COMP 9727 Recommender systems Project: Project Design

Zhuangyuan Fan z5432930

Scope

1.1 introduce

The main idea of this project originates from my daily life experiences: I often feel helpless when deciding with choosing a restaurant. It is difficult to make distinguish between paid advertisements by merchants and genuine user word-of-mouth recommendations. Therefore, the core design scenario of this project are:

I. Daily meal scenario (not know what to eat)

II. Takeaway recommendation scenario (quick and easy ways to find delicious takeaways)

III. Healthy diet Scenario(health dietary choices to facillitate fat loss)

IV. Multiple modules with different flavor options(providing personalized restaurant recommendations)

It mainly targets the following user pain points:

- Difficulty choosing, especially when traveling or living in a new city, or when having limited time to select a restaurant.
- Lack of credible, personalized recommendations and skepticism regarding the authenticity of platform suggestions;
- People with special dietary preferences(e.g., Preparation for competition, religious beliefs or personal reasons) have difficulty quickly locating the appropriaterestaurant

We will create detailed profiles for each user include age, preferences, evaluating history(since we are not an order app), and dietary goals to generate tailored recommendations (see the Dataset section for details)

User-name	Age	Preferences	History	Goals
user1	20	burger	KFB,MbDoalds	NULL
user2	30	pizza	Dcmino	enjoy
user3	25	salad	xxsaladbar	lose weight

(this is an example of user data, In actual project, preferences and history can include multiple entries)

1.2 target and example of using environment

This project is targeted at the long-term of serving users of all ages; however, given the data volume constraints of the start-up phase, three different states of gradual expansion have been defined:

State	User-Age	Recommended Strategies
State 1(Begin)	User aged 18-35	Popular resturant among young users, healthy restaurants, and late-night snacks
State 2(Medium-term)	User aged 10-50	low sugar low oil restaurant for adolescent and recommended family restaurant for adults.

State 3(Long-term)	User aged 10-70	Recommended offering light, soft foods that are easy to chew for elderly users

Example Use cases:

1. “Late-night Snacks Module:”

   Priority is given to restaurants that are closer to user, haveclosing times more than 1 hour away, and offer delivery, combined with evaluation ranking to recommende.

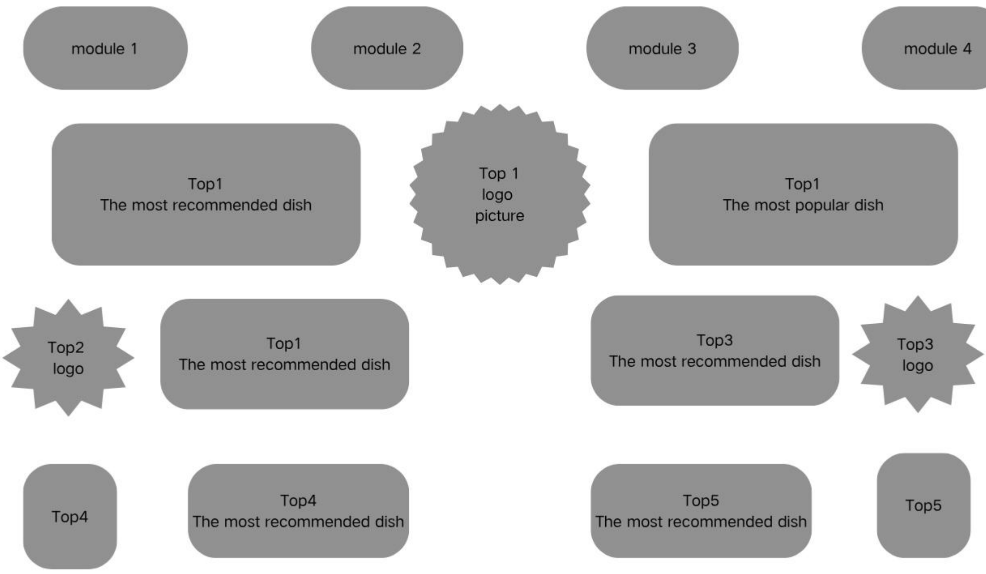
2. “Healthy Restaurants Module:”

   This module recommends restaurants or dishes that focus on low-fat and low-calories options. If a user order more than 3 time in week, the system will matches them with health tags and their personal health goals.

