourcing". In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA).* IEEE. 2016, pp. 194–203.

[50] Christoph Dorn et al. "Interaction mining and skill-dependent recommendations for multi-objective team composition". In: *Data & Knowledge Engineering* 70.10 (2011), pp. 866–891.

[51] Luis Castillo et al. "samap: An user-oriented adaptive system for planning tourist visits". In: *Expert Systems with Applications* 34.2 (2008), pp. 1318–1332.

[52] Idir Benouaret and Dominique Lenne. "Recommending diverse and personalized travel packages". In: *International Conference on Database and Expert Systems Applications.* Springer. 2017, pp. 325–339.

[53] Gang Chen et al. "Automatic itinerary planning for traveling services". In: *IEEE transactions on knowledge and data engineering* 26.3 (2013), pp. 514–527.

[54] Aristides Gionis et al. "Customized tour recommendations in urban areas". In: *Proceedings of the 7th ACM international conference on Web search and data mining.* ACM. 2014, pp. 313–322.

[55] Rashmi Hti and Maunendra Sankar Desarkar. "Personalized Tourist Package Recommendation Using Graph Based Approach". In: *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization.* ACM. 2018, pp. 257–262.

[56] C Abilash Reddy and V Subramaniyaswamy. "An enhanced travel package recommendation system based on location dependent social data". In: *Indian Journal of Science and Technology* 8.16 (2015), p. 1.

[57] Chang Tan et al. "Object-oriented travel package recommendation". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.3 (2014), p. 43.

[58] Cheng-Yao Fu et al. "Travelbuddy: interactive travel route recommendation with a visual scene interface". In: *International Conference on Multimedia Modeling.* Springer. 2014, pp. 219–230.

[59] Shenglin Zhao et al. "STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation". In: *Thirtieth AAAI conference on artificial intelligence.* 2016.

[60] Chenyi Zhang et al. "Personalized trip recommendation with poi availability and uncertain traveling time". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.* ACM. 2015, pp. 911–920.

[61] Hsun-Ping Hsieh and Cheng-Te Li. "Constructing trip routes with user preference from location check-in data". In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM. 2013, pp. 195–198.

[62] Daniel Herzog, Hesham Massoud, and Wolfgang Wörndl. "Routeme: A mobile recommender system for personalized, multi-modal route planning". In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM. 2017, pp. 67–75.

[63] Wolfgang Wörndl, Alexander Hefele, and Daniel Herzog. "Recommending a sequence of interesting places for tourist trips". In: *Information Technology & Tourism* 17.1 (2017), pp. 31–54.

[64] Wörndl Laß Herzog. "Context-Aware Tourist Trip Recommendations". In: *RecTour 2017 2nd Workshop on Recommenders in Tourism*. RecTour. 2017, pp. 18–25.

[65] Daniel Herzog and Wolfgang Wörndl. "A Travel Recommender System for Combining Multiple Travel Regions to a Composite Trip." In: *CBRecSys@ RecSys* 1245 (2014), pp. 42–48.

[66] Chenyi Zhang, Hongwei Liang, and Ke Wang. "Trip recommendation meets real-world constraints: POI availability, diversity, and traveling time uncertainty". In: *ACM Transactions on Information Systems (TOIS)* 35.1 (2016), p. 5.

[67] Hongwei Liang and Ke Wang. "Top-k route search through submodularity modeling of recurrent POI features". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM. 2018, pp. 545–554.

[68] Wolfgang Wörndl, Michael Ludwig, and Daniel Herzog. "Recommending Customized Trips Based on the Combination of Travel Regions". In: *ENTER 2015 Conference, Lugano, Switzerland, February 3-6, 2015*. 2015.

[69] Roberto Interdonato and Andrea Tagarelli. "Personalized recommendation of points-of-interest based on multilayer local community detection". In: *International Conference on Social Informatics*. Springer. 2017, pp. 552–571.

[70] Jitao Sang et al. "Probabilistic sequential POIs recommendation via check-in data". In: *Proceedings of the 20th international conference on advances in geographic information systems*. ACM. 2012, pp. 402–405.

[71] Ling-Yin Wei, Yu Zheng, and Wen-Chih Peng. "Mining popular routes from social media". In: *Multimedia Data Mining and Analytics*. Springer, 2015, pp. 93–116.

[72] Eric Hsueh-Chan Lu, Shih-Hsin Fang, and Vincent S Tseng. "Integrating tourist packages and tourist attractions for personalized trip planning based on travel constraints". In: *GeoInformatica* 20.4 (2016), pp. 741–763.

[73] Gang Hu et al. "Unifying multi-source social media data for personalized travel route planning". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 2017, pp. 893–896.

[74] Jiayuan He, Jianzhong Qi, and Kotagiri Ramamohanarao. "A Jointly Learned Context-Aware Place of Interest Embedding for Trip Recommendations". In: *arXiv preprint arXiv:1808.08023* (2018).

[75] Lei Yang et al. "A Tourist Itinerary Planning Approach Based on Ant Colony Algorithm". In: *International Conference on Web-Age Information Management*. Springer. 2012, pp. 399–404.

