**AIE425 Intelligent Recommender Systems, Fall Semester 24/25**

Course Project: Nutrition-aware Food Recommender Engine

**A20000021, Ahmed Ashraf Mohamed Ali & A20000726, Mohamed Ibrahim Fekry**

**1. Introduction**

This report presents the design, development, and evaluation of a Nutrition-Aware Hybrid Food Recommender Engine. The system combines content-based and collaborative filtering approaches to provide personalized recipe recommendations based on nutritional content and user preferences. The goal is to help users discover recipes that align with their dietary needs and taste preferences. This project addresses the growing demand for personalized nutrition and healthier eating habits by leveraging machine learning techniques.

**2. Data Collection & preprocessing**

**2.1. Data Collection**

**Datasets Used:**

1. Recipes Dataset: Contains recipe details, including nutritional information (e.g., calories, fat, protein) and ingredients.
2. Reviews Dataset: Contains user reviews and ratings for recipes.

**Source**: The datasets were downloaded from [[Kaggle](https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews/data)].

**Dataset Size:**

Recipes Dataset: 522,517 entries with 28 columns.

Reviews Dataset: 1,401,982 entries with 8 columns.

**2.2. Data Preprocessing**

**Handling Missing Values:**

- Missing values in the datasets were filled with `0` to ensure consistency.

- For example, missing `CookTime` or `AggregatedRating` values were replaced with default values.

**Filtering Recipes:**

- Recipes were filtered based on maximum nutritional values to exclude unrealistic or unhealthy recipes.

- For example, recipes with calories > 2000 or fat > 100g were removed.

**Normalization:**

- Nutritional data (e.g., calories, fat, protein) was normalized using `StandardScaler` to ensure all features contribute equally to the similarity calculations.

**Text Feature Extraction:**

- Ingredients were processed using `TfidfVectorizer` to create a TF-IDF matrix for text-based similarity.

- This step converts ingredient lists into numerical vectors for similarity computation.

**3. Dataset Description**

**3.1 Recipes Dataset**

- **Size**: 522,517 entries with 28 columns.

1. **Key Columns**:

- `RecipeId`: Unique identifier for each recipe.

- `Name`: Name of the recipe.

- Nutritional columns: `Calories`, `FatContent`, `ProteinContent`, etc.

- Text columns: `RecipeIngredientParts`, `Description`.

1. **Example Data:**

- RecipeId: 38

- Name: Low-Fat Berry Blue Frozen Dessert

- Calories: 170.9

- FatContent: 2.5

- ProteinContent: 3.2

**3.2 Reviews Dataset**

- **Size**: 1,401,982 entries with 8 columns.

1. **Key Columns:**

- ReviewId: Unique identifier for each review.

- RecipeId: Links reviews to recipes.

- AuthorId: Unique identifier for each user.

- Rating: User rating (1-5).

1. **Example Data:**

- ReviewId: 1

- RecipeId: 38

- AuthorId: 2008

- Rating: 5

**3.3 Modelling User Interests**

- User interests are modeled through ratings and interactions with recipes.

- Nutritional preferences are inferred from the recipes users interact with.

- For example, if a user frequently rates low-calorie recipes highly, the system infers a preference for low-calorie foods.

**4. Data Analysis and Interpretation**

**4.1 Nutritional Data Distribution**

- Histograms were plotted to visualize the distribution of nutritional values (e.g., calories, fat, protein).

- **Observations**:

- Most recipes have low to moderate nutritional values.

- Outliers exist, such as recipes with extremely high calorie or fat content.

**4.2 User Interactions**

- The reviews dataset shows a wide range of user ratings, with an average rating of ~4.0.

- **Observations**:

- Popular recipes (e.g., A-To-Z Bread) have high ratings and frequent interactions.

- Users tend to rate recipes they enjoy highly, providing valuable data for collaborative filtering.

**4.3 Insights for Recommender System**

- The data suggests that users prefer recipes with balanced nutritional content and high ratings.

- Combining nutritional data with user preferences can improve recommendation quality.

