Machine Learning in Dinner Suggestions

DBM130

I.H.W. Beekmans
Eindhoven university of technology
Industrial Design
i.h.w.beekmans@student.tue.nl

L. C. de Jong
Eindhoven university of technology
Industrial Design
l.c.de.jong.1@student.tue.nl

M. Knust Graichen
Eindhoven university of technology
Industrial Design
m.knust.graichen@student.tue.nl

M.C. Rijkers
Eindhoven university of technology
Industrial Design
m.c.rijkers@student.tue.nl

ABSTRACT
The application (DDT) uses machine learning (ML) to help households explore their dinner meal possibilities and keeps track of one's preferences. DDT generates a personalized profile for the user by making a decision tree. DDT knows which recipes the user is most likely to enjoy and stimulates a healthier varied dietary pattern. DDT is designed for (young) adults, who cook on a daily basis for themselves or others, with a focus on student households. The main aim is to eliminate the seven 'safe meals' and create a repertoire of more diverse dinners. Based on a training period a first version of DDT is created for fourteen different participants. The possibilities and limitation of the realized application have been evaluated and possibilities for the future are explored.

AUTHOR KEYWORDS
Daily dinners; recommended system; machine learning; Artificial intelligence; varied dietary.

INTRODUCTION
Millions of Britons spend most of their lives eating the same seven 'safe meals'. Moreover, sixty percent eats the same dishes each week. One in four adults even prepares and eats the same meal on the same day [17]. On top of this a research among 3344 Dutch participants has shown that they know five meals by heart [15]. The Nutrition Centre of the Netherlands published the Wheel of Five. The Wheel of Five is a model on varied dietary pattern. In line with this model the Nutrition Centre advises to vary one's eating pattern [11]. It can be a challenge to find new (tasty)

recipes which support a varied dietary pattern as Comber [4] stated "Household food practices are a complex set of routines, including food purchasing, preparation, and consumption".

Daily Dinner Tip (DDT) aims at helping people in young households, between the age 18 till 35 years old, to maintain a varied dietary pattern. The application (DDT) uses machine learning (ML) to help households explore their dinner meal possibilities and keeps track of one's preferences. From the input of the user DDT generates a personalized profile for the user by making a decision tree. DDT knows which recipes the user is most likely to enjoy and help them to have a healthier varied dietary pattern. DDT is designed for (young) adults which cook on a daily basis for themselves or others, with a focus on student households. The main aim is to eliminate the seven 'safe meals' and create a repertoire of more diverse dinners. Taking both their busy and messy lifestyles as experience into account.

In this report, the concept of DDT and the final prototype will be explained. A closer look will be taken into the machine learning DDT uses. During two user study periods, fifteen participants used 'dinner tinder' in which they liked or disliked recipes. From the input of this study, personal decision trees were made which would predict whether the user would like or dislike certain recipes. The predictive ability of the current DDT version is evaluated and further possibilities for the future are discussed.

RELATED WORK

Almost everyone has dealt with a recommender system in one way or another. The most familiar ones are those from online webshops, whereby one gets personalized suggestions based on the bought products. The algorithm estimates the likelihood of other products in their database based on a number of factors including the interaction with the service, other members with similar interests and product information [1].

In today's busy society there is a growing market for services such as foodboxes [3]. This service stimulates healthy and varied dining with minimum effort required. By selecting some preferences online, a box with all the ingredients and recipes needed (for a week) is delivered at the user's doorstep [2]. One of the most known examples are HelloFresh boxes [12]. However, some people, especially non-regular subscribers, experience trouble with those type of boxes because of misalignment with delivery timing and their schedule [8]. Furthermore, many people experience those type of subscriptions as too expensive in comparison to 'general' grocery costs [16].

The more traditional way of selecting (new) recipes is by using cookbooks. However, these books don't offer the possibility to quickly filter information based on preferences. Currently there are hundreds of smartphone applications for searching recipes, which allow more searching and filtering functionalities [10]. Most of these applications work in a similar way; Users can select one or more ingredients, the type of dinner (starter, lunch, main dish etc.), the kitchen, certain diets or even the available time available to cook. Based on these preferences recipes are presented, usually ordered by rating. Some applications also have a form of user feedback, where one can score a dish, save favourites or like. Often these reviews are visualized with the dish as well, creating a sense of community feeling (for instance, Allerhande shows a five-star rating system with every dish [2]). Most of these functionalities are offered through websites as well, however applications are more phone-friendly and thus more appropriate for its context (supermarkets and kitchens). Whilst these recommender systems are frequently used the content is often not personalized to the preferences, routine and needs of one specific user.

Next to these many websites and applications there are a few physical (more personalized) designs. One example is SuChef—an in-kitchen display that shows a list of everyday meal suggestions to help users find cooking inspiration [12]. However often, similar designs are kept conceptual due to its high development and production costs. Resulting in various research prototypes and only a limited number of commercial products.

