Recipe Prediction - Kitchen Assistant

Introduction to Natural Language Processing
AIT 526-004
George Mason University

Group 3
Kasey Howlett
Sai Sushmitha Kadambari
Chi Quinn
Evans Sarker

October 21, 2023

Abstract

Since early 2021, the world has been facing steady inflation. As a result, grocery prices are steadily increasing. In addition, choosing what to make for dinner, with an abundance of options, can lead to decision fatigue. These challenges create an avenue for a solution to alleviate decision fatigue while also reducing spending. This project utilizes natural language processing and cosine similarity to provide suggested recipes based on pantry stock. A corpus of recipes from the Epicurious website contains recipe titles and ingredients. The data is preprocessed using several NLP techniques such as lemmatization, tokenization, and pos tagging. A given list representing pantry stock is compared iteratively against all existing data in the corpus to provide a Top 5 list of most probable recipes. Testing involved comparing an existing recipe from the original corpus to the entire corpus for accuracy. The value provided by this project includes a working system for providing recipe recommendations for a given user. The working system makes the most of available ingredients while also taking away the need to make decisions from a user.

Key Words

Natural language processing, Machine Learning, Cosine Similarity, Python, Recipe, Ingredients

Introduction

Deciding what to have for dinner is a daily task. It is reasonable that one might develop decision fatigue about choosing what to have night after night. Decision fatigue "occurs when we're overwhelmed by the number of choices we have to make throughout the day," (Pignatiello). In addition, since early 2021, the world has been facing a steady increase in inflation. As a result, picking what to eat is also hindered by a need to monitor spending. This opens the avenue for machine learning to be utilized to create a solution to make the average person's day more manageable.

The intent of our project is to utilize machine learning coupled with natural language processing to provide a tool that will make the task of choosing what to cook for dinner, or any meal, easier. The idea is that a person would be able to maximize their existing pantry to choose what to cook. This could enable a person to relieve themselves of decision fatigue while also utilizing their pantry more effectively, reducing spending. Our group will be utilizing a large source of data from the Epicurious website, a popular source of dinner recipes to create a tool that will predict possible recipes given recipe ingredients.

Related Work

Throughout the years, multiple studies have explored the intersection between machine learning techniques and natural language processing in predicting recipes from a given set of ingredients. This section provides an overview of prior and continued efforts in the field, shedding light on how various researchers have leveraged these technologies to forecast recipes based on ingredient lists.

In the study "Machine Learning-Based Food Recipe Recommendation System" by M.B. Vivek, N. Manju, and M.B. (2017). The team explores the application of machine learning and collaborative filtering techniques to predict food recipes based on user preferences and ingredient lists. The authors use two approaches: item-based and user-based recommendation systems. They discuss the significance of collaborative filtering and content-based approaches and suggest the potential for hybrid recommendation systems. Lastly, the study significantly contributes to understanding machine learning for addressing recommendation challenges and making improvements to model performance.

Galanis and Papakostas (2022) also proposed a similar approach, essentially they explore the generation of creative cooking recipes with input ingredients and cooking instructions. However, they used Seq2Seq models and Transformer-based architectures to improve recipe generation. Their work also focuses on generating recipes with specific attributes like cuisine, dietary restrictions, and serving size. They emphasize the importance of fine-tuning models with cooking datasets to eventually generate recipes. Moreover, through their work, the authors showcase the progress and challenges in Machine Learning (ML) and Natural Language Processing (NLP) approaches for the generation of recipes.

The paper titled, "ChefAI.IN: Generating Indian Recipes with AI," (Chaudhary et al., 2022) employs ML and Artificial Intelligence (AI) approaches to generate new Indian recipes. The authors noted a lack of AI and ML approaches in culinary applications. Although recipe

generators exist, not many focus specifically on Indian cuisine. Therefore the study employs an evolutionary genetic algorithm to generate new indian recipes. They also created a web application to generate recipes based on ingredients the user selects. Furthermore, this study demonstrates the potential of AI/ML in creating cuisines that can be diverse and it paves the way for automation in this sector.

Majumder et al. (2019) developed models that can generate personalized recipes for users with incomplete ingredient knowledge. They deployed historical user preferences extracted from consumed recipes to achieve personalization. The main idea is that their models use a specific encoder-decoder architecture with attention mechanisms. Their models are better than basic models at making recipes that match a user's preferences and are easy to follow. Additionally, their work contributes to the advancement of AI-generated recipes and possibly opens the door for future improvements in the modeling.