**5. Background of the Chosen Algorithm**

**5.1 Content-Based Filtering**

- **Objective**: Recommends recipes based on nutritional content and ingredient similarity.

**- Key Steps:**

1. **Feature Extraction**: Extract nutritional features (e.g., calories, fat, protein) and text features (ingredients).

2. **Similarity Calculation**: Use \*\*cosine similarity\*\* to measure similarity between recipes.

3. **Recommendation**: Rank recipes based on similarity scores and return the top N recommendations.

**- Advantages:**

- Focuses on recipe attributes (e.g., nutrition, ingredients).

- Suitable for users with specific dietary needs.

**- Limitations:**

- May lack diversity in recommendations.

- Relies heavily on the quality of feature extraction.

**5.2 Collaborative Filtering**

- **Objective**: Recommends recipes based on user ratings.

- **Key Steps**:

1. **Matrix Factorization**: Use SVD (Singular Value Decomposition) to decompose the user-item interaction matrix into latent factors.

2. **Prediction**: Predict user ratings for unseen recipes based on latent factors.

3. **Recommendation**: Rank recipes based on predicted ratings and return the top N recommendations.

- **Advantages**:

- Leverages user behavior data.

- Suitable for users who value community ratings.

- **Limitations**:

- Requires a large amount of user interaction data.

- May struggle with cold-start problems (new users or recipes).

**5.3 Hybrid Approach**

- **Objective**: Combines content-based and collaborative filtering to provide balanced recommendations.

**- Key Step:**

1. **Content-Based Recommendations**: Generate recommendations based on nutritional and ingredient similarity.

2. **Collaborative Filtering Recommendations**: Generate recommendations based on user ratings.

3. **Combination**: Merge recommendations from both approaches and remove duplicates.

- **Advantages**:

- Provides diverse and personalized recommendations.

- Balances nutritional relevance with user preferences.

- **Limitations**:

- Requires careful tuning of weights for combining recommendations.

**6. Design of the Recommender Engine**

**6.1 System Architecture**

1. **Input**: User ID and nutritional preferences.

2. **Content-Based Filtering**: Recommends recipes based on nutritional similarity.

3. **Collaborative Filtering**: Recommends recipes based on user ratings.

4. **Hybrid Recommendation**: Combines both approaches and removes duplicates.

5. **Output**: Top 10 recommended recipes.

**6.2 Key Features**

- **Personalization**: Tailored recommendations based on user preferences.

- **Nutritional Awareness**: Focus on recipes with balanced nutritional content.

**7. Implementation of the Recommender Engine**

**7.1 Tools and Libraries**

- **Python**: Primary programming language.

- **Libraries**:

1. `pandas`, `numpy`: Data manipulation.
2. `scikit-learn`: Machine learning and preprocessing.
3. `surprise`: Collaborative filtering.
4. `matplotlib`, `seaborn`: Visualization.

**7.2 Implementation Process**

1. **Data Loading**: Load and merge datasets.

2. **Preprocessing**: Handle missing values, filter recipes, and normalize data.

3. **Content-Based Filtering**: Compute cosine similarity for nutritional and text features.

4. **Collaborative Filtering**: Train SVD model using user ratings.

5. **Hybrid Recommendation**: Combine results from both approaches.

**8. Testing Methodology and Results Representation**

**8.1 Testing Method**

- **Cross-Validation**: Used to evaluate the collaborative filtering model.

- **Test Cases**:

- Input: User ID and nutritional preferences.

- Output: Top 10 recommended recipes.

