Weekend Getaway

Group Members: Anthony Ho, Nina Yang and Ashley Anderson
Git Repository: https://github.com/ninayang8/weekendgetaway

Goals for Project

The goal of this project was to provide informative insight of movies and restaurants (in the Ann Arbor area) to help people decide what to do for a weekend getaway. This would be conducted by accessing data on the Yelp and OMDb API and using beautiful soup to scrape data from the IMDb website.

Goals Achieved

Working with Yelp, OMDb, and IMDb allowed us to plan a weekend getaway in Ann Arbor including restaurants and movie suggestions. From Yelp, we were able to extract the name, location including coordinates, category, and price. From IMDb, we extracted the title, platform for streaming, and reviews of the top 100 rated movies. From OMDb, we additionally extracted the rating and box office. Through these data points, we were able to create visualizations to see if there were correlations between the quantitative and qualitative data we obtained. For example, we found that the rating of the movies on IMDb had no correlation with the amount of reviews left on IMDb for the movie.

Problems Faced

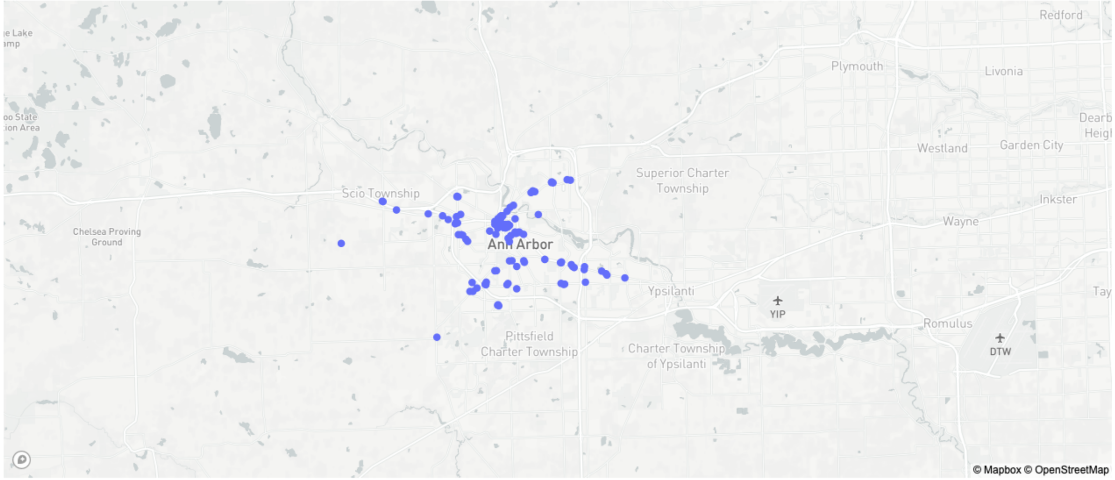
Throughout the project, we had a multitude of issues arise. The first issue we encountered was being unable to use Tripadvisor because their API was only able to be used by app developers and not for academic research purposes. This caused us to pivot and eventually decide to use both IMDb and OMDb as a joint component of the project. In addition, our second biggest issue was figuring out how to limit the input into database tables. We solved this problem by implementing a “count” system to take into account how many rows were present in a database at a specific time.