1.3 UI and user feedback

This project plan supports both web and mobile applications. When a user uses the system, each recommendation board displays the first 20 restaurants by default; each user's scroll down action then loads an addition 10 recommendations.

The interface is designed to be very easy to use. On the web version, users log in, select their preferred module, browse restaurants of interest, or search for a specific restaurant. (forget to put the search tool in the below picture)



(The example of web Top5 in first page, other recommend will same top4 and top5)

In addition to ensure the authenticity of reviews, we ask that users include at least the following information in their submissions review:

- At least one photo of an authentic meal showing the restaurant’s name or logo
- A photo of the receipt(for authenticity verification)

Simultaneously, to further improve recommendation accuracy, users allow to turn on the 'dish rating' option, which allows them to rate individual dishes (this option is turned off by default to avoid impacting user experience).

1.4 data updata plan and cold start strategy

The data of this recommend system are updated in every Wednesday and Saturday, and the system uses a "time decay weighting" method, which gives hight weight to the most recent evaluations. This approach ensure timeliness and accuracy of restaurant recommendations(see the Method section for details):

- New evaluations are weighted highly; the more recent an evaluation, the greater its impact on the recommendation weights.
- Old evaluations receive progressively lower weights, and evaluations beyond a certain certain period receive very low weights.

The cold start strategy is specified below:

- ◆ For new restaurant(within two weeks of launch): recommendation rankings are limited to the top5 positions to ensure that enough genuine reviews are accumulated
- ◆ For new or anomalous users(suspected of fraudulent activity or other anomalies): review weight are decreased to avoid obvious misdirection of the recommender system.

2. Datasets

2.1 Sources of data

Sources of currently used dataset: from kaggle "Restaurant Reviews" dataset, we select the following:

- Evaluation number: approximately 10000
- Restaurant number: approximately 1500
- Member number: approximately 4000

Ideal data sources in real scenarios include:

- State 1(when user volume is small): crawl users' publicly visible evaluation in compliant manner from public review platforms
- State 2&3(when user volume is large): develop community platforms independently, directly obtain user evaluation through the web and APP, and conduct strict authenticity verification.

Additionally, to ensure that evaluation are authentic, we employ the following methods:

- Combined cross validation of data from multiple review platforms
- User-submitted consumption voucher review mechanism
- Offline blind testing for benchmark calibration

2.2 Archiving of data and preprocessing

For kaggle dataset and its associated keywords:

Class name	Who to use
Restaurant name	Unique of Restaurant identification, used to associate records with that restaurant.

Revierwer	Unique of user identification, used to construct user profiles and perform behavioral analysis.
Review text	User evaluation text can be used for senetiment analysis, keyword extraction, etc
Rating	User rating(1 = poor - 5 = excellent, use for model training and evaluation
Timestamp	Time of review, used to calculate evaluation weight.
Metadata	Additional information from evaluation using in cross validation

In real scenarios user and restaurant data archiving methods include:

User:

- Basic information: age, gender, current residence, place of birth
- Preference profiles: eating habits(e.g preference for healthy or heavy flavors), and historical review records.

Restaurant:

- Basic information: address, name, type, and price range
- User Feedback: Real-time rating, comment text, the number of users who have tried the restaurant.

Data Preprocessing:

- Text data: using sentiment Analysis, RNN or other NLP techniques to determine positive/negative sentiment, and extract keywords for dishes and restaurant features using TF-IDF.
- Time data: Transform to a same format, and apply a time decay weighting function to assign different weights to reviews based on their timestamps
- Rating data: detects rating anomalies (e.g. outlier data and noisy data) and flags or rejects them.