**8.2 Results Representation**

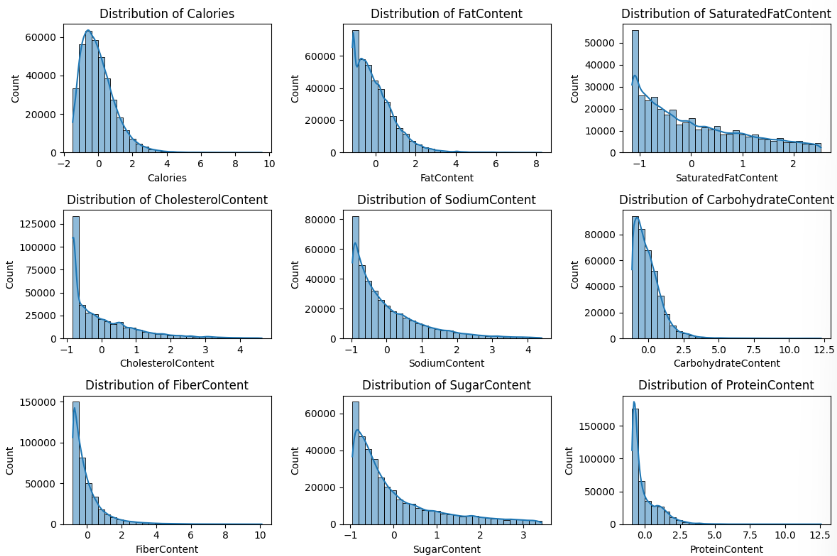
- **Tables**: Display recommended recipes with nutritional details.

- **Visualizations**: Bar plots for top recommended recipes.

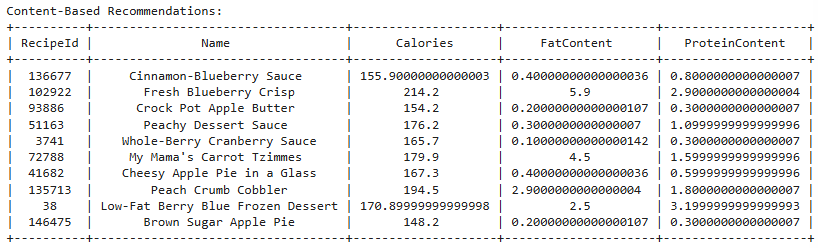
**9. Results**

**9.1 Nutritional Data Distribution**

The following histograms show the distribution of nutritional values in the dataset:



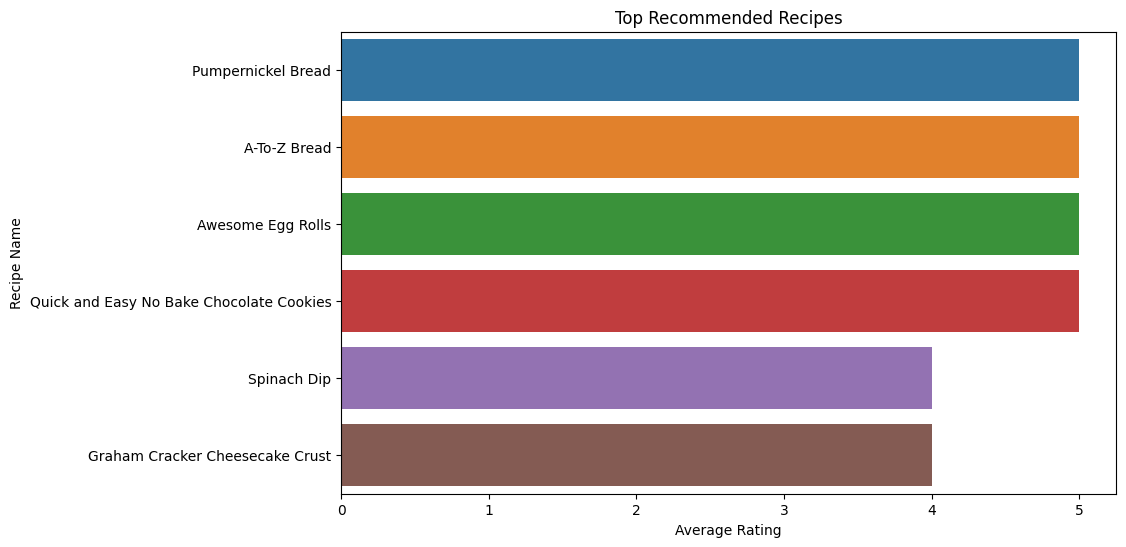
**9.2 Content-Based Recommendations**



**9.3 Collaborative Filtering Recommendations**

A list of recipe IDs predicted to have the highest ratings for a specific user based on their past interactions.