THE CONCEPT

The final prototype differs from the idealized product which will be referred to as the concept. The concept targets young adults aging from 18 till 35 years old, but people outside the target group can use the app as well. The concept will be installed as an app on the smartphone of the user. After installation, the user will set up their personal profile by filling in several questions about influential factors. The influential factors can be divided into biological (taste), economical (cost, income), physical (access, skills, time), social (culture, peers), weather conditions and psychological (mood) determinants [5,6]. The more or less static factors like kitchen appliances, and allergies will be used as hard filter for the dinner database. The user will never cook these meals anyhow, and so the potential recipes will be reduced beforehand. The other, more dynamic factors will function as a soft filter for the machine learning algorithm. The soft filter will be firstly introduced by a short questionnaire in which the user rates several recipes. Afterwards the soft filter receives the input for whether the user rejects or accepts the suggested recipes. The suggested recipes are based on the hard filter, the user's decision tree and their time available for that specific day to prepare the meal. Each day a recipe will be suggested on a preferred time. When a recipe gets declined by the user a new recipe will be suggested until the user finds a recipe which he/she wants to prepare that evening. The declined and accepted recipes will be considering by the decision tree to keep improving the tree and make it more accurate. The user can also save a recipe for later. Doing that will result with a pop-up questions why the user has done this, reasons can be: no time for it now, want to try it in the future, like it but do not want to eat it right now, etc. When a recipe is accepted, the app will open the recipe. The user can save a shopping list for which ingredients he/she will need. After having prepared the meal, the app will ask if the user enjoyed the meal and ask if he/she want to rate the recipe. This input will be used for the decision tree to make it more accurate.

REALIZED PROTOTYPE

"Garbage in, garbage-out" has plagued analytics and decision making for generations, this especially applies to machine learning. The quality demands of machine learning are steep, and need to meet exceptionally high standards, in order to guarantee its quality [13]. However, there is one mistake that can single-handedly ruin a machine learning algorithm, namely, overfitting. If an algorithm gets too complex or flexible (e.g. it has too many input features or it's not properly regularized) it can end up memorizing the noise instead of finding the underlying pattern. On the other-hand a machine learning algorithm can be underfit by providing too few features or regularizing too much. Typically, underfitting machine algorithms learn fast but are more biased towards wrong outcomes [18]. Contrariwise, complex machine learning algorithms tend to have more variance in their predictions and need more data in order to train properly. Error from bias can be reduced but might increase error from variance as a result, or vice versa [19]. Before going in-depth the report will discuss the kind of desired outcomes in order to select the most appropriate machine learning algorithm for the given problem.

There is a wide range of machine learning algorithms available, each with its own strengths and weaknesses. There is no single learning algorithm which works best on all problems. In order to select the most appropriate one, deliberate decisions need to be made. The first distinguishing between the different machine learning algorithms is supervised, unsupervised or reinforcement learning. Looking at the concept, it will

$$E(s) = \sum_{i=1}^{c} -Pi \cdot \log_2(Pi)$$

Formula 1: Calculate the entropy op the leaf node.

$$E(attribute1, attribute\ 2) = \sum_{c \in X} P\ (attribute1) \cdot E(like, dislike) + P\ (attribute2) \cdot E(like, dislike)$$

Formula 2: Calculate the entropy of two attributes.

output the probability of liking or disliking the recipe based on different categories. The goal is to approximate the mapping function to the extent that when new data is provided, the algorithm can predict the output based on the variables. The main reason to go for supervised learning is that algorithm tries to find the relationship between liking or disliking and the independent variables. Whereby, unsupervised learning is more about relationships between variables. Besides, when using unsupervised learning one has to have generally more data points in order to make it work.

The data consists of the recipes including all ingredients, and other variables like cuisine and cooking time. However, some machine learning algorithms need a certain degree of heterogeneity of the data. Support vector machines, linear regression, logistic regression, neural networks, distance (k-NN) methods all require that the input features (variables) be numerical and scaled to similar ranges in order to perform well. However, it is impossible to do this for the kind of meat in the dish. Obviously, it is not completely impossible, there is an inconvenient way to do this namely, create a binary categorical variable for each categorical data point. This inefficient chaotic way can be used but is not preferred. Secondly, some learning algorithms (e.g., linear regressions, distance-based methods) will perform quite poorly when features contain redundant information [18,19]. (e.g., highly correlated features). Lastly, the presence of interactions and non-linearities play a role in the decision making. When each feature makes an independent contribution to the end output, then machine learning algorithms based on linear functions generally perform better [9,19]. However, when there are complex interactions among the features, machine learning algorithms like *decision trees* and *neural networks* perform better. These algorithms have been built to discover these interactions.

But how do we decide, a recipe greater than the sum of its parts. It is the combination, and so interaction, between different ingredients that makes or break a dish. It is no surprise that the preference therefore goes to a machine learning algorithm that is built to excel these interactions. Due to the fact that the concept consists of categorical data (nominal) the preference goes to the decision tree algorithm. Moreover, a decision tree is easy to understand and make sense of. While, a neural network does not present an easily-understandable model. Both try to find the golden truth of the data-set.

Decision Tree

Before making the leap towards how to gather, store and clean the data, the functioning of a decision tree needs to be understood. For the sake of understanding, the decision tree will be explained with examples from the concept. A decision tree classification or regression model tries to break down a dataset into smaller and smaller subsets, while at the same time an associated decision tree is incrementally developed. The final results is a tree with *decision nodes* and *leaf nodes*. A *decision node* has two or more branches (e.g., beef, pork, chicken, fish and vegetarian). A *Leaf node* is the end result, in this case a binary decision namely like (1) or dislike (0). The *root node* is the topmost decision node, the one which has the most influence for the end result [5,7].

