

Recipe Recommender:

A Nutritional Distance-Based Approach

Team name : Tech Titans

Team members :

Benitha Gadupudi - G31142871

Divya Sree Vadlamudi - G48698217

Radhika Raghuwanshi - G32395729

Introduction:

- What is Recipe recommendation?
- Aim to enrich recipe recommendations by integrating nutrition-based feature extraction
- Central to our approach is the extraction of nutrition values from the dataset, serving as the cornerstone for recipe similarity comparison.

Objective:



Feature Extraction



Distance calculation
method



Evaluation and
Optimizing

Dataset information:

- raw-data_recipe.csv
 - Dataset has 49698 items.
 - Features include recipe id, recipe name, average rate, image url, review numbers, ingredients, cooking directions, nutritions and reviews.
 - The Raw dataset images would be used.
 - <https://www.kaggle.com/datasets/elisaxxygao/foodrecsysv1>



Preprocessing Data



Dropping
columns



Average rating



Unique recipes



Data extraction

Model Implementation:

- The k-Nearest Neighbors (KNN) recommender is used for our model to find similar recipes.
- Based on a selected recipe (recipe_id) our model calculates the distance between nutrition values of the selected recipe and all the members of the dataset.
- The KNN recommender returns the nutrition data of the selected recipe (recipe_id) and the top recommended recipes.
- The number of recipes to be recommended and which distance metric to be used from the list of Euclidean, Hamming and Cosine similarity along with the selected recipe id to call KNN recommender..

The output below shows the recommended recipes using cosine similarity as a distance metric and recommending three recipes

```
▶ knn_recommender(cosine, 79774, 3)
```

time cost: 9.43067 sec

| | calories | fat | carbohydrates | protein | cholesterol | sodium | fiber |
|-----------|----------|----------|---------------|----------|-------------|----------|----------|
| recipe_id | | | | | | | |
| 79774 | 0.000461 | 0.000398 | 0.000703 | 0.000338 | 0.0 | 0.001298 | 0.001680 |
| 159723 | 0.001383 | 0.001061 | 0.002110 | 0.000790 | 0.0 | 0.003570 | 0.005039 |
| 25725 | 0.000922 | 0.000663 | 0.001407 | 0.000451 | 0.0 | 0.002434 | 0.003079 |
| 88986 | 0.000691 | 0.000398 | 0.001055 | 0.000451 | 0.0 | 0.001623 | 0.002239 |

159723 Richard and Suzanne's Famous Spaghetti Sauce



25725 Garden Tomato Soup



88986 Gazpacho IV



Hybrid Nutrition Recommender:

- This model serves as a comprehensive recommender system that leverages multiple distance metrics to generate diverse and potentially more accurate recipe recommendations based on nutritional data.
- It combines the strengths of different distance approaches and allows users to prioritize recommendations based on specific criteria such as rating and review count.
- This hybrid recommender system provides a flexible and customizable solution for recommending recipes based on nutritional similarity and user-defined preferences.
- We implemented four distance metrics where two recommendations from each metric are taken into account and based on the average rating or review numbers the user specified number of recommendations would be chosen from the eight recommended.

Metrics used for Distance calculation:

1. Cosine Distance: The cosine distance measures the cosine of the angle between two vectors in a multidimensional space.
2. Euclidean Distance: The Euclidean distance (L2 norm) measures the straight-line distance between two points in a multidimensional space.
3. Levenshtein Distance : The Levenshtein distance (edit distance) measures the minimum number of single-character edits (insertions, deletions, substitutions) required to transform one string into another.
4. Hamming Distance: The Hamming distance measures the number of positions at which corresponding bits differ.

The output below shows the recommended recipes by a hybrid nutrition recommender using review numbers as a metric to select top three recipes.

```
nutrition_rn, topN_rn = nutrition_hybrid_recommender4(22886, ['review_nums'], 3)
```

78007 Mediterranean Vegetable Stew

22886 Grilled Mediterranean Vegetable Sandwich

231104 Sandy's Greek Pasta Salad

time cost: 80.40569 sec

22886 Grilled Mediterranean Vegetable Sandwich 78007 Mediterranean Vegetable Stew 231104 Sandy's Greek Pasta Salad



nutrition_rn

| | calories | fat | carbohydrates | protein | cholesterol | sodium | fiber |
|-----------|----------|----------|---------------|----------|-------------|----------|----------|
| recipe_id | | | | | | | |
| 22886 | 0.004148 | 0.003051 | 0.005627 | 0.002031 | 0.000143 | 0.001623 | 0.006158 |
| 22886 | 0.004148 | 0.003051 | 0.005627 | 0.002031 | 0.000143 | 0.001623 | 0.006158 |
| 78007 | 0.002535 | 0.001724 | 0.003869 | 0.001467 | 0.000000 | 0.001623 | 0.008958 |
| 231104 | 0.003917 | 0.002520 | 0.005627 | 0.002369 | 0.000428 | 0.001298 | 0.005599 |

topN_rn

| | recipe_id | aver_rate | review_nums |
|-------|-----------|-----------|-------------|
| 30897 | 22886 | 4.58 | 108 |
| 34962 | 78007 | 4.22 | 39 |
| 35766 | 231104 | 4.25 | 8 |

Model Evaluation

- The evaluation of Recipe recommendation system is done by establishing ground truth dataset based on predefined criteria like average rating and review numbers.
- Key evaluation metrics are computed to assess the quality and coverage of the recommendations compared to the ground truth dataset.
- We computed metrics like Precision, Recall, Coverage, True positives, False positives and False Negatives to evaluate our model.
- Precision: Precision measures the accuracy of the recommended items among all recommended items.
- Recall measures the ability of the system to find all relevant items.
- Coverage measures the proportion of items from the ground truth dataset that are recommended by the system.

Challenges:

- **Handling Large Datasets Efficiently:**

Working with a large dataset posed significant challenges, particularly with uploading and managing images. Uploading all images initially was not feasible due to resource constraints, requiring iterative approaches for data processing and image handling.

- **Evaluation Model Selection and Implementation:**

Determining the appropriate methods to evaluate the performance of the recommendation model proved to be a complex task. Identifying suitable evaluation metrics and understanding how to interpret them accurately .

Thank You!