Emmi Carr
Madeline Richards
Final Project Report
4/22/2019

Yelp vs Zomato Analysis

1. Goals

- Our goal for our project was to collect data from at least 100 restaurants in Ann Arbor for each website including star rating, price range, restaurant category, and the number of reviews.
- Then we wanted to compare the results and trends we observed in the data between Yelp and Zomato. Specifically, from each database, we wanted to compare the following:
    - Number of restaurants in each restaurant category
    - Category with the most restaurants
    - Restaurant category with the highest average star rating
    - Price range with the highest average star rating
    - Overall average star rating and price range for all restaurants in the database
- For each database, we wanted to create at least two of the following graphs/charts:
    - Scatterplot of star rating vs. number of reviews for each restaurant
    - Histogram showing the star rating for each restaurant category including the overall average star rating
    - Histogram showing the number of restaurants in each restaurant category

2. Goals Achieved

- We obtained all the data we set out to achieve from 100 restaurants from each website.
- From our original data analysis goals, we accomplished the following:
    - Number of restaurants in each restaurant category
        - Example: Yelp had 4 out of 100 restaurants in the “Italian” category, while Zomato had 10.
        - Able to find the category with the most restaurants
            - Yelp: tie between “American (New)” (7), “Coffee & Tea” (7), and “Korean”
            - Zomato: “American” (28)
    - Average star rating based on restaurant category
        - Able to find the restaurant category with the highest average star rating
            - Yelp: "Local Flavor" (5.0/5.0)
            - Zomato: “Cuban” (4.6/5.0)
    - Average star rating based on price range

        - Price range with the highest average star rating
            - Yelp: 1 out of 4 / $ out of $$$$
            - Zomato: 4 out of 4 / $$$$ out of $$$$
    - Overall average star rating
        - Yelp: 4.0292 / 5
        - Zomato: 4.071 / 5
- For our data visualizations, we generated the following:
    - Scatterplot of star rating vs. number of reviews for each restaurant
    - We revised the histogram to display the distribution of star ratings for each database

3. Issues Faced

- Our first issue was with getting information on the city level for the Zomato data. For example, Ann Arbor was under the locality of Detroit, and we did not know, until after carefully studying the documentation and testing with print statements, exactly how to obtain only Ann Arbor restaurants.
- The most difficult issue we faced was adding 20 outputs at a time without dropping the table. Our original code dropped the table if it already existed, and using a loop with offsets, it grabbed 20 restaurants at a time, but the complete code only needed to be run once to get 100 rows of data. After realizing this conflicted with what was asked of us for this project, we struggled on how we could modify our code to fit the requirements. To solve this issue, we replaced our offsets with a count system, and we added “INSERT OR IGNORE INTO” statements to add each restaurant’s data only if it is not in the database.

