Restaurant Recommender - Functional Specifications

Group Members: Poornima Muthukumar, Juhi Choubey

Background

Recommendation Systems return the most relevant and accurate results (products, restaurants, books, travel plans, movies, tv-shows) to the user by filtering useful results from a huge pool of information. Recommendation systems discover data patterns in the data set by learning about customers' choices and produce the outcome that correlates to their interest or preference.

As a developer implementing a recommendation system can be a daunting task. With this project, we simplify integrating a recommendation system for recommending restaurants to users. Businesses can benefit from a good recommendation system as it increases user engagement and creates stickiness on the platform. In this project we build a python package that provides developers with the ability to easily integrate two kinds of recommendation systems - one is the collaborative filtering model and the other is a content based filtering model.

Goals

The goal of the project is to use the yelp dataset to build a restaurant recommendation system that recommends restaurants to users based on two different approaches

- The **collaborative filtering recommendation** model works by searching a large group of users and finding a smaller set of users with tastes similar to the particular user. It looks at the restaurant they like and creates a ranked list of suggested restaurants.
- The **content based filtering recommendation** model works by recommending restaurants to users based on similar restaurant categories and dominant topic keyworks, thus suggesting restaurants that align with a user's preferences.

Data Source

We use the Yelp dataset which has been made publicly available for personal, educational and academic purposes.

Data Structure:
Reviews

user_id	`string, 22 character unique user id, maps to the user in user.json`

business_id	string, 22 character business id, maps to business in business.json
stars	integer, star rating

Business

business_id	string, 22 character unique string business id
name	string, the business's name
categories	// an array of strings of business categories "categories": ["Mexican", "Burgers", "Gastropubs"],

attributes	// object, business attributes to values. note: some attribute values might be objects
	{
	"RestaurantsTakeOut": true,
	"BusinessParking": {
	"garage": false,
	"street": true,
	"validated": false,
	"lot": false,
	"valet": false
	},
	}

city	string, the city
stars	float, star rating, rounded to half-stars

Users

user_id	string, 22 character unique user id, maps to the user in user.json
name	string, the user's first name
average_stars	float, average rating of all reviews

Users

Users can range from developers to researchers, students or restaurants, or yelp or anyone with proficiency in programming knowledge. For the scope of this project we are scoping to 2 main types of personas described below.

Persona #1

Oliver How can I incorporate existing libraries to build a website to recommend restaurants to users traveling to a new city ?	**Behavior** ● Loves to develop web pages using existing libraries or API ● Prefers to focus on the front-end rather than backend.
Facts ● 25 years old ● UX/UI Developer at tripadvisor/expedia travel site.	**Goal** ● Building UI for a travel site. ● Save time by using existing library ● Build a web page to recommend restaurants for users traveling to a new city.

Persona #2

Liam How can I use existing libraries to incorporate additional features ?	**Behavior** ● Likes to explore existing libraries on Github. ● Adds additional features to the existing model.
Facts ● 35 year old ● Data Scientist working for a famous restaurant company.	**Goal** ● To understand how the existing model works. ● Learn by incorporating additional features to existing libraries.

Use Cases

1. Users will get 10 recommendations from the system based on their previous ratings.
   Training Input: Users, Restaurants, Reviews
   User Input: User ID
   Output: List of top 10 Restaurants.
   ML Algorithm - Collaborative / Content

2. Users will be able to provide a restaurant name and get similar restaurants.
   Training Input: Users, Restaurants, Reviews
   User Input: Restaurant ID, Top Results to return.
   Output: List of top (N) Restaurants.
   ML Algorithm - Collaborative / Content.