The core algorithm for building decision trees makes uses of two principles namely, entropy and information gain. The algorithm starts with dividing the dataset into different subsets that contain homogenous values. This is done by calculating the entropy of the leaf node, entropy can be calculated using formula 1. Secondly, it calculates the entropy of two attributes with formula 2. In both formulas P stands for probability of the attribute and the entropy of like vs dislike [14].

Information Gain

The information gain is based on the decrease in entropy (formula 1) after a dataset is split on a specific attribute. Constructing a decision tree is all about finding attributes that returns the highest information gain. This can be found by calculating both the entropy of one attribute, and then one without the attribute. The steps that need to be taken in order to calculate the information gain are as followed:

1. Calculate the entropy of the leaf node, using formula 1.
2. Split the dataset on all different attributes. The entropy for each branch will be calculated.
3. The information gain can be calculated with *gain(t,x) = Entropy(t) – entropy(t,x)*. The one which returns the largest information gain will be selected as decision node.
4. Divide the dataset by its branches and repeat the same process on every branch.

A branch with entropy of zero is a leaf node. A branch with entropy which is greater than zero needs further splitting. The algorithm is run recursively on the non-leaf branches, until all data is classified.

In order to prevent overfitting pre- or post-pruning can be used. Pre-pruning stops the growing of the three, before it perfectly classifies the training set. Post-pruning allows the three to perfectly classify the training set, and then post prune the tree. The algorithm makes use of post-pruning, meaning that it will reduce the tree, based on a defined threshold. More-over when the error-rate increases with the child (next branch), reject the child.

Data Cleaning

The influential factors can be divided into biological (taste), economical (cost, income), physical (access, skills, time), social (culture, peers), weather conditions and psychological (mood) determinants [5,6]. The more or less static factors like kitchen appliances, and allergies will be used as hard filter for the dinner database. The user will never cook these meals anyhow, and so the potential recipes will be reduced beforehand. The other, more dynamic factors will function as a soft filter for the ML algorithm.

The data needs to be cleaned in order to make sure that the machine learning finds the golden truth. However, when trying to do one will face different issues throughout the process. The first issue that arises is the trade-off between *bias* and *variance*. A learning algorithm with low bias must be *flexible* so that it can fit the data well. But if the learning algorithm is too flexible, it will memorize each data point. Therefore, the input data got mapped back to reduce the complexity and increase the bias of the algorithm [18]. Secondly the amount of training data available relative to the complexity of the machine learning algorithm. Due to the complex interactions among many different input features and different behaviours among them the algorithm needs a large amount of training data. *(e.g. Recipes consisting of beef and tortillas are been graded well, however together not)*. In order to let the algorithm train, a database with different recipes has been created. Due to feasibility reasons a dataset of 180 different dishes has been created. Thirdly, if the input feature vectors have very high dimensions, the learning algorithm will confuse itself and this causes a high variance. In order to avoid this problem, all nominal data will be mapped within categories. The problem here is when creating too small problems the dataset doesn’t have enough points to say something meaningful. However, when making the categories too big the groups don’t say something meaningful at all. The fourth issue is the degree of noise in the desired output values. The learning algorithm should not attempt to find a function that exactly matches the training examples. By asking the user to provide the reason why declining, we hope to force the user to think about their decision and therefore reduce the noise within their answers.

Figure 1: Screenshot 'DinnerTinder' application.

Considering these four elements the following mapping has been taken place. The recipes got translated into both nominal and ordinal data to increase the bias and lower the variance. Starting with the carbs (*potatoes, puff pastry, pasta, bread, tortillas, rice, etc.)*, followed by meat (*beef, pork, chicken, fish or vegetarian)*, cuisine (*Dutch, French, Italian...)*, dairy *(Cow cheese, goat cheese, crème and other-cheese)*, herb *(bouillon, none, dried, fresh and pre-mixed)*. On top of this there are some ordinal factors which are taken into consideration namely, spiciness (scale 0 to 4) and cooking time (scale 0 tot 6). On top of this some binary categories have been added to inform the machine about the leading ingredient. (e.g. mushrooms, beans, coriander). These leading ingredients are been added later on in the process. All dishes are stored in a .csv file, as an example a subset of this database (27 out of 183 recipes) can be found in appendix A.

R-environment

All decision trees are created with the aid of R. R is a free software environment for statistical computing and graphics. In order to create the decision trees, a standard library within the R-environment is used. The graphics are made with the aid of the R.plot library. The code (& walkthrough) can be found in appendix B.

USER STUDY

The conduct user study consisted of two parts, one training period and one evaluation period. The first part was conducted among fifteen participants. Due to too many missing data points one participant was excluded. Therefore, the evaluation study was conducted by only fourteen participants. The scope of this study focused on the 'soft filtering' element as described earlier. Participants and recipes were selected in such a way that the 'hard filtering' could be excluded (so no diets/ allergies and no need for 'special' cooking equipment).

