Meet Fresh Recommender System Report

Fenny Jin

Yunying Liang

Yangfeng Peng

Baomei Wen

Yibo Wang

Qiu Zheng

Introduction

Meet Fresh is a Taiwanese dessert restaurant chain with locations in Asia, Australia, New Zealand, UK, Canada and the United States. The corporate headquarters are based in New Taipei City. Meet Fresh was established in 2007, the chain specializes in fresh Taiwanese desserts including soft taro balls and delicate herbal jelly by utilizing traditional Taiwanese processes and selection of the finest ingredients. On the menu, customers can find soft taro balls with red beans, grass jelly with boba, purple rice with sweet potato, barley and boba. Other options include tofu pudding, egg waffle, hot almond soup or shaved ice. A selection of milk or herbal teas, as well as pineapple or winter melon teas, is offered as well.

A recommendation system is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. In simple words, it is an algorithm that suggests relevant items to users as shown in figure 1.

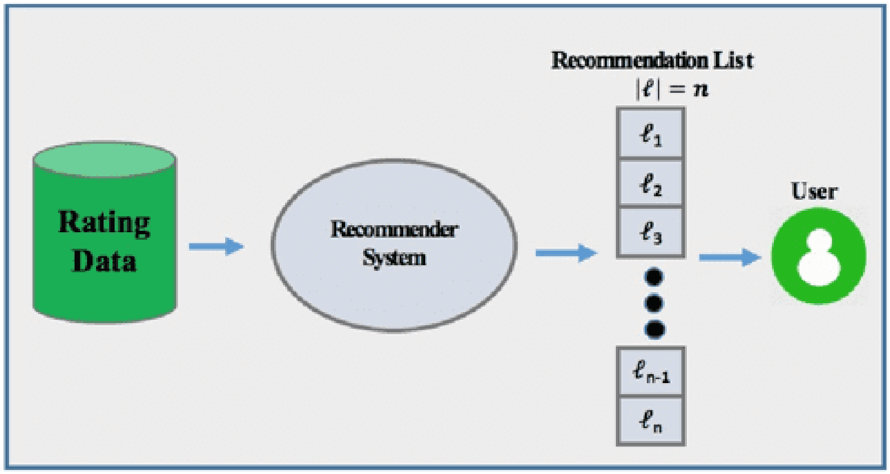


Figure 1. A general schema for recommender systems[1]

The objective of our project is to build a recommendation system for Meet Fresh. It is expected to improve the business performance and increase revenues or profitability of the company. In our project, we have used two different types of recommender algorithms, popularity based recommender system and collaborative filtering recommender system. Popularity based is a great strategy to target the new users with the most popular items sold and is very useful to deal with the cold start issue of a recommendation engine. It can not be personalized so the popularity recommender system recommends the same items for all users. Collaborative filtering is a technique that can filter out items that a customer might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. Most websites like Amazon, YouTube, and Netflix use collaborative filtering as a part of their sophisticated recommendation systems.

Discovery summary

For the Meet Fresh need-finding outcomes, we employed reviews, user interviews, and post-event protocols. The need-finding approach looks into the requirements and pain issues associated with ordering milk tea and desserts. It was determined that a recommendation system was required to simplify and facilitate ordering. A high-level summary of the need-finding exercises and results is provided below.

- Need-finding execution 1: Reviews observations
    - Participant observation was our initial need-finding exercise. As a starting point for the need-finding process, we gathered data from customer reviews and did some visualization. We concentrated on the negative reviews (with low ratings) to investigate the concerns and figure out how to construct a recommendation system. The store was out of various items or ingredients, according to the negative reviews, and the menu was not updated. A better platform with a real-time and improved menu is the opportunity behind these criticisms (ingredients).
- Need-finding execution 2: User interview
    - Workflow: form a script -> practice with group members -> add/edit for clarity -> finalize the script -> recruit users -> conduct user interviews -> summarize the results.
    - Users are pleased with the products they purchase from Meet Fresh and return on a frequent basis. The most crucial aspects while placing an order are the portion size, photographs, and descriptions.
- Need-finding execution 3: Post-event protocols
    - After user interviews, we asked the users to go to the Meet Fresh website and app to experience the ordering process and collected the feedback and pain points in 1-3 days. The comments and pain points primarily centered on the item recommendation system and the website and app's design flaws. The users also offered some suggestions for improving and increasing the company's earnings in the future.

Brainstorming summary

The group first conducted individual brainstorming first, then carried out the group brainstorming section to narrow down the ideas. Each team member shared their own ideas, and the group chose the three promising ideas and narrowed down to the most interesting one. The selection criteria the team used are:

1. Address customers' needs for ordering
2. Technology feasibility
3. Dataset accessibility

Despite having ideas to improve the sales and rating of Meet Fresh, we decided to narrow our scope for our recommendation system to improve customer experience when placing an order. Thus, we eliminated all the ideas regarding improving the store, employees, and product. We also took technology feasibility into consideration. While having think-outside-the-box ideas, there were certain technology and hardware limits that our group could not solve during this project, so we decided to move these ideas into future explorations. Dataset accesibility is also a big concern for the team due to the reason that certain sensitive information may dramatically improve the recommendation system, but may cause privacy concerns for the customers.

Low fidelity prototyping

Verbal prototype

Based on our brainstorming idea, the team used three prototypes: verbal prototype, paper prototype and wizard of oz. The first prototyping method we used is the verbal prototype. We narrated our idea to the interviewers and collected feedback to improve our prototype.

Paper Prototype

The paper prototype includes AI recommendation. Once the user selects the AI recommendation method, the app would lead the user to different scenarios on the second page, such as picking up a drink to go shopping. It also enables users to input anything they like in the 'Other' button. We expected this app could let the user input anything that fits their mood, such as 'I am happy because it is a good day.' The recommendation system finds a link between a recommended order and the user input. To go one step further, the system will also recommend the best snack to come with the drink or vice versa. In the brainstorming procedure, we agreed to develop a 'combo' option for the recommendation system. Although simplified, the paper prototype illustrates most functionalities proposed in the need-finding and brainstorming procedure. In addition, it targets either new or returning users, and it is easy to use. Most of the effort will be put into the AI recommendation algorithm to match the exterior and user profiles.

Wizard of oz

The functionality of the AI recommendation system is to recommend food or drinks to the users. The user can interact with the recommendation system through voice dialogues. The wizard of oz prototyping technique is used here to illustrate the process.

A "smart" recommendation system could facilitate the ordering processes. Users can customize their orders through their communications with the recommendation system. "No topping", "no iced", "no sugar", "pick up an hour later", "go back", various choices are available here, as well as other options if users have extra requests. What's more, the recommendation system could adopt optimization algorithms and adopt different scenarios to provide the best choice to each user. The goal of the AI interface is to provide the smartest choice in a timely fashion.

Evaluation: user interviews on the design

Our interviewees think this is an interesting system to use, but most of them are concerned about privacy issues, some of them are unwilling to share their activities and others will use it if the privacy policy is clear. Another issue pointed out by our interviewers is that our system may be time-consuming for customers to select so many scenarios. With this feedback, we are going to improve our system based on more external factors, such as weather, seasons, ingredients, and so on. We also need to consider the steps the customers are willing to take to make our system more efficient and user-friendly.

After the interviewees reviewed the paper prototype, the feature we failed to provide is the "back" bottom for navigating to the menu or the regular order page or going back to the last page. Some interviewers also wanted broader scenarios than those provided, while others wanted to group the scenarios into more general cases. This requires more user interviews to determine the right description and the number of scenarios. Additionally, the interviewees would not want to use the recommendation system to adjust the topping, ice, and sugar level. They said this is more of a personal choice themselves. An overall design deficit highlighted by the users is simplicity. Interviewees demonstrated the need for a system to be simpler and faster instead of comprehensive.

The wizard of oz prototype is a representation of what the product could be in the team's vision. To our surprise, many interviewers are attracted to the high-technology interface and have positive thoughts toward the recommendation system. However, after walking through the dialogue, they think the process is too tedious if they have to complete more than two steps to make a purchase. In some extreme cases, some users don't even want to use the recommendation system, they just want to click on the screen several times and finish the ordering processes.

Based on the user's feedback, the interviewees are more intrigued by the recommendation system that selects drinks or snacks based on external factors such as weather and season than a scenario-based system. Another critical finding is that interviewees could be driven away by a comprehensive but complex system. Simplicity and user-friendliness are critical UX/UI features we need to consider. Overall, the prototyping and follow-up user interviews provided a great opportunity to re-evaluate the features and refine our ideas.

Final product solution design before the technical development

We conducted and gathered research to cater to the people/customers for whom we are designing our product (recommendation system), including reviews observations, user interview, and post-event protocols. Then we created a point of view that is based on user needs and insights. It is to build a recommendation system for Meet Fresh, which also can improve the business performance and increase revenues or profitability of the company. In order to generate a wide range of potential solutions, design thinking and brainstorming sessions are implemented during this phase of product development. Address customers' needs for ordering, technology feasibility, and dataset accessibility are the three most important selection criteria to consider. After brainstorming, we decided to narrow our scope for our recommendation system to improve customer experience when placing an order. Based on our brainstorming idea, the team built three prototypes to test our hypothesis: verbal prototype, paper prototype and wizard of oz. The first prototyping method we used is the verbal prototype. We narrated our idea to the interviewers and collected feedback to improve our prototype. We created these prototypes to evaluate if they're on the right track, and it often sparks different ideas that we wouldn't have come up with otherwise to further streamline product development.

Creating and building great products (recommendation system) is contingent upon forward-thinking design implementation. Two different types of recommendation systems were developed here: popularity based recommender system and collaborative filtering recommender system.

We want to recommend the most popular items to customers, they are highly rated and highly ranked. That's what the popularity based recommender system do. It works with the trend, recommends the same items for all the users, it's not a customized system.

According to users' consumption records, we want to recommend similar items to similar consumers. Collaborative filtering can be used to filter out items that a customer might like on the basis of reactions by similar users.

At the interview phase, we got the private data for each interviewer. We are going to feed in the private data into the deep learning model and implement the Cross-Deep-Net retrieval model(CDN) to improve the prediction accuracy.