2.3 Abnormal Data Detection and Handling

Abnormal data definition include:

- Multiple evaluations of the same restaurant by the same users within a short period;
- Repeated submissions from same IP address or using the same consumption ticket;
- All rating are either the maximum or minimum without any reason(review text);
- Accounts with indicators of fraud (like registering for a short period of time submitting extreme ratings).

Handling methods:

- Reduce the weight of ratings from anomalous users (each detected anomaly halves the weight, cumulatively reducing it to as low as 1% of a normal user's weight).
- Automatic exclusion rating data based on predefined by rule (e.g., short-term multiple ratings);
- Manual review accounts flagged for severe anomalies.

Offline verification method:

- Regularly select top-ranked restaurants for offline blind testing;
- Compare offline results with online ratings
- Integrate offline blind test data into the model optimisation process to ensure recommendation better reflect real user experience.

3. Method

3.1 Rating System and design

For increase the timelines s and accuracy of restaurant recommendation, we designed a scorning system based on user evaluation time and user weights, implemented as follows:

1.Time decay function:

The latest reviews(up to 7 days old) are assigned a weight of 120%

For each additional 7 day's period beyond the first week, the weight decrease by 10%, until it reaches a minimum of 1%. All evaluation older than 6 months use a uniform minimum weight of 0.1%

This function effectively incentivizes restaurants to maintain high-quality food and service.

$$\mathrm{TimeWeight(t)} = \begin{cases} \max\{1\%, 120\,\% - 10\% \times \left\lfloor \frac{t}{7} \right\rfloor\} & t \le 180 \\ 1\%\,, & t > 180 \end{cases}$$

T is number of days elapsed since the review

2.New-user and New-restaurant Weight Handling:

New Users and new restaurant are assigned a reduced evaluation weight(e.g 50% of the standard weight) for the first two weeks.

From the third week onwards, the weights are restored to normal level.

3. Paid Collaboration Strategy:

Restaurant can increase their referral ranking weights by paying for cooperation; The maximum ranking is capped at 5th place to ensure fairness and preserve user experience. We will denote this paid-collaboration factor as $(x).

4. Main Algorithms:

Favorable Rate:

$$S^{+} = \left(\frac{\sum e_i^{+} \times w_i \times TimeW(e)}{N + K} \times (1 + 0.5\% \times C(i))\right) * g(p) + \$(x).$$

In this function:

- e_i^{+} is $i-th$ user's positive evaluation,
- w_i is user's weight,
- $TimeW(e)$ is time decay function,
- N is total number of reviews,
- K is smoothing constant.
- $C(i)$ is the number of time for this user reviews at this restaurant.

- $g(p)$ is favorable ration default 1.0
- $(x) is the paid-collaboration factor.

Poor evaluation rate:

$$S^{-} = (\frac{\sum e_i^{-} \times w_i \times TimeW(e)}{N + K} \times (1 + 1.5\% \times C(i))) \times P(n) - 0.3\$(x)$$

In this function:

- e_i^{-} is $i-th$ user's negative evaluation
- other same with above one
- P(n) is penalty function define $1 + N(1 - e^{-\alpha^{n}})$, where α control the rate of decay.
- If there are a large volume of negative reviews
- P(n) can increase N by up to 5.

Final Scoring algorithms:

$$\text{score} = 1 + 4 \times \frac{S^{+}}{S^{+} + S^{-} + \varepsilon}$$

This algorithm ensures that the final score lies between 1 and 5.

3.2 Recommendation System Method and Design

This project will choose 3 difference recommendation method to ensure system accuracy and precision.

Method 1: Collaborative Filtering

Collaborative Filtering (CF) analyzes historical user behaviour data to identify associations between user and items, thereby generating personalized recommendations. It employs both of the following CF approaches:

User-based CF:Identify users with similar evaluation histories and profile characteristics, then recommend restaurants favored by those similar users.