During the training period each participant received six sets of fifteen dishes via a self-created web application. Within this 'dinner tinder' application the participants needed to swipe trough the presented dishes. Dishes were shown with a picture, the cooking time and list of ingredients (see figure 1). Each day the participant needed to accept (by pushing the like-button) or decline (via the dislike-button) fifteen different dishes. The answers were saved as a CSV-file, consisting of all 90 dishes and a like (1) or dislike (0) per participant. This dataset was used in order to train an individual decision tree for each participant. A subset of the generated data can be found in appendix C.

Based on these decision trees predictions could be made on whether that specific participant would like or dislike a certain dish. A dataset of 85 recipes (which were not part of the training set) was split into three subsets of 28 dishes. During the evaluation period all participants completed three new 'dinner tinder' of one subset per day. Based on the results of each day an iteration was made on the decision tree of each participant. An example of such a decision tree can be found in appendix D. In this way the learning capability of our algorithm could be evaluated. Different than before each 'dislike' was followed by an open question for explanation. These reasons could be used as an exploration for more categories in the dataset. On top of this a personalized survey was created per participant. Via this survey ten 'categories' were randomly selected from each last generated decision tree. Participants were asked to rate each category on a five-point-likelihood scale from 'Hate it' to 'love it'. See appendix E for one of the surveys (including results) All other surveys were similar to this example but with different categories. The results from these surveys were used to evaluate whether the found decision nodes correspond with the self-defined preferences of each user.

FINDINGS & DISCUSSION

Given the scope of this project all findings are mainly for explorative purposes. However, the generated decision trees and personalized predictions do suggest that the application serves it purpose (see appendix F).

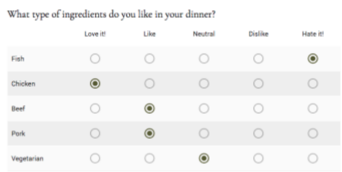


Figure 2: Snapshot survey result participant 2.

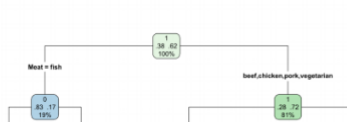


Figure 3: Snapshot generated decision tree participant 2.

The application works best for users who (strongly) dislike a whole type of food, for instance all types of fish. This claim can be confirmed by using the completed end surveys to do a spot check on the generated decision trees (data from appendix D and E). As an example, figure 2 shows a part of the survey results and figure 3 shows a snapshot of the decision tree of this participant. When comparing these figures, it can be noted that the user's opinion 'I hate fish' corresponds to 0.83 likelihood for rejection. However, the user's distinction of 'Loving chicken' and 'liking beef and pork' cannot be found extracted from the decision tree. Due to the scattered data and the small sample size it is impossible to do a meaningful statistical analysis of these results. But from an intuitive point of view the results indicate that the generated decision trees roughly correspond to the user's preferences.

When looking at the last set, some results indicate signs of overfitting. These results seem to be influenced by single recipes instead of the recognizable patterns in decision making. For future development this effect should be compensated for (via pruning). On top of this the selected categories, as described earlier, can be evaluated based on the given explanations for rejection. While some categories are very similar to the given reasons, some are slightly different or even completely new. For instance, we make use of the category 'fish', which is often indicated as a reason for rejection (92 times). However, some users indicate that they only reject a specific type of fish, eg. squid or codfish. Due to the chosen categories the developed prototype cannot recognize this type of preferences.

On the other hand, some categories can be found in both the user's reasons and the developed prototype. As an example, mushroom is given as a reason (11 times) and is one of the chosen categories. However, none of the generated decision trees uses this category as one of the decision nodes. As described earlier, one of the reasons for this might be that vegetables can often be replaced of removed when preparing a dish. Therefore, the presence of a specific vegetable, such as a mushroom, has less influence on the user's decision as the carb- or meat-type of that dish. Next to this the number of dishes including mushrooms in our database was rather limited, which makes it harder to recognize a pattern. It must be noted that the given reasons indicate more 'leading ingredients' than initially defined. For instance: lentils (18 times), avocado (5 times), eggplant (6 times), tomatoes (6 types) e.a. For future development these type of categories could be added, but a larger recipe database is needed for this.

Finally, the indicated reasons showed signs of some other possible categories as well. For instance, the way of preparing (deep-fried, cooked, cold); the costs or a specific combination (eg. two type of carbs together). Another reoccurring reason related more to the perception of a certain 'type of dish'. Some people do not believe that a salad or a soup can function as a proper dinner, simply because it will not be enough to saturate them. If a bigger recipe database is used these types of reasons might come up in the decision tree as indicated carb-type. Lastly many reasons were more intuitive 'it doesn't look nice' or 'I don't know these ingredients' or the like. This type of reasons will be very hard to find by the created prototype. However, there might be some underlying patterns among these dishes that the participant is not aware of.

FUTURE POSSIBILITIES

The current prototype was developed to be sufficient to be used for user testing, but there are still areas that need improvement for it to work like described in the concept.

In the user testing phase data was gathered from 14 users, which was used to develop personalized decision trees. A couple of results showed promise of an opportunity of clustering different user profiles to enhance the potential of the decision tree with less individual data points. When more data would be gathered the next step would be to see how many profiles can be created by clustering the personal decision tree.

One of the strengths of using the decision tree machine learning algorithm is that the intermediate steps that were taken are very visible. This attribute could be applied by letting the user see and edit his/her own decision tree to enhance and accelerate the learning period that is needed to create a decision tree that fits to the users own needs.

