Gang Chen z5469391

Project Design

FoodLike: A Personalized Food Recommendation System

1. Scope

(1) Client

I propose to design a personalized food recommendation system called FoodLike, targeting young urban users who frequently face the dilemma of "what to eat." The system helps users make efficient dining decisions by recommending dishes and restaurants tailored to their individual taste preferences and historical behavior.

Target User Profile:

- Age: 20–35 years old
- Characteristics: Busy lifestyle, frequent food delivery users, interested in healthy eating or specific dietary preferences (e.g., spicy, vegetarian)

Usage Scenario & Interaction:

- Upon launching the app, the system recommends 5 dishes using a card-based interface;
- Each card shows dish image, name, tags (e.g., spicy, vegan), restaurant name, rating, and price;
- Users can interact via "Like" "Dislike" and "Save" buttons

(2) Domain

The proposed recommender system falls under the **food and dining domain**, with a focus on **personalized dish and restaurant recommendations** for young urban consumers.

System Purpose:

FoodLike helps users quickly decide what to eat by suggesting dishes and restaurants tailored to their personal tastes, dietary needs, and previous interactions. It combines collaborative filtering with content-based approaches to provide accurate, diverse, and engaging food suggestions.

Why This Domain?

Choosing food is a daily decision that many users find time-consuming or frustrating—especially when they are hungry, tired, or overwhelmed by too many options. A personalized recommender system can significantly improve user satisfaction by reducing decision fatigue and encouraging culinary exploration. Moreover, food-related platforms (delivery, dine-in, reviews) are booming, making this a highly practical and impactful area for real-world application.

(3) Display

Given that our target users primarily interact with the system on mobile devices (e.g., during commutes or right before meals), the system will present Top-4 dish recommendations at a time in a swipeable card interface. This display design prioritizes clarity, quick scanning, and lightweight interaction.

- Each card displays the dish image, name, tags (e.g., spicy, vegan, low-fat), rating, price, and associated restaurant.
- Recommendations are ranked based on personalized relevance scores, from most to least relevant.
- There is no traditional pagination; instead, infinite scrolling (lazy loading) allows users to continuously browse suggestions.
- Users may also refresh the recommendation list to retrieve a new batch of suggestions, supporting multi-round recommendation exploration.

This presentation strategy helps avoid cognitive overload while maintaining user engagement, making it easy for users to find appealing food options quickly and intuitively.

(4) User Interface

FoodLike will be delivered as a mobile application (iOS / Android), as our target users (young professionals and students) primarily use mobile platforms for food ordering and browsing. The user interface is designed to be minimal, intuitive, and optimized for quick decision-making.

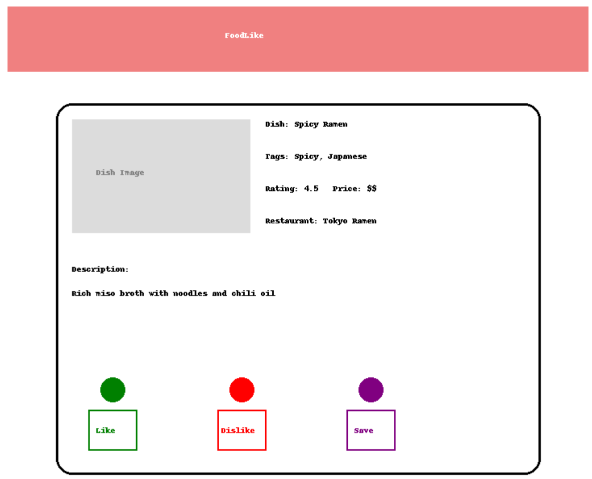
UI Features:

- A swipeable card layout, where each card presents one dish recommendation.
- Each card includes dish image, name, cuisine tags, restaurant info, rating, and price.
- Four feedback buttons below each card: Like, Dislike, Save

Interface Mockup:

A mockup has been created (see next page) to visually represent the recommendation view. The layout is optimized for one-handed interaction, with minimal taps and quick swiping. Feedback signals are captured in real-time to adapt recommendations dynamically.

This UI design ensures the system is easy to interact with even in short sessions, such as during lunch breaks or commutes.



(5) User Interaction

FoodLike adopts a swipe-based card interface where users interact with one recommendation at a time and provide feedback using intuitive buttons. The system supports the following user actions:

- Like: Indicates strong interest. This action boosts the recommendation probability of the current dish and other similar dishes.
- Dislike: Indicates disinterest. The system will reduce the exposure of the current item and similar items.
- Save: Considered a strong positive signal. It is used to refine the user embedding vector, reinforcing personalized preferences.