[76] Hsun-Ping Hsieh, Cheng-Te Li, and Shou-De Lin. "Measuring and recommending time-sensitive routes from location-based data". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.3 (2014), p. 45.

[77] Shuhui Jiang et al. "Personalized travel sequence recommendation on multi-source big social media". In: *IEEE Transactions on Big Data* 2.1 (2016), pp. 43–56.

[78] Vineeth Rakesh et al. "Probabilistic social sequential model for tour recommendation". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM. 2017, pp. 631–640.

[79] Jie Li, Yujiu Yang, and Wenhuang Liu. "Exploring personalized travel route using POIs". In: *International Journal of Computer Theory and Engineering* 7.2 (2015), p. 126.

[80] Qi Liu et al. "A cocktail approach for travel package recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 26.2 (2012), pp. 278–293.

[81] Ramesh Baral, Tao Li, and XiaoLong Zhu. "CAPS: Context Aware Personalized POI Sequence Recommender System". In: *arXiv preprint arXiv:1803.01245* (2018).

[82] Ramesh Baral et al. "CLoSe: C ontextualized Lo cation Se quence Recommender". In: *Proceedings of the 12th ACM conference on recommender systems*. ACM. 2018, pp. 470–474.

[83] Zhiwen Yu et al. "Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints". In: *IEEE Transactions on Human-Machine Systems* 46.1 (2015), pp. 151–158.

[84] Ramesh Baral et al. "HiCaPS: hierarchical contextual POI sequence recommender". In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM. 2018, pp. 436–439.

[85] Wong Biwas Creed. "Itinerary Planning Using Top-k Package Recommendation and Multiple Constraints". In: ().

[86] Xinhuan Chen et al. "A package generation and recommendation framework based on travelogues". In: *2015 IEEE 39th Annual Computer Software and Applications Conference*. Vol. 2. IEEE. 2015, pp. 692–701.

[87] Marwa Boulakbech et al. "Linked Data Graphs for Semantic Data Integration in the CART System". In: *International Conference on Web Engineering*. Springer. 2018, pp. 221–226.

[88] Haitao Xiong and Zhengbin Liu. "A situation information integrated personalized travel package recommendation approach based on TD-LDA model". In: *2015 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC)*. IEEE. 2015, pp. 32–37.

[89] Hsien-Tsung Chang, Yi-Ming Chang, and Meng-Tze Tsai. "ATIPS: automatic travel itinerary planning system for domestic areas". In: *Computational intelligence and neuroscience* 2016 (2016), p. 1.

[90] Sergei Mikhailov and Alexey Kashevnik. "An Ontology for Service Semantic Interoperability in the Smartphone-Based Tourist Trip Planning System". In: *2018 23rd Conference of Open Innovations Association (FRUCT)*. IEEE. 2018, pp. 240–245.

[91] Maha Alsayasneh et al. "Personalized and diverse task composition in crowdsourcing". In: *IEEE Transactions on Knowledge and Data Engineering* 30.1 (2017), pp. 128–141.

[92] Chao Chen et al. "TripPlanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints". In: *IEEE Transactions on Intelligent Transportation Systems* 16.3 (2014), pp. 1259–1273.

[93] William Jeffries and Alexander Brodsky. "Composite Alternative Pareto Optimal Recommendation System with Individual Utility Extraction (CAPORS-IUX)." In: *ICEIS (1)*. 2018, pp. 328–335.

[94] Nava Tintarev and Judith Masthoff. "Evaluating the effectiveness of explanations for recommender systems". In: *User Modeling and User-Adapted Interaction* 22.4 (Oct. 2012), pp. 399–439. ISSN: 1573-1391. DOI: 10.1007/s11257-011-9117-5. URL: https://doi.org/10.1007/s11257-011-9117-5.

[95] Carl Shapiro. "Optimal pricing of experience goods". In: *The Bell Journal of Economics* (1983), pp. 497–507.

[96] Yooncheong Cho et al. "The impact of product category on customer dissatisfaction in cyberspace". In: *Business Process Management Journal* 9.5 (2003), pp. 635–651.

[97] David N Laband. "An objective measure of search versus experience goods". In: *Economic Inquiry* 29.3 (1991), pp. 497–509.

[98] Patrick E Murphy and Ben M Enis. "Classifying products strategically". In: *Journal of Marketing* 50.3 (1986), pp. 24–42.

[99] Nava Tintarev and Judith Masthoff. “Over-and underestimation in different product domains”. In: *Workshop on Recommender Systems associated with ECAI*. 2008, pp. 14–19.

[100] Claudio Baccigalupo and Enric Plaza. “Case-based sequential ordering of songs for playlist recommendation”. In: *European Conference on Case-Based Reasoning*. Springer. 2006, pp. 286–300.

[101] Hao Liu, Jun Hu, and Matthias Rauterberg. “Music playlist recommendation based on user heartbeat and music preference”. In: *2009 International Conference on Computer Technology and Development*. Vol. 1. IEEE. 2009, pp. 545–549.

[102] Elad Liebman, Maytal Saar-Tsechansky, and Peter Stone. “Dj-mc: A reinforcement-learning agent for music playlist recommendation”. In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2015, pp. 591–599.

