Design Document

Yelp Reviews

Group Members: Tommy Chen, Emily Yamauchi and Sindhu Madhadi

Project Introduction

This project focuses on the Yelp data. The motivation behind our original idea is that the Yelp Fusion API does not support filtering businesses by rating. So we asked ourselves the following questions:
→ Can we build a visualization app that allows for easy filtering (rating, price, category)?
→ Can rating be predicted from the review text?
→ What else can we extract from the text data? How accurate is the restaurant categorization?
The python packages involved in this project are **Requests, BeautifulSoup, Folium and NLTK**.

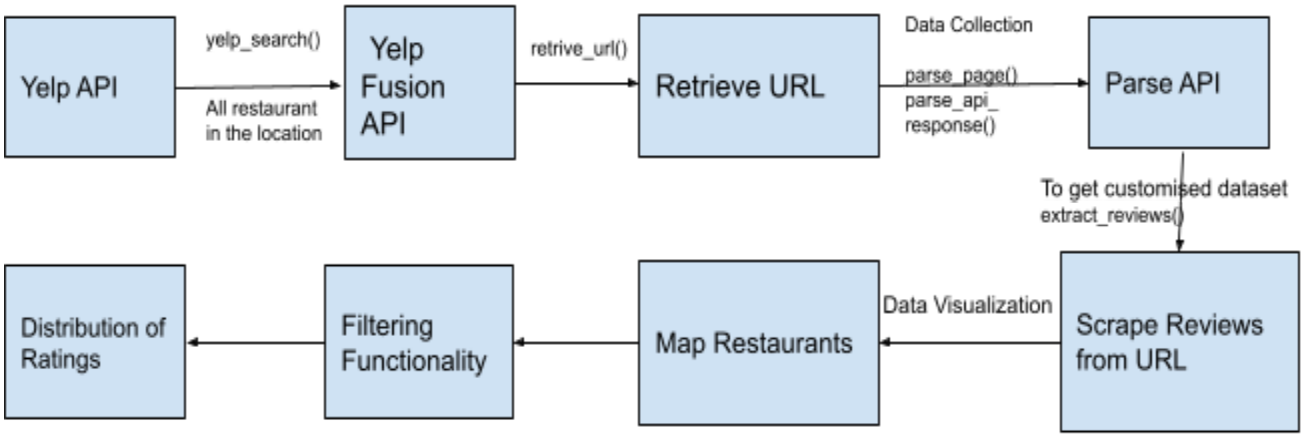
Data

The data is obtained from both the Yelp Fusion API and the web scraper that we implemented.

- Yelp Fusion API is a free, easy-to-use API to get information on businesses on Yelp.com. It uses a private key for authentication to authenticate all endpoints. The user could search for businesses based on the query constructed and the API response is in JSON format. You can find more info about the API here: https://www.yelp.com/fusion.
- The resulting dataset obtained from the API includes the following columns: Name, Category, Coordinates (longitude and latitude), Price, Rating, Url, and Review Count.
- However, this API has many limitations. It only returns 20 results for a single request and returns up to 1000 businesses for the same search query. It also has a limited request rate. Most importantly, the API only supports up to 3 reviews per business, which would be insufficient for our project.
- To resolve this issue, our group implemented a web scraper using the **Requests** package to retrieve HTML from url and the **Beautiful Soup** package to parse the HTML. The resulting scraper would find all the

JSON-formatted objects within the HTML and extract all 20 reviews on the first page of the business website.

Architecture



Components

- Data collection and parsing tool using Yelp Fusion API
    - Return JSON file of restaurant meeting criteria based on search parameters
    - Read JSON file to a pandas DataFrame, keeping just relevant data points (as specified in above “Data” section).
- Web scraper for the 20 newest reviews
    - Using the “URL” from the above DataFrame, scrape latest 20 reviews for the selected restaurant, and returns list of the reivews
- Data visualization map
    - Using the coordinate data from the above DataFrame, plot restaurants in a map visualization
    - TBD: Interactive visualization allowing filtering by category, price range, rating
    - TBD: Other visualization components in tooltip, showing review text data for each restaurant
    - TBD: Show separately distribution of ratings based on restaurants displayed in map
- Data visualization for text reviews

    - Using web scraper function, return a visualization for the text review (word cloud). TBD

Use Cases

Case 1: Search for and plot restaurant data based on user-selected parameters without filtering (same functionality as existing Yelp searches)

- Terms: “Pizza”, “Tacos”, “Boba”, etc.
- Location: “Seattle”, or specific neighborhoods such as “University District, Seattle”

Case 2: Search for and plot restaurant data based on user-selected parameters, but include filtering

- Same parameters as above Case 1
- Returned plots can be filtered by ratings (Yelp “stars”), price range, Boolean “Open Now?”, restaurant category/cuisine

Case 3: View latest reviews based on searches from either Case 1 or Case 2

- Using the search function from either Case 1 or Case 2, populate the map with restaurants meeting criteria. Upon selection, the user could view the latest 20 reviews for the selected restaurant.
- Visualization of the review text. Currently considering word cloud, but if time allows, attempt to extract keywords from the review list.

Testing

We have used a unit test framework to test our cases. A unit test folder is created, with the same structure as the application implementation folder. Consider the test cases as a python package same as the implementation folder. We are using a pycharm integrated development environment for testing the cases.The functions we are using are test_parsing, test_scraping .

Test Cases:

- In this scenario we tested the parsing function(test_parsing) to check whether the data is parsed into a panda dataframe correctly, by comparing the no of restaurant results in json response with the panda dataframe.
- We are working on the implementation of the rest of the test cases.