The item information is available in our project, content based filtering model is a good choice to deal with such kind of data.

In order to feed these machine learning models, different kinds of data should be collected. We'll talk about them much more later.

Product technical solution design

1. Data Collection (Qiu Zheng)

In order to support the recommender, we may need information for different users and available items (i.e., the menu). Possibly, user rating could be used as an indicator to rank the recommended items. Transaction data is also necessary to link the user and item information to identify patterns of the purchase behavior of different users.

(1) User information

The user information dataset was established based on an external source. The available dataset includes 92 users' information, including features such as gender, age, ethnicity, and preference for ice or hot drinks. Due to the data size limitation, we determined to mock more data based on these 'real' users. Two methods were used in this project.

(a) Random Sampling

In this method, the mocked dataset is developed through a sampling of a roughly uniform distribution based on the range of the original dataset. Ultimately, 500 users were generated.

(b) Oversampling

Bootstrapping and Synthetic Minority Oversampling Technique (SMOTE) [3] were tested in this part of the data processing procedure.

It was found that Bootstrapping, as a 'sample with replacement' methodology, increases the data size but fails to increase the variability of data. It is useful when estimating the data variance but not suitable for this phenomenon.

The other method, SMOTE, is usually used to oversample the minority population in a classification problem. The original dataset (92) data were taken as the minority class to apply this method. In the following, for a picked instance 'A' in the data, the algorithm randomly selects another instance 'B' that is within the set of the k-nearest (k usually takes 5) neighbor. A synthetic instance (convex combination) is created between instances' A' and 'B' in feature space. SMOTE increases the actual number of data and retains the distributions of each feature as the original data. Therefore, 500 users were generated using the SMOTE sampling method.

Different methods mentioned in the following may use either of the two datasets. Since these are all mock-up data, the trained model(s) did not apply to the real world. However, the performance of the recommender provides a reference to the feasibility of the product.

(2) Item Information

Meet Fresh website (https://meetfresh.us/) provides a menu for their products. There are 19 categories, including multiple drinks, snacks, and available toppings. Then the user can purchase an item with/without additional toppings. The total number of items is 155 (without the toppings).

A simple web-scrape engine was built in Python using the package Scrapy to facilitate the collection of the menu data. The scraped data contains features such as series name, item name, calories, and contains. Some contain information that is not available online and thus was filled up through manual input.

(3) Transaction Information

Due to the confidentiality of the business data, the actual transaction data was not available from Meet Fresh at this point. The transaction data was thus mocked through the dataset in Kaggle (Link: https://www.kaggle.com/datasets/marian447/retail-store-sales-transactions). The feature introduction and wrangling procedure using Spark with Scala are presented as the following:

Record ID: There are in total 130k data available in this dataset.

Date: The time for each of the records.

Customer ID: A virtual ID of each customer. There are 22k different customers, and then this information was converted to the user ID (1 - 500) in the user information dataset through modulo operation.

Transaction ID: An ID that indicates a specific user buys different items simultaneously.

SKU: A unique ID for each item in the dataset, which has a form combination of 5 characters. These characters can be capital A to Z or numbers 0 to 9. Firstly, each of the characters was transformed to UTF-8 Unicode. Next, it was assumed that the first character in the original dataset indicates a category of the merchandise; the second to the fourth character defines a specific item in a category; the last character refers to a possible additional topping. Using the modulo operation, the first character was mapped to the series; the sum of the second to the fourth character was mapped to an item in the corresponding series; the last character was mapped to a topping if the previous specified item is qualified to have topping (e.g., some kind of milk tea).

Quantity: The quantity of each item purchased.

(4) Review information

Additional information was gathered from the Yelp website. There are about 20 Meet Fresh stores in the U.S. The typical yelp academic dataset does not include those stores. To collect those data, an automatic download engine was built in Python using PyAutogui. This engine downloads the HTML text data from each yelp website page, and then the postprocessing code extracts review data from the HTML text data.

2. Popularity based recommendation system

Popularity based recommendation system works with the trend. It basically uses the items that are in trend right now. In our case, it ranks items based on the rating count. If an item is highly rated then it is most likely to be ranked higher and will be recommended. Figure 2 shows distribution of rating count grouped by items. We can see some items have been rated more than 3000 times. These items have been highly rated i.e. they are the most popular ones and will be recommended.

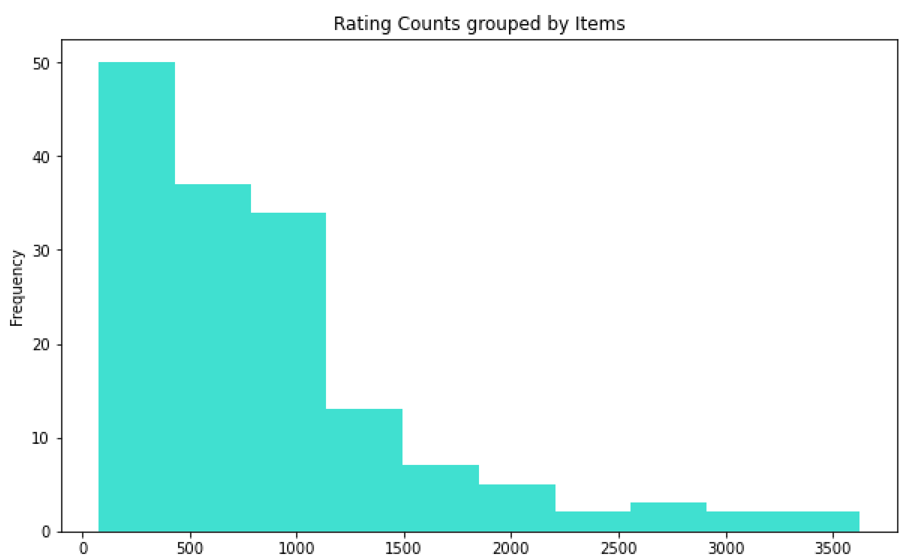


Figure 2: Distribution of rating count grouped by items

We split our dataset into training and testing sets by assigning 70% data points to the training set and 30% data points to the testing set. Then we trained the model using the training set and applied the model to the testing set. In this way, we can evaluate the performance of our model.

Since this is a popularity based recommender model, we get the same result for both users i.e. the model recommends the same items for all the users.

3. Collaborative filtering recommendation system

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). It uses similarities between users and items simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. So collaborative filtering is based on the historical data. The core assumption here is that the users who have agreed in the past tend to also agree in the future. In terms of user preference, it is usually expressed by two categories. Explicit Rating, is a rate given by a user to an item on a sliding scale, like 5 stars for Titanic. This is the most direct feedback from users to show how much they like an item. Implicit Rating, suggests users preference indirectly, such as page views, clicks, purchase records, whether or not to listen to a music track, and so on.

In our case, we used explicit rating values which directly rate from users to items ranging from 1 to 5. Then we explored a model-based approach which involves a step to reduce or compress the large but sparse user-item matrix.

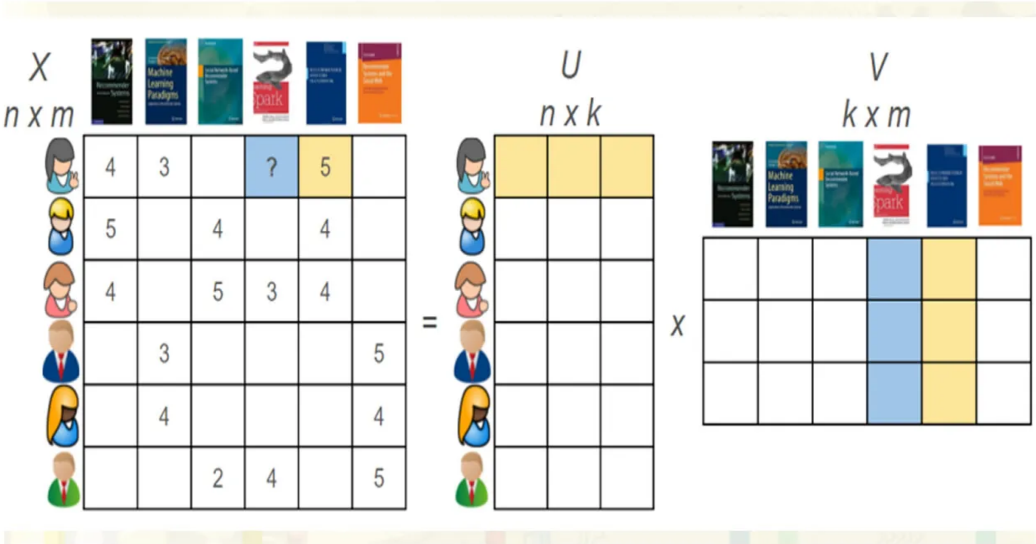


Figure 3: A schematic visualization of collaborative filtering[2]

One of the popular algorithms to factorize a matrix is the singular value decomposition (SVD) algorithm which We used in our model. SVD is a matrix factorization technique that is usually used to reduce the no.of features of a dataset by reducing space dimensions from N to K. It provides another way to factorize a matrix into singular vectors and singular values.

$$R = U\Sigma V^T$$

Based on Linear Algebra, any real matrix R can be decomposed into 3 matrices U, Σ, and V. In our case, U is an n × r user-latent feature matrix, V is an m × r item-latent feature matrix. Σ is an r × r diagonal matrix containing the singular values of the original matrix, simply representing how important a specific feature is to predict user preference.

When we implemented the algorithm by the Surprise library of scikit-learn, our data set was also split to the training and testing sets. We used the training set to train the model and applied the model to the testing set to predict the ratings.

We also evaluate the model using recall. The recall is the measure of our model recommending relevant items. Mathematically:

$$Recall@k = \frac{Recommended\ items\ that\ are\ relevant}{Relevant\ items}$$

An item is considered relevant if its true rating is greater than a given threshold. An item is considered recommended if its estimated rating is greater than the threshold, and if it is among the k highest estimated ratings.

Collaborative Filtering provides strong predictive power for recommender systems, and requires the least information at the same time. However, it has a few limitations in some particular situations.