[103] Jesús Bobadilla et al. “Recommender systems survey”. In: *Knowledge-based systems* 46 (2013), pp. 109–132.

[104] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. “Performance of recommender algorithms on top-n recommendation tasks”. In: *Proceedings of the fourth ACM conference on Recommender systems*. ACM. 2010, pp. 39–46.

[105] Guy Shani and Asela Gunawardana. “Evaluating recommendation systems”. In: *Recommender systems handbook*. Springer, 2011, pp. 257–297.

[106] Agung Toto Wibowo. “Generating Pseudotransactions for Improving Sparse Matrix Factorization”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys ’16. Boston, Massachusetts, USA: ACM, 2016, pp. 439–442. ISBN: 978-1-4503-4035-9. DOI: 10.1145/2959100.2959107. URL: http://doi.acm.org/10.1145/2959100.2959107.

[107] Yifan Hu, Yehuda Koren, and Chris Volinsky. “Collaborative filtering for implicit feedback datasets”. In: *2008 Eighth IEEE International Conference on Data Mining*. Ieee. 2008, pp. 263–272.

[108] FO Isinkaye, YO Folajimi, and BA Ojokoh. “Recommendation systems: Principles, methods and evaluation”. In: *Egyptian Informatics Journal* 16.3 (2015), pp. 261–273.

[109] Qian Zhao et al. “Explicit or implicit feedback? engagement or satisfaction?: a field experiment on machine-learning-based recommender systems”. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM. 2018, pp. 1331–1340.

[110] Jürgen Buder and Christina Schwind. “Learning with personalized recommender systems: A psychological view”. In: *Computers in Human Behavior* 28.1 (2012), pp. 207–216.

[111] Sandra Clara Gadanho and Nicolas Lhuillier. "Addressing uncertainty in implicit preferences". In: *Proceedings of the 2007 ACM conference on Recommender systems*. ACM. 2007, pp. 97–104.

[112] Yehuda Koren. "The bellkor solution to the netflix grand prize". In: *Netflix prize documentation* 81.2009 (2009), pp. 1–10.

[113] Gábor Takács et al. "Matrix factorization and neighbor based algorithms for the netflix prize problem". In: *Proceedings of the 2008 ACM conference on Recommender systems*. ACM. 2008, pp. 267–274.

[114] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5 (1988), pp. 513–523.

[115] Nick Littlestone and Manfred K Warmuth. "The weighted majority algorithm". In: *Information and computation* 108.2 (1994), pp. 212–261.

[116] Allen Kent et al. "Machine literature searching VIII. Operational criteria for designing information retrieval systems". In: *American documentation* 6.2 (1955), pp. 93–101.

Appendix 1

BIJBEL VOEDING FFN

Karine Hoenderdos en Mary Stottelaar, diëtist

6 mei 2019

Inhoudsopgave

1. Uitgangspunten FoodFirst Network

Bij FoodFirst Network geloven we in de kracht van puur en onbewerkt eten en een gezonde leefstijl. Dit is de korte samenvatting:

1. We eten verse, onbewerkte producten. Zoals groenten, fruit, peulvruchten, volkoren granen, eieren, noten, kip, vis, zuivel en olijfolie. Deze producten vormen de basis van al onze recepten.
2. Er is een hoofdrol weggelegd voor plantaardige producten, zoals groenten, fruit en noten. Per dag eten we minstens 300 tot 400 gram groenten, 1-2 stuks fruit en een handvol noten.
3. We vinden vet een waardevolle voedingsstof. We kiezen bij voorkeur voor de natuurlijke, nauwelijks bewerkte vetten. Voorbeelden zijn olijfolie, vette vis, avocado, noten, volle zuivel, arachideolie en een beetje roomboter.
4. De voedingswaarde van elk recept wordt berekend en is – op verzoek – ook aan te klikken. We vermelden bij de receptuur wel de macronutriënten (eiwitten, vetten, koolhydraten, vezels), maar we communiceren niet in calorieën.
5. De recepten gaan uit van een gemiddelde eter.
6. We promoten een "rustig" eetpatroon van 3 hoofdmaaltijden per dag. Tussendoortjes zijn niet per se noodzakelijk, maar we hebben wel recepten en richtlijnen voor gezonde tussendoortjes en voor eiwitrijke tussendoortjes.
7. We hanteren bij FFN algemeen de 80/20 regel**.**
8. Haalbaarheid (in budget en tijd) is belangrijk. Daarom bieden we ook goedkope recepten en gebruiken we kleine hoeveelheden bewerkte producten als dat nuttig, goedkoop of tijdbesparend is. Voorbeelden zijn ingeblikte peulvruchten en vis, diepvriesfruit en – groenten, kant-en-klare mayonaise, currypasta, bouillonblokjes, brood.

