**Extract, Transform, Load**

**Restaurant Ratings from Google Places and Yelp**

Kristen Hyman and Kyle Lee

**Overview**:

Our team’s goal was to obtain data on restaurant ratings for restaurants in the DMV area, transform and clean the data, and load the information into an easily accessible MySQL database.

**Extract**

To extract our data, our team performed API calls using the Google Places API and the Yelp Developer API.

*Google Places:*

To successfully perform Google Places API calls, we first obtained a Google Places API key.

We set the parameters of the API call to request data for restaurants from several different locations around Washington D.C. and Virginia.  The API Request we performed included parameters that specified the coordinates (latitude and longitude), a radius for the search area, and the establishment type (restaurant).

Each API Call provided us with 20 results within a JSON format each time we ran the the request.  To obtain more than the set 20 results from one query, users must obtain a “token” key from the JSON result, and pass the token within a second API call using the same parameters. This process can be done up to 3 times, to allow for a maximum of 60 results per API request.

This posed some difficulties in the extraction phase, as the process proved to be rather time consuming.  Additionally, after passing in the token keys, after 60 results, the user must input different coordinates to obtain new results. As such, our team used locations all around the DMV area (i.e., Penn Quarter, Georgetown, Adams Morgan, Arlington). After performing several successful requests to obtain a total of 340 results, we reached our API call maximum for the 24 hour period, and were unable to continue requesting more data.

*Yelp:*

I signed into my yelp account and obtained a yelp API key.

I set the parameters of the call to be Restaurants in Washington, DC.

The Yelp API has a generous call limit, so I was able to gather around 1000 restaurants. However, I was only able to call 50 at a time. I had to figure out how to incorporate the offset parameter into the code so that I could view results 1-50 then 51-100….and so on up to 1000.

I decided to make the API call it’s own function so that it would be easier to get 50 results at a time as I would pass in the new offset parameter to the function call.

**Transform:**

*Google Places:*

For each JSON received from Google Places, our code was written to extract the Restaurant Names, Addresses, and Ratings, and append the information into a list. After we completed collecting all of the information, we transformed this list into a dataframe that was ready to be loaded into MySQL.

*Yelp:*

The yelp API data was received in a JSON format.

Some of the data was already in a good format, such as the name and rating, but I had to change the location data because it would return 3-5 addresses for the same restaurant. To overcome this obstacle, I added the location to the dataframe at a separate time by creating a list of addresses to append to the dataframe. I also made sure the format would match up with the addresses from the google places API by removing some unnecessary commas and removing ‘DC’, since google places didn’t incorporate that.

**Load:**

Our team first created an empty database in MySQL, with two empty tables (one for the Google Places data and one for the Yelp data).

Our team then loaded both data frames into the MySQL Database using sqlalchemy. We created a connection to the database in MySQL, and used the “to\_sql” function to push the data frames into the tables.

For the Yelp data, there was an odd encoding mismatch as the yelp data was in UTF-8 and MYSQL default is Latin-1. MYSQL couldn’t interpret some of the characters when loading the data. To overcome this, we made sure to specify the encoding when building the engine to UTF-8.

Additionally, in the SQL query, we had to specify the encoding on the columns of the MYSQL Yelp table.

Upon loading the Google data and Yelp data into separate tables, we then used the SQL UNION function to combine the information. Our team also considered using joins to combine the data, however, we had substantially more records from the Yelp data source than from the Google data source. We believed that if we had used an inner join, we would have lost a significant amount of the information within the database. We determined that both data sets provided reliable information, and should both be maintained in the database using the UNION function.

We then deleted duplicate rows with the same name and address.

We also deleted rows with incomplete addresses.

**Future Considerations:**

In a future project where more time is allotted, our team has considered performing more Google Places API calls to obtain a higher number of records from this data source. Once an equal number of results from Google and Yelp are obtained, an inner join could be performed to combine the data within one table, that includes the Restaurant Names, Addresses, and both Google Rating and Yelp Rating.

We could also use this combined table, and join it with a csv, such as health inspection records. Then we could identify the highest rating restaurants that are also the cleanest, with no incidents.