- Collaborative filtering can lead to some problems like cold start for new items that are added to the list. Until someone rates them, they don't get recommended.
- Data sparsity can affect the quality of recommenders and also add to the cold start problem mentioned above.
- Scaling can be a challenge for growing datasets as the complexity can become too large.
- With a straightforward implementation, you might observe that the recommendations tend to be already popular, and the items from the long tail section might get ignored.

For the popularity based recommendation system, we also conducted user interviews with my two friends. They both said that they definitely would try the most popular items if they visit the restaurant or order dessert online on their first time.

4. Deep Learning Model: Cross-Deep-Net (CDN) Retrieval Model using TensorFlow (Qiu Zheng)

Recommender systems are often composed of two components:

- A retrieval model retrieval an initial list of candidates from the whole list of available items;

- A ranker model ranks the retrieval list, predicts the possible rating for each user and item pair, then recommends positively-rated items to users.

This work presented herein focused on the first stage, i.e., building a retrieval model to help Meet Fresh recommend products for their customers. The goal of the model is to recommend roughly ten items per search; then, if these items do not match the customers' needs, the engine will recommend ten more items.

A retrieval model facilitates efficient retrieval of candidates from large corpora by maintaining a two-tower, factorized structure: separate query and candidate representation towers, joined at the top via a lightweight scoring function. Therefore, in order to build a retrieval model, we need two sub-models:

- A query model computing the query representation (usually a fixed-dimensionality embedding vector) using query features, such as user information;
- A candidate model computing the candidate representation (an equally-sized vector) using the candidate features, such as item information.

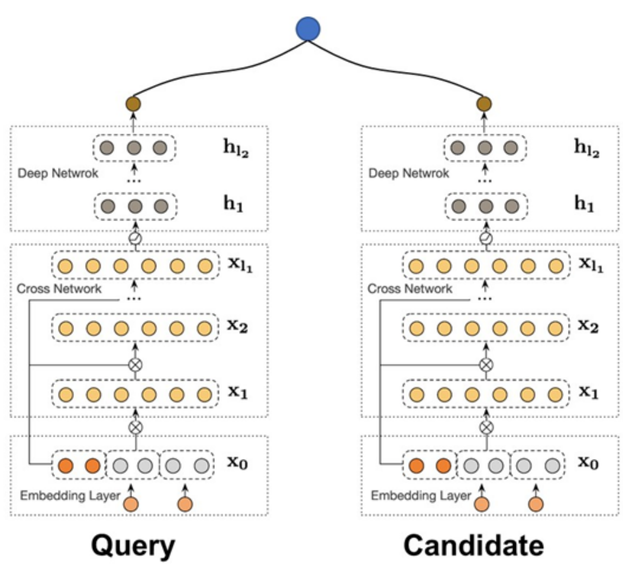


Figure 4. Illustration of the CDN Retrieval Model

This task used string and number data to construct embedding vectors for query and candidate models, then fed the embedding model into a cross net. The features and embedding methods are listed below. Lookup layers treat each of the same values as the same vector. Vectorization layers work with the long string features that will split a text string into multiple words (or characters) to do embedding. For example, if two items share a few same words in a feature but are not identical, they will be treated as two different values in string lookup embedding. However, they will have a few identical entries in vectorization embedding. Normalization transforms the features into normal distribution while discretization buckets continuous features by ranges. For instance, a predefined bucket can help discretize the user into a teenager, mid-age, and older adult.

Unlike the usual feedforward multilayer perceptron, a cross net explicitly applies feature crossing at each layer, so a cross net can have a smaller number of layers to achieve high accuracy. Based on previous studies [4,5], the projection dimension of cross-net should be smaller than half of the input size to reduce the computational cost. More specifically, in practice, it is observed that a quarter of the input size is to preserve the accuracy of a full-rank deep neural network.

Finally, the cross-net was fed into a deep network with a Dropout layer and ended up with a 64-dimension L-2 regularization layer. Both dropout and regularization layers are used to prevent overfitting.

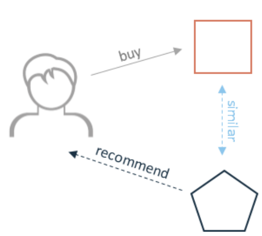
Table 1. Embedding features and processing methods

	Features	Embedding methods				
		String Lookup	Vectorization	Integer Lookup	Normalization	Discretization
Query Model	User ID	√				
	Age			√	√	√
	Gender	√				
	Ethnicity	√				
	Preference	√				
Candidate Model	Item Id	√				
	Ice/Hot	√				
	Series	√	√			
	Name	√	√			
	Contains	√	√			
	Calories				√	√

Solution development and results (3-6 pages)

4. Content Based Filtering Model (Yunying Liang)

Content-Based Filtering is one of the popular ways to recommend useful information to users. This technique is based on the description of the item and a profile of the user's preferences. It's best suited in situations where there is known information on an item, but not much known information about the user. That being so, the Content-based filtering approach teats recommendations as a user specific classification problem.



In our case, we don't have enough real user data but we have accurate menu data which allows us to build a content-based filtering model. The data we used was the menu data scraped from MeetFresh official website. We preprocessed the menu by using one-hot-encoding and word embedding techniques like tfidf for item descriptions. By creating a vectorized menu matrix, we could calculate the correlation of items based on contents and descriptions of items. The recommender will recommend every old user for the 5 most similar items based on their previous purchases.

1. Results for popularity based recommendation system

Popularity based recommender system is simple and very straightforward. But it is not personalized. It recommends the same items for all users. Table 2 shows the top 10 most popular items which are hot almond drinks, mini Q & melon jelly, fluffy oolong tea, coffee smoothie, creamy milk, and so on.

	user_id	Unique_ID	Name	score	Rank
0	6	16-1	Hot Almond Drink	2505	1.0
1	6	5-2	Mini Q & Melon Jelly	2471	2.0
2	6	13-3	Fluffy Oolong Tea	2232	3.0
3	6	15-3	Coffee Smoothie	2058	4.0
4	6	5-1	Creamy Milk	1965	5.0
5	6	12-2	Mixed Fruit Green Tea Slush	1893	6.0
6	6	13-7	Fluffy Mini Q (Mini Taro Ball) Winter Melon Tea	1871	7.0
7	6	16-3	Mini Q (Mini Taro Ball) Hot Almond Drink	1657	8.0
8	6	5-3	Boba & Caramel Pudding	1653	9.0
9	6	15-2	Coffee Milk Tea	1446	10.0

Table 2: The top 10 most popular items

Figure 5 shows a bar chart and scatter chart of rank for the top 10 most popular items. The score represents the rating times of one particular item. The more times it is rated, the higher it is ranked.

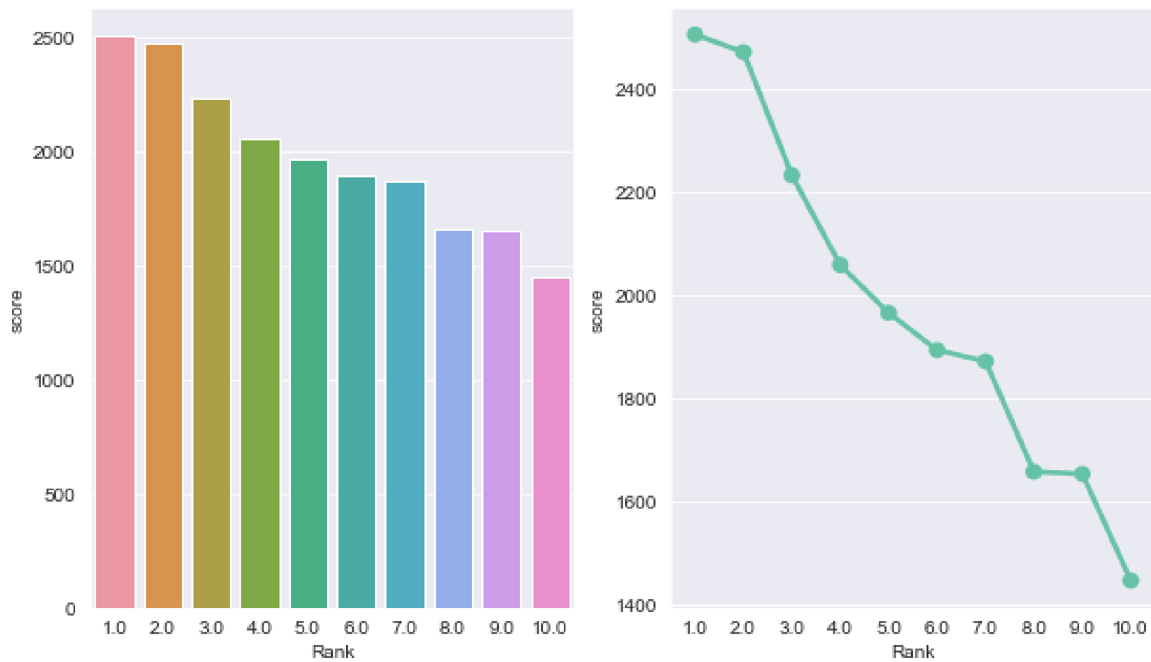


Figure 5: Rank of top 10 most popular items

We used Root Mean Squared Error (RMSE) as the metric measurement. The RMSE of the popularity based model is 1.53, which is small, the model works well.