How Feedback Improves the System:

- In collaborative filtering, likes and saves help identify similar users and enhance recommendation relevance.
- In content-based filtering, tag and feature weights associated with liked/saved dishes are boosted.
- User embeddings are periodically updated using interaction logs to reflect evolving preferences, improving recommendation precision over time.

Simulated Interaction Flow:
Although a fully functional frontend is not required, we design a simulated user journey to mimic real usage:
The user is shown a set of recommended dishes → interacts with them via Like/Dislike/Save → the system records these as feedback → recommendation results adapt accordingly.
This "pretend-to-use" interaction scenario allows for effective user research and system evaluation, forming a feedback loop that refines the model continuously.

(6) Model Updates and Cold Start Strategy
Cold Start Problem:

- To address the cold-start issue for new users or unseen dishes, FoodLike applies a hybrid approach:
- Initial Content-Based Recommendations: New users receive suggestions based on general popularity and dish metadata (e.g., cuisine, tags).
- Onboarding Survey: New users fill out a short preference quiz (e.g., favorite cuisines, spice tolerance).
- Exploration Phase: The system introduces some random dishes early on to explore user preferences and build a profile quickly.

Model Updates:

- FoodLike adopts a hybrid update strategy combining batch and online learning:
- Weekly Offline Updates: The system retrains user and item embeddings weekly using accumulated interaction data.
- Online Weight Adjustments: Lightweight updates to ranking weights or preferences are made in real-time based on like/dislike/save feedback.

This design ensures that FoodLike stays responsive to user behavior over time while remaining computationally efficient.

(7) Business Model

FoodLike aims to be more than just a helpful tool for users - it has the potential to create real-world business value. Here are several monetization strategies that could support its commercial viability:

1) Promotional Partnerships with Restaurants

FoodLike can collaborate with local restaurants by offering paid promotion slots. Restaurants can pay for boosted visibility in the recommendation list, appearing more frequently to relevant users.

2) Premium Membership Subscriptions

The platform can offer a subscription plan with advanced features such as:

- More frequent and personalized recommendation updates
- Access to high-accuracy "top picks"
- Syncable favorites list and detailed taste profile analytics

3) Restaurant Analytics Services

Using the platform's aggregated user interaction data (e.g., likes, saves, skips), FoodLike can offer restaurants insights into customer preferences, dish popularity trends, and data-driven menu optimization - as a paid service.

4) Affiliate Commission and Order Integration

If integrated with food delivery or reservation platforms, FoodLike can earn commissions by redirecting users to partner services when they act on a recommendation.

This business model creates a value loop between users (personalized experience), restaurants (marketing and data), and the platform (monetization), allowing FoodLike to scale as a useful and sustainable service.

2. Datasets

(1) **Dataset Name & Source**

This project uses the **“Restaurant Reviews – Yelp Open Dataset”**, available on Kaggle:

Yelp Dataset (https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset).

It includes restaurant reviews, ratings, and metadata from various cities in the US and Canada, making it suitable for building personalized food recommendation systems.

(2) **Dataset Scale & Type**

- Number of users: ~1,000,000

- Number of restaurants (items): ~200,000 (some with menu/category data)
- Total interaction records: over 8 million reviews
- Included data fields:
  - Star ratings (1–5 stars)
  - Review text
  - Timestamps
  - Business categories (e.g., Chinese, Japanese, Fast Food)
  - Restaurant location and city tags

(3) **Structure & Usage**

Field	Usage Description
user_id, business_id, stars	Used to build the user-item rating matrix (collaborative filtering)
categories, name, address	Used for item-side embeddings (content-based recommendation)
review_text	Used to generate text embeddings for enhanced semantic modeling
date	Used to analyze time-aware preference or simulate cold-start

(4) How Will It Be Used?

- The dataset will be split into training and testing sets (chronologically) to simulate real-world deployment.
- Restaurant categories will be cleaned and grouped into cuisine types (e.g., Western, Dessert) for better interpretability.
- In cold-start simulations, only a user's first 5 interactions will be used to construct an initial preference vector.

(5) Is the Dataset Enough? How to Compensate?

While the dataset is somewhat limited, combining collaborative and content-based methods (e.g., text embeddings, categories) can significantly improve performance. In future real-world use, additional behavioral data (e.g., likes, saves, clicks) could be integrated to support online fine-tuning and feedback learning.

(6) **Three Limitations of the Dataset**

- The data is not complete – only a portion of active users are represented.
- Though cleaned, the dataset includes skewed categories (e.g., more fast food than niche cuisines) and vague cold-start cases.
- The original dataset is designed for rating prediction – it lacks direct support for ranking or exploration tasks.

3. Methods

(1) Method Overview

FoodLike is a personalized food recommendation system that aims to recommend restaurants and dishes based on user preferences, feedback interactions (like/dislike/save), and contextual features such as time and location. We adopt a hybrid strategy that combines collaborative filtering and content-based methods to enhance cold-start handling and recommendation accuracy.