4. Calculation File

- Data-Analysis.py

```python
#analysis
import sqlite3
import json
import matplotlib
import matplotlib.pyplot as plt

#create database connection
def main():
    #create database connection

    conn = sqlite3.connect('rest_data.sqlite')
    cur = conn.cursor()

    #open json file for writing to
    write_file = open("data_analysis.json", "w")

    #function to get the number of restaurants in each category
    #takes the column of categories as input
    def get_num_of_rests_by_cat(category_data):
        categories = dict()
        for row in cur:
            cat = row[0]
            categories[cat] = categories.get(cat, 0) + 1
        return categories
        print(categories)
    #YELP
    categories_yelp = get_num_of_rests_by_cat(cur.execute("SELECT category FROM Yelp"))
    categories_yelp = {"Yelp Restaurant Category Counts" : categories_yelp}

    #ZOMATO
    categories_zomato = get_num_of_rests_by_cat(cur.execute("SELECT category FROM Zomato"))
    categories_zomato = {"Zomato Restaurant Category Counts" : categories_zomato}

    #Average Star Rating by Category
    #takes the category and rating columns from the database as input
    def get_rating_by_cat(data):
        rating_by_cat = {}
        for row in cur:
            cat = row[0]
            rating = row[1]
            if cat in rating_by_cat:
                rating_by_cat[cat].append(rating)
            else:
                rating_by_cat[cat] = [rating]
        averages = {}
        for cat in rating_by_cat:
```

```python
        averages = {}
        for cat in rating_by_cat:
            avg = round(sum(rating_by_cat[cat])/len(rating_by_cat[cat]),2)
            averages[cat] = avg
        return averages

    #YELP
    rating_by_cat_yelp = get_rating_by_cat(cur.execute("SELECT category, rating FROM Yelp"))
    rating_by_cat_yelp = {"Yelp Average Star Rating by Restaurant Category" : rating_by_cat_yelp}

    #ZOMATO
    rating_by_cat_zomato = get_rating_by_cat(cur.execute("SELECT category, rating FROM Zomato"))
    rating_by_cat_zomato = {"Zomato Average Star Rating by Restaurant Category" : rating_by_cat_zomato}

    #Average star rating based on price range
    #takes the star rating and price range columns from the database as input
    def get_rating_by_price(data):
        rating_by_price = dict()
        for row in cur:
            rating = row[0]
            price = row[1]
            if price in rating_by_price:
                rating_by_price[price].append(rating)
            else:
                rating_by_price[price] = [rating]
        averages = {}
        for price in rating_by_price:
            ratings = rating_by_price[price]
            avg = sum(ratings)/len(ratings)
            averages[price] = round(avg, 2)
        return averages

    #YELP
    rating_by_price_yelp = get_rating_by_price(cur.execute("SELECT rating, price FROM Yelp"))
    rating_by_price_yelp = {"Yelp Average Star Rating by Price Range" : rating_by_price_yelp}

    #ZOMATO
    rating_by_price_zomato = get_rating_by_price(cur.execute("SELECT rating, price FROM Zomato"))
    rating_by_price_zomato = {"Zomato Average Star Rating by Price Range" : rating_by_price_zomato}

    #Overall average star rating for all restaurants
    def get_overall_average_rating(ratings):
        ratings = list()
        for row in cur:
            ratings.append(row[0])
        avg_rating = round(sum(ratings)/len(ratings),4)
```

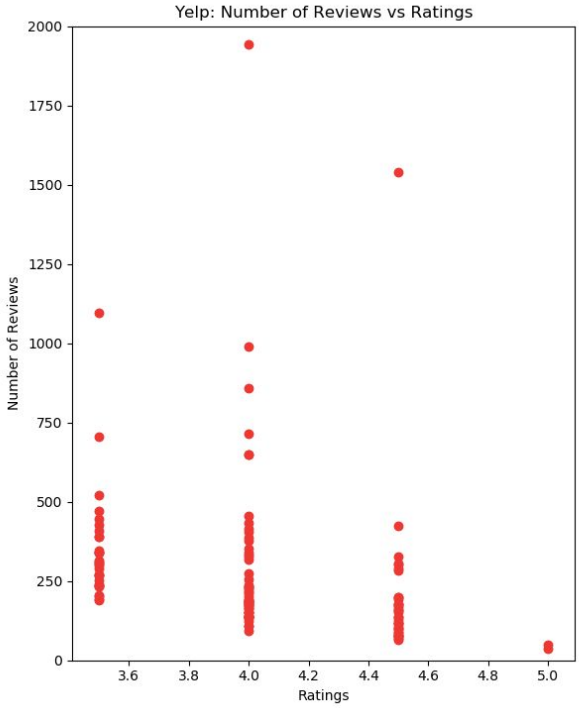
```
    return avg_rating

#YELP
overall_avg_rating_yelp = get_overall_average_rating(cur.execute("SELECT rating FROM Yelp"))
overall_avg_rating_yelp = {"Yelp Overall Average Star Rating for All Restaurants" : overall_avg_rating_yelp}

#ZOMATO
overall_avg_rating_zomato = get_overall_average_rating(cur.execute("SELECT rating FROM Zomato"))
overall_avg_rating_zomato = {"Zomato Overall Average Star Rating for All Restaurants" : overall_avg_rating_zomato}

#write to json file
yelp_data = {"Yelp Data" : (categories_yelp, rating_by_cat_yelp, rating_by_price_yelp, overall_avg_rating_yelp)}
zomato_data = {"Zomato Data" : (categories_zomato, rating_by_cat_zomato, rating_by_price_zomato, overall_avg_rating_zomato)}

json.dump((yelp_data, zomato_data), write_file, indent = 4, sort_keys = True)

write_file.close()


#VISUALIZATIONS

#scatterplot of star rating vs number of reviews

#YELP
cur.execute("SELECT rating, num_reviews from Yelp")
ratings = list()
num_reviews = list()
for row in cur:
    ratings.append(row[0])
    num_reviews.append(row[1])

fig = plt.figure()

ax1 = fig.add_subplot(121)
ax1.scatter(ratings, num_reviews, color = 'red')
ax1.set_xlabel("Ratings")
ax1.set_ylabel("Number of Reviews")
ax1.set_title("Yelp: Number of Reviews vs Ratings")
ax1.set_ylim(0, 2000)

#ZOMATO
cur.execute("SELECT rating, num_reviews from Zomato")
ratings = list()
num_reviews = list()
for row in cur:
    ratings.append(row[0])
```

