Food Recommendations

Anooj Shah
Texas A&M University
College Station, Texas USA
anooj.shah@tamu.edu

Brian Lu
Texas A&M University
College Station, Texas USA
brian.lu32@tamu.edu

Abstract

Many are unaware of their diet and the nutritional values they are receiving in their meals. This may be about the individual nutritional content or also overall calories an individual consumes. There is quite a large range of literature about restaurant food recommendation systems. We wanted to take a slightly different view in this broad field by thinking about users trying to maintain healthy diets and cook their meals at home. Often times, users cook meals they do not enjoy or they do not know which new recipes they should try out. Our goal is to get a better understanding of the user's tastes and preferences to recommend food similar to the user's "flavor profile". Our plan is to first retrieve the user's taste by receiving input of various foods. The system will then output various recommendations that will provide the user with recipes they will enjoy.

1. Introduction

According to the Centers for Disease Control and Prevention (CDC), about 36.5% of U.S. adults age 20 and older and 17% of children age 2-19 were considered obese in America. Even today, the rate has not decreased. The CDC has classified this issue as a national epidemic. It is not a simple weight issue as it can cause serious health issues to the person's physical, metabolic and psychological health.

Overweight and obesity are defined by the Body Mass Index (BMI). It is calculated by dividing the weight (kilograms) by the square of the height (meters). Although BMI is not a perfect measurement, it allows us to gauge our current health.

Our belief is that people are only becoming obese because they do not know how to cook or they think cooking takes up too much time. Also the process of buying ingredients and putting together meals that you may not enjoy does not seem to be worth it for most people. Knowing that, people would consider ordering take out food for a cheap and quicker alternative to cooking. They could also be unaware of particularly healthy and delicious food. They may also think that exercise is the only way when it comes to losing weight, or they are simply unaware of their diet and the nutritional values they are receiving on a daily basis. Although there are several factors to consider, we consider these reasons to be the primary cause of obesity.

Often times, people think maintaining a healthy diet and eating food they enjoy cannot coexist. Our goal is to bring more light on the topic of food recommendations based on more than taste. We hope to eventually think about several parameters of age, weight, goals, and taste to determine the best recipes for different users. For example, every user has different goals - some want to lose weight while others want to maintain or gain weight. We want to recommend foods that not only meet a user's taste profile, but it should also help bring a user closer to their goals.

The objective is to develop a system that develops a "taste profile" that will predict what foods an individual may like. We can also calculate nutritional values based on the food the user has entered. Using this information we can recommend a food for the user that is tasty and healthy. We hope to make the food eating and health maintaining experience as friendly, effortless, and accurate to keep track of their meals and diet and help achieve one's goals.

For the purpose of this paper, we will mainly focus on the first aspect of determining tasty recipes for individuals that have started getting into cooking meals.

2. Preliminary Literature Survey

In the past few years, many researchers have developed systems to help with obesity. Many current systems use the technique of taking a picture of a food and then displaying the appropriate nutrition information. This is helpful in many ways such as users can now monitor the food they eat in an easy way. They can manage how many calories they are consuming and how much is their fat, sugar, cholesterol intake per day. However, in the advent of the fast food age, fewer people are concerned with health over taste. There has been minimal research in the field of taste and nutrition.

There are also numerous algorithmic approaches in this field of food recommendations. For example, some methods use content based approaches, collaborative filtering, classifiers, etc.

In terms of content based methods, we see this as a major player in the field of personalized and user's individual tastes. These recommendations can be done by looking at the individual ingredients that make up various foods. For example, say user A enjoys a food X. Food X has an ingredient G. Thus, it would be predicted that user B might also like food Y, which contains ingredient G as well.

Collaborative Filtering-Based systems have also been widely used, and the results have been promising. Freyne and Berkovsky tried both a nearest neighbor approach [5, 4, 3]. They found that the performance was quite poor compared to a content-based method. Additionally, Trattner and Elsweiler [7] found that the best performing Collaborative Filtering-Based methods were Latent Dirichlet Allocation and Weighted matrix factorization.

After researching these articles, we decided it would be best to focus in on a slightly different, but also related aspect of making decisions and recommendations for food. We will focus on delivering recipes and meals that match meals that the user enjoyed in the past.