2. Aanbevolen hoeveelheden volgens FFN

Per dag:

300 – 400 gram groenten
1-2 stuks fruit
1 tot 2 handjes noten
2-3 porties zuivel per dag
Eiwitrijke producten (vlees, vis, ei, tahoe) 125 gram
Volle granen naar behoefte*
Natuurlijke vetten naar behoefte, maar minimaal 30 tot 40 gram per dag*

Per week:

200 gram gare peulvruchten per week
1 tot 2 keer vis, minstens 1 keer vette vis

* de behoefte is afhankelijk van de activiteit, lichaamsgewicht en gezondheidsdoel.
Sporters zullen meer graanproducten nodig hebben dan mensen met diabetes.

3. Algemene richtlijnen maaltijden en dagmenu's

Warme maaltijd

De helft van het bord is gevuld met groenten, ¼ met eiwitbron en maximaal ¼ met koolhydraatbron

Minstens 200 gram groenten in de warme maaltijd, maar meer mag ook

Voor de koolhydraatbron geldt: bescheiden porties in de receptuur. Per persoon is dat circa: 100-150 g aardappelen / 75-100 g ongekookte pasta of 60-80 g ongekookte rijst, couscous, gierst. Wij kiezen voor de volkoren producten.

Variatie in weekmenu

Richtlijn eiwitbron: 1-2 x per week vis, 2-3 x vegetarisch, 1-2 x kip/kalkoen, 1-2 x rood vlees (rund, varken, lam etc)

Lunch

Altijd groenten bij de lunch

Minstens 100 gram groenten (kan ook een flinke tomaat of stuk komkommer zijn)

Niet te vaak en te veel brood: maximaal 2 sneetjes.

Ontbijt

Altijd iets van fruit of groenten bij het ontbijt

Niet te vaak en te veel brood: maximaal 2 sneetjes.

4. Toekenning persoonlijke content

Bij binnenkomst vullen mensen hun doelstellingen en een kort gezondheidsprofiel in (zie bijlage 1). Op basis daarvan krijgen zij persoonlijke content toebedeeld, met recepten, dagmenu's, video's en artikelen.
Voor de recepten en de ingrediënten maken we afkapwaarden en tags, zodat bekend is voor welke doelgroep ze geschikt zijn (zie bijlage 2). Als er nieuwe content is, zoals video's, recepten en artikelen, worden deze opnieuw gecodeerd. Dat geldt ook voor nieuwe ziektebeelden of doelgroepen.
We streven ernaar om iedereen voldoende content te bieden, maar hoe meer voorkeuren iemand invult bij binnenkomst, hoe minder keuze. Bij het laten zien van de keuzes worden ook minder geschikte recepten getoond, zodat iemand zelf een keuze kan maken (het Netflix principe).

Bijlage 1 INTAKE

Vragen bij binnenkomst:

Wat is je doel of wat zijn je doelen?

Ik wil gezonder eten en…

- Afvallen [A1]
- Gezond blijven [A2]
- Meer bewegen/fitter worden [A3]
- Beter slapen [A4]
- Ontspannen/minder stress [A5]
- Meer energie [A6]
- Betere stoelgang [A7] buikpijn hier weggehaald, dan is het puur vezelnorm
- Meer spiermassa krijgen [A8] aankomen hier weggehaald, want kan bij ondervoeding en dan wordt dit puur eiwit afkapwaarde
- Beter willen koken [A9]

Waar moeten we verder rekening mee houden?

- Ik eet vegetarisch [B1]
- Ik heb een allergie [B2]
  - Glutenvrij
  - Koemelk
  - Noten
  - En verder
- Ik heb een aandoening
  - Diabetes type 2 [B3]
  - Hart- en vaatziekten [B4]
  - Hoge bloeddruk [B5]
  - Ondervoeding [B6]
  - Kanker [B7]
  - Prikkelbare darm [B8]
- Ik ben een fanatieke sporter (minimaal 4x per week intensief)
- Het grootste deel van de dag breng ik zittend/staand/in beweging door (aanklikken wat van toepassing is)
- Ik ben zwanger
- Ik geef borstvoeding
- Ik zit in de overgang
- Ik ben/word geopereerd

- Ik heb weinig tijd om te koken [C5]

Ik ben:

- Naam
- Man/vrouw
- Leeftijd
- Lengte
- Gewicht (automatisch BMI berekenen)
- Middelomtrek <hierbij nog een pop-up hoe je dat moet meten>

Uit deze intake kan content op maat worden geselecteerd. Hiervoor maken we afkapwaarden op zowel receptniveau als op ingrediëntniveau. De afkapwaarden kunnen we gebruiken voor het selecteren van de content, voor het maken van challenges (bv hoeveel groenten eet je, hoeveel stappen zet je?) en voor het maken van weekmenu's.

