**Recipe Recommendation**

**Team Members and Roles:**

Blessy Kuriakose (11230255) [blessykuriakose@my.unt.edu](mailto:blessykuriakose@my.unt.edu) team lead, coding, report

Aniv Chakravarty (11424113) [anivchakravarty@my.unt.edu](mailto:anivchakravarty@my.unt.edu) model design, coding, report

Sathwik Reddy Rajidi (11509963) [sathwikreddyrajidi@my.unt.edu](mailto:sathwikreddyrajidi@my.unt.edu) testing, coding, report

Meetings are scheduled for Fridays at 12:30pm. (Time is subject to change.) Most, if not all, materials (coding, ppt, reports) will be completed though the cloud (Google Drive) for ease of collaboration.

**Project Abstract:**

People can spend hours looking up a workable recipe on the web only for them to find that they are missing crucial ingredients or for it to end up not suiting their palette. The job of this system is to alleviate these concerns and make cooking an easy and fruitful process. This project aims to develop a recipe recommendation system that suggests new recipes to users based on their ingredient preferences and the preferences of other users with similar tastes. By leveraging data with computational prowess, we can provide users with a personalized list of recipes that are highly correlated to their tastes. The system is designed to save users time and effort when looking for new recipes and help them discover new and exciting meals that they'll love.

This system should be able to…

1. Filter recipes by ingredients.
2. Analyze user preferences using sentiment analysis on past reviews.
3. Cluster reviewers to predict new recipes the user might enjoy.
4. Rank recipes using the above factors and display them as options.

**Data Abstraction:**

We will be using a dataset from Kaggle with 18 years of data collected from food.com. This contains two files, the recipes csv and the interactions csv, both of which will be utilized in this project.

The recipe csv is a 280 MB file with 231637 rows and 12 columns. The details of the data are below:

1. Name: (String) Name of the recipe
2. Id: (Integer) A unique identification number given to the recipe.
3. Minutes: (integer) The time it takes to cook in minutes.
4. Contributer\_id: (Integer) A unique identification number of the person who contributed the recipe.
5. Submitted: (Date) The date the recipe was added in mm/dd/yyyy format.
6. Tags: (list) A list of possible tags describing the recipe.
7. Nutrition: (list) A list of seven floating point values describing the nutritional values of the recipe.
8. N-steps: (Integer) The number of steps in the recipe.
9. Steps: (list) An order list of text describing each step of the recipe.
10. Description: (String) Text describing personal notes of the contributor.
11. Ingredients: (list) ingredients in the form of strings.
12. N-ingredients: (Integer) number of items in the Ingredients.

The second csv consists of a record of collaborative interactions and is a 333MB file with 1048575 rows and 5 columns. The details of the data are below:

1. User\_Id: (Integer) the unique identifier of the user
2. recipe\_Id: (Integer) the identifier from Id in the other dataset.
3. Date: (Date) the date of the review
4. Rating: (Integer) A rating system of value from 1 to 5 where 5 means really good and 0 means really
5. Bad with 0 used for not rated.
6. Review: (String) Textual description of a review from the user.

**Literature Survey:**

Recommendation systems are implemented in many fields to successfully help lower the resistance for completing a task or process, but despite the varied use-cases, the considerations given to creating such a system have common layouts that can be categorically understood. There are two such categories to speak of: content-based and collaboration-based. Of the two, the latter has two further subcategories: item-based and user-based. Content-based refers to recommendations made based on the subject's own historical data. Item-based is similar, but from a collaborative standpoint, that is, the perspective of multiple users is taken into account when considering similarities between items and the more similar items to that in the target user’s profile are pushed forward as a recommendation. User-based is where user-clusters are made and recommendations are given based on the experience of users similar to the target user. (This last type is what we will be covering within this project.)

“Personalized Recipe Recommendation System using Hybrid Approach” claims to take a step further from the conventional recommendation approach by adding in various categories of filter. This paper uses a web crawler to retrieve 10,971 recipes and their reviews from food.com. (Same source as the dataset from Kaggle.) Their task takes the approach of predicting user ranking given recipe content and historical user ranking, through which they cluster users via a KNN model. They also implement a second approach using stochastic gradient descent. Our approach is similar in the clustering ideology, however we aim to cluster groups based on reviews instead of recipe content.

When it comes to building clusters and calculating distance on a large scale is time consuming especially when incorporating collaborative recommendation and preferences. So instead we obtain a subset of the preferences and their centroid representing the users latent vectors which are represented on the item latent vector. Essentially finding key overlapping points that can be used to save on the overall computation time. They evaluated their model on recommendations and predictions on some well known datasets like Wikipedia and Amazon to see the computation speed up rate which resulted in almost double to triple speed up compared to other methods like GMIPS, SVDS, FGD and L2S.

Hierarchical Graph attention network model (HGAT) uses relation level attention module on embeddings on a heterogeneous recipe graph where each is of three types which are recipe, ingredients and users while edges are relations between the nodes. Attention modules are used on a node and relation level. They claim that the use of a neural network approach by considering the relations helped them achieve better performance to their baseline approaches like BPR, IngreNet, GAT etc., since they were able to capture more feature details along with their score and ranking based optimization functions.

Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). Sentiment analysis has gained much attention in recent years. In this paper, we aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. Sentiment polarity categorization is nothing but classifying the statements (reviews,reports etc) as positive, negative or neutral. A general process for sentiment polarity categorization is proposed with detailed process descriptions. Data used in this study are online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes. At last, we also give insight into our future work on sentiment analysis.

**Project Design:**

Libraries and technologies:

* Python
* Google colab
* Pandas, numpy, scikit-learn, matplotlib, seaborn
* Google drive
* Github

We will be using python as our base coding language and google colab so that we will be able to keep track and work on the coding collaboratively among us. Key libraries that will be used are pandas for handling data, numpy for working on arrays, scikit-learn to implement models and matplotlib along with seaborn for visualization. More libraries will be added as required during the coding portion in the future. We first plan on having a working system before we can build on it further.

The flowchart below gives an overview of the model functioning process:

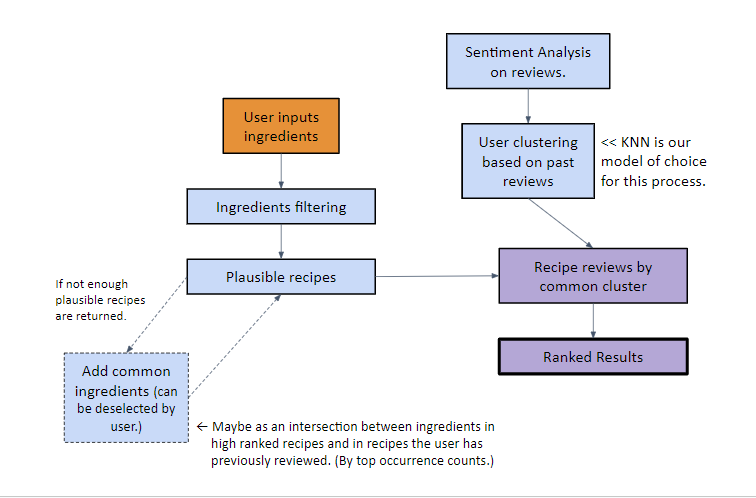
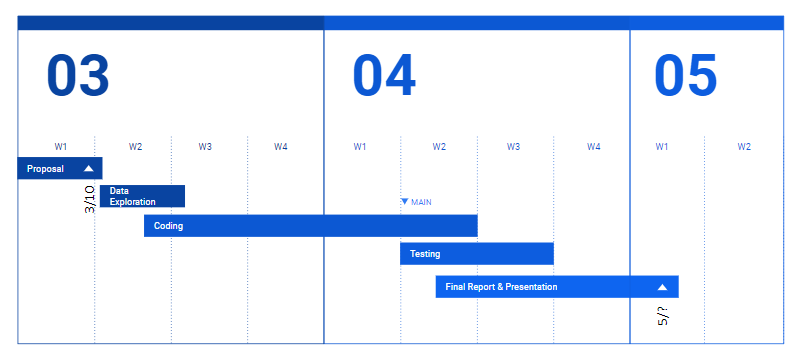


Figure 1: (workflow/architectures)

**Project Milestones:**

The following gives a rough outline of our workflow, in terms of a timeline for project completion:



The coding portions will be broken down as follows:

1. Sentiment analysis on reviews -- 24th of March.
2. KNN clustering of users (based on sentiment analysis) -- 7th of April.
3. Fusing for final output given input -- 14th of April.
4. Testing. -- 21st of April.
5. User Interface. (If time allows) -- 28th of April.
6. Finalization. -- 5th of May

The user interface part will be an extended goal. Finalization includes the final report and any changes necessary to the code.

**References**

Shubham Kumar Agrawal. “Recommendation System -Understanding The Basic Concepts.” *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/07/recommendation-system-understanding-the-basic-concepts/> [Accessed Mar. 10, 2023]

Gaudani, Hetal. (2016). “Personalized Recipe Recommendation System using Hybrid Approach.” *International Journal of Innovative Research in Computer and Communication Engineering*. 5. 192-197. 10.17148/IJARCCE.2016.5642. [Accessed Mar. 10, 2023]

Jyun-Yu Jiang, Patrick H. Chen, Cho-Jui Hsieh, and Wei Wang. 2020. Clustering and Constructing User Coresets to Accelerate Large-scale Top-K Recommender Systems. *In Proceedings of The Web Conference 2020* (WWW '20). Association for Computing Machinery, New York, NY, USA, 2177–2187. <https://doi.org/10.1145/3366423.3380283> [Accessed Mar. 10, 2023]

Tian Y, Zhang C, Metoyer R and Chawla NV (2022) Recipe Recommendation With Hierarchical Graph Attention Network. Front. Big Data 4:778417. doi: 10.3389/fdata.2021.778417 <https://www.frontiersin.org/articles/10.3389/fdata.2021.778417/full>

Fang, X., Zhan, J. Sentiment analysis using product review data. Journal of Big Data 2, 5 (2015). <https://doi.org/10.1186/s40537-015-0015-2>