3. Goal

Our goal is to provide a user with several meals and recipes that may match what they would be interested in. We hope to do this using data from many other users. This would allow the system to recommend the top ten recipes that a user may enjoy from the knowledge that other users that had similar taste profiles also enjoyed.

This will only be possible with a large data set that has the ratings of many users on various different recipes/meals.

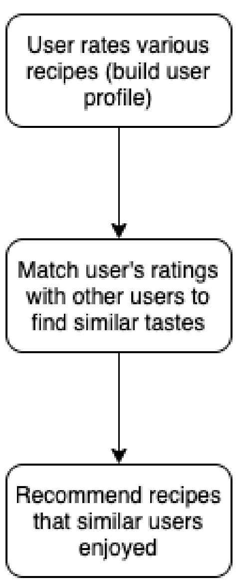

Figure 1. A general flow of the recipe recommendation

4. Methods and Approach

4.1. Data Sets

We will use two data sets, provided by Kaggle [2], that have been merged together to provide the necessary data. These data sets were from Allrecipes.com. This website was chosen as it contains one of the largest food specific social networks with over 1.5 billion visits per year. This data set used a web crawler to get 52,821 recipes from 27 categories. The data is from the years 2000 - 2018. The data set does not include recipes with zero ratings. The first data set had the following attributes: user id, recipe id, rating, and date. The other data set had the following attributes: recipe id, recipe name, ingredients, nutrition facts, etc.

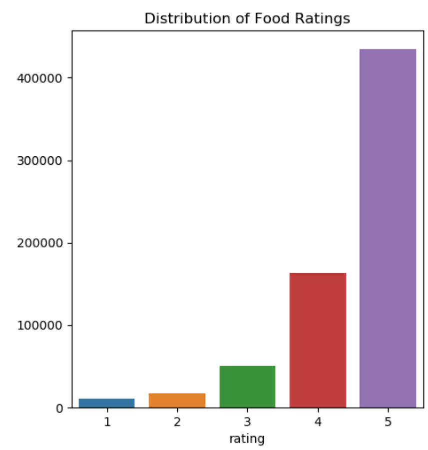

Figure 2. Distribution ratings among users

After merging the two data sets, the data set consists of the recipe name along with its recipe ID, a user ID that has reviewed that specific recipe, the user's rating on that specific recipe, and the list of ingredients associated with the recipe. This data will be useful for item-based collaborative filtering. Item based filtering is dependent on data from different users. For example, if Users A, B, C like food X and Y then when a User D enjoys a food X, they would receive a recommendation to also try food Y. This is because multiple users that enjoyed food X also liked food Y. This could potentially mean that the taste profile between these individuals are similar. The user rating distributions can be referred to in Figure 2.

4.2. Classifiers and Techniques

The technique and classifier we used are Collaborative Filtering and K-Nearest Neighbor in our system to develop the user's personal flavor profile.

Collaborative Filtering filters information by using the preferences of other users. Collaboration Filtering works primarily on the assumption that users have similar tastes