The related works mentioned above, out of countless other applications, have significantly contributed to the development of recipe modeling with the help of NLP and ML. By building on the progress made by the mentioned studies, the present research looks to contribute to the field of AI-generated recipes. While the above studies focus on different ways to deploy ML and NLP, such as recommendations, generations, and personalizations, this project looks to simplify meal decision-making and optimize the use of ingredients generally found in pantries. Essentially the goal is to aid decision making when selecting recipes and reducing the spending for users.

Objective

This tool is designed to address the common challenges individuals face in meal planning and decision-making when it comes to cooking. The primary goals of our project are to alleviate decision fatigue and mitigate the impact of rising food costs by providing users with a practical, data-driven solution.

The central focus of our project is the creation of a robust recipe prediction system. This system will empower users to make informed choices about what to prepare for their meals, be it dinner or any other culinary endeavor. Users will input a list of available ingredients from their pantry, and our tool will respond by generating personalized recipe recommendations that align with their ingredient inventory.

By undertaking this project, the ultimate objective is to simplify the meal planning process for individuals by offering a practical and user-friendly solution that maximizes the use of available ingredients while reducing decision fatigue. We believe that our innovative kitchen assistant tool will enhance the daily lives of users, making meal preparation more convenient and cost-effective.

Selected Dataset

The food ingredients and recipes dataset was discovered through Kaggle’s online platform (Goel et al., 2019). The dataset was originally scraped from the Epicurious website (Epicurious, n.d). Epicurious, created in 1995, is a subsidiary of the American media company,

Condé Nast becoming one of the first websites for food resources (Carmody, 1995). The website contains thousands of recipes of different types of cuisines of meals.

The processed dataset contains three columns and 13,480 rows of food and drinks: title, ingredient, and instructions. The title column is the named dish or drink referenced from the website. The ingredient column lists measured items for food preparation. The Instruction column describes the direction of food preparation in a sequential format.

Preprocessing Dataset

The data type of each column in the dataset is text-based. None of the data are categorical or numerical. Preprocessing the dataset using natural language processing techniques includes transforming all strings to lowercase, tokenizing and removing stop words, punctuations, numerical values, or the measurements under the Ingredient columns. Lastly, we applied the Part-Of-Speech tagging from the Spacy package to extract non-noun tokens (Honnibal, 2017).

Exploratory Data Analysis

When examining the recipe dataset, a few things were noted. First about 72% of the recipes included salt compared to 46% of recipes containing sugar (figure 3). Second, most of the recipes contain about 10-15 ingredients (figure 4). To truly evaluate the types of recipes was limited in our study since the dataset did not contain any attributes describing the recipe including the type of cuisine (by region, country, etc.), dietary restrictions (vegan, dairy-free, etc.), and user ratings.

The word cloud (figure 1) returns tokens including measurement. “Tablespoon oil”, “kosher”, “salt” “cup”, and “sugar” (cup of sugar) indicated they are frequently mentioned in most recipes. Further removing measurements text (cup, tablespoon, teaspoon) from the list returns mostly flavoring ingredients (figure 2). The frequency of the most ingredients in the dataset, in Figure 3, implies most recipes are rich in flavor since most ingredients consist of salt (9659 recipes, 72% of recipes list), oil (9023, 67%), and sugar (6179, 46%). Applying bigrams and counting the frequency, returns a more detailed ingredient list. Salt is divided into two types: kosher salt (4757, 35%) and sea salt (1286, 10%). Lemon juice (1973, 15% of recipes), stick butter (stick of butter) (1544, 11%), and “extra-virgin oil” (1085, 8%) continue as the most referenced ingredient. Applying trigrams to the ingredients list returns mostly seasonings. But apple cider vinegar (268, 2%), leaf tender stem (217, 2%), and vanilla ice cream (195, 1%) are the most trigram words listed in the “Ingredients” column dataset