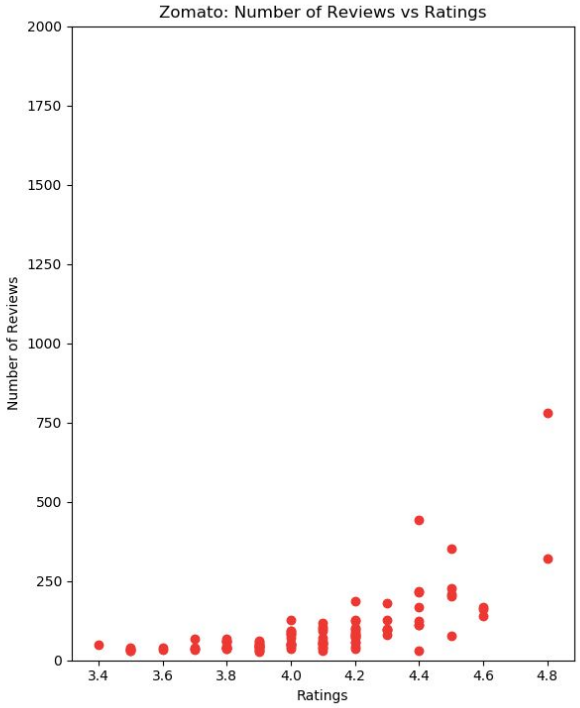
```
    num_reviews.append(row[1])

ax2 = fig.add_subplot(122)
ax2.scatter(ratings, num_reviews, color = 'red')
ax2.set_xlabel("Ratings")
ax2.set_ylabel("Number of Reviews")
ax2.set_title("Zomato: Number of Reviews vs Ratings")
ax2.set_ylim(0, 2000)

fig.savefig("ratings_by_reviews.png")
plt.show()


#HISTOGRAM DISTRIBUTION OF STAR RATINGS

#YELP
cur.execute("SELECT rating from YELP")
ratings = list()

for row in cur:
    ratings.append(row[0])

fig = plt.figure()

ax1 = fig.add_subplot(121)
ax1.hist(ratings, color = 'red')
ax1.set_xlabel("Ratings")

ax1.set_title("Yelp: Distribution of Star Ratings")
ax1.set_ylim(0, 55)

#ZOMATO
cur.execute("SELECT rating from Zomato")
ratings = list()

for row in cur:
    ratings.append(row[0])


ax2 = fig.add_subplot(122)
ax2.hist(ratings, color = 'red')
ax2.set_xlabel("Ratings")

ax2.set_title("Zomato: Distribution of Star Ratings")
ax2.set_ylim(0, 55)
```

```
    fig.savefig("ratings_dist_hist.png")
    plt.show()

#call function
if __name__ == "__main__":
    main()
```