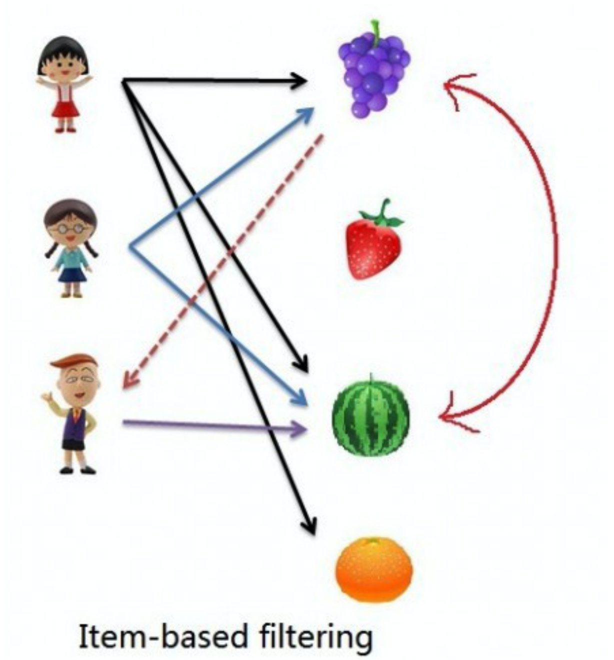

Figure 3. Collaborative Filtering: item based

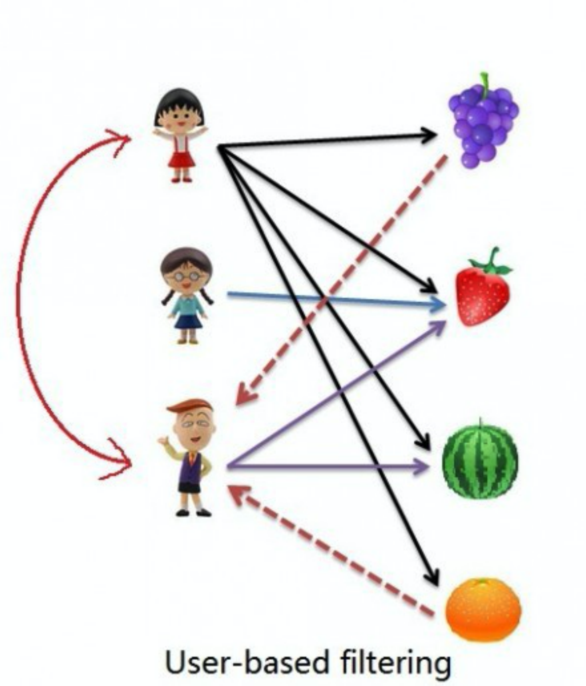

Figure 4. Collaborative Filtering: user based

if they liked the same foods/products. As shown in figures 3 and 4, there are 2 different types of collaborative filtering - User-based and Item-based. User based collaborative filtering measures the similarity between various users. Item based collaborative filtering measures the similarity between the ratings by various users [6]. In our approach we decided to use item-based collaborative filtering.

Figure 1 shows the overall workflow that we have chosen to proceed with using the item-based collaborative filtering approach.

We will build the actual classifier using the K Nearest Neighbors Algorithm. KNN (Kth nearest neighbors) is a simple supervised machine learning algorithm that can be used to solve both classification problems. KNN does this by assuming that similar data points exist closer to each other. Therefore, KNN captures similarity as a measure of distance.

We started of with reducing the number of parameters on the two original data sets. One data set specifically contained: the user ID, recipe ID, the user's rating, and the date last modified. The other data set contained: the recipe ID, the recipe name, the recipe's image URL, the recipe's ingredients, the recipe's cooking directions, and the recipe's nutrients. To provide the user with a recipe, we did not need the user's last modified review, the image URL, the cooking directions, nor the nutrients. So, we simply created a merged data set and wrote the data to a new data set.

We then dropped any user ID and any recipe ID duplicates if any. Doing this allows us to pivot the data set. Pivoting the data set reshapes the data based on column values and uses unique values from a specified index/columns to form axes of the resulting data set. We used the recipe name as it's index, columns as the user ID, and the values as the user's rating. With this we are able to tell how popular the current recipe is based on how many users have reviewed this recipe and what their ratings are, giving us a weighted value to allow the system to give a reasonable recommendation. A representation can be referred to Figure 3. We then stored the data and information from the pivot table into a matrix for our next classifier that allows us to give the appropriate recommendation.

We later thought about how we could weight different ratings. For example, a recipe with one 5 star rating should not be better than another recipe with a 4.9 rating. We decided to also do an analysis on the number of ratings for each recipe to remove any that were smaller than a certain pre-determined value.

We can use the pivot data set mentioned previously for using K-Nearest Neighbor to classify how popular the recipes are based on users' ratings. However, there is an issue with using K-Nearest Neighbor. K-Nearest Neighbor will classify a value based on how close it is to other node, its neighbors, by Euclidean distance. But with such large data set, Euclidean distance is unhelpful because all vectors are almost equidistant to the search query vector. Therefore, we used cosine similarity for nearest neighbor search.

