CIS 550 Project Outline

Detailed Functionality Description

Group 79 Mar 19, 2023

Motivation

The aim of this project is to add social features and a friend recommendation system to Yelp. Yelp is a popular website and app that allows users to search for and rate local businesses, such as restaurants, bars, and shops. The website is known for its extensive collection of user-generated reviews and ratings, which can help others make informed decisions about where to go and what to do.

However, despite the wealth of information on Yelp, the platform currently lacks social features. Our project will aim to fill this gap by allowing users to connect with friends, see where they have been, and receive personalized recommendations based on their interests and preferences.

Note: Since the Uber Eats dataset we mentioned in our proposal does not contain enough overlapping restaurants' information as we have in the Yelp dataset. Therefore, we changed direction a little bit to focus on the Yelp dataset while adding a social component in our project, such as finding friends based on a user's taste.

Features to be Implemented

- The first feature we will implement is the ability for users to filter restaurants based on their input. This will allow users to find restaurants based on specific criteria, such as price range, location, and review.
- The second feature will be to display the top 5 rated restaurants in a specific zip code on the homepage. This will provide users with a quick and easy way to find the best restaurants in their area.
- The third feature will be to display restaurants that a user's friends have reviewed and sort them by their friend's review star rating. This will allow users to see where their friends have been and what they thought of the place.
- The fourth feature will be to find users who share similar taste with the current user via review stars. This will provide users with personalized recommendations based on their preferences and interests.
- **Possible Advanced Features (if time allows)**
  - One possible advanced feature is to sort restaurants by distance to the user, which would require the user's current location input.
  - Another possible advanced feature is to provide recommendations of similar restaurants by quantifying them and calculating similarity. This would involve

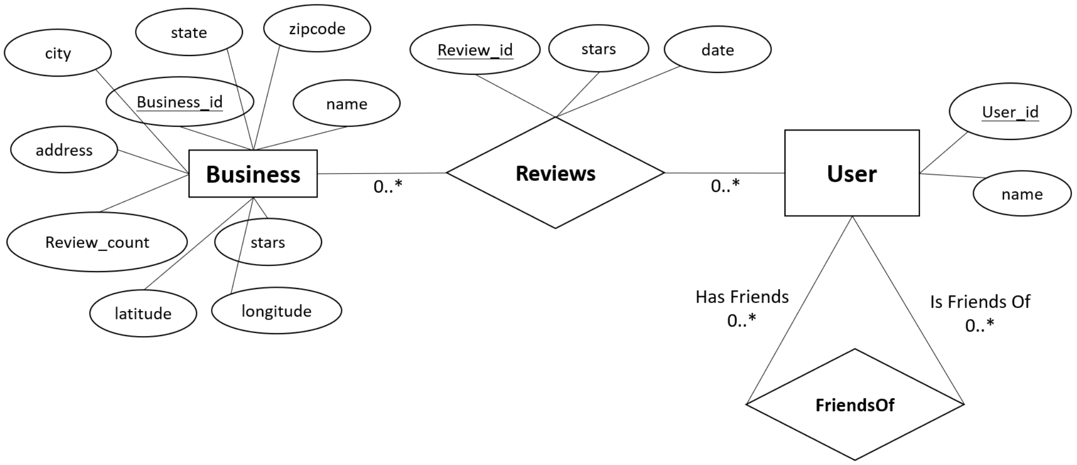
giving scores to each restaurant based on rating from Yelp, total amount of reviews, and other measures.

- Finally, we could find restaurants based on the user's given cuisine type, although this would require a significant amount of cleaning of the Business-Category data.

Pages and Descriptions

- Page 1: Best Restaurants in Your Area (Static)
  - This page will display the top 5 rated restaurants in the user's area based on their zip code.
  - This page may also display the top 5 rated restaurants with a set of zip codes selected by users. The user can input a set of zip codes and the page will display the top 5 in the combination of the input zip codes.
- Page 2: Filter Restaurants (based on user input)
  - This page will allow users to filter restaurants based on specific criteria, such as cuisine type, price range, and location.
  - Ideally, there will be a few slide bars on top of the page through which the users can select the ranges for the criterion, similar to the song page in HW2.
  - The user may also select how many results they want to display in one page, that is 5, 10, or 15 results.
- Page 3: Where Did Your Friends Go? (friends matching)
  - This page will display the restaurants that a user's friends have reviewed and sort them by their friend's review star rating.
  - In this page, the user needs to input his or her user ID, then the restaurants' information his or her friends reviewed will be displayed.
  - The user may also input the restaurant that he or she thinks is good, then the all of the users who rated highly of that restaurant will be displayed. This function enables the user to find friends who have similar taste.

ER Diagram



SQL DDL

```
Business(business_id, name, address, state, city, zipcode,
review_count, stars, latitude, longitude)

User(user_id, name)

Reviews(review_id, business_id, user_id, stars, date)
  business_id FOREIGN KEY REFERENCES Business(business_id)
  user_id FOREIGN KEY REFERENCES User(user_id)

FriendsOf(user_id_1, user_id_2)
  user_id_1 FOREIGN KEY REFERENCES User(user_id)
  user_id_2 FOREIGN KEY REFERENCES User(user_id)
```

Data Cleaning and Preprocessing (David)

For the data cleaning part, we first load our data into a python workbook via Google Collab. After loading the JSON file and defined the , we first conducted basic data cleaning procedure such as drop any rows with Null/NA values in it. This is because as we have a very large dataset, we have enough sample size (over 10,000 rows) for our project. After identify any missing value, we also looked for duplicated value for rows. Next, we identified businesses in Pennsylvania by using the function “df.iloc[:]” with conditions on restaurants/bar/food type of categories to select the businesses in Pennsylvania that are identified as food/restaurants/bar to us. We saved this table in **yelp_business_PA_df (~16,000 rows)**. Moreover, since our yelp_review table has millions of rows, we only selected those rows where its business id in the **yelp_business_PA_df.**

Next, we also dropped any missing rows that's in **yelp_review_PA_df**. Additionally, we conducted the same procedure to identify users which have used **Yelp** for providing a review for a restaurants/bars/food provider. Finally, we output the three files **yelp_business_PA_df, yelp_review_PA_df** and **yelp_user_PA_df** in the csv format before using AWS and MySQL.

Technologies

We will be using AWS, MySQL, Python/R, and React for our project. AWS will be used for hosting the application, MySQL will be used for the database, Python/R will be used for data cleaning and preprocessing, and React will be used for the front-end development.

Distribution of Work

Our team will be divided into front-end development, back-end development, and data cleaning/preprocessing. Each member will be responsible for their assigned tasks but will collaborate when necessary to ensure a smooth and successful project.

- Frontend/Client - UIUX *Weiyu & Ting*
- Backend/Server - SQL *Ruixi & Yang*
- Data preprocessing and joins - *Together*