2. Results for Collaborative filtering recommendation system

Model-based Collaborative filtering is a personalized recommender system, the recommendations are based on the past behavior of the user. In our case, each user has different items recommended to them as they are inferred based on the ratings provided by the similar users.

	user_id	Unique_ID	Name	Rating
0	8	7-5	Oolong Tea	3.810004
1	8	3-5	Purple Rice Soup Combo C	3.809523
2	8	18-2	Red Bean Almond Pudding	3.808935
3	8	8-4	Fresh Milk Herbal Tea	3.808759
4	8	3-6	Hot Almond Purple Rice Soup	3.804683
5	8	10-8	Fresh Milk Jin Xuan Tea	3.799658
6	8	6-1	Plain Tofu Pudding	3.799541
7	8	3-13	Purple Rice Drink with Boba & Fresh Milk	3.798839
8	8	2-2	Icy Grass Jelly Signature	3.796997
9	8	9-8	Jin Xuan Oolong Milk Tea	3.794788

Table 3: 10 recommendation items to the user 8

Table 3 shows 10 items which are recommended to user 8 based on users' rating behavior in the past. The highest rating item using the testing set to predict is item 7-5 which is Oolong Tea, followed by purple rice soup combo C, and Red bean almond pudding.

	user_id	Unique_ID	Name	Rating
0	300	9-9	Boba Milk Tea	3.828150
1	300	9-13	Pudding Milk Tea	3.809703
2	300	6-10	Taro Ball Tofu Pudding	3.809255
3	300	13-1	Fluffy Black Tea	3.804753
4	300	4-1	Red Bean Milk Shaved Ice	3.803188
5	300	2-4	Hot Red Bean Soup Signature	3.801411
6	300	7-5	Oolong Tea	3.800911
7	300	18-2	Red Bean Almond Pudding	3.799468
8	300	6-11	Taro Ball Tofu Pudding	3.799403
9	300	8-4	Fresh Milk Herbal Tea	3.799230

Table 4: 10 recommendation items to the user 300

Table 4 shows 10 items which are recommended to user 300 based on users' rating behavior in the past. The highest rating item using the testing set to predict is item 9-9 which is Boba Milk Tea, followed by Pudding Milk Tea, and Taro Ball Tofu pudding.

Here, MRSE and cross validation are used for model evaluation. Our SVD model has a test RMSE value of 0.438 and cross_validation RMSE value of 0.433. Recall is also a measure of our model's performance. The model achieved the Recall@10 of 0.8071, which means that about 80% of relevant items in the test set were ranked by our model among the top-10 items. The model works pretty well.

3.Results for Deep Learning CDN Retrieval Model (Qiu Zheng)

The embedding input dimension was determined as a value larger than the vocabulary size to enable a cold start when there is a new user. The output size was taken as 64 to achieve both relatively high accuracy and computation efficiency. Therefore, the generated input dimensions are 385 for the query model and 577 for the candidate model. Following the one-quarter rule, the projection dimensions for cross net were set as 100 for the query model and 150 for the candidate model. The other configurations of the final model are shown below.

In this model, the Retrieval layer defined the task of the training procedure and measured metrics, including multiple Top-K-categorical accuracies. This model has no time features, so the model uses a random split on the total of 133k data. The training, validation, and testing used typical 80-10-10 splits. Adagrad was used as the optimizer.

A procedure followed [6] was conducted to determine the learning rate. The trick is to train a network starting from a low learning rate and increase the learning rate exponentially for every batch. The result demonstrated herein indicates a reasonable learning rate is about 0.01 to 0.015, where the loss is relatively low. When the learning rate is larger than 0.015, the loss suddenly increases a big amount.

Table 5. CDN retrieval model configurations

	Query Model	Candidate Model
Embedding dimension	385	577
Cross Net projection dimension	100	150
Dropout layer rate	0.1	0.1
Deep net dimensions	256, 128	512, 256, 128
L2 regularization layer dimension	64	64

Figure 6: Choose learning rate

Finally, the learning started from 0.015 and used the callback function ReduceLROnPlateau to reduce the learning rate when validation loss entered a plateau. Early stopping was also considered in the training procedure to prevent overfitting.

After 436 episodes, the model converged and achieved 12.7%, 51.2%, and 84.5% for Top-10, Top-50, and Top-100 categorical accuracy respectively. The testing set also achieved a similar accuracy, indicating the model is robust for our sample recommender.

Table 6. Top-K categorical accuracies

Top-K Categorical Accuracy	Training	Validation	Testing	Random Selection
K = 1	1.2%	1.4%	1.2%	0.6%
K = 5	6.8%	7.0%	6.6%	3.2%
K = 10	12.7%	13.3%	12.6%	6.4%
K = 50	51.2%	51.3%	50.7%	32.2%
K = 100	84.5%	84.6%	84.3%	64.5%

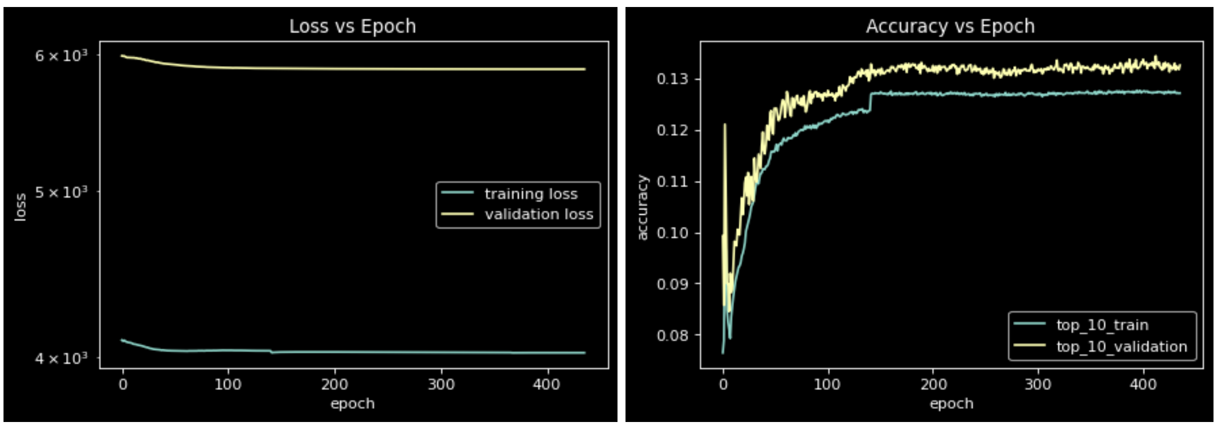


Figure 7: Training history: a) loss vs episode; and b) accuracy vs episode

Let us assume ten items as a group of recommendation items when the user received feedback from the engine once. Compared to a random selection, the CDN model improves 100% the chance to match the user's preference within ten items. If the user selects four more groups to recommend during the order procedure, the recommender will even achieve a 50% chance of predicting the customers' needs. The actual situation would be better than this result.

A randomly picked testing case recommended a user with ten items, including Mango & Passion Fruit Green Tea (3rd), Orange & Passion Fruit Green Tea (6th), Fresh Milk Green Tea (7th), and Fresh Milk Green Tea (9th). Compared to the customer's actual purchase item, Mango Green

Tea Slush, this result indicates the recommender predicts the user may like something made with green tea and would like to mix it with fruit flavor. Especially for the third item, a customer who chose Mango Green Tea Slush may also be willing to order Mango & Passion Fruit Green Tea. Another test case predicts the actual purchase item in the top-10 list. In addition, it is observed that the recommender captured that this user likes the Tofu Pudding very much and thus has 5 Tofu Pudding items in the top-10 list.

Table 7. Illustration of two testing results

Actual Purchase Item	Mango Green Tea Slush	Mini Q Tofu Pudding
Top-10 Recommendation	Fresh Milk Black Tea	Rice Ball Tofu Pudding
	Icy Taro Ball Combo C	Mini Q Tofu Pudding
	Mango & Passion Fruit Green Tea	Potaro Ball Tofu Pudding
	Red Bean Almond Pudding	Iced Fruit Black Tea
	Herbal Tea	Taro Paste Volcano Shaved Ice
	Orange & Passion Fruit Green Tea	Fluffy Winter Melon Tea
	Fresh Milk Green Tea	Potaro Ball Tofu Pudding
	Potaro Ball Tofu Pudding	Win Win (Boba & Lychee) Milk Tea
	Fresh Milk Green Tea	Red Bean Tofu Pudding
	Jin Xuan Oolong Tea	Jin Xuan Oolong Tea

The built model did require some information from the user, and no doubt this may not be easy to gather in practice. For example, this model used age as an input feature in the query model. Some users may not like to share such private information. In the solution, user information like this was taken as known due to technical difficulty. However, based on the results of the built retrieval model, Meet Fresh can achieve one (or more) trade by recommending their commodities to eight different customers, given that each recommendation lists ten items. Let us again assume that Meet Fresh feeds each customer a daily email with ten recommended items. Compared to a random recommendation, Meet Fresh might double their everyday trade number by using the recommendation system built here. After gathering more true-to-life data, the model is anticipated to achieve higher accuracy. This model will then be ready to serve in an application like a web page order system.

4. Results for Content-Based Filtering (Yunying Liang)

The content-based filtering is pretty straight forward and very useful when we don't have enough user data. The similar content in the items allows the recommender to recommend the best matches based on the taste of users for some specific ingredients.

	Name	Unique-ID	correlation	avg_rating	num_of_ratings
0	Icy Taro Ball Combo B	1–2	0.891326	3.170792	404
1	Icy Taro Ball Combo C	1–3	0.788955	3.143296	977
2	Icy Grass Jelly Combo A	1–4	0.757855	3.033333	270
3	Hot Red Bean Soup Combo B	1–11	0.689812	3.279446	866
4	Hot Red Bean Soup Combo C	1–12	0.652847	3.040724	221

Table 8. Similar food for 1-13 Hot Almond Soup Combo A

Figure 1 shows 5 most similar items of item 1-13 Almond Soup Combo A. We can see they are all combos. It indicates that the user who purchased hot almond soup combo A may try on other combos. Items 3 and 4 are both hot, so the user who purchased 1-13 may like hot soup combos so both item 3 and 4 may be good recommendations for users who purchased 1-13.

Summary (2-3 pages)

Model assessment

For the popularity based model, Root Mean Squared Error (RMSE) is used as the metric measurement. The RMSE of our model is 1.53, which is small, the model works well. It is also proved by the user interviews. In general, people tend to try the most popular items when they visit a restaurant or dessert shop for the first time.

In the top 10 most popular items, we noticed there are 8 items related to the different drinks which is quite surprising. People seem to be more interested in the drinks of Meet Fresh than their desserts. Maybe Meet Fresh can adjust their product design and marketing strategies to achieve a better performance.