Our initial approach as mentioned was using the Eu-

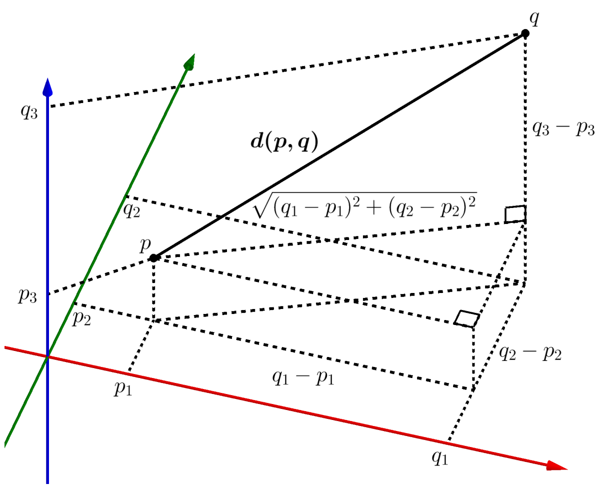

Figure 5. Euclidean Distance in 3D Space

clidean Distance. This is essentially a straight line distance between two points in a Euclidean Space. If the dimension is greater than 1, the straight line is simply the distance between two points in an n dimension plane space. We can extend this concept to any number of dimensions. Figure 5 shows an example of this.

The approach we ended up pursuing is cosine similarity. Cosine similarity is the measure of calculating the difference of the angle between two vectors. The length of the vectors do not matter for cosine similarity. From the formula in figure 6 [1], we can see that we normalize the vectors to a unit vectors and then calculate the inner product of the vectors.

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

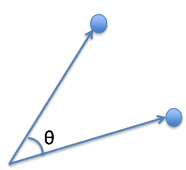

Figure 6. Cosine Similarity Formula

We can compare figure 6 to figure 7, which is the the formula for the Euclidean Distance.

$$D(a,b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2}$$

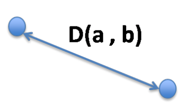

Figure 7. Euclidean Distance Formula

With that said, we used the approaches mentioned to then use the KNN classifier to test it on recipes.

4.3. Training and Testing

For testing, KNN (Kth nearest neighbors) does not require training. This is one of the advantages of KNN along with many others. KNN is simple and easy to implement. There is no need to build a model and tune parameters as needed. The algorithm is also versatile. It can be used for numerous use cases. However, one disadvantage is that the KNN algorithm does get slower as the number of predictors or independent variables increase.

The way we tested is we randomly selected a recipe from the data set and tested to see if any recipes were relatively similar to the one tested. There may be some loss of accuracy of this approach that we will consider in our future steps.

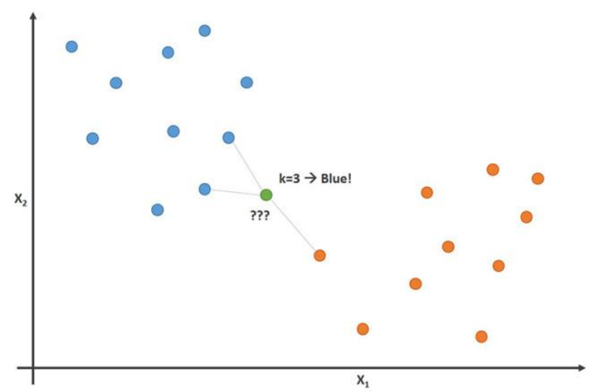

Figure 8. The distribution ratings among other users

4.4. Evaluation

There are also various methods to measure the performance of these algorithms. We can use Mean Absolute Error to calculate the error in the predictions.

The goal of the system is based on two main factors - a food item that a user would enjoy, and a food item that will complement the nutritional goals of a user. This would be done by using item-based collaborative filtering where the ratings of foods by other users can develop "taste profiles" which would help recommend certain foods to the user.

Additionally, we can populate multiple foods determined by the model and go down the list to find the one that closely matches the nutritional goals of the user. For example, we will test to see if our results are correct by using the same data set to predict the recipes for users that were already seen before. This will help test the accuracy of the "taste profile". Then we can go through the top foods and find what complements the nutrition for the day. For example, if you already went over your total calories for your day, you may want a low calorie dinner. Or perhaps you have consumed too much sodium for the day, you may want to consider eating something that has much lower sodium.

5. Overall Process

5.1. Initial Plan

Our initial plan was to provide a much more detailed recipe recommendation system. The original goal was to give the user a recommended recipe based on their flavor profile and maintain a healthy diet with proper nutrition. However, a lot of problems came along as we progressed into the project.