[6536, 3370, 8953, 3877, 9974, 8674, 10205, 5335, 7537, 3748]



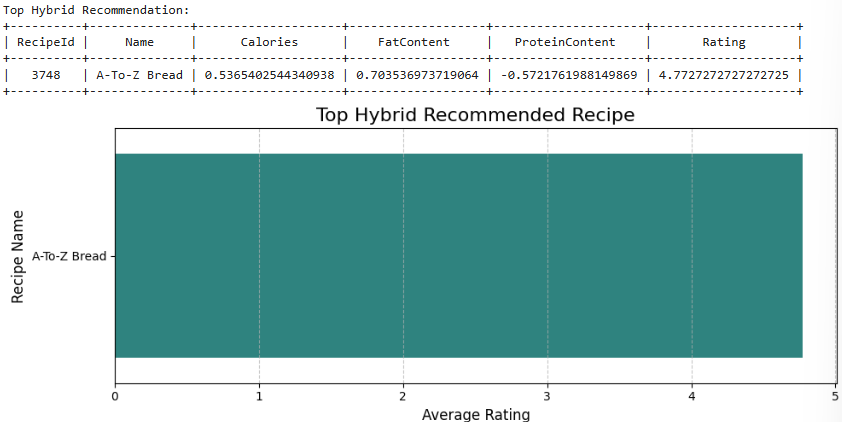
**9.4 Hybrid Recommendations**

A combined list of recipe IDs from both content-based and collaborative filtering approaches, prioritizing nutritional similarity and user preferences.

[500481, 6536, 444942, 314004, 469399, 9116, 336285, 373791, 3748, 3877]

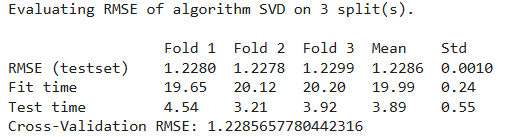
**9.5 Top Recommended Recipe**

The top recipe based on both nutritional similarity (content-based) and user preferences (collaborative filtering), visualized in a bar plot showing recipe name and it’s corresponding rating.



**9.6 Model Evaluation**

- **RMSE**: 1.2286 (mean across 3 folds).



**10. Evaluation and Comparison of Results**

**10.1 Content-Based vs. Collaborative Filtering**

- **Content-Based**: Focuses on nutritional similarity but may lack diversity.

- **Collaborative Filtering**: Focuses on user preferences but may ignore nutritional content.

**10.2 Hybrid Approach**

- Combines the strengths of both approaches.

- Provides balanced and personalized recommendations.

**10.3 Model Performance**

- The collaborative filtering model performs well, with a low RMSE.

- The hybrid approach ensures diverse and relevant recommendations.

**11. Conclusion**

The hybrid recommender engine, combining content-based and collaborative filtering approaches, effectively balances nutritional relevance and user preferences to deliver personalized food recommendations. The content-based component ensures that recommendations align with dietary needs by analyzing nutritional and ingredient data, while the collaborative filtering component leverages historical ratings to capture user preferences. The hybrid model successfully integrates these strengths, offering diverse and meaningful suggestions.

With an RMSE of 1.2286 on a 1-5 rating scale, the model demonstrates reasonable predictive accuracy but leaves room for improvement.

While the engine is a promising tool for nutrition-aware and user-centric recommendations, further refinement is necessary to reduce prediction errors and improve alignment with user expectations. Iterative testing and validation, along with the inclusion of additional data such as user feedback or contextual information, would be valuable for enhancing the model's accuracy and relevance.

**12. Enhancements**

1. **Dietary Restrictions**: Add support for gluten-free, vegan, and other dietary preferences.

2. **User Feedback**: Incorporate user feedback to refine recommendations.

3. **Real-Time Recommendations**: Implement real-time recommendation updates based on user interactions.