**ETL Project Report**

Lauren Gama, Tahseen Chisti, Myke London

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. Upon a thorough search, we found relevant datasets, provided by Yelp, Google, YellowPages and Zomato, among others. We found a Yelp dataset on Kaggle that had a few different JSON files. We downloaded the yelp\_academic\_dataset\_business.json file because it had restaurant names, addresses, and ratings. We renamed the file to yelp\_business.json and saved it to a local folder.

We also found a YellowPages CSV on Kaggle with restaurant names, addresses, and ratings. We renamed the file to yellowpages.csv and saved it to a local folder. For time purposes, and the fact that Zomato’s API would only allow twenty results upon each request, we focused our first dataset on Kaggle’s YellowPages dataset.

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 Yelp JSON file was in an 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:



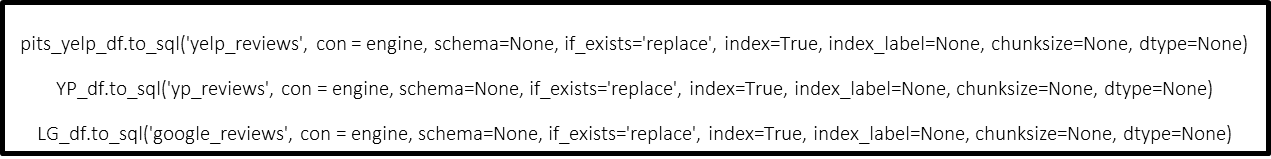
Once this was complete, we created a new dataframe with only the following columns: Restaurant\_Y, Categories\_Y, Address\_Y, City\_Y, State\_Y, Postal\_Code\_Y, and Rating\_YP. We then used the loc function to limit the dataframe to only businesses in Pittsburgh. Then, as the dataset had information for different types of businesses, we used the loc function to filter out the restaurants.

To manipulate our YellowPages CSV data, the file needed to be first saved to a local folder. Once saved, we connected the raw YellowPages CSV data and saved the data into a pandas dataframe within a Jupyter Notebook. With the CSV file now converted to a pandas dataframe, we included only restaurants found in Pittsburgh, PA by using the *loc* function. Next, we removed all unnecessary columns to include only the restaurant’s name, street, city, state and rating and renamed the columns Restaurant\_YP, Address, City\_YP, State\_YP, and Rating\_YP.

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. Finally, Pittsburgh was dropped from the Address\_G column by splitting the address at the comma.

*Load*

With our Yelp JSON dataset, YellowPages CSV dataset and Google JSON dataset now converted into their respective pandas dataframes within a Jupyter Notebook, we wanted to combine all three pandas dataframes into one common dataset. To do so, we first attempted to combine the datasets with a common identifier by using the pandas merge function inside a Jupyter Notebook. After some time, we pivoted our course of action upon realizing this task would be easier using SQL, rather than python. Once we shifted strategies, we converted each dataframe into SQL using the below code.



However, when we tried to move on to export the Yelp dataframe to MySQL, we received an error message that indicated an incorrect string value in the Restaurant\_Y column at row 1038. Upon further inspection, we found that there were accent marks in the restaurant name in that location. Although we tested a number of solutions, replacing the restaurant name with an un-accented version worked best. We then had to reset the dataframe index in order to export the dataframe without throwing another error.

Next, we created a new SQL database, entitled restaurant\_db, and ran a number of queries on each table in the database. We listed which restaurants were rated at a 5, counted how many restaurants scored a 4 or above, how many restaurants were categorized as Italian, and the average rating of seafood restaurants. We also performed joins. We had originally hoped to join all three tables, however, there were not enough restaurants in the Google table that matched the other two tables. Nevertheless, we successfully used an inner join on the Yelp and YellowPages tables and compared the restaurant ratings.

Our team set out to create a database that would allow users to see restaurant ratings from multiple websites at once. We encountered challenges with finding a common city across each of our datasets as well as being limited to only 20 restaurants from the Google Places API call. However, we were able to pull together a database that contains ratings for some of the same restaurants so users can see ratings from our three sources in one place.