5.2. Finding the Necessary Data Set

Our attempts of find a large enough data set did provide us some trouble. Most of the data sets found on Kaggle did not provide the necessary data we were seeking. Most data sets that included both food and ratings were for restaurants. However, we knew that we wanted to reach the average user who will be cooking these meals at home. They did not contain the recipe ingredients nor the user rating. When we did come across a data set that did contain a user rating, it did not provide with another data set to follow the recipe name that came along with the recipe ID. We eventually came across one that contained what we needed with a few extra data. From that point on we need to simply reduce down the data and combine them to retrieve what we sought for. A separate program was created to break down the needed data and combine them. The data set now consists of: user ID, recipe ID, recipe name, the user's rating, and the list of ingredients.

5.3. Initial Approach and Conflicts

The user was to answer a series of questions that contained an image of the recipe along with its ingredients. The user simply either gave the displayed recipe a rating from 1 - 5 or gave it a thumbs up or a thumbs down. The system will take in those responses and use the recipe's nutrients as a method to categorize the recipe, we would label that recipe with a tag for the system to determine what kind of food the user likes, e.g. when sodium is greater than 500 mg, it would be considered 'salty' and would be tagged as so.

We also would define the proper and healthy nutritional diet according to the U.S. Food & Drug Administration (FDA) into the system as a guideline to properly give a balanced diet. Continuing from the user's responses, the system would have generated a very basic flavor profile for the user. The user would then be prompted to enter their calorie limit of the day. The system would use this information to recommend a recipe for the user and ensure the user stays withing the defined calorie limit as well as it gives the user a proper healthy diet. The user could then select the recipe they would like and provide the list of ingredients with its measurements and the cooking directions.

Unfortunately, we were too ambitious with our goal and realized the goal was not feasible within the given time constraint and finding all the necessary data sets to provide the proper recipe. We attempted to research about how to develop a flavor profile for the user as well as providing the correct recipe and concluded the approach was too difficult and needed a new method.

5.4. Breaking Down the Approach

With such a difficult scope, we decided to not consider proper diet and nutrients as a factor for our food recommendation system. Although we have lowered the scope, finding and researching a way to recommend a recipe while maintaining the user's calorie limit was not an easy feat. From this point, finding a way to implement and learn the user's flavor profile was also starting to prove too difficult. We had to consider each ingredient within the recipe and determine if it closely related to the flavor profile but also deemed it to be too difficult.

We came across an article describing the logistics behind a restaurant recommendation system and how it uses both content-based filtering and collaborative-based filtering with K-Nearest-Neighbor. We then decided this should be the bare minimum scope of our project. The plan from this point on was to use content-based filtering as a method to recommend food to the user. The user would enter in some ingredients they like and the system would output several recipes that contained that ingredient. However, this implementation was flawed as the system would return any recipe that contained that ingredient and not a recipe that contained a similar ingredient. We found that this did implementation would not be beneficial for us to achieve the goals and accuracy we were wanting. The final approach was to use collaborative-based filtering along with K-Nearest-Neighbor.

5.5. Solution and Testing

Since the article mentioned previously had used both collaborative-based filtering and K-Nearest-Neighbor for restaurant recommendations, this was our final solution. New data sets had to be found with the proper data and had to implement a similar concept with that data (as stated in section 4). The following is some sample outputs.

```
Recommendations for Recipe: 'Seafood Sandwich' on priority basis:

1: Black Friday Pie
2: Creamy Ham and Beans
3: Joe's Favorite Hamburgers
4: Awesome Crab Soup
5: Marinated Mushrooms II
6: Old German Honey Cookies
7: Best Scrambled Eggs Ever!
8: Richard's Breakfast Scramble
9: Hot and Tangy Broccoli Beef
10: South-of-the-Border Stuffed Peppers
```

```
Recommendations for Recipe: 'Lemon Horseradish New Potatoes' on priority basis:

1: Oriental Cold Noodle Salad
2: Corned Beef Hash
3: New England Bean Dip
4: Shannon's Smoky Macaroni and Cheese
5: Tomato Soup III
6: Chinese Chicken Salad III
7: Garlic Linguine
8: Soft Garlic Parmesan Breadsticks
9: Adult Watermelon for BBQ's
10: BBQ Salmon over Mixed Greens
```

```
Recommendations for Recipe: 'Ronaldo's Beef Carnitas' on priority basis:

1: Potato and Pork Bake
2: Stuffed Pepper Soup III
3: Hugh's Dry Rub
4: California Chicken Soup
5: Bunwiches
6: Pork and Pepper Stew
7: Tex-Mex Burger with Cajun Mayo
8: Yummy Honey Chicken Kabobs
9: Pepperoncini Beef
10: Pork Chops with Italian Sausage
```