Bijlage 2 Te noteren bij ingrediënten en recepten

Dit moet worden genoteerd bij elk ingrediënt:

	Afkapwaarde
Vegetarisch	Geen vis of vlees, dan “vegetarisch”
Vegan	Geen vis, vis, vlees, ei of zuivel, dan “vegan”
Allergenen	Geen gluten, noten, koemelk, ei et cetera, dan “glutenvrij” et cetera
FODMAP vrij	*Dit volgt nog (zie bronnen en onderbouwing)*
Zwangerschap	*Dit volgt nog (zie bronnen en onderbouwing)*

Dit moet genoteerd of berekend bij elk recept <u>per portie</u> (dus per persoon):
Energie (calorieën), in tientallen afgerond (dus 376 calorieën wordt 380 calorieën, ook op de website)
Eiwitten (in gram afgerond)
Vetten (in gram afgerond)
Koolhydraten (in gram afgerond)
Vezels (in gram afgerond)
Hoeveelheid groente (in tientallen gram afgerond)
Broodmaaltijd ja of nee
Tijd
Moeilijkheid
Zout/natrium (zout in tienden van een gram, natrium in milligrammen afgerond op tientallen)
Kalium (in milligrammen, afgerond op tientallen)
Maaltijdmoment
Chef
Gelegenheid
Seizoen

Bijlage 3 Afkapwaarden voor persoonlijk profiel

A1 = Afvallen

CHECK/VOORWAARDE: BMI ≥ 23

Totaal voor 3 hoofdmaaltijden per dag: maximaal 1700 kcal iets verhoogd

Afkapwaarde ontbijt: ≤ 400 kcal

Afkapwaarde lunch: ≤ 650 kcal iets verhoogd

Afkapwaarde diner: ≤ 650 kcal iets verhoogd

A2 = Gezond blijven

Geen richtlijnen/afkapwaarden

A3 = Fitter worden

Geen richtlijnen/afkapwaarden voor receptuur en dagmenu's

Extra content aanbod tussendoortjes en beweegvideo's

A4 = Beter slapen

Geen richtlijnen/afkapwaarden voor receptuur en dagmenu's

Extra content aanbod slaapartikelen, ontspanning en yoga

A5 = Ontspannen/minder stress

Geen richtlijnen/afkapwaarden voor receptuur en dagmenu's

Extra content aanbod slaapartikelen, ontspanning en yoga

A6 = Meer energie

Geen richtlijnen/afkapwaarden voor receptuur en dagmenu's

Extra content aanbod slaapartikelen, ontspanning en yoga

A7 = Betere stoelgang

Afkapwaarde vezels: streef naar 24 gram vezels per 3 hoofdmaaltijden, plus adviseer vezelrijke tussendoortjes

Ontbijt: 6 gram of hoger

Lunch: 8 gram of hoger

Diner: 10 gram of hoger

Tussendoortjes: 3 gram of hoger

- ⇨ Voldoet een maaltijd of tussendoortje aan de vezelclaim, dan kun je er "Rijk aan vezels" bij vermelden

A8 = ik wil meer spiermassa hier aankomen weggehaald en energieafkapwaarden. Eiwit verlaagd

Afkapwaarde: totaal voor 3 hoofdmaaltijden per dag: 65 gram eiwit + eiwitrijke tussendoortjes van ≥ 10 gram eiwit

Ontbijt: ≥ 15 gram eiwit dit verlaagd (was 25) want volgens mij niet haalbaar

Lunch: ≥ 25 gram eiwit

Diner: ≥ 25 gram eiwit

Tussendoor/snack: minimaal 6 gram eiwit (dit komt overeen met 1 ei). Aanraden dat men 2-3 eiwitrijke tussendoortjes neemt op een dag.

- ⇨ Voldoet een tussendoortje aan deze afkapwaarde, dan kan er "Rijk aan eiwit" bij staan

A9 = ik wil beter leren koken

Geen richtlijnen/afkapwaarden voor receptuur en dagmenu's

B1 = Ik eet vegetarisch kan met alles combineren

Afkapwaarde: alle ingrediënten van het recept hebben het kenmerk "vegetarisch" of "vegan"

- ⇨ Voldoet een recept aan het predikaat vegan, dan kun je er "100% plantaardig" bij zetten

B2 = Ik heb een allergie kan met alles combineren, maar de keuze daalt snel

Afkapwaarde: alle ingrediënten van het recept zijn vrij van het allergeen

B3 = Ik heb diabetes combi mogelijk met A7, maar wel lastig

Afkapwaarde: 75 gram koolhydraten voor de 3 hoofdmaaltijden

Ontbijt: ≤ 25 gram koolhydraten

Lunch: ≤ 25 gram koolhydraten

Diner: ≤ 25 gram koolhydraten

- ⇨ Bevat een recept minder dan 10 gram koolhydraten, dan kun je er "extra lowcarb" bij zetten

B4 = Ik heb hart- en vaatziekten kan met alles combineren

Afkapwaarde: Per 3 hoofdmaaltijden maximaal 5 gram zout én minimaal 3000 mg kalium

Ontbijt: ≤ 1 gram zout en ≥ 750 mg kalium

Lunch: ≤ 2 gram zout en ≥ 1000 mg kalium

Diner: ≤ 2 gram zout en ≥ 1250 mg kalium

B5 = Ik heb hoge bloeddruk kan met alles combineren

Afkapwaarde: Per 3 hoofdmaaltijden maximaal 5 gram zout én minimaal 3000 mg kalium

Ontbijt: ≤ 1 gram zout en ≥ 750 mg kalium

Lunch: ≤ 2 gram en ≥ 1000 mg kalium

Diner: ≤ 2 gram en ≥ 1250 mg kalium

B7 = Ik ben ondervoed <geen combinatie mogelijk met A1>

Afkapwaarde: totaal voor 3 hoofdmaaltijden per dag: minimaal 2000 kcal en 65 gram eiwit + eiwitrijke tussendoortjes van ≥ 10 gram eiwit