Next to the improvements for the machine learning another factor that could use improvement is the connectivity with the people around the user. Very often a person does not cook solely for him/herself but is eating together with other people in the household or friends. By combining the preferred attributes of every person that is joining the dinner the cook can make the dinner an enjoyable experience for all the guests.

CONCLUSION

In conclusion the created prototype shows promising results for the development of a personalized daily dinner suggestion system. Towards the end of the study some results showed signs of overfitting. For future development this could be compensated for by pre- or after pruning. While the current application is quite good at predicting personal food preferences, the predications are not yet time-specific. More research is needed on the temporary influences on one’s dinner preference, such as mood; weather; time available; day of the week and dinner company.

REFERENCES

1. Amazon's Recommendation Engine: The Secret To Selling More Online. (2018, August 28). Retrieved from http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/
2. Allerhande - recepten (2018, June 14). from https://www.ah.nl/allerhande/
3. C. (2016, March 09). Claire doet van TestAankoop: Foodboxen vergelijken. Retrieved April 18, 2018, from http://vreeverweg.be/claire-doet-van-testaankoop-foodboxen-vergelijken/
4. Comber, R. (2013). Food Practices As Situated Action: Exploring and Designing for Everyday Food Practices with Households. SIGCHI, CHI2013, 2457-2466.
5. Conner, M. T. (1993). Understanding Determinants of Food Choice: Contributions from Attitude Research. British Food Journal, 95(9), 27-31. doi:10.1108/00070709310045059
6. Eufic. (2016, June 06). The determinants of food choice. Retrieved November 31, 2018, from https://www.eufic.org/en/healthy-living/article/the-determinants-of-food-choice
7. Jeffcock, P. (2018, feb 01). Decision Trees in Machine Learning, Simplified. Retrieved January 30, 2019 from https://blogs.oracle.com/bigdata/decision-trees-machine-learning
8. Kiel J. (2018). Learning from the Veg Box: Designing UnpredictabilityinAgencyDelegationInCHI‘18 Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, Paper No.447 DOI: https://doi.org/10.1145/3173574.3174021
9. Le, J. (n.d.). KDnuggets. Retrieved January 30, 2019 from https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html
10. Live-in nanny. (2018). 10 iPhone Apps to Help You Choose What’s for Dinner. Retrieved April 18, 2018, from http://www.liveinnanny.org/blog/10-iphone-apps-to-help-you-choose-whats-for- dinner/

11. Nutrition Centre of The Netherlands. (n.d.). Achtergronden Schijf van Vijf. Retrieved December 1, 2018, from https://www.voedingscentrum.nl/encyclopedie/achtergronden-schijf-van-vijf.aspx
12. Palay, J. (2009). SuChef: An In-kitchen Display to Assist with "Everyday" Cooking. CHI 2009, EA09, 3973-3978.
13. Redman, T. C. (2018, April 03). If Your Data Is Bad, Your Machine Learning Tools Are Useless. Retrieved January 30, 2019 from https://hbr.org/2018/04/if-your-data-is-bad-your-machine-learning-tools-are-useless
14. Sayad, S. (n.d.). Decision Tree - Classification. Retrieved February 3, 2019, from https://www.saedsayad.com/decision_tree.htm
15. Schwartz, K. (2016, September 06). Niemand kookt korter dan de Nederlander. Retrieved June 13, 2018, from https://www.trouw.nl/home/niemand- kookt-korter-dan-de-nederlander~acb3d436/
16. Skwarecki B. (2017). Are Meal Kits Really Cheaper Than Groceries? (May 2017). Retrieved June 7, 2018 from https://lifehacker.com/are-meal-kits-really-cheaper-than-groceries-1794922297
17. Spillett, R. (2014, July 24). 60% of people eat the same seven regular meals every week. Retrieved December 1, 2018, from https://www.dailymail.co.uk/news/article-2703772/What-s-dinner-tonight-Lasagne-just-like-week-How-60-people-seven-regular-meals-everyweek.html
18. Wakefield, K (n.d). A guide to machine learning algorithms and their applications. Retrieved January 30, 2019 from https://www.sas.com/en_ie/insights/articles/analytics/machine-learning-algorithms.html
19. Wu, X., Kumar, V., Ross Quinlan, J. et al. Knowl Inf Syst (2008) 14: 1. https://doi.org/10.1007/s10115-007-0114-2

