Machine Learning Term Project

Team 10 : 202135546 송영우, 202135596 현관, 202135599 황성민, 202234908 오예진

1. The objective of the system

Enhance customer experience and satisfaction by recommending personalized sandwich combinations at a sandwich chain where customers can select multiple ingredients, while also optimizing ingredient freshness and inventory management efficiency.

Assumptions

- All sandwiches are priced the same.
- Only one ingredient can be selected per category.
- Assume a linear relationship between the user's ingredient preferences and the sandwich combination.

2. Datasets to use

Creating a Rating Dataset for 500 Users (Each Rated 30 Sandwich Combinations)

1. Initialize User Preferences
    - For 500 users, assign a preference score (0–5 in 0.5 increments) for each of 20 ingredients.
    - Set initial weights for each of the 5 ingredient categories.
2. Apply Variations
    - Add user-specific average bias.
    - Apply random noise.
    - Introduce ingredient-specific variability (e.g., bread scores are similar across users, meat scores have larger variance).

3. Select Representative Combinations
    - From 625 possible combinations, select 50 representative combinations.
    - Assign 30 combinations per user, considering anchors.
4. Include Long-Tail Combinations
    - For the remaining 575 combinations, assign 10 combinations per user to capture long-tail preferences.
5. Compute Combination Ratings
    - Derive combination ratings as a linear combination of the ingredient preference scores.
6. Apply Demographic and Dietary Adjustments
    - Add bias for gender and age group.
    - Apply dietary restrictions (e.g., vegetarian or allergies) by subtracting 1 point for restricted ingredients

Generate Training and Testing Sets

- Repeat the above process twice to create separate training and testing datasets.

Dataset Composition Results

1. User Dataset
    - Basic user information
    - Ingredient preference scores
    - Category weights
    - User-specific average bias
2. Ingredient Dataset (4 categories × 5 ingredients per category)

Bread	Vegetable	Meat	Sauce
White / 195	Lettuce / 2.9	Roasted Chicken / 90	Sweet Onion / 40.1
Wheat / 195	Tomato / 7.7	Ham / 40	Sweet Chili / 40

Parmesan Oregano / 207	Pickle / 0.4	Meatball / 210	Smoke BBQ / 32.8
Honey Oat / 237	Onion 2.8	Bacon / 90	Honey Mustard / 38.4
Flatbread / 232	Avocado / 56.5	Pepperoni / 150	Ranch / 116

3. Sandwich Composition Dataset (625 combinations × 20 ingredients)
4. Final Training Table
   - 20,000 user–sandwich entries
   - Columns : user_id, sandwich_id, rating

3. Filtering method to use

1. User-Based Collaborative Filtering
   - Used for constructing the predicted rating table
2. Item-Based Collaborative Filtering
   - Analyze similarity between individual sandwiches
   - Recommend final candidates with different compositions
3. Rule-Based / Attribute-Based Filtering
   - Apply dietary goals and allergy information

Recommendation System Workflow

1. Construct the predicted rating table (item-based) using the rating table and item table through an ALS-based Matrix Factorization (MF) model.

Variable	Size	Meaning
U	(500 × 20)	User latent preference vector
V	(20 × 20)	Ingredient latent feature vector
S	(625 × 20)	Sandwich composition
C = S·V	(625 × 20)	Sandwich embedding
R_hat = U·C^T	(500 × 625)	Predicted ratings

2. User-Based CF
    - Construct the predicted rating table using user-based collaborative filtering.
3. Hybrid Predicted Ratings
    - Combine the user-based and item-based predicted rating tables using a weighted average.
    - Select the top 50 sandwiches.
4. Filtering
    - Apply filters based on the user’s health information.
    - Exclude sandwiches the user has already tried.
5. Final Recommendation
    - Recommend 3 sandwich combinations.
    - Use item-based CF to propose candidates with diverse ingredient compositions.

4. Machine Learning model to use

ALS-Based Matrix Factorization (MF) Model

- Decompose the user–sandwich rating matrix to construct a predicted rating matrix.
- Consider latent feature vectors of users, sandwiches, and ingredients.
- Iterative training allows minimizing the loss function.
- Performs well on large, sparse matrices.