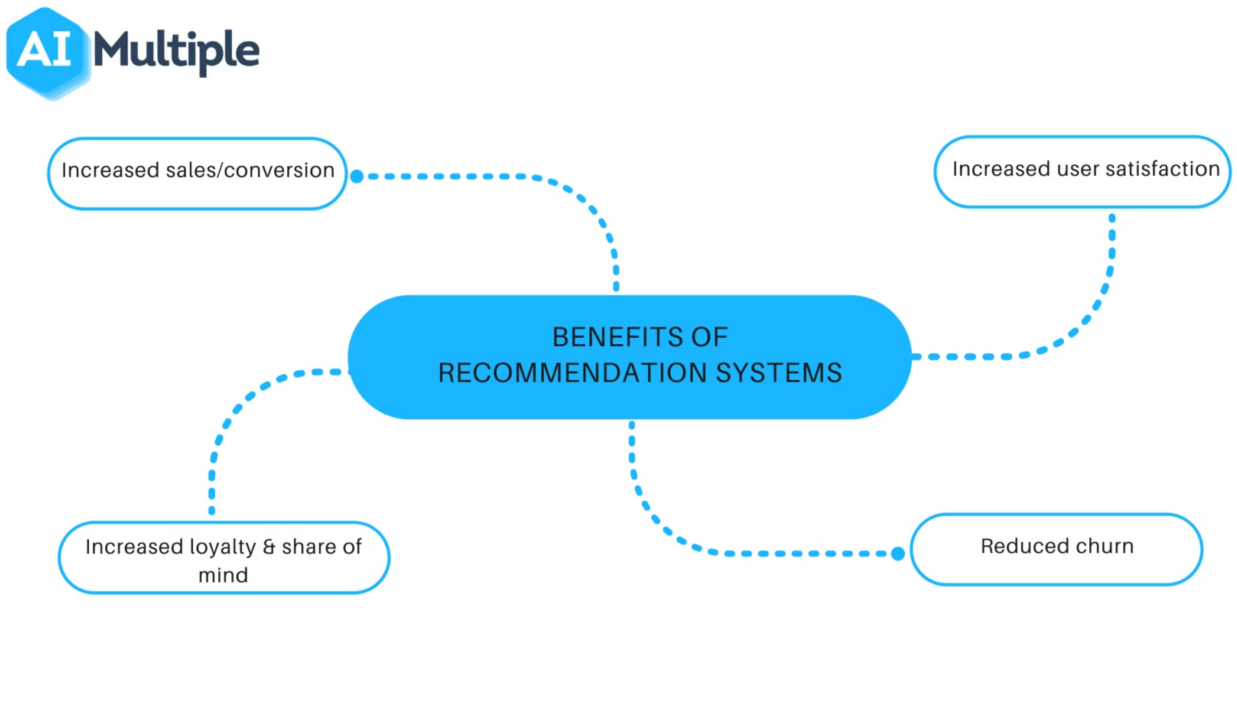
For collaborative filtering recommendation, MRSE and cross validation are used for model evaluation. Our SVD model has a test RMSE value of 0.42 and cross_validation RMSE value of 0.417. We also calculated the recall value. The model achieved Recall@10 of 0.8088, which means that about 80% of relevant items in the test set were ranked by our model among the top-10 items. Table9 shows the prediction ratings to the testing data set are very close to the real ratings.

uid	iid	r_ui	est	details
332	12-1	4	3.826586	{'was_impossible': False}
122	4-4	1	1.651725	{'was_impossible': False}
138	16-3	5	4.507333	{'was_impossible': False}
392	14-1	4	3.789763	{'was_impossible': False}
40	7-3	5	4.527209	{'was_impossible': False}
402	10-2	4	3.764331	{'was_impossible': False}
22	4-4	3	3.078964	{'was_impossible': False}
204	15-4	1	1.616849	{'was_impossible': False}
136	10-13	3	2.99226	{'was_impossible': False}
270	5-1	5	4.528693	{'was_impossible': False}

Table 9: An example of prediction ratings for 10 users and items

The results can help Meet Fresh get a better understanding of existing customers, and recommend the most potential personalized items to an existing customer to boost revenue of the company.

Buiness implication



One of the business problem we identified in the early stage is that customers have difficulty making an order despite large number of selection on the menu. Some customers would be hesitant to try toppings, drink or snakc they are not familiar with or they stick to the items they always order. By implementing the recommender system, Meet Fresh introduces the opportunity for customers to see the most popular items as well as personal items. By making successful and new recommendations, this would increase customer loyalty and share of mind. Customers would be more willing to try items they have not tried, if they have a positive experience, they would come back to Meet Fresh for more. This path would not only lead to increase in user satisfaction, but also improve sales in store. The other business impact the recommender system will bring is that it is a great way to re-engage customers to prevent churn. Coupled with coupons or discount can increase more profitability of conversion.

Next steps

The bigges limitation we have for developing our model is accessibility to real Meet Fresh transaction and customer info dataset. Because the dataset are mocked based on hypothetical transaction and customer info, our model's accuracy is limited. Thus, a more relevant and predictive recommendations would need more accurate transaction data as well as knowing customers' interests and preferences. In our initial brainstorming session, we envisioned that we would be able to reach existing customers or potential customers through a range of channels starting with social media, then to email or phone number. This future recommender system will allow Meet Fresh to target new user segments on website and social media. Meet Fresh could also promote new items, a specific range of merchandise to record customer's footprint as it occurs. In our product design, we also had hope to make the recommendations also based on geological location and weather information of the store. Adding weather information and geological locations would be a customized program for all the in-store retail experience that we hope to achieve in the future.

Final thoughts

In order for the recommender system to be practically successful, breadth of data and depth of data are two critical factors. For the case of Meet Fresh, the recommender system wold have a higher chance of success once it has established some customer base. If the business is only serving a handful of customers, considering the financial and operational cost, it may not be beneficial to implement such high tech solution. Additionally, have single data point on each customer is not sufficient or helpful for a recommender systems. Having customer preferences and consistent transaction history can guide towards a more accurate recommendation.

References and Acknowledgement

1. Himan Abdollahpouri, Popularity Bias in Recommendation: A Multi-stakeholder Perspective, 2020, ArXiv

2.https://www.slideshare.net/seldon_io/tensorflow-london-by-oliver-gindele-recommender-systems-in-tensorflow

3. He, Haibo, and Yunqian Ma. *Imbalanced learning: foundations, algorithms, and applications*. Wiley-IEEE Press, 2013.

4. Wang, Ruoxi, et al. "DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems." *Proceedings of the Web Conference 2021*. 2021.

5. Wang, Ruoxi, et al. "Deep & cross network for ad click predictions." *Proceedings of the ADKDD'17*. 2017. 1-7.

Annex

Meet Fresh Need-Finding Report

Group 4: Fenny Jin, Baomei Wen, Qiu Zheng, Yibo Wang, Yunying Liang, Yangfeng Peng

Abstract. This report includes the need-finding results using reviews observations, user interviews, and post-event protocols for Meet Fresh. The need-finding process investigates the needs and the pain points of ordering milk tea and desserts. The need for a recommendation system that can simplify and facilitate ordering was identified. The below provides a high-level summary of the need-finding exercises and results.

Need-finding execution 1: Reviews observations

Our first need-finding exercise was participant observation. We collected data from customer reviews and did some visualization as a start point of the need-finding process. Specifically, we focused on the negative reviews (with low ratings) to study the complaints, where we might seek a way to develop a recommendation system.

There are in total 9101 review data collected from yelp. These reviews covered 38 cities in 13 states. Ratings ranging from 1 to 5 show that 2749 reviews are negative (rating is 1 or 2) and 4929 reviews are positive (rating is 4 or 5). The average rating based on all reviews is 3.3.

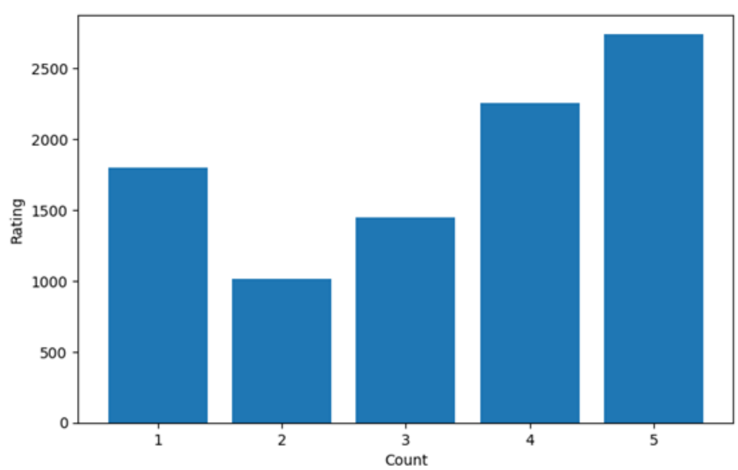


Figure 1.Rating counts

Looking at cities having more than 100 reviews, Chapel Hill is the best store, with the most positive reviews and the least negative reviews. Figure 2 is plotted in order of performance, ranging from good on the top to bad at the bottom. Figure 3 shows a word cloud in negative reviews. Some findings are listed below:

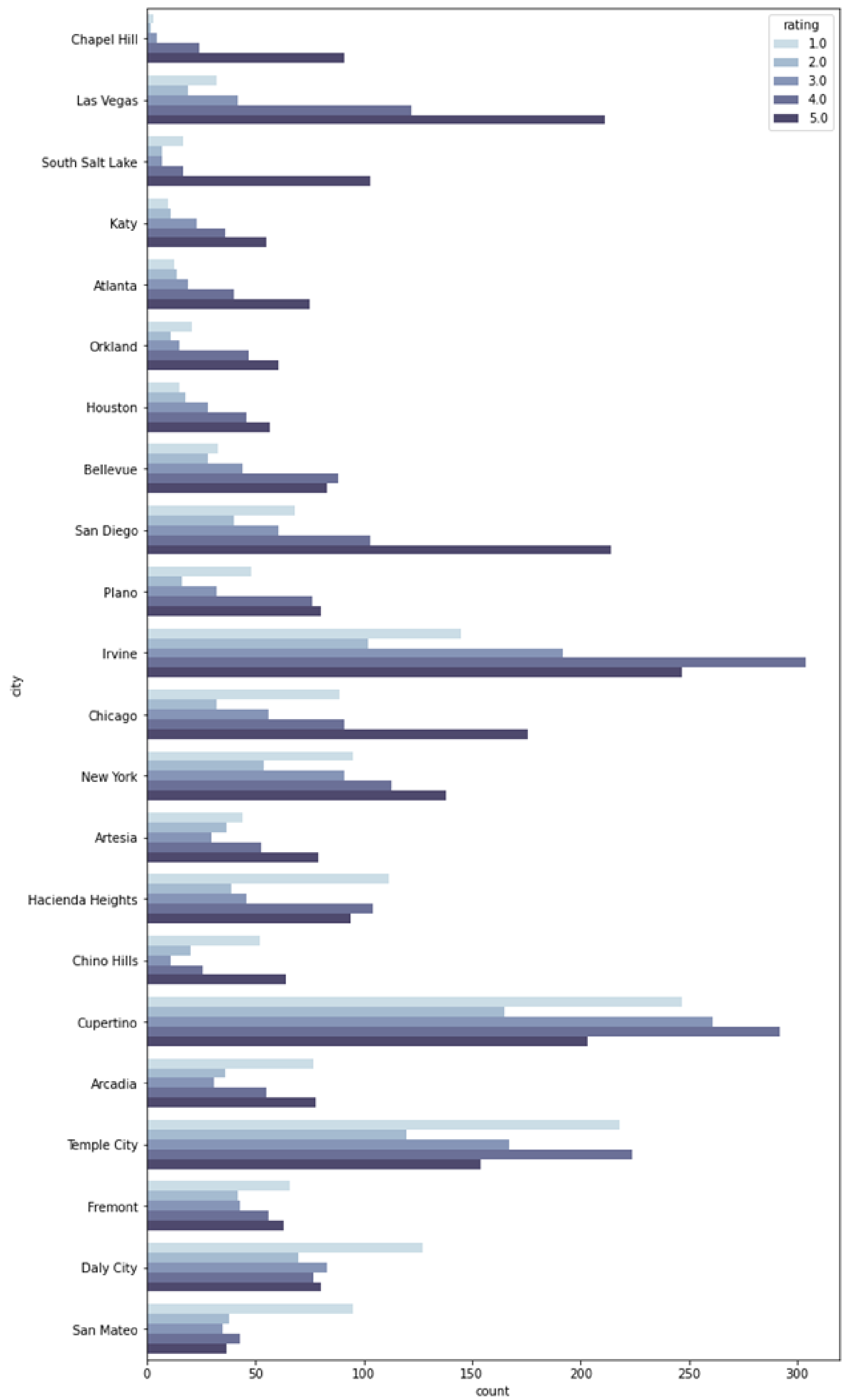


Figure 2.Rating counts for different cities

Figure 3. Word cloud of negative reviews

1) Those words considering customer services, including "cashier", "staff", "customer service", "worker", and "manager" are mentioned in more than 50% of the negative reviews.
2) Waiting time is the second issue mentioned in most of the negative reviews. 43% of the negative reviews include words such as "minutes", "busy", "slow", and "wait". Some of the reviews are like:
    a) "...We waited a good 30 minutes, and I was extremely sad to open my grass jelly to find a sad amount of barley and bits of mung bean."
    b) "... The place is pick up only, we waited about half-hour after ordering. ..."
    c) "... I ordered 2 mochis and a drink; it's currently 40 minutes of waiting for my order. ..."

    Many people are unsatisfied with the waiting time, which indicates the need to develop a better platform for ordering.
3) About 15% of the negative reviews indicate the issue located in the menu. Those reviews include words such as "menu" or "ingredients". These reviews could read like:
    a) "The rice balls, taro, and other ingredients were all absent...."
    b) "We did not because you guys ran out of the ingredients and did not tell us about it. ..."
    c) "... They had a limited menu selection when we visited and it was clearly not indicated on any menu. ..."
    d) "... Their menu seems full of options but it's not as the majority of options available are limited on certain days. ..."

e) "I Used To come here once a while to get these grass jelly desserts. However I found out today they've changed their menu. ..."

These reviews majorly complained that the store was out of some items or ingredients that they did not update in the menu. The opportunity behind these complaints is a better platform with a real-time and better menu (ingredients).

In terms of recommendations, the reviews also indicate that the most popular items are icy grass jelly, shaved ice, and signature items, based on the findings in Figure 4. It is noticed that the red bean soup and hot grass jelly appear the most in negative reviews.

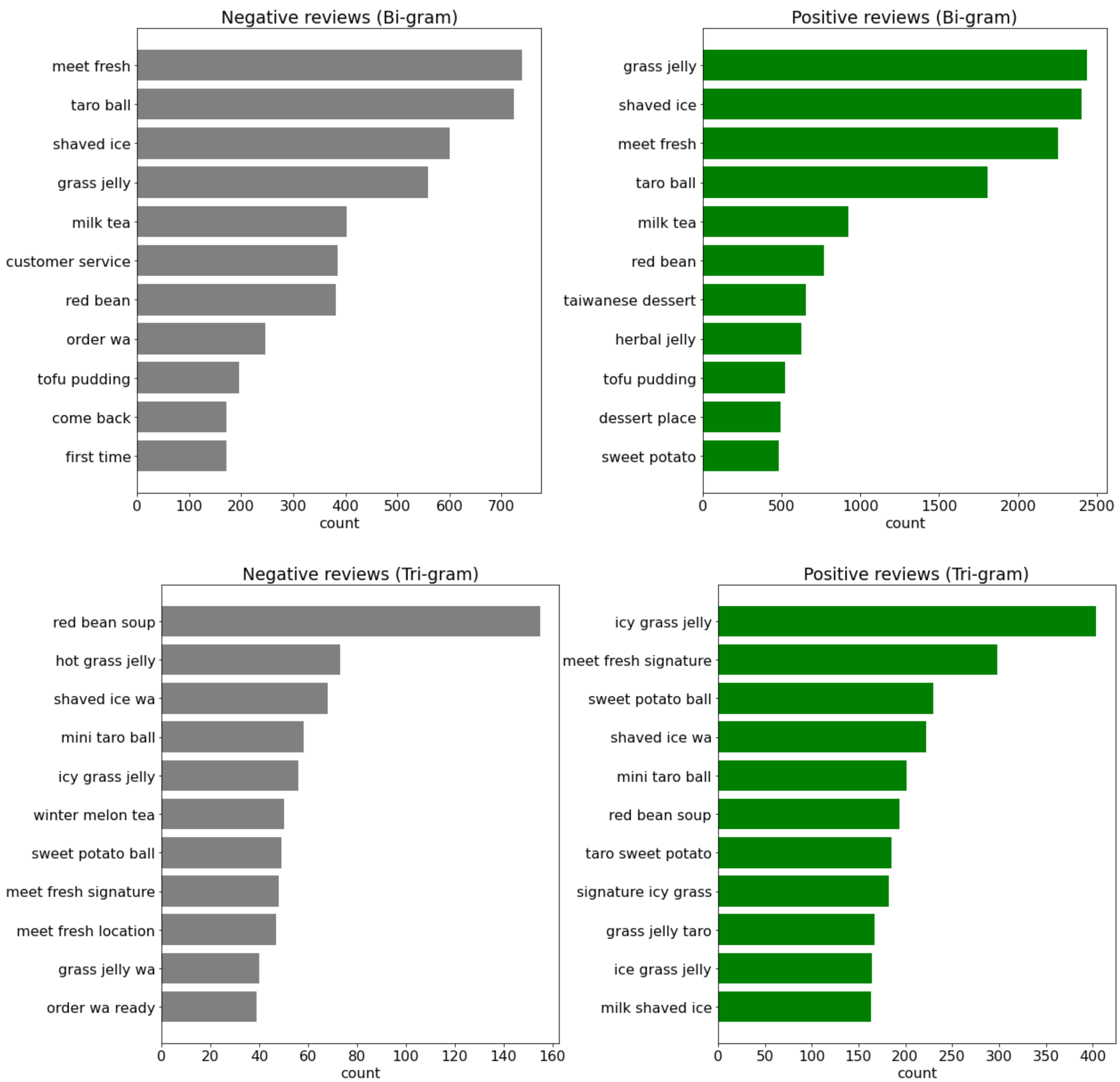


Figure 4. Bi-grams and Tri-grams word frequency in the reviews.

Need-finding execution 2: User interview

Workflow: form a script -> practice with group members -> add/edit for clarity -> finalize the script -> recruit users -> conduct user interviews -> summarize the results

Refined script:

- Greeting and brief intro to the interview
- Basic user info: age, gender, occupation, location, workstyle
- How often do you drink milk tea or have dessert?
- What is your favorite thing to order and why?
- Do you know about Meet Fresh? If so, how do you know?
- What is your preference of ording? Dine in or take out?
- When do you like to have milk tea or dessert?
- When you go, do you go with somebody or by yourself?
- When you purchase, are you buying for yourself only or for somebody else?
- How many items would you order per person?
- What is the budget per person for your order?
- What helps when you are ordering?
- When you order milk tea or dessert, do you eat it as a snack or do you treat it as a meal?
- Would you order anything else with milk tea?
- What is your motivation for going into a dessert shop?
- How has the pandemic impacted your food options?
- What is the most difficult part of placing an order?
- What could be optimized when placing an order?

Our second need-finding exercise was user interviews. We recruited users who have experience ordering at Meet Fresh and conducted the interview. We had 8 users for user interviews and 5 users for finding the post-event protocols. The results are summarized in the Data Inventory session.

Almost all the users are less than 30 years old, and reside in the urban areas. For their purchasing behavior, they usually purchase Meet Fresh on weekends and prefer take out. Each person purchases no more than three items at each time.

Half of users purchased the Meet Fresh at least once per week, another half at least one time per month. Six of eight users would use no more than 20 dollars to purchase these items, two users would use $20-$50. As for awareness, half of the users know about Meetfresh from their friends, three by themselves, and one from family members. Additional purchases with milk tea was inconclusive as the decision was split between yes, no and depends. Additional interviews may be needed to determine the need as it could be crucial to our recommendation system.