Figure 1: Word cloud, including measurements

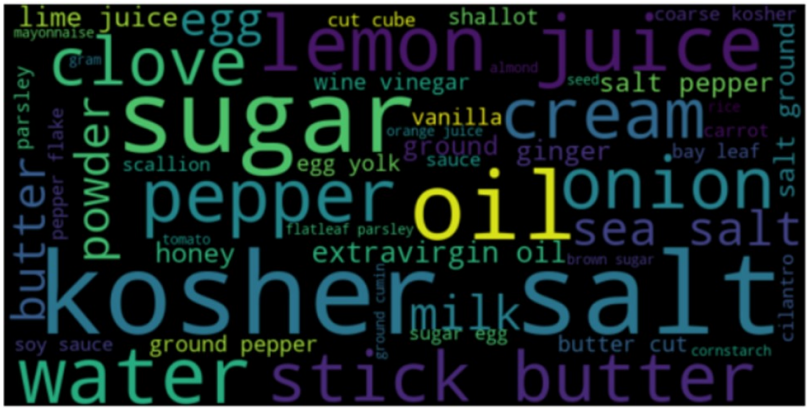


Figure 2: Word cloud, without measurements

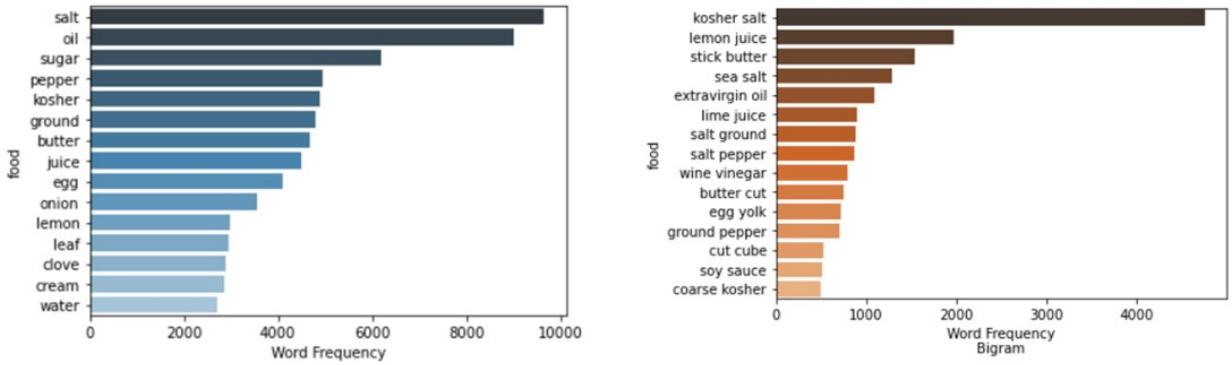
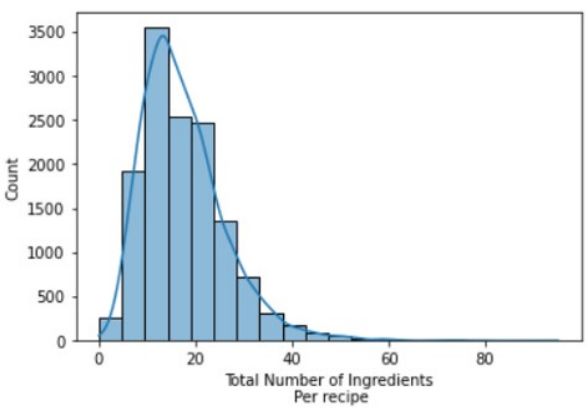


Figure 3: Word frequency of ingredients: unigram and bigram

The Kernel density in the histogram (figure 4) displays an uneven bell-shaped curve, positively skewed. This indicates there are more than 20 tokens or a potential list of ingredients per recipe. The peak falls around 10-14 tokens per recipe. The boxplot (figure 5) further shows there are many recipes with over 40 ingredients or tokens. One recipe includes 95 tokens. Whereas, 16 recipes only include one ingredient or token.



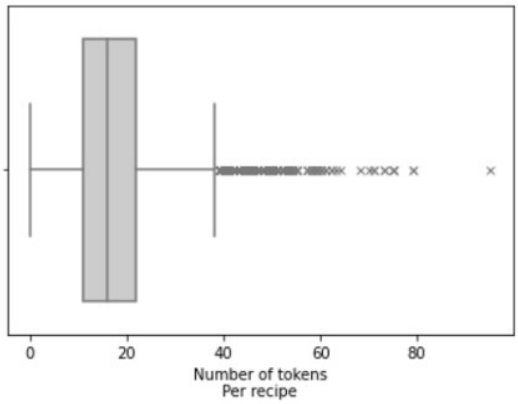


Figure 4: Histogram of the total number of ingredients, per recipe; Figure 5: Boxplot of tokens per recipe

Proposed System

A high-level overview of our proposed system architecture can be seen below. For this project, our intent is to focus more on the functionality of the tool, so there is no intention for a working user interface. Our intended system architecture includes a user providing input to the recipe prediction tool in the form of food items/ingredients. The back end of the tool will include a data store of known recipes and the working code that will generate recipe predictions.

