

UIUC CS 410 – Project Proposal

Customized Dish Recommendation at Restaurants Using Yelp Reviews and BestDish.py

Yu Liu [yul10] & Qinjingwen Cao [qcao11]

Introduction

People always have difficulty in selecting which dish to order in a restaurant they have never been to before. To address this pain point, we decided to develop a python program BestDish.py. This program utilized the abundant data points from the Yelp review database. We will first identify the dish name from the review and recommend the best dish to order based on popularity and rating. In our **BestDish.py v1.0**, we will provide both global dish recommendations and restaurant-specific dish recommendations. In our **BestDish.py v2.0**, we will ask users a few cold start questions and personalize our dish recommendations accordingly.

Team Members

There are two members on our team, Corn Spider.

Yu Liu, yul10@illinois.edu (captain) and Qinjingwen Cao, qcao11@illinois.edu

Background and Problem Definition

“What should I have for breakfast?”, “What should I have for lunch?”, “What should I have for dinner?” have become the top 3 most difficult questions in modern society. Having a great meal can significantly improve health, boost mood, and increase happiness. However, having a meal in the restaurant can also easily become a painful torture if you get awful dishes, bad experience and expensive check in the end. Without external input, people usually have limited restaurants/dishes in their mind to choose from, and the selections can be limited or outdated if chefs have changed the menu. To allow customers to keep up with recent menu info and decrease the risk of ordering dishes that taste terrible, we plan to develop a python program BestDish.py to help users find the best dish in each restaurant. We hope all our users, before going to bed, can have a big, satisfying smile on their face when they are thinking of the dish they had in the daytime, which is recommended by our BestDish.py.

In **BestDish.py v1.0**, we plan to include the following function:

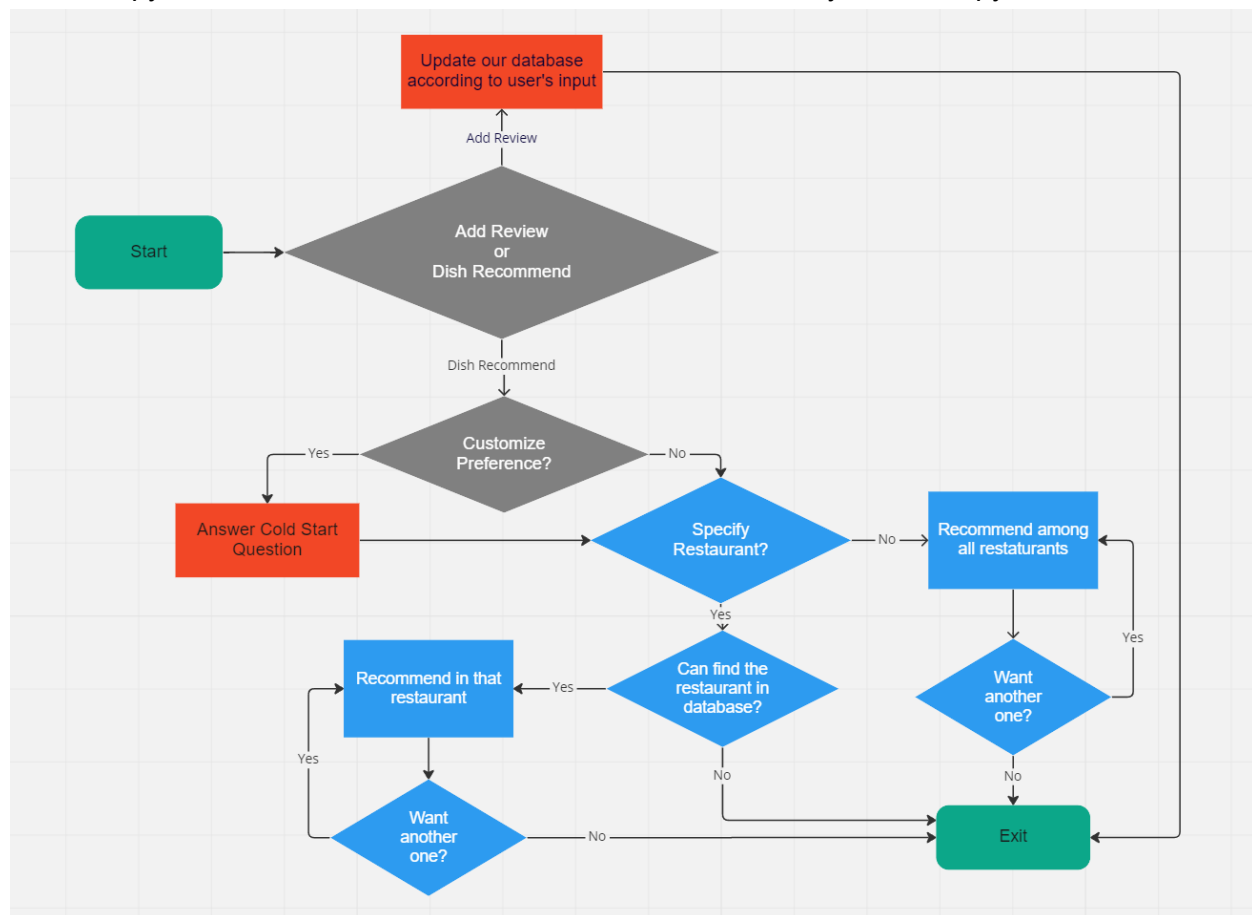
1. Based on the reviews on Yelp, our program will generate dish recommendations to users for those restaurants covered by the Yelp database.

2. The users can find the best dish of all the restaurants as well as the best dish at a specified restaurant.

In **BestDish.py v2.0**, we plan to include the following function:

1. We will support the user to add their review of any new/existing dish to our database.
2. The user can answer several questions at the beginning to include their personal flavor preferences. In this way we can personalize our dish recommendation.

Below is the interaction flow chart of BestDish.py. The blue part is the function covered by BestDish.py v1.0, while the red ones are the function covered by BestDish.py v2.0.



As we mentioned above, our primary programming language is python.

The database is <https://www.yelp.com/dataset>, the documentation can be found at <https://www.yelp.com/dataset/documentation/main>.

Below are details of the development plan of our core recommendation system.

In **BestDish.py v1.0**,

1. Data pre-processing
 - a. Remove stop words
 - b. Emoji replacement

2. Dish Name Identification
 - a. N-gram
3. Sentiment Analysis
 - a. Naive Bayes
4. Recommendation Design

We have several options here

 - a. Based on popularity
 - b. Based on popularity + rating
 - c. Based on popularity + rating + sentiment analysis

In **BestDish.py v2.0**,

1. Design of cold start question
2. Recommendation based on collaborative filtering

The expected outcome will be a functional python program for dish recommendation. We will evaluate our dish recommendation performance by manually reviewing the Yelp reviews, and compare with other recommendation algorithms if there are any.

Milestones

Week#	Task	Workload
Week9	Finalized project topic; dataset; project proposal	5h
Week10	Finish data pre-processing; Identify Dish names	8h
Week11	Finish Sentiment Analysis	8h
Week12	Publish BestDish.py v1.0, finish the project progress report	10h
Week13	Design cold start questions; Implement collaborative filtering.	8h
Week14	Publish BestDish.py v2.0, finish the project final report	10h

The total workload is 49 hours, which meets the project requirement.