Almost all the users treated the food from Meetfresh as snacks. This indicated the importance of portion to the customers. There is an equivalent amount of users who like to go to Meet Fresh or similar places by themselves and with friends. Users go to the brick-and-mortar store only exclusively for purchasing. This contradicted our initial assumption that a dessert shop could be a place for socializing. However, the user's choices have been singular under the influence of the pandemic and could change post-pandemic.

When it comes to decision making, pictures of the items play the most crucial influence for users. Almost all the users said pictures help them make decisions. Descriptions and prices are not the top priority, but still helpful. Half of the users claimed that descriptions and prices helped.

Overall, the users are satisfied with the product they purchase at Meet Fresh and have made regular visits to the store. When placing an order, the portion size, pictures and descriptions are the most influential factors.

Avoiding the bias

To avoid biases, the questions were designed to be open-ended and semi-structured. The script provided structured guidance for the interviewers, however, the interviewers did not have to strictly follow the order. The conversations were interactive and the users asked questions for clarification. The interviewers only took notes of the key points of the conversations. Additionally, the interviewers are not affiliated with meet fresh and we intentionally avoided mentioning our intention to develop the recommendation system for Meet Fresh when doing the interviews.

Need-finding execution 3: Post-event protocols

After user interviews, we asked the users to go to the Meet Fresh website and app to experience the ordering process and collected the feedback and pain points in 1-3 days. The personal and practical experience helped them get first hand information about pros and cons of the website and app during the ordering process. I also asked the users to keep exploring the website and app when they are available. After multiple experiences, they gave me more interesting feedback points.

We didn't recruit new users for this activity as it will take much more time to collect the information we wanted. One benefit of using the same users is that they will focus on the topics and questions we discussed, which makes their feedback more relevant and reliable. But meanwhile, this could introduce some potential bias. As we are going about needfinding, we want to make sure we take a broader approach, understanding the entire problem space with interests, not just focus on narrowly under user direction particular interface. We could actually draw some very useful insights based on the users' feedback. The major feedback points are listed below:

- Most popular items recommendation, new items recommendation on the home page.
- Daily deals recommendation, every day specials recommendation on the home page.
- No size options for most series (signature series, grass jelly, red bean soup and teas).
- Scroll down very long to check all the items, sliding left to right may be a better choice.
- No price on the main menu, no calories or confusing calorie labels.
- Boost recommendation means and ways (like WeChat official account) to increase the population of the potential users.
- Increase regional flavored new items to attract more users.
- No pick-up time options except asap when you check out by using a mobile app.

In conclusion, the feedback and pain points mainly focused on the recommendation system of the items and design drawbacks of the website and app. The users also provided some perspectives which could enhance and boost the profit of the business in the future.

Data inventory

Table 1 data inventory contains the synthesis from user interview and post-event protocols. The results of the interview also include pain points and user expectations. All recruited users are experienced customers and have had experience ordering them on a regular basis. The bias in the user interview is that the users recruited are affiliated with the interviewers. We were not able to interview users with diverse ethical backgrounds due to the accessibility of these users in a global pandemic.

Table 1: Data Inventory from need finding 2(user interview) and 3(post-event protocols)

Who	Details	User #1	#2	#3	#4	#5	#6	#7	#8
	Age	28	29	28	45	30	29	29	27
	Gender	Male	Male	Female	Female	Female	Female	Female	Male
	Ethnicity	White	White	Asian	Asian	Asian	Asian	Asian	Asian
	Occupation	Working profession al	Working profession al	Lawer	Financial Analyst	Postdoc	Student	Student	Working professional
	Location	Urban	Urban	Urban	Urban	Urban	Urban	Urban	Rural
	Commute or WFH	WFH	WFH	Commute	Commut e	Commute	Hybrid	Mostly WFH	WFH
Context	**Frequency**	2 times per week	once every 2 months	4-5 times per week	once per week	more than 1 times per week	once per week	twice per month	once per month
	Specific visiting time	Saturday AM	After work	After meal	weekend s	weekends	weekends	weeken ds	weekends or weekday night
	Preference and reason	Pastry and milk tea	Thai tea	Red bean soup	Double Taro	Icy Taro Ball Series	Icy Taro Ball Series	Hot milk tea	Non-sweet tea
	Budget	$10 - $15	$5	< $15	$30-45	$20-$50	$10-$20	$10-$2 0	$15
	Where do you know about Meet Fresh?	Friends	Friends	Friends	Family	By myself	By myself	By myself	Friends
	Dine in/Take out	Take out	Both	Dine in and take out	Take out	Take out	Take out	Take out	Take out

	Snack with milk tea	Depends	No	Yes	Yes	No	Yes	Sometimes	No
	items to order per person	1-2	1	2	1-2	>3	2 to 3	2	3
Goal	**Snacking**		Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Meal	Yes							
	By yourself	Yes	Yes	Yes				Yes	
	With friends	Yes	Yes	Yes		Yes	Yes		Yes
	Purchasing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Socializing								
Need	**Pictures help?**	Yes		Yes	Yes	Yes	Yes	Yes	Yes
	Descriptions help?	Yes	Yes		Yes		Yes	Yes	
	Price help?		Yes	Yes	Yes	Yes	Yes		
	Wait Time estimator	Yes							
Pain points	**too many options**	Yes			Yes	Make the purchase in store	Yes		
	long lines prior to ordering		Yes	Yes				Yes	
	not enough descriptions	Yes	Yes						
	reduce food waiting time		Yes	Yes					

Defining Requirements:

Based on the results of user interviews and post-event protocols, we are quite confident in our original idea about building a recommendation system which should meet users' needs in a fast, simple, concise way. We will consider 1) Most popular items recommendation 2) Daily deals/promotions recommendation 3) New items recommendation 4) Every day specials recommendation.

Continued need-finding:

The design is a life cycle from needfinding to brainstorming design alternatives to prototyping to evaluation. Needfinding on its own can be a cycle by itself. We would like to carry on the second round of user interviews when the time is right. This time we will more focus on quantitative questions like:

- What items do you want to be recommended to you every day?
- What new items do you want to launch?
- What daily deals do you think would be good and useful?

We can even browse other websites and apps to inspire more thoughts and ideas. We may have some surprising gains.

Group 4 Meet Fresh Prototype User Interview Report

Fenny Jin, Baomei Wen, Qiu Zheng, Yibo Wang, Yunying Liang, Yangfeng Peng

Abstract

The report documents the brainstorming results of the proposed recommendation system for Meet Fresh. After brainstorming and discussion, only 1 idea was determined to pursue further and 3 low fidelity prototypes were built upon it. Verbal prototype, paper prototype, and wizard of oz were used to represent the solution. User feedback was analyzed for each prototype to extract insights to modify the next iteration.

Brainstorming plan

To achieve the best result we conducted individual brainstorming first, then carried out group brainstorming later. The brainstorming plan is shown as below:

- **Individual brainstorming:**
    - Have 2-3 individual brainstorming sessions at different times. Each session shouldn't take too long.
    - Try to propose ideas as much as you can.
    - Try to propose ideas that use traditional and nontraditional tech.
    - Think about customers' needs.
    - Think about the company's revenue/profit.
    - Think about the scenarios that are beyond the scope.
    - Think about the customers who are beyond the scope.
    - Think about how to solve the problem more broadly/wisely. At the first stage, don't set any constraints to limit your imagination.

- **Group brainstorming:**
    - Glance through individual ideas.
    - Inspire more shining points through group discussion.

- Choose three promising ideas, then narrow down to the most interesting one.

Some ideas from the individual/group brainstorming list below:

1. Recommends items to users with personalized promotion.
2. Recommends customized items to users based on location, and season.
3. Recommends items to users based on ethnicity/cultural background.
4. Recommends ‘Frequently bought/order together’ item bundles with a special discount.
5. Recommends items with voice message/search.
6. Sort historical popular items & recommendations to new users; Recommend based on old users' historical data.
7. Say something to show recommended food: ‘I am happy’; ‘tea’; ‘going to work’; etc.
8. Link to Twitter or yelp, etc, to find history posts about the users; analyze users' positive reviews on food, extract menu and what they like, and recommend similar textures/tastes.
9. Drink scene option: drink for working/shopping/studying/party.
10. User face recognition to link users’ faces and their favorite food.
11. Recommendation changes daily, based on weather, your mood, and your favorite color.
12. Top 5 Low-calories food
13. Instagram connected.
14. Live stream from other stores.
15. Word clouds, use keywords description from customers to describe the ingredient or item (flavor, texture, look, taste)
16. Staff training and rating system to improve staff services.
17. Some specialties during holidays or seasons.
18. More stores in urban areas and universities.
19. More options when buying milk tea, such as 0 sugar, 25% sugar, half sugar, 75% sugar and full sugar, no ice, less ice, and full ice.
20. More attractive take-away packaging.

21. The location option (ZIP code): Meetfresh stores are located in urban areas, and people from rural areas have to take-out.
22. Many customers prefer purchasing the Mettfresh at weekends —— coupons & promotions on weekdays.
23. Separate the milk tea bar & snacks preparation area: the milk bar could be semi-open, pedestrians can purchase the milk tea on the side of the street, and it can attract more customers.
24. Pictures, descriptions, and prices are helpful for customers, especially the pictures. Adding this information into the menu.
25. Customers treat Meetfresh as a snack rather than a meal, launching some products like the meal?

Selection Criteria

The selection criteria the team used:

1. Address customers' needs for ordering
2. Technology feasibility
3. Access to dataset

Despite having ideas to improve the sales and rating of Meet Fresh, we decided to narrow our scope for our recommendation system to improve customer experience when placing an order. Thus, we eliminated all the ideas regarding improving the store, employees, and product. We also take technology feasibility into consideration. While having think-outside-the-box ideas, there are certain technology and hardware limits that our group can not solve during this project, so we decide to move these ideas into future explorations. Access to dataset is also a big concern for the team due to the reason that certain sensitive information may dramatically improve the recommendation system, but may cause privacy concerns for the customers.