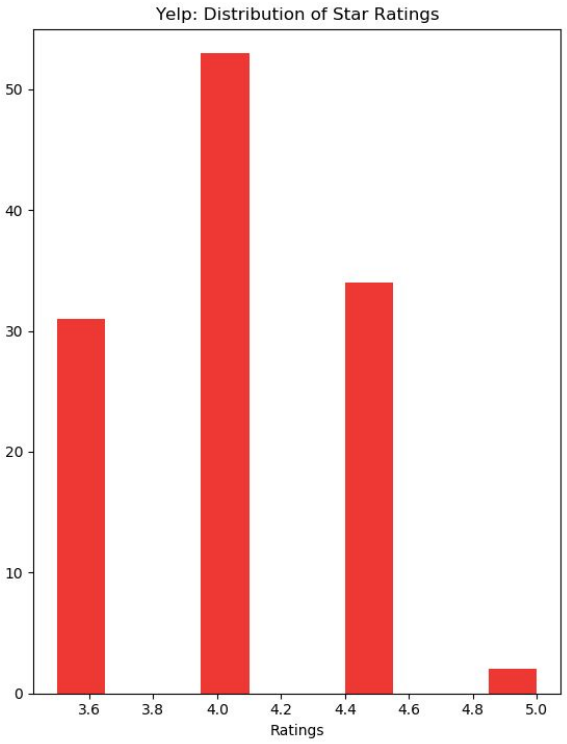
5. Visualizations

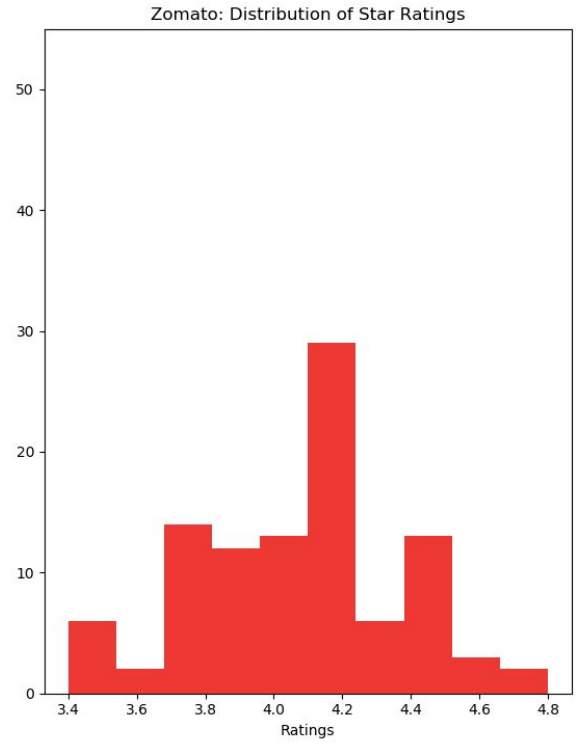
- ratings_by_reviews.png





- ratings_dist_hist.png





6. Instructions for Running the Code

1. *Run the python file five times: Gathering_Data.py*
    a. 100 rows of data will now be in the SQL database: rest_data.sqlite
2. *Run the python file: Data-Analysis.py*
    a. The two matplotlib visualizations are generated
    b. The JSON file is generated: data_analysis.json

7. Documentation for Each Function

Function	Inputs	Outputs
get_num_of_rests_by_cat (Data-Analysis.py)	Category column for each Yelp and Zomato	Dictionary named "categories." Keys are categories, values are the number of restaurants in the category
get_rating_by_cat (Data-Analysis.py)	Category and Rating columns for each Yelp and Zomato	Dictionary named "averages." Keys are categories, values are average star rating of each category
get_rating_by_price (Data-Analysis.py)	Price and Rating columns for each Yelp and Zomato	Dictionary named "averages." Keys are the price ranges, and values are the average star rating for each range
get_overall_average_rating (Data-Analysis.py)	Rating column for each Yelp and Zomato	Float values of average star rating for each Yelp and Zomato
main function (Gathering_Data.py)	NA	Function makes the database connection, calls both APIs and inserts restaurant data into the database. Must call 5 times in order to populate database with 100 rows.
main function (Data-Analysis.py)	NA	Selects data from the database, performs all the calculations specified and writes them to a json file and makes visualizations.

8. Documentation of Resources Used

Date	Issue Description	Location of Resource	Result (Did it fix the issue?)
4/17/19	Needed to make visualizations	matplotlib	Yes; we were able to make scatterplots and histograms using the module
4/14/19	Needed to insert data into a SQL database and later select it from the database	sqlite	Yes; we created a database with two tables for each website
4/20/19	Needed to be able to load in requested data from APIs and to be able to write our calculations to a json-formatted file	json	Yes; used json.loads() and json.dump()
4/14/19	Needed to know documentation of the data for the site in order to extract the information we needed	https://developers.zomato.com/documentation#/ (Zomato API Documentation)	Yes; we extracted the data we need
4/14/19	Needed to know documentation of the data for the site in order to extract the information we needed	https://www.yelp.com/developers/documentation/v3/get_started (Yelp API Documentation)	Yes; we extracted the data we needed

Link to Repository: https://github.com/escarr/Final-Project