# Yelpflix: Creating Netflix-Like Recommendations for Yelp

By Nicole Lee, Scott Marler, Lisa Minas

## Recap of Project

We are creating a recommendation engine that will improve the Yelp customer’s experience by intelligently proposing restaurants on the customer’s homepage. Our recommendation engine will be reminiscent of Netflix’s movie suggestions: given the customer’s current location, personal preferences based on historical data, and similar users’ experiences, our engine will suggest 5 new restaurants. Our final product is both feasible with Yelp’s enormous data on its users and desirable as a way to enhance the customer’s ease of finding new restaurants that are more tailored to his or her personal tastes.

Our dataset is from Yelp’s annual Dataset Challenge. The data consists of 5 main tables containing information on Businesses, Check-ins, user Reviews, user Tips, and the User themselves. This dataset is a snapshot of Yelp’s full dataset and limited in scope to a handful of cities in the US, UK, Germany, and Canada. Though our proposed recommendation enhancement would require real-time computation and live streaming, we can use the static data as a proof-of-concept.

The schema we are applying to Yelp’s raw dataset to will clean the raw data and filter out data that is beyond the scope of our project. We are only looking at “Restaurant” type businesses. Since the data is static, we are simulating the current user’s location as the average lat and log of the user’s most frequent checked-in locations.

## What We’ve Accomplished

In the past couple of weeks, we have achieved several milestones that will bring us closer to our final product. We have moved through most of a ETL framework and progressing into the analysis phase of our project. Steps we have accomplished are:

1. We have extracted (downloaded) the data as json files from the online source and moved it onto a Hadoop cluster. Following the examples set by the lab homeworks, we are using the W205 Spring AMI and saving the data to our volume.
2. We transformed the data into csv files to better manipulate and clean the data. The original data had many formatting inconsistencies and unexpected data types which made the process difficult; fixing one issue seemed to uncover another.
3. We loaded the data from the csv files to hive ddl tables. As part of the loading process we created schema-on-read for our database tables and applied our appropriate filters in order to optimize our analysis process.
4. Now that we have base tables to work with, we passed is simulating approximate locations for every user in our data. Since the sample data provided by Yelp did not give us user locations, we estimated them by calculating a radius from the locations of the user’s most recently visited businesses. The specific calculations for this are provided in our original proposal.

## What We Have Left

Unfortunately, we hit a minor roadblock when we discovered the data were not as granular as we initially assumed. Some of the data are aggregated, which hinders our initial plan to calculate user visits by number of checkins. Number of checkins would have originally helped us determine user preferences by finding which types of restaurants a user frequently visits. Therefore, our immediate next step will be to adjust our model for determining user preferences. Considering this limitation of data, our model will instead factor in:

1. the number of tips given to a restaurant as a measure of “liking” a restaurant, under the assumption that users tend to write more tips for restaurants they enjoy
2. the most frequented type of restaurant
3. the highest rated type of restaurant

In addition to determining user preferences, another milestone is to determine ‘similar users’ for each user. Our final step is to produce the output table with recommendations for each user. We are considering using PySpark for this given its flexibility and ease-of-use as experienced in our homeworks.