Verbal Prototype

The first prototyping method we used is the verbal prototype. We used the following description to narrate our idea to the interviewers and collected feedback to improve our prototype.

"*We are going to build a recommender system based on scenarios. This system can recommend different products of Meet Fresh for different scenarios. Firstly, it can recommend drinks or snacks depending on what you are going to do, including: going shopping, studying/working, or hanging out and so on. After you make the selection, it will recommend the suitable drink or snack for your scenario. Then, it is going to recommend more specific drinks based on the local temperature. More than that, it will also recommend different toppings based on different seasons. Of course, after selecting your drinks, you can choose to let it recommend other products like snacks based on weather conditions. During the whole process, you can always let our system decide for you and it will randomly choose one of the recommended products."*

Our interviewees think this is an interesting system to use, but most of them are concerned about privacy issues, some of them are unwilling to share their activities and others will use it if the privacy policy is clear. Another issue pointed out by our interviewers is that our system may be time-consuming for customers to select so many scenarios. With this feedback, we are going to improve our system based on more external factors, such as weather, seasons, ingredients, and so on. We also need to consider the steps the customers are willing to take to make our system more efficient and user-friendly.

Paper Prototype

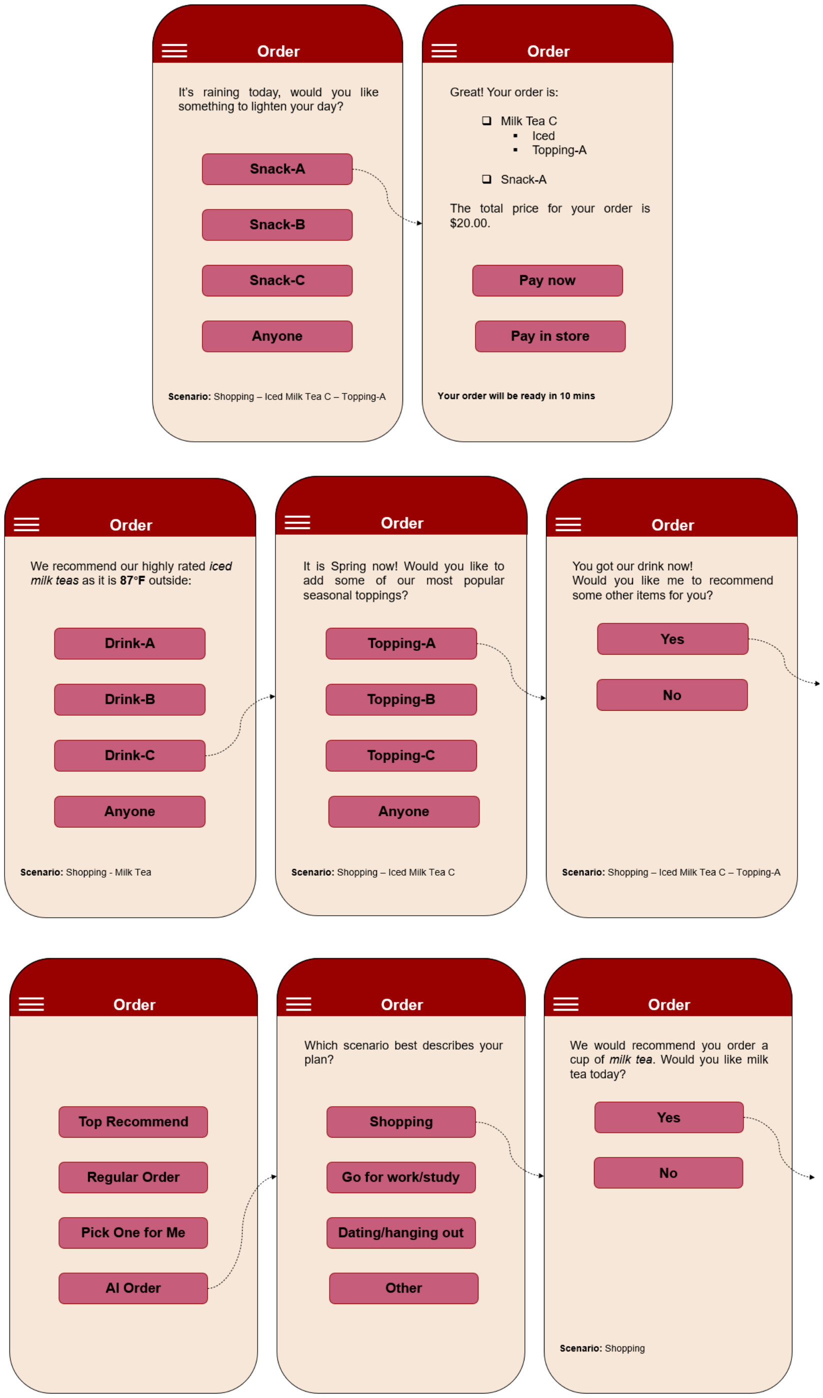


Figure 1: Paper Prototype

The paper prototype is trying to implement the recommendation system into the current interface, primarily focusing on the order step.

The first page enables the user to select different methods when placing an order. There are four types listed: top-recommended, regular order, random order, and AI recommendation.

- Items in this the top-recommended can be seasonal special or monthly/weekly best sellers. (Many interviewees stated that this can help new users know where to start and help returning users learn what is new.)
- The regular ordering system in this prototype enables users to choose base, topping, sugar, and ice levels step by step, incorporating the top-sells/random/recommendation in every step. Regular orders will include the menu with unavailable items grayed out for each store. We anticipated this menu could be updated every day for every store. As illustrated in the need-finding procedure, review observations showed that many negative reviews complained that they had waited for out-of-stock items after ordering.
- Random selection helps users with difficulty to choose among many items. We wish this selection could also function like a Mystery Box to prompt profit.
- AI recommendation selection refers to the recommendation system we decide to develop, which will be detailed in the following.

Once the user selects the AI recommendation method, the app would lead the user to different scenarios on the second page, such as picking up a drink to go shopping. It also enables users to input anything they like in the 'Other' button. We expected this app could let the user input anything that fits their mood, such as 'I am happy because it is a good day.' The recommendation system finds a link between a recommended order and the user input. The three listed options could be the most scenarios that customers come across.

The following could be step-by-step like a regular order, but with additional considerations of exterior environment or user profiles.

- On the fourth page of this paper prototype, we list the exterior environment to select the ice milk tea as it detects relatively high temperature. This step narrows down the menu to only a few options for the users. In addition, it can also go the other path named user profiles, such as a girl trying to lose weight; then it will recommend something with low calories.

- Next, the prototype showing the selection of toppings is based on the season. For example, customers may like grass jelly more than red beans during summer. Such a trend is going to be found in the data analysis step.
- We would try to add sugar or ice level options in the procedure based on either exterior environment or user profiles.

To go one step further, the system will also recommend the best snack to come with the drink or vice versa. In the brainstorming procedure, we agreed to develop a ‘combo’ option for the recommendation system. Although simplified, the paper prototype illustrates most functionalities proposed in the need-finding and brainstorming procedure. In addition, it targets either new or returning users, and it is easy to use. Most of the effort will be put into the AI recommendation algorithm to match the exterior and user profiles.

After the interviewees reviewed the paper prototype, the feature we failed to provide is the “back” bottom for navigating to the menu or the regular order page or going back to the last page. Some interviewers also wanted broader scenarios than those provided, while others wanted to group the scenarios into more general cases. This requires more user interviews to determine the right description and the number of scenarios. Additionally, the interviewees would not want to use the recommendation system to adjust the topping, ice, and sugar level. They said this is more of a personal choice themselves. An overall design deficit highlighted by the users is simplicity. Interviewees demonstrated the need for a system to be simpler and faster instead of comprehensive.

Wizard of oz

The functionality of the AI recommendation system is to recommend food or drinks to the users. The user can interact with the recommendation system through voice dialogues. The wizard of oz prototyping technique is used here to illustrate the process. A hypothetical will proceed as follows:

Context: Spring morning, user ordering processes, purchase recommendation.

AI: what scenario most accurately describes your following plan? Going shopping(to go), dating/hanging out with friends (dine in), or taking it home.

User: I would like something to go when shopping.

AI: We would recommend you to order milk tea. Would you like milk tea today?

User: Yes

AI: The current temperature is 30 degrees celsius. So hot right! We recommend our highly rated iced milk tea A, B and C. Would you like to order one of these or let me randomly choose between the three?

User: I chose A.

AI: OK. It is Spring now! Our most popular seasonal topping is A, B and C. Would you like to add some?

User: Yes, I want B and C.

AI: Sure! You got it.

OR

AI: Would you like me to recommend items based on weather?

User: Yes.

AI: It's raining today, would you like something to lighten your day? We recommend these A, B and C. Please choose one.

AI: Based on the time of the day, we recommend XXXX or xxxx

User: Perfect!

AI: Thank you. Your order is completed.

A “smart” recommendation system could facilitate the ordering processes. Users can customize their orders through their communications with the recommendation system. “No topping”, “no iced”, “no sugar”, “pick up an hour later”, “go back”, various choices are available here, as well as other options if users have extra requests. What’s more, the recommendation system could adopt optimization algorithms and adopt different scenarios to provide the best choice to each user. The goal of the AI interface is to provide the smartest choice in a timely fashion.

The wizard of oz prototype is a representation of what the product could be in the team’s vision. To our surprise, many interviewers are attracted to the high-technology interface and have positive thoughts toward the recommendation system. However, after walking through the dialogue, they think the process is too tedious if they have to complete more than two steps to make a purchase. In some extreme cases, some users don't even want to use the recommendation system, they just want to click on the screen several times and finish the ordering processes.

Reflection

Based on the user’s feedback, the interviewees are more intrigued by the recommendation system that selects drinks or snacks based on external factors such as weather and season than a scenario-based system. We need to conduct more user interviews to determine the need to modify or remove the idea of the scenario-based system. Another critical finding is that interviewees could be driven away by a comprehensive but complex system. Simplicity and user-friendliness are critical UX/UI features we need to consider. Overall, the prototyping and follow-up user interviews provided a great opportunity to re-evaluate the features and refine our ideas.