(2) Baseline Method

Method: User-Based Collaborative Filtering (UBCF)

Rationale:

- The system is interaction-driven — users frequently like, dislike, or save food items. Therefore, similar users tend to have similar preferences.
- The user-item interaction matrix (like/dislike/save) allows for fast prototyping using cosine similarity between users.
- Easy to implement and serves as a reasonable starting point for system evaluation.

(3) Method 2: Hybrid — Collaborative Filtering + Food Descriptions/Tags

Method: Hybrid — Combine collaborative filtering with food metadata (e.g., cuisine, tags, ingredients)

Rationale:

- Many food items come with rich descriptive content (title, cuisine type, tags, short descriptions), which helps in cold-start cases where interaction data is sparse.
- Combining user similarity with content similarity enables fallback logic when user history is short.
- We can use models like LightFM or late fusion strategies (score-level fusion) to combine the results of both branches.

(4) Method 3: Context-Aware Recommendation

Method: Context-Aware Recommendation — Add Time and Location Embeddings

Rationale:

- Food choices are heavily influenced by time (e.g., breakfast vs. dinner) and location (nearby restaurants).
- We can model a 3D user-time-food preference tensor to dynamically adapt recommendations.
- Alternatively, contextual filters (e.g., recommend within a 2km radius + dinner hour) can boost relevance and freshness.

(5) Strategy Comparison & Execution Plan

We plan to use UBCF as our baseline model and incrementally introduce content features and contextual embeddings to build a hybrid model. We will compare the performance of different strategies using metrics like Precision@K, Hit Rate, and Recall@K, and iteratively select the best-performing one.

(6) Summary of Strategic Thinking

- **Start Simple**: UBCF with basic user-item matrix
- **Add Content**: Food tags, text, images → content-based embeddings
- **Add Context**: Use time and GPS/location as filters or embedding inputs
- **Fuse**: Hybrid methods to improve cold-start and overall ranking

4. Evaluation

(1) Evaluation Objectives

We aim to evaluate not only the **predictive power** of the recommendation model, but also the **user experience** of the entire FoodLike recommendation system.

- **Model Layer**: Based on historical interaction data, can the model accurately predict the dishes a user will like?
- **System Layer**: When facing real users, do they find the recommendations helpful and decision-friendly?

(2) Model Evaluation Metrics (Offline Evaluation)

To assess the performance of the recommendation model based on historical data, we will adopt the following metrics:

- Precision@4: Among the Top-4 items recommended to the user, how many are dishes the user actually liked?
- Recall@4: Among all the dishes the user liked, how many were successfully recommended in the Top-4?
- NDCG@4 (Normalized Discounted Cumulative Gain): Evaluates the ranking quality of recommended items.
- Coverage: Measures how many different dishes/restaurants are recommended, reflecting exploration ability and diversity.

Training/Test Strategy:
We use a leave-one-out method, leaving one interaction out for each user as the test case. This ensures personalized and rigorous evaluation.

(3) System Evaluation (User-Centric Evaluation)

We evaluate the user's perception of the system in real or simulated environments through:

- Behavioral Feedback: Click-through rate (CTR), skip rate, and collection rate.
- Subjective Feedback: User questionnaires rating relevance, satisfaction, and diversity of recommendations.
- Conversion Behaviors: Whether users try, order, or save the recommended dishes.

Evaluation Unit:

Metrics are calculated per user and averaged across all participants to reflect personalization capabilities.

(4) Trade-off in Metric Selection

- Precision vs. Diversity: High precision may lead to homogeneity; diversity ensures variety.
- Novelty vs. Recall: High recall might rely on popular items; novelty ensures exploratory recommendations.

We use Precision@4 and NDCG@4 to evaluate accuracy, and Coverage to measure diversity, aiming to balance personalization with novelty.

(5) Computational Response Mechanism

- Offline Stage: Train/update user and item embeddings weekly.

- Online Stage: Retrieve Top-N candidates based on real-time behavior, compute embedding similarity, and return recommendations quickly for near real-time interaction.

If temporal or contextual models are introduced later, incremental updates or streaming pipelines will be required.

(6) User Study Design

To supplement offline evaluation, we designed a small-scale user study:

1. Display Top-4 recommendation cards via paper or Figma prototype.

2. Invite 10 students to interact by clicking/skipping/saving the items.

3. Collect feedback through a questionnaire with the following questions:

- Do the recommendations match your interests?
- Are the results diverse and novel enough?
- Would you be willing to continue using this system?
- What improvements would you like to see?

This experiment collects both behavioral data (clicks/saves) and subjective feedback (questionnaires), providing a complete evaluation loop for further system optimization.