Appendix A: Subset of Recipe Database*

ID	Title	Carb	Meat	Cuisine	Spiciness	Dairy	Herb	coriander	mushrooms	beets	beans	cabbage	time
1	Hutspot Stamppot	potatoes	beef	dutch	0	none	boullion	0	0	0	0	0	2
2	Boerenkool Stamppot	potatoes	pork	dutch	0	none	boullion	0	0	0	0	2	1
3	Zuurkool hamrolletjes	potatoes	pork	dutch	0	cow-cheese	none	0	0	0	0	2	2
4	Andijvie Stamppot	potatoes	pork	dutch	0	none	boullion	0	0	0	0	0	1
5	Bietenquiche	puff pastry	vegetarian	french	0	other-cheese	fresh	0	0	2	0	0	5
6	Pasta Bolognese	pasta	beef	italian	0	cow-cheese	boullion	0	0	0	0	0	1
7	Risotto Kip	risotto	chicken	italian	0	creme	dried	0	0	0	0	0	2
8	Pasta Pesto	pasta	chicken	italian	0	cow-cheese	none	0	0	0	0	0	1
9	Pasta Geitenkaas	pasta	vegetarian	italian	0	goat-cheese	dried	0	0	0	0	0	1
10	Rodebieten stampot	potatoes	pork	dutch	0	none	dried	0	0	0	0	0	1
11	Courgette soep	bread	fish	french	0	none	dried	0	0	2	0	0	1
12	Ovenpasta	pasta	pork	italian	0	cow-cheese	pre-mixed	0	1	0	0	0	2
13	Shoarma pasta	pasta	beef	turkish	0	cow-cheese	dried	0	0	0	0	0	1
14	Quiche (v)	puff pastry	vegetarian	french	0	other-cheese	fresh	0	0	0	0	0	2
15	Lassagne	pasta	beef	italian	0	creme	dried	0	0	0	0	0	3
16	Wraps Kip	tortillas	chicken	tex-mex	1	cow-cheese	pre-mixed	0	0	0	0	0	1
17	Burritos	tortillas	beef	tex-mex	1	cow-cheese	pre-mixed	0	0	0	1	0	1
18	Burritos (v)	tortillas	vegetarian	tex-mex	1	cow-cheese	pre-mixed	0	0	0	2	0	1
19	Wraps Crunchy Chicken	tortillas	chicken	tex-mex	1	cow-cheese	pre-mixed	0	0	0	0	0	2
20	Visstick burgers	bread	fish	american	0	none	none	0	0	0	0	0	1
21	Broodje Gyros	bread	beef	greek	0	none	none	0	0	0	0	0	1
22	Broodje Shoarma	bread	beef	turkish	0	none	none	0	0	0	0	0	1
23	AVG	potatoes	pork	dutch	0	none	pre-mixed	0	0	0	0	0	2
24	Bami	noodles	chicken	indonesian	1	none	pre-mixed	0	0	0	0	0	1
25	Boboti	rice	beef	surinamese	1	none	pre-mixed	0	0	0	0	0	1
26	Nasi	rice	pork	indonesian	1	none	pre-mixed	0	0	0	0	0	1
27	Hamburgers	bread	beef	american	0	none	dried	0	0	0	0	0	1

*full dataset: https://docs.google.com/spreadsheets/d/1H_qgLWNzOXd3WuqcZWD5SkyTQMM-sum0owKswzKSjSg/edit?usp=sharing

Appendix B: Subset of recipe

1) download Rgui via https://cran.r-project.org/bin/windows/base/

2) instal extra library “rpart.plot” by
- opening ‘packaging’ menu
- click on ‘install packaging’
- Search for rpart.plot, select it and click on download

4) In RGui go to your file and Change Working Directory and select the location where your data is located.
5) Use the following code to create the decision tree

library(rpart)

library(rpart.plot)

data_train <- read.csv("user5.csv", TRUE, ";")
//user5.csv is filename

data_test <- read.csv("user5Test.csv", TRUE, ";") /
/user5Test.csv is filename

dim(data_train)

dtm<- rpart(results~., data_train, method="class")

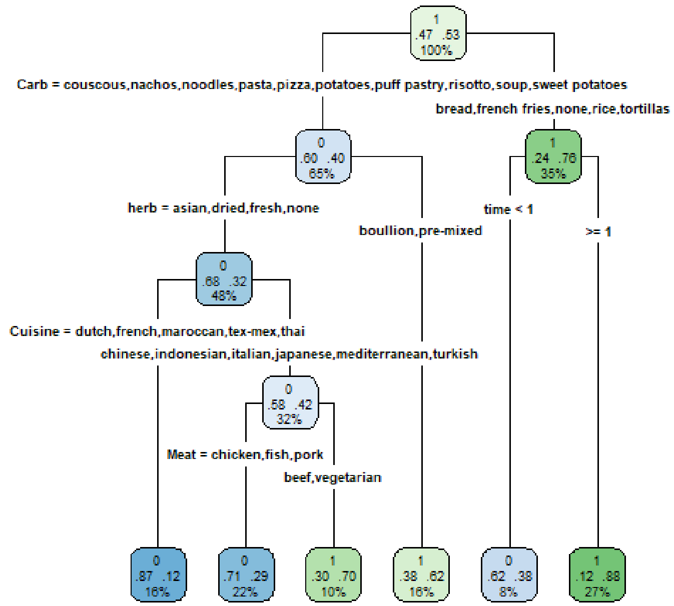
rpart.plot(dtm, type=4,extra=104)

Appendix C: Subset of Generated Data*

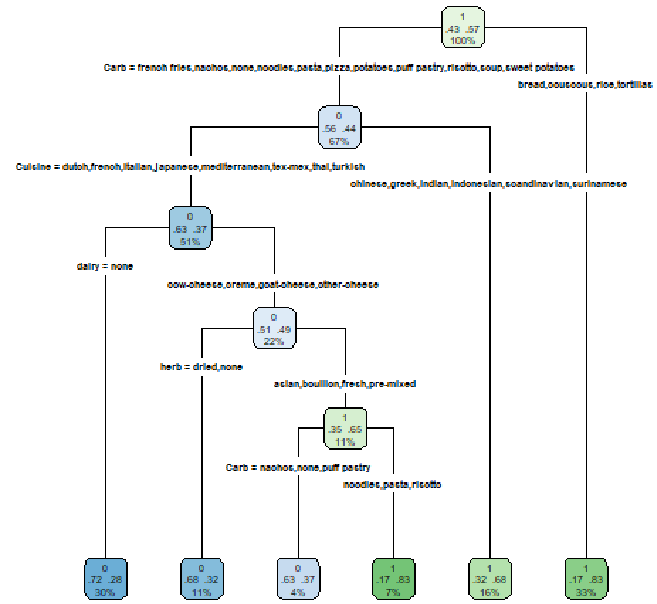
ID	Title	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant -	Participant 8	Participant 9	Participant 10
1	Hutspot Stamppot	0	0	0	1	0	0	1	-	0	1	0
2	Boerenkool Stamppot	1	0	0	1	1	1	1	-	0	1	1
3	Zuurkool hamrolletjes	0	1	0	1	1	1	0	0	0	1	1
4	Andijvie Stamppot	0	0	1	0	0	0	0	1	0	1	0
5	Bietenquiche	0	0	1	0	0	0	0	0	0	0	1
6	Pasta Bolognese	1	1	0	0	1	1	1	-	1	1	1
7	Risotto Kip	0	1	1	0	1	1	-	1	0	1	0
8	Pasta Pesto	0	1	1	0	1	1	-	1	1	1	1
9	Pasta Geitenkaas	0	0	1	0	0	1	1	-	0	1	1
10	Rodebieten stampot	0	0	0	0	1	0	0	0	0	0	1
11	Courgette soep	0	0	1	0	0	0	1	1	0	0	1
12	Ovenpasta	1	1	1	0	1	1	1	1	1	1	0
13	Shoarma pasta	1	1	1	0	1	0	0	-	1	1	0
14	Quiche (v)	0	1	1	0	1	1	0	-	0	0	1
15	Lassagne	1	1	0	0	1	1	1	1	1	1	1
16	Wraps Kip	1	1	1	1	1	0	1	-	0	1	0
17	Burritos	1	1	1	1	1	0	1	-	0	1	1
18	Burritos (v)	1	1	1	0	1	1	1	0	0	1	1
19	Wraps Crunchy Chicken	1	1	1	1	1	1	1	1	0	1	0
20	Visstick burgers	1	1	0	1	0	0	0	1	0	1	0
21	Broodje Gyros	1	1	1	1	1	1	0	1	0	1	0
22	Broodje Shoarma	1	1	1	1	1	1	0	1	0	1	0
23	AVG	1	1	0	1	1	1	0	1	1	1	0
24	Bami	1	1	1	0	1	1	1	-	0	1	0
25	Boboti	1	1	1	0	1	1	1	-	0	1	0
26	Nassi	1	1	1	0	1	1	-	1	0	1	0
27	Hamburgers	1	1	1	1	1	0	-	1	1	1	1

*full dataset: https://docs.google.com/spreadsheets/d/1iQKdCCyTTnD0TV8jLcm6pkZ5HnTvGTeugR_yg-pP2nY/edit?usp=sharing

Appendix D: Subset of Generated Decision Trees*



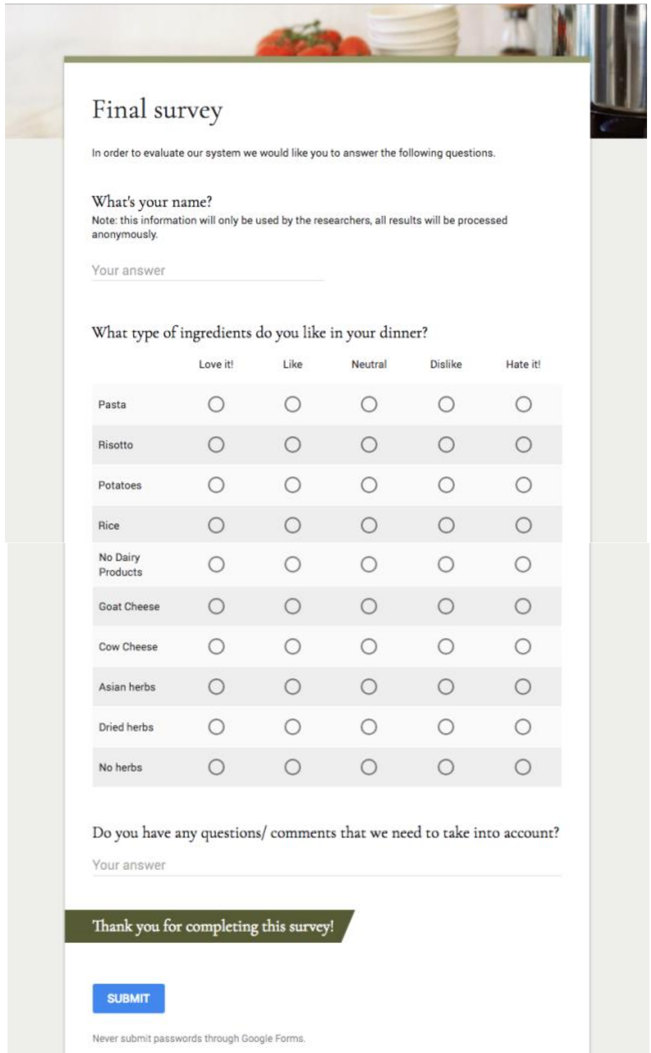
Decision Tree Participant 1, Iteration 1



Decision Tree Participant 1, Iteration 4

*full dataset: https://drive.google.com/drive/folders/1rPkyX3QkTrVIbofwdX8KU_3FFNogeazD?usp=sharing

Appendix E: End Survey (participant 1)*



Final survey

In order to evaluate our system we would like you to answer the following questions.

What's your name?
Note: this information will only be used by the researchers, all results will be processed anonymously.

Your answer

What type of ingredients do you like in your dinner?

	Love it!	Like	Neutral	Dislike	Hate it!
Pasta	○	○	○	○	○
Risotto	○	○	○	○	○
Potatoes	○	○	○	○	○
Rice	○	○	○	○	○
No Dairy Products	○	○	○	○	○
Goat Cheese	○	○	○	○	○
Cow Cheese	○	○	○	○	○
Asian herbs	○	○	○	○	○
Dried herbs	○	○	○	○	○
No herbs	○	○	○	○	○

Do you have any questions/ comments that we need to take into account?

Your answer

Thank you for completing this survey!

SUBMIT

Never submit passwords through Google Forms.

Pasta	Neutral
Risotto	Neutral
Potatoes	Like
Rice	Love it!
No Dairy Products	Neutral
Goat Cheese	Like
Cow Cheese	Like
Asian Herbs	Like
Dried Herbs	Like
No Herbs	Like

*full dataset:
https://docs.google.com/spreadsheets/d/1HOMz2KgO1Nk6d_EyCPHmqgA8SduXkURBZGDIcC5FIP8/edit?usp=sharing

Appendix F: Personalized Predictions*

decision 1		**0**	**1**
	0	8	7
	1	5	7
decision 2		**0**	**1**
	0	2	4
	1	8	13
decision 3		**0**	**1**
	0	1	9
	1	2	16

decision 1			**decision 2**			**decision 3**		
	0	*1*		*0*	*1*		*0*	*1*
1	0.1555556	0.8444444	1	0.2187500	0.78125000	1	0.7142857	0.2857143
2	0.1555556	0.8444444	2	0.2187500	0.78125000	2	0.6250000	0.3750000
3	0.1555556	0.8444444	3	0.4666667	0.53333333	3	0.3000000	0.7000000
4	0.1555556	0.8444444	4	0.2187500	0.78125000	4	0.8750000	0.1250000
5	0.1555556	0.8444444	5	0.7586207	0.24137931	5	0.7142857	0.2857143
6	0.2941176	0.7058824	6	0.2187500	0.78125000	6	0.8750000	0.1250000
7	0.1555556	0.8444444	7	0.2187500	0.78125000	7	0.8750000	0.1250000
8	0.1555556	0.8444444	8	0.2187500	0.78125000	8	0.1153846	0.8846154
9	0.1555556	0.8444444	9	0.7586207	0.24137931	9	0.8750000	0.1250000
10	0.1555556	0.8444444	10	0.7586207	0.24137931	10	0.6250000	0.3750000
11	0.2307692	0.7692308	11	0.7586207	0.24137931	11	0.8750000	0.1250000
12	0.2307692	0.7692308	12	0.7586207	0.24137931	12	0.1153846	0.8846154
13	0.2307692	0.7692308	13	0.2187500	0.78125000	13	0.8750000	0.1250000
14	0.2307692	0.7692308	14	0.2187500	0.78125000	14	0.8750000	0.1250000
15	0.2307692	0.7692308	15	0.9166667	0.08333333	15	0.1153846	0.8846154
16	0.8235294	0.1764706	16	0.2187500	0.78125000	16	0.1153846	0.8846154
17	0.2307692	0.7692308	17	0.7500000	0.25000000	17	0.1153846	0.8846154
18	0.8235294	0.1764706	18	0.2187500	0.78125000	18	0.8750000	0.1250000
19	0.2307692	0.7692308	19	0.9166667	0.08333333	19	0.1153846	0.8846154
20	0.8235294	0.1764706	20	0.3000000	0.70000000	20	0.8750000	0.1250000
21	0.4285714	0.5714286	21	0.2187500	0.78125000	21	0.1153846	0.8846154
22	0.1555556	0.8444444	22	0.2187500	0.78125000	22	0.3000000	0.7000000
23	0.1555556	0.8444444	23	0.7586207	0.24137931	23	0.3000000	0.7000000
24	0.1555556	0.8444444	24	0.7586207	0.24137931	24	0.1153846	0.8846154
25	0.1555556	0.8444444	25	0.2187500	0.78125000	25	0.3000000	0.7000000
26	0.1555556	0.8444444	26	0.2187500	0.78125000	26	0.1153846	0.8846154
27	0.1555556	0.8444444	27	0.2187500	0.78125000	27	0.3000000	0.7000000
28	0.2941176	0.7058824						

*full dataset: https://docs.google.com/spreadsheets/d/16EzyYAFbanD9PW9BcEzyKAdJYOD48aQ-J067f2BzUvY/edit?usp=sharing