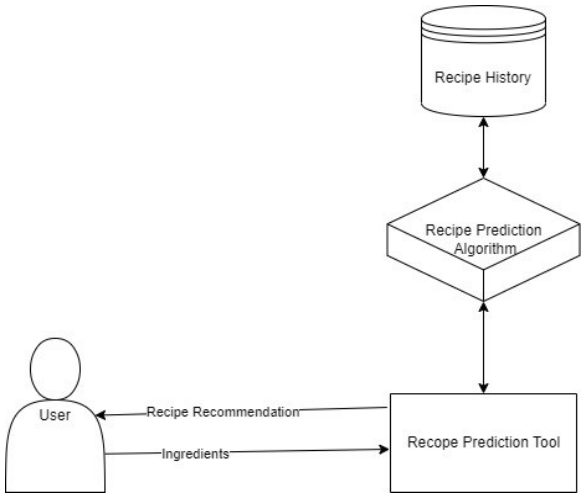


Figure 6: Architecture of the system

Our approach to this problem space will include NLP methods and machine learning. We expect that some data visualization will also be included to show insights in the data, once we have completed the analysis. Some of the NLP techniques involved include tokenizing, stemming, and stopwords to name a few. The intent is then to utilize an algorithm for comparing the similarity of strings such as cosine similarity. We hope that our solution may be different than other similar tools, based on our large corpus of recipes. We are also open to the idea of utilizing cuisine labels as an additional layer of complexity that allows for some customization to our tool.

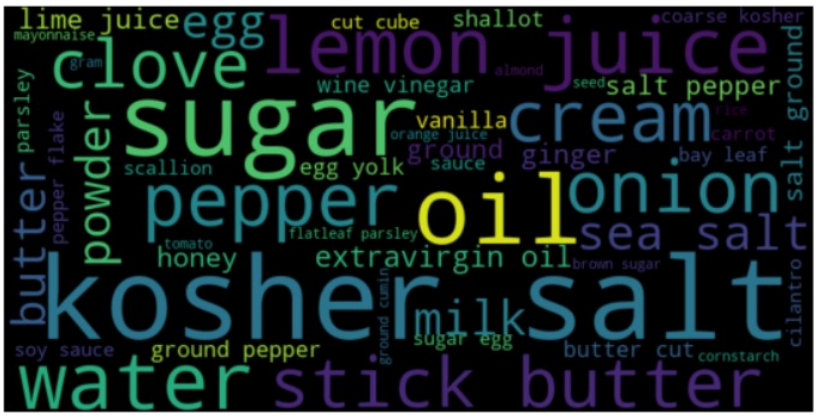
Proposed Development Platforms

The primary platform utilized for our recipe prediction tool is Python. There are numerous useful libraries within Python that aid with machine learning and natural language processing. A few of these include Pandas, NLTK, and Beautiful Soup, in addition to Matplotlib and Seaborn for data visualizations. Jupyter Notebook will be utilized for testing. Microsoft Excel will also be utilized for examining results, and possible visualizations.

Baseline solution

The proposed solution for our recipe prediction tool is to utilize natural language processing to use lists of recipe ingredients to predict likely recipes possible to cook for a given user. A given user would be able to provide ingredients they have, in the form of a list, each item separated by a comma. The tool then outputs five suggested recipe titles. This is the design of our initial solution.

We performed some initial exploratory analysis to get a feel for the dataset we were working with. Our dataset included over 13,000 rows of ingredient lists and recipe names. There was no single cuisine that our recipes were sourced from, so the recipe data includes food from all over the world. The below visual represents an initial look into the most common recipe ingredients. Kosher salt is a common ingredient among the corpus of ingredients, in addition to sugar, oil, pepper, lemon juice, etc.



The solution itself utilizes cosine similarity as a method for comparing a new list of ingredients to our existing corpus of recipes. Cosine similarity is a natural language processing "method of calculating the similarity of two vectors by taking the dot product and dividing it by the magnitudes of each vector," (Chamblee). The vectors in this case are the lists of ingredients.