Item-based CF: Calculate similarity between restaurants using evaluation data, then recommend restaurants similar to those the user has historically preferred.

In practice, we design the following dynamic update algorithms:

User-dynamic update:

- If a user selects restaurants of a new type more than 3 time within a short period, we trigger the “taste-change hypothesis” and increase the recommendation weight for those new restaurant types.
- if user evaluation more than 5 evaluations for new type of restaurant, then system will set new type of restaurant for this user.

Restaurant-dynamic update:

- if a restaurant receives multiple recent negative ratings from a frequent patron, this restaurant’s recommendation weight is reduced, and it is added to the offline scouting list to further quality restaurant.

Method 2: Content-based Recommendation

For improve recommend accuracy, we also design a content based approach using menu data and content characterisation. The steps are:

1. Collect and tag each restaurant's menu dish, ingredients, flavour profiles and relevant metadata.
2. Use natural language processing(NLP) techniques (like TF-IDF, RNN models) to extract features from menu data and generate embedding vectors for each dish.
3. Computer similarity between user preference vectors and dish embedding (e.g cos similarity, dot product, or Euclidean distance, like ass1) to generate a ranking list of personalized recommendation.

This method may suitable for specialized modules such as healthy eating or late-night snacking that can provide accurate recommendations.

Method 3:Hybrid Recommendation:

Considering the limitation of a single recommendation algorithm, we propose a hybrid strategy combining Method 1 and Method 2:

First, using content-based filtering to preselect restaurant that match explicit user preferences (like healthy meals or other). This serves as Pre-Ranking phase in recommend system.

Next, apply collaborative filtering to the preselected set to refine the recommendation further based on user similarity and historical evaluation. This constitues the ranking and re-ranking phase in recommend system.

Additionally, during the initial stage when user data are sparse, greater weight is given to Method 2; as more behavioral data accumulate, the weight of Method 1 is gradually increased to enhance model accuracy.

4. Evaluation:

To ensure the effectiveness of the recommend system, we designed a comprehensive evaluation framework that assesses three dimension: model accuracy, user experience and practical application effectiveness

3.1 Model performance evaluation:

We will use the following standard metrics to evaluate the performance of models (all *k* default 20 each user scroll down action add 10):

Precision@k: the percentage of the top K recommended restaurant that are relevant to the user's interests.

Recall@k: the proportion of relevant restaurant that appear in the top k recommendations.

F1@k: Precision@k and Recall@k harmonic mean, provides a model performance

NDCG(Normalised Discounted Cumulative Gain): measures ranking quality by assigning higher scores to more relevant restaurants appearing earlier in the recommendation list.

RMSE & MAE: metrics for evaluating the accuracy of model prediction user ratings.

3.2 User Behaviour and Experience Assessment:

To optimise the user profile in real time, we also analyse the following online user behaviour metrics:

CTR(Click-through rate): the proportion of impressions that result in clicks, calculated as:

$$\mathrm{CTR} = \frac{\text{number of clicks}}{\text{number of exposures}}$$

Conversion rate: Measure the percentage of users who after clicking on recommended restaurant, to submit an evaluation.

User Taste Tracking: Records and analyses changes in user preferences over time to update user preferences and profile in real time.

In addition, we will conduct regular user surveys to collect feedback.

3.3 Offline Real Experience and Verification

For ensure the consistency between online recommendations and real-world experiences, we will conduct the following offline evaluation activates:

Regular on-site blind testing of top 10 restaurant in the main list and top 3 restaurants in each recommendation modules.

Design of on-site blind testing: randomly package food samples from various restaurants and invite participants to score them blindly to obtain unbiased feedback.

Special offline inspection of sample visits to verify alignment between online evaluation and actual situation.

Finally, we will combine the offline experience data with online user evaluation and conduct a comprehensive analysis according the weighting of 4(offline):6(online), to further improve recommendation quality and ensure consistency of user experience.