Week 3 PROJECT REPORT

This week's progress:

We used the reviewerId, restaurantId and rating to construct a sparse matrix, feed the data set into a collaborative filtering based algorithm , so that we can generate item recommendations for users.
We considered the first user as an example to see whether our recommendations are calibrated or not. Once we're familiar with the procedure for one user, we repeated the process for all of the users.
Here is the definition of the problem. We are given the distribution categories gg for each restaurant ii, p(g|i)p(g|i), what we are interested is whether p(g|u)p(g|u) is similar to q(g|u)q(g|u). Where:

- p(g|u)p(g|u) is the distribution over categories gg of the set of restaurants HH reviewed by user uu in the past.

$$p(g|u)=\sum i\in Hp(g|i)p(g|u)=\sum i\in Hp(g|i)$$

- q(g|u)q(g|u) is the distribution over categories gg of the set of restaurants II we recommended to user uu.

$$q(g|u)=\sum i\in Ip(g|i)q(g|u)=\sum i\in Ip(g|i)$$

→look at the first user
→ finding the index that the user interacted with,
→ Then mapping this to a list of Item, before that we need to first map the recommended index to the actual item_col/restaurandID first.

For the same user, we used the .recommend method to recommend the topn recommendation for him/her, note that we also passed in the original sparse matrix, and by default, the items/restaurants that the user has already reviewed are filtered from the list (controlled by a **filter_already_liked_items** argument, which defaults to True).

→it returned the recommended index and their corresponding score
→Mapped index to item
The next we defined a function to obtain the categories distribution for a list of items.
Given that we now have the list of interacted items and recommended items, we passed it to the function to obtain the two categories distributions.

Next Week plan:

Looking at the results above, we saw that according to p(g|u)p(g|u), the user has interacted with categories such as Cuban, AsianFusion, however, they are nowhere to be seen in the topn recommendation to the user.

To scale this type of comparison, we'll now define our calibration metric CC. There are various methods to compare whether two distributions are similar to each other, and we chose to go ahead with KL-divergence for the next week.