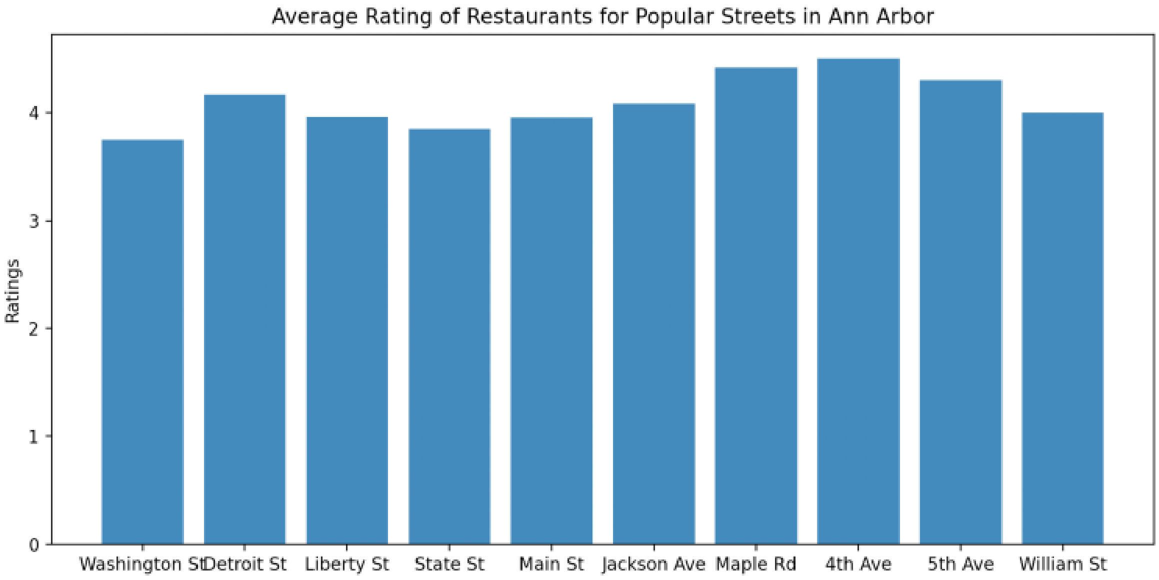
Calculations Done

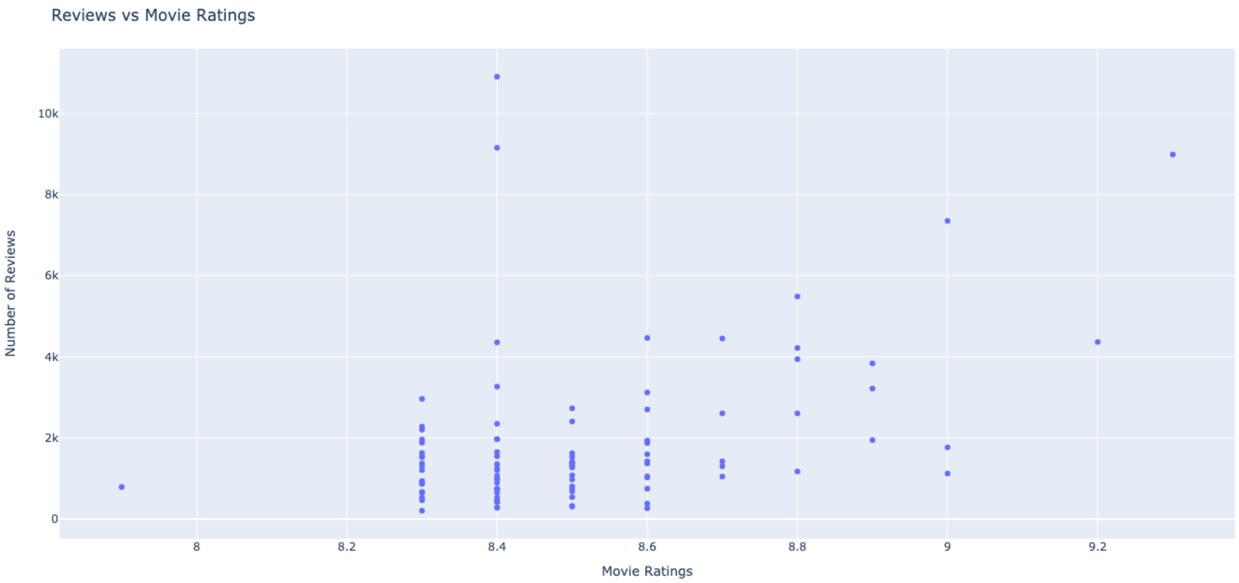
Yelp:
https://drive.google.com/file/d/1YhH0ExOiWR7HoYVi_f1h7NBxtGJ9T0JF/view?usp=sharing
IMDb/OMDb:
https://drive.google.com/file/d/1VV59pUdG_PdTgs7BNL7CdbDw6g8eLzV8/view?usp=sharing
(ratings were taken from OMDb and platforms available was scraped off of IMDb)

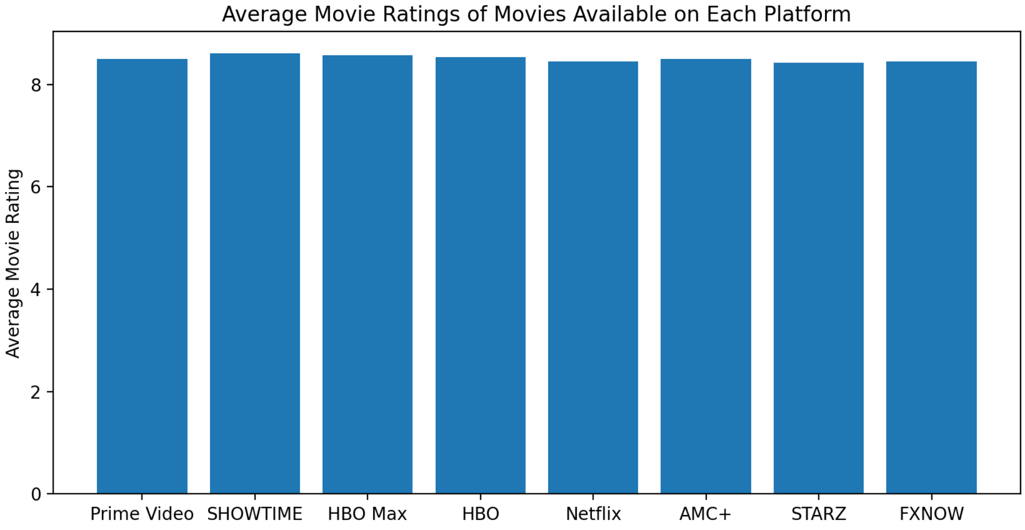
Visualizations

