**Project Report**

At the end of the week, your team will submit a Final Report that describes the following:

* **E**xtract: your original data sources and how the data was formatted (CSV, JSON, MySQL, etc).
* **T**ransform: what data cleaning or transformation was required.
* **L**oad: the final database, tables/collections, and why this was chosen.

Please upload the report to Github and submit a link to Bootcampspot.

When searching for restaurant ratings, users must typically search for the same restaurant on different websites to see how it is rated on each site. In order to simplify this process, our team decided to create a database of restaurant ratings. We wanted to create a database that would include ratings from multiple review sites so users could simply search for a restaurant once and find the information they need. In order to make this task more manageable, we limited our database to restaurants within the city of Pittsburgh, Pennsylvania (which was the only common city across our individual datasets).

Extract

We began by looking for review site data through Kaggle and APIs. We used the Google Places API to pull restaurant ratings from Google. This API requires geographic coordinates in order to obtain data for a specific city, therefore, we first used the Google Geocode API to find the coordinates for Pittsburgh, PA. Once we had the coordinates, we included them, as well as the radius for our search, “restaurant” as our location type, and the API key, as parameters for our Google Places API call. This returned our data in a JSON format.

Transform

The next step with the Google Places data was to store it in a pandas data frame. This was done by creating empty lists for restaurant name, address, and rating, and then looping through the JSON data to append name, vicinity, and rating to these lists, respectively. These three lists were then used to create the data frame. In order to more closely match the data retrieved for other websites, these columns were renamed to Restaurant\_G, Address\_G, and Rating\_G. Also, Pittsburgh was dropped from the Address\_G column by splitting the address at the comma.

Load

Finally, we loaded this data into a SQL database.

Yelp

Used data from Kaggle

JSON file was in a unorthodox format in where the records were not separated by commas. Hence we had to enter the below code to get the data in a pandas readable format



As the dataset had information for a lot of businesses, we had to perform the below query to filter out the restaurants in the dataset



We also had to encode some individual columns of the dataframe to export it into SQL

