Project 18: Food Recommender System

The main objective of this project is to build a personalized food recommendation system that incorporates user reviews, ratings, and food contents (e.g., calories, macronutrients, ingredients) to recommend recipes or food items. Consider the Dataset of [Food Recommendation Systems](https://www.kaggle.com/code/ngohoantamhuy/food-recommendation-systems) which is recommender system dataset collected from Food.com website. In this project you will use these data from this Kaggle repository:

• **RAW\_recipes.csv**: Contains the raw dataset with detailed recipe information including name, id, minutes, contributor\_id, submitted, tags, nutrition, n\_steps, steps, description, ingredients, and n\_ingredients.

• **RAW\_interactions.csv**: Contains the raw dataset with detailed recipe information including user\_id, recipe\_id, date, rating, and review.

1. **Initial Data Exploration**: Explore the distribution of recipes based on key features such as minutes, tags, n\_steps, and n\_ingredients, and visualize the distribution of recipes for each of these features.
2. **User Profile Generation**: Using the two datasets, RAW\_recipes.csv and RAW\_interactions.csv, create a new dataset named User\_Data.csv, where each row corresponds to a user in the system. The columns should include rated\_recipes (a list of all recipes rated by the user), ingredients (a list of all ingredients in the recipes rated by the user), and rating\_list (the list of ratings given by the user). Based on this generated user profile, explore the distribution of users across key features such as the number of rated items, the total number of ingredients per user, and the average of recorded ratings. Visualize the distribution of users for each of these features.
3. **Sentiment Analysis on User Reviews**: Perform Sentiment Analysis on the user reviews using pre-trained models (e.g., VADER, TextBlob, or BERT) to analyzes the sentiment polarity of each review and categorizes them into positive, neutral, or negative sentiments.
4. **Sentiment polarity between different recipe groups**: Explore the average sentiment for different recipes and tags by analyzing the sentiment polarity of user reviews. Visualize the results to identify any interesting patterns or findings regarding recipes or tags with notable sentiment polarity. Discuss any significant insights, such as recipes or tags that show strong positive or negative sentiment trends.
5. **Tag and ingredients-based Recipe Similarity Calculation**: Propose a method for calculating the similarity between different recipes, such as TF-IDF, Jaccard, Levenshtein Distance, Semantic Similarity, or Doc2Vec, by considering recipes' ingredients and tags. Next, calculate the recipe similarities based on their ingredients using the selected method.
6. **Description-based Recipe Similarity Calculation**: Repeat Task 5 for recipe similarity calculation based on the Tags of each recipe.
7. **Review-based Recipe Similarity Calculation**: Repeat Task 5 for recipe similarity calculation based on the descriptions of each recipe.
8. **Recipe Clustering**: Based on the recipe similarities calculated in the previous task of 5 and 6 (Content and Review), apply a suitable clustering algorithm (e.g., K-Means, DBSCAN, or Agglomerative Clustering) to group similar recipes. Perform the clustering task separately for ingredients and descriptions and compare the clustering outcomes.
9. **Cluster Analysis**: For each of the identified recipe clusters, find the most frequent tags, ingredients, and topics using suitable topic modeling techniques (e.g., LDA, LSA, or other methods). Additionally, explore the average sentiment and average ratings within these clusters. Visualize the results to identify any interesting patterns or trends related to the recipe groups. Discuss any significant insights, such as clusters or tags that show strong positive or negative sentiment trends, and highlight any notable patterns in user preferences and ratings.
10. **Content-Based Filtering**: Implement a content-based recommendation system that recommends food items similar to those a user has previously rated highly, based on:   
     o Tag and ingredients-based Recipe Similarity (Task 5)   
     o Food descriptions similarity (Task 6)   
     o Review-based similarity (Task 7)  
    To evaluate recommendation model, the dataset should be divided into two sets called training and test. For thetraining set, 80% of ratings of each user are selected randomly. The remaining 20% ratings for each user are used as the test set. The training set is used as the input of the models in the training phase. Whilst, the test set is used to evaluate the performance of the models. Three evaluation metrics: mean absoluteerror (MAE), and root man squared error (RMSE), should be used to evaluate the models.
11. **Collaborative Filtering-Based Recommendations**: Repeat task 10 using the Collaborative Filtering method, where the similarity between users is calculated based on the training set using the Pearson Correlation Coefficient as follows:   
    A black lines with black text

    Description automatically generated with medium confidence  
    which calculates the similarity between two users, 𝑢 and 𝑣, where 𝑚 is the number of commonly rated items, 𝑟𝑢,𝑖 is the rating user 𝑢 has given to item 𝑖, and 𝑟̅𝑢 is the average of all the ratings that this user has given.
12. **Suggestions**: Feel free to suggest any extra analysis that can elucidate some aspects of the previous specifications using some state-of-the-art approaches for the recommendation. Use appropriate literature of corpus linguistic literature to justify your findings and comment on the obtained results. Also, comment on the limitations and structural weakness of the recommendation systesm.
13. Identify appropriate literature to discuss the findings and comment on the strength and weakness of the data processing pipeline.