As we tested our data set, we realized there is an issue. With such a large data set, pivoting the entire data set will cause the system to run what seems like forever. The training data set size was then reduced to about 10% of the original training data set and has completed the recommendation within a fair amount of time. We have decided not to push it too much further as it could take a while to run a good portion of the data. For testing, we used the same data set to see if any of the recipes are similar to the one given. Because we have no method of validating if the ten suggested recipes are indeed similar, the measured accuracy of the system is unknown. Currently, assuming the recipes are relatively close to each other, the actual accuracy of the recipe recommendation could be fairly low.

6. Future Steps

Our system uses user ratings to recommend the user a recipe based on how similar the inputted recipe is and how popular it was for other users of similar tastes. However, we would like to create what we had in mind with the initial approach.

Currently, there is no method of receiving an input from the user of what recipes are similar to what they have entered. To add on to it, there is also no method of providing the user to know what recipes are currently within the data set and is limited what is only within the data set. In the future, we first plan on having an interface with the recipe names and allow the user to select that recipe and have the system recommend a similar recipe. With this approach, we are able to provide the user with some knowledge of what recipes are available within the data set and have the system be more flexible with the user.

Our next step would be to allow the user be even more flexible with their options. Instead of inputting a recipe, perhaps the user wants any ingredient containing a specific ingredient or a few. Our system currently does not support this approach, but we would like to incorporate this feature as well.

Additionally, we would like to develop a continuing taste profile for users. For example, a user would choose a recipe that they enjoy. Then, the system would recommend a meal/recipe that they may enjoy. However, if the user says they already know they do not enjoy that food, the system should adjust its parameters to understand that. This would not be possible using the KNN approach we used. This is because in KNN, we choose the smallest ”distance” point to see what it is similar to. However, this may not always work. Individuals may have different preferences.

One aspect related to developing a taste profile is using the actual ingredients. In our system, we did not analyze the individual ingredients to see that certain ingredients may be why certain users do not like certain foods. The idea is that certain ingredients can be substituted easily and if the user inputs that they do not like certain ingredients, the system should recommend a substitution on a certain ingredient.

Our final end goal is to develop a system that learns from all the recipes and meals, to understand which ingredients go well together. This would allow for a more advanced approach (as mentioned earlier) where a user can input ingredients that they have available and the system would recommend certain meals that could be made from those ingredients that also match their specific taste profile.

Figure 9. Interface Sample

We can also go deeper by creating a flavor profile with tags, as mentioned previously, and take that a step further. There can be a method of providing a flavor profile of specific ingredients and the system will recommend the appropriate recipe. For example, if a user likes foods with pumpkin seeds mixed with blueberries as ingredients, more recommendations will come to individuals that enjoy this combination of ingredients. We hope to also look into substituting ingredients for commonly substituted ingredients. For example, if a user is allergic to peanuts, we would not want to discard all the recommendations of foods with peanuts. We could rather create a recipe to replace the ingredient the user is allergic to with another ingredient. From the example above, the user may not be allergic to cashews and cashews could be a good substitute for any recipe with peanuts.

References

[1] Carleton CS Computer Science Comps Project Netflix Prize - K Nearest Neighbors.

[2] Elisa. foodrecsys-v1, 2018. foodRecSys-V1 Kaggle Data Set from Allrecipes.com.

[3] Berkovsky S. Smith G Freyne, J. Recipe recommendation: Accuracy and reasoning. page 99–110, 2011. In International conference on user modeling, adaptation, and personalization.

[4] Berkovsky S. Baghaei N. Kimani S. Smith G. Freyne, J. Personalized techniques for lifestyle change. page 139–148, 2011.

[5] J. Berkovsky Freyne. Intelligent food planning: Personalized recipe recommendation. pages 321–324, 2010. In Proceedings of the 15th international conference on intelligent user interfaces.

[6] Onel Harrison. Machine learning basics with the k-nearest neighbors algorithm. 2018.

[7] D. Trattner, C. Elsweiler. Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems. pages 489–498, 2017. In Proceedings of the 26th international conference on world wide web.