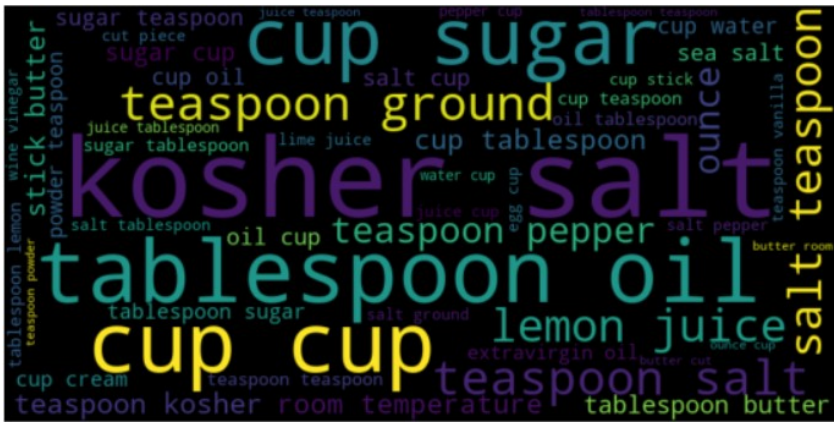
Our designed tool stores an existing corpus of ingredients and recipe titles to compare to user input. The corpus of recipe ingredients has been pre-processed including steps such as tokenization, parts-of-speech tagging, and stop-word removal. We do not expect these steps to often be repeated. It is our expectation that the existing corpus may be refreshed in the future, to include more recipes. However, at this time no further preprocessing needs to occur when the user provides input. When the user interacts with the tool, the tool will convert the input into a string (removing commas) and then vectorize the string. Vectorization is an essential step, as cosine similarity is the comparison of two vectors. The new ingredients string will be compared

to a vectorized format of every other existing ingredients list in the corpus. The top three recipe suggestions are provided to the user by sorting the list of scores gathered from this process. The user will receive three recipe names that contain the most similar ingredients to what they have listed as the food in their pantry. An example of initial results is as follows:

[0.99, 'Miso-Butter Roast Chicken With Acorn Squash Panzanella']
[0.66, 'Grilled Corn on the Cob with Salt-and-Pepper Butter']
[0.65, 'Garlic-Brown Butter Croutons']

In our testing, we used the first recipe, miso-butter roast chicken with acorn squash panzanella as the test case. What we can see from the initial development is that our method for computing cosine similarity is effective. The results show that there was an exact match with the recipe itself, which confirms the method's accuracy for us.

Our initial thoughts from this first iteration include a need for further data preprocessing. While the text was thoroughly preprocessed to remove special characters, and numerical text, and to remove some parts of speech, we found that an issue arose related to cooking measurement terms. There are numerous terms in the original ingredient text that are not ingredients but instead the measurement of an ingredient. Some examples include a cup, tablespoon, pound, or large. See more examples below.



These words all represent units of measure for the ingredients in the recipes. For the next iteration of testing, we will be implementing a method similar to stop word removal, where we will design a custom list of words to remove. The existing cleaned corpus of ingredients will be run against this list to remove additional unnecessary words.

Proposed solutions and present preliminary results

While our solution was effective at establishing a working tool for providing suggested recipes, it was not ideal. We made the observation that there was still a quantity of words that did not provide any value that remained in the corpus of ingredients. Measurements such as cups, pounds, ounces, and tablespoons are an issue for our tool because two recipes could have the same ingredients but different amounts. We did not want the units of measurement to affect the cosine similarity scores. Our proposed solution included implementing a function to remove words like these. We created a custom list of words to provide to this function as input to

compare against each of the records of ingredients. After re-running the program, these were the results of the second iteration:

[1.0, 'Miso-Butter Roast Chicken With Acorn Squash Panzanella'],
[0.66, 'Grilled Corn on the Cob with Salt-and-Pepper Butter'],
[0.66, 'Garlic–Brown Butter Croutons']

While there was not a significant change in the scores for the top three most similar recipes, we found that we were satisfied with this second iteration. The scores were impacted, though minimally, but they increased. The reason this is significant is that our assumption for this improvement was that similarity could be negatively impacted by these measurement words. If our hypothesis had been proven incorrect, the similarity scores would have either stayed completely the same or decreased. The scores increased, which tells us that the measurement words were causing the issue we had thought. We are satisfied with the preliminary results of our proposed recipe prediction tool.

The idea for the next iteration of work would be to add functionality to the output to the user. A possible idea of ours is to include more information about the ingredients that are in similar recipes or build the functionality to provide the recipe overall for the user to use. Additionally, giving the user more than three options to choose from is another idea for our next iteration of work.