Ontbijt: ≥ 15 gram eiwit + ≥ 500 kcal

Lunch: ≥ 25 gram eiwit + ≥ 750 kcal

Diner: ≥ 25 gram eiwit + ≥ 750 kcal

Tussendoor/snack: minimaal 6 gram eiwit en 150 kcal

- ⇨ Voldoet een tussendoortje aan deze afkapwaarde, dan kan er "Rijk aan eiwit" bij staan

B8 = Ik heb kanker kan met alles combineren

Afkapwaarde: geen (omdat ondervoeding apart onderdeel is)

Wel extra content over kanker

C1 = Ik ben een fanatieke sporter (meer dan 4x per week intensief) bij combi met A1 daalt het energiepeil

Afkapwaarde: totaal voor 3 hoofdmaaltijden per dag: minimaal 2000 kcal en 65 gram eiwit + eiwitrijke tussendoortjes van ≥ 6 gram eiwit en 150 kcal. Bij combi met A1 wordt het maximaal 1700 kcal voor 3 hoofdmaaltijden.

Ontbijt: ≥ 15 gram eiwit + ≥ 500 kcal

Lunch: ≥ 25 gram eiwit + ≥ 750 kcal

Diner: ≥ 25 gram eiwit + ≥ 750 kcal

Tussendoor/snack: minimaal 6 gram eiwit en 150 kcal

- ⇨ Voldoet een tussendoortje aan deze afkapwaarde, dan kan er "Rijk aan eiwit" bij staan

C2 = Ik ben zwanger

CHECK/VOORWAARDE: alleen bij vrouwen + leeftijd tussen 18 en 45 jaar + niet zwanger + niet borstvoeding

Afkapwaarde: weglaten van recepten met ingrediënten die niet geschikt zijn voor zwangeren (zoals rauwmelkse kaas en gerookte zalm, zie bronnen en onderbouwing)

Geschikte content over zwangerschap en borstvoeding

C3 = Ik geef borstvoeding

CHECK/VOORWAARDE: alleen bij vrouwen + leeftijd tussen 18 en 45 jaar + niet zwanger + niet borstvoeding

Geschikte content over zwangerschap en borstvoeding

C4 = Ik zit in de overgang

CHECK/VOORWAARDE: alleen bij vrouwen + leeftijd tussen 40 en 60 jaar

Geen richtlijnen receptuur, wel content over hormonen/overgang/afvallen/hart- en vaatziekten

C5 = Ik ben/word geopereerd

Afkapwaarde: totaal voor 3 hoofdmaaltijden per dag: minimaal 2000 kcal en 65 gram eiwit + eiwitrijke tussendoortjes van ≥ 6 gram eiwit en 150 kcal

Ontbijt: ≥ 15 gram eiwit + ≥ 500 kcal

Lunch: ≥ 25 gram eiwit + ≥ 750 kcal

Diner: ≥ 25 gram eiwit + ≥ 750 kcal

Tussendoor/snack: minimaal 6 gram eiwit en 150 kcal

- ⇨ Voldoet een tussendoortje aan deze afkapwaarde, dan kan er "Rijk aan eiwit" bij staan

C6 = Ik heb weinig tijd

Afkapwaarde: alle recepten met ≤ 20 minuten bereidingstijd

Mogelijke combinaties

A1 (calorieën) is leidende afkapwaarde bij combinaties

A2 en A3 en A4 en A5 en A9 kunnen altijd met alles gecombineerd worden (geen afkapwaarden)

B2 (allergie) kan ook met alles, maar wordt wel meteen flink beperkt in keuze

B8 kan ook met alles

C6 kan ook met alles

C4 kan ook met alles

A1 + A7 = A1 + vezelnorm van A7

A1 + A8 = A1 + eiwitnorm van A8

A1 + A7 + A8 = afkapwaarde calorieen plus eiwit plus vezel

A1 + B1 + A7 + A8 = afkapwaarde kcal, vega, eiwit en vezel

A1 + B3: kcal en koolhydraten

A1 +

Moeilijk/niet te combineren

A1 en B7

A1 en C5

A1 en C2 (zwanger)

A1 en C3 (borstvoeding)

B3 en A7 (door koolhydraatbeperking daalt vezelgehalte, het kan wel maar de keuze zal laag zijn)

A1 en C1 (maar je kunt dan ook automatisch “doorschakelen” naar A1 en A8)

Bijlage 4: Allergie

We scoren bij FFN de recepten op de volgende 12 meest voorkomende allergenen:

1. **Koemelkeiwitallergie**

Geldt voor alle melkproducten, dus ook voor kaas, room, yoghurt, volle, halfvolle en magere melk, karnemelk, geitenmelk, schapenmelk, paardenmelk en ezelinnenmelk, geiten- en schapenkaas, chocolademelk, yoghurt, kwark, yoghurtproducten met en zonder suiker, fruitzuiveldranken zoals Taksi en Rivella, vla, pap, pudding, mousse, slagroom, zure room, crème fraîche, kaas, smeltkaas, smeerkaas, buitenlandse kaas, verse kaas, hüttenkäse, cottage cheese

2. **Glutenallergie/coeliakie**

Gluten: alle recepten waarin tarwe, spelt, kamut, rogge of gerst voorkomen.
Niet alleen brood, maar ook crackers, koek, gebak, bloem als bindmiddel, bakpoeder, couscous, bulgur, griesmeel, grutten, vermicelli, pasta, sauzen op basis van een roux, gebonden soepen. Bij glutenvrij dieet zijn haver(mout), quinoa en boekweit toegestaan, indien speciaal glutenvrij verpakt (vermeld op het etiket)> glutenvrij logo. Kant-en-klare producten (dus ook vleeswaren e.d.) moeten altijd gecontroleerd worden op gluten, geraspte kaas kan ook gluten bevatten.

3. **Allergie voor rauwe tomaat, paprika, wortel en rode peper**

Allergenenmix: rauwe tomaat, rauwe paprika, rauwe rode peper (nachtschade) en rauwe wortel. Liever ook niet de gedroogde variant gebruiken of deze moet mee verwarmd worden.

4. **Allergie voor steen-en pitvruchten**

Vruchten uit de categorie steen-en pitvruchten:
Aardbei, abrikoos, appel, appelbes, braam, framboos, kers, kweepeer, mango, mispel, nectarine, peer, perzik, pruim, rozebottel, kakifruit en kiwi. Deze fruitsoorten zijn vers niet toegestaan maar wel als ze verhit zijn geweest. Gedroogd liever ook niet gebruiken.

5. **Allergie voor pinda**

Ook geen pindakaas en pindasaus

6. **Allergie voor noten**

Alle soorten

7. **Allergie voor sesamzaad**

8. **Allergie voor soja**

Geen spoortje: geen sojasaus, sojamelk, sojaboon, tahoe, sojameel et cetera

9. **Allergie voor selderij**

Dit geldt voor blad-, bleek- en knolselderij, zowel rauw als verhit. Let op met kant-en-klaar smaakmakers, er zit bijvoorbeeld bijna in alle bouillontabletten selderij

10. **Ei**

11. **Schaaldieren**

Kreeft, krab, crayfish, garnalen et cetera

12. **Koriander**

Zowel rauw en gedroogd

Bronnen en onderbouwing

Aanbevolen hoeveelheden:

Richtlijnen goede voeding van de Gezondheidsraad

Richtlijnen PuurGezond

Energiebehoefte bij afvallen

Voedingscentrum (wij zitten wat hoger omdat we geen dieet willen voorschrijven)

Advies wel of niet afvallen

https://www.voedingscentrum.nl/nl/mijn-gewicht/heb-ik-een-gezond-gewicht.aspx (wij hebben de grens op 23 gezet, omdat veel mensen "vanzelf" al wat zullen afvallen als ze volgens FFN gaan eten)

Vezels

Darmgezondheid. De aanbeveling is 30-40 gram per dag. 9 op de 10 Nederlanders haalt dit niet. We hebben gekozen voor minimaal 24 gram uit de 3 hoofdmaaltijden. Met fruit en noten tussendoor kom je dan wel bij de 30 gram.

FODMAP

Als jullie FODMAP arme recepten gaan maken, dan kun je deze lijst aanhouden. http://www.fodmapdieet.nl/pdf/Lijst%20met%20FODMaParm-rijk.pdf. Het betekent dat je dan alle ingrediënten dan ook zou moeten scoren op FODMAP-arm. Je kunt er voor kiezen om dat nu al te gaan doen.

Zwangerschap

Voedingscentrum

Neem vanwege gevaar op besmetting met bacteriën (waaronder listeria) en de parasiet Toxoplasma gondi de volgende producten **niet**:

- Rauw vlees of vleeswaren gemaakt van rauw vlees
- Rauwe en gerookte vis
- Rauw ei
- Rauwe kiemgroente
- Kaas gemaakt van rauwe melk
- Rauwe melk rechtstreeks van de boer

Neem vanwege mogelijk schadelijke stoffen deze producten **niet**:

- Roofvissen zoals zwaardvis, haai en tonijn (ook niet uit blik of gerookt)

- Lever, leverworst of leverpaté
- Alcohol
- Kruidenpillen
- Kalebaskalk (pimba)

Bij sommige producten adviseren we om er **niet meer dan** een bepaalde hoeveelheid van te nemen:

- Maximaal 1 kopje koffie per dag
- Maximaal 2 keer vette vis per week
- Niet meer dan 1 à 2 kopjes venkelthee en anijsthee
- Geen overmatig gebruik van keukenkruiden
- Maximaal 2-3 dropjes per dag.

Ondervoeding

Bron: Stuurgroep Ondervoeding

In het algemeen wordt bij ziekte een minimale hoeveelheid van 1,2 g eiwit/kg lichaamsgewicht aangehouden.

Voor de eiwitopbouw is voldoende beweging essentieel. Verder is met name de hoeveelheid essentiële aminozuren in de voeding is van belang. Voor een goede anabole respons is ongeveer 10 gram essentiële aminozuren per maaltijdmoment nodig. Om die hoeveelheid te halen, is per maaltijd 20 gram hoogwaardig eiwit (dierlijk eiwit) nodig, of 25-30 gram van gemiddeld voedingseiwit.

Voor energie geldt dat dit persoonlijk moet worden berekend, maar een heel grove richtlijn is dat er 30% boven de normale energiebehoefte moet worden geteld. Voor vrouwen betekent dit een energiebehoefte van 2000 kcal naar 2600 kcal.

Omdat hier echt een individueel advies nodig is, hebben we een algemeen advies genomen van 2000 kcal uit de 3 hoofdmaaltijden en minstens 25 gram eiwit per maaltijd.

We adviseren dringend dat jullie mensen altijd adviseren naar een diëtist te gaan.

Spieropbouw/aankomen

Voor spieropbouw rekent men circa 20 gram eiwit per maaltijd, te verdelen over 4 maaltijdmomenten

https://www.frieslandcampinainstitute.nl/gezondheid/voeding-en-bewegen/de-theorie-hoeveel-eiwit-heeft-een-sporter-nodig/

ook wordt wel een hoeveelheid eiwit van 2,2 gram per kilogram vetvrije massa genoemd

https://www.menshealth.com/nl/voeding/a23099211/hoeveel-eiwitten-moet-je-eten-om-spiermassa-op-te-bouwen/

1,1 g eiwit per kg VVM staat ongeveer gelijk aan 0,9 g eiwit/kg lichaamsgewicht •

1,5 g eiwit per kg VVM staat ongeveer gelijk aan 1,2 g eiwit/kg lichaamsgewicht •
1,9 g eiwit per kg VVM staat ongeveer gelijk aan 1,5 g eiwit/kg lichaamsgewicht
Voor het gemak hebben we de afkapwaarden voor ondervoeding en aankomen/spieropbouw gelijk gesteld.

Vezelrijk en koolhydraatbeperkt eten

Het kan, maar het is lastiger omdat de graanproducten grotendeels ontbreken. Je moet dan wel werken met producten als lijnzaad, hummus en peulvruchten. Een voorbeeld:

Ontbijt

Schaaltje aardbeien	1,1
2 eetlepels lijnzaad	3,5
Handje noten	1,4
Yoghurt	

Lunch

2 tomaten	1,8
Stuk komkommer	0,6
25 g sla	0,3
2 opscheplepels kikkererwten	8,6
Feta	

Avondeten

200 g gegrilde groenten	3 g
100 gram bloemkoolrijst	2,2

Tussendoor

Worteltjes schaaltje	2
Hummus 2 eetlepels	2,2

Dan heb je 26,5 gram vezels binnen en 52 gram koolhydraten

Appendix 2

Ingredient	Gram
Snee brood	30
Paprika	140
Ei	55
Theelepel	3
Lente-ui	10
Banaan	100
Eetlepel	12
Munt (tak)	2
Appel	150
Bosje bieslook	5
Roggebrood	55
Sinaasappel	150
Mango	400
Volkoren cracker	15
Limoen	80
Granaatappel	150
Takje basilicum	2
Portobello	125
Takje tijm	2
Tomaat	150
Komkommer	400
Mozarella (bol)	125
Avocado	180
(rode) ui	70
Jalapeno	30
Mandarijn	30
Bosui	10
Nectarine	120
Takjes	2
Teen knoflook	5
Stengel bleekselderij	40
Saffraandraadje	1
Rode peper	15
Citroen	90
Laurierblad	1
Prei	110
Augurk	10
Bosje radijs	40
Vleestomaat	250
Bakje tuinkers	20
Little gem	300
Gemberwortel (per cm)	12
Snufje	1
Lollo rosso	160
(gele) Courgette	350
Koolrabi	200
Hollandse nieuwe haring	75
Sjalot	10
Witlof (stronk)	100
Puntpaprika	120
Ansjovisfilet	4
Paksoi	500
Venkelknol	200
Eidooier	18
Perzik	140
Vijg	32
Peer	150
Atsina cress	95
Abrikoos	50
Kardemompeul	2
Aubergine	280
Romeinse sla	400
Citroengras	10
Galia meloen	1000
Beschuit	10
Maiskolf	275
Romanesco	400
Mosterd cress	95
Madame jeanette	10
Kaneelstokje	2
Rijstwafel	7
Tortilla wrap	40
Boerenkoolblad	20
Kiwi	75
Rijstvel	7
Grapefruit	150
Pruim	30
Raddichio	100
Norivel	3
Passievrucht	15
Mini bolpompoen	400

Steranijs	2
Mini bloemkool	250
Broccoli	240
Kaki	125
Groene peper	15
Krop sla	360
Kokoscreme	200
Shanghai paksoi	200
Mini tortilla	25
Bosje bladpeterselie	10
Gedroogde abrikoos	3
Munt bosje	10
Basilicum bosje	10
Pita broodje	80
Snoepgroente bakje	400
Bospenen bos	600
Cherrytomaat	15
Koriander bosje	10