Final Work

As outlined previously, the next step of our work was to enhance the output for the user. This initial project of work does not include the creation of a user interface for the user. However, for the initial model, we tested the method of providing a list of recipes that are output to the user once the tool runs. In addition to this, we desired to provide the recipes themselves to the user,
for their ease of use. The code now outputs the top five recipes to the user in the console. The top five are selected based on sorting the list of similarity scores to determine which recipes match the same ingredients provided by the user. However, in addition to this, the recipe instructions and details are also output to a text file for the user to utilize.

Complications and Errors

The most notable complication encountered involved removing excess words. While removing stopwords is a great tool provided by the NLTK library, the existing list of stopwords is not custom to any particular problem space. We discovered through our development that we needed to expand on the existing list of stopwords. For this case, we care about recipe ingredients but not their quantities. As a result, we were required to include a custom list of measurement words to remove after removing stopwords. Another complication we discovered was the existence of words or ingredients that had the same meaning but were interpreted differently. A perfect example of this is the ingredients salt and kosher salt. Some recipes required kosher salt specifically. However, after performing the necessary cleaning, these words were still interpreted as different words. We can see this in the word cloud shown previously. One can see salt and kosher salt represented differently. More custom pre-processing would need to be included to mitigate this risk.

Conclusion

We deem our solution a success. While we did not go so far as to create a custom user interface, the tool does exactly what we hoped it would do. A user is able to provide ingredients from their pantry and receive suggestions for the most probable recipes. We have thus concluded that our idea to generate recipe suggestions using NLP and ML was a valid hypothesis. Methods such as lemmatizing, pos tagging, and cosine similarity, among other things, were effective tools for this solution space. Cosine similarity's methodology of vectorizing recipe ingredient lists was the ideal methodology for our goal. While cosine similarity typically compares two strings for similarity, we were able to use cosine similarity in a loop to compare new user input to our entire existing corpus of recipes for the best possible recipe suggestions.

Future Work

Through the development of the initial model of our tool, we have brainstormed some enhancements for future development. Possibilities for the future of our recipe prediction tool include further customization for the user. For example, one of our ideas for the next phase of work would include adding an additional data source to add specific diet information for the user to choose from. While we are happy with our current tool, we know that a lot of people have different food plans or macronutrient requirements. An additional layer, allowing the tool to filter to only recipes in a specific category would be one of our future goals.

References

Carmody, Deidre (1995, May). “THE MEDIA BUSINESS; Conde Nast to Jump into Cyberspace”. The New York Times. Retrieved September 25, 2023, from https://www.nytimes.com/1995/05/01/business/the-media-business-conde-nast-to-jump-into-cyberspace.html

Chamblee, B. (2022, February 12). What is Cosine Similarity? How to Compare Text and Images in Python. *Medium*. https://towardsdatascience.com/what-is-cosine-similarity-how-to-compare-text-and-images-in-python-d2bb6e411ef0

Chaudhary, S., Soni, B., Sindhavad, A., Mamaniya, A., Dalvi, A., & Siddavatam, I. (2022). ChefAI.IN: Generating Indian recipes with AI algorithm. *2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT)*. https://doi.org/10.1109/tqcebt54229.2022.10041463

Epicurious. (n.d.). Epicurious – Recipes, Menu Ideas, Videos & Cooking Tips. Epicurious. Retrieved September 25, 2023, from https//www.epicurious.com/

Galanis, N.-I., & Papakostas, G. A. (2022). An update on cooking recipe generation with machine learning and Natural Language Processing. *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*. https://doi.org/10.1109/aic55036.2022.9848929

Goel S., Desai, A. & Tanvi. (2019, February). Food Ingredients and Recipes Dataset with Images. Version 1. Retrieved September 11, 2023, from https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images/data

Majumder, B. P., Li, S., Ni, J., & McAuley, J. (2019). Generating personalized recipes from historical user preferences. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/d19-1613

Oesper, L., Merico, D., Isserlin, R., & Bader, G. D. (2011). WordCloud: a Cytoscape plugin to create a visual semantic summary of networks. Source Code for Biology and Medicine, 6 (1), 7.

Pignatiello, G. A., Martin, R. J., & Hickman, R. L., Jr (2020). Decision fatigue: A conceptual analysis. *Journal of health psychology*, *25*(1), 123–135. https://doi.org/10.1177/1359105318763510

Vivek, M. B., Manju, N., & Vijay, M. B. (2017). Machine learning based food recipe recommendation system. *Proceedings of International Conference on Cognition and Recognition*, 11–19. https://doi.org/10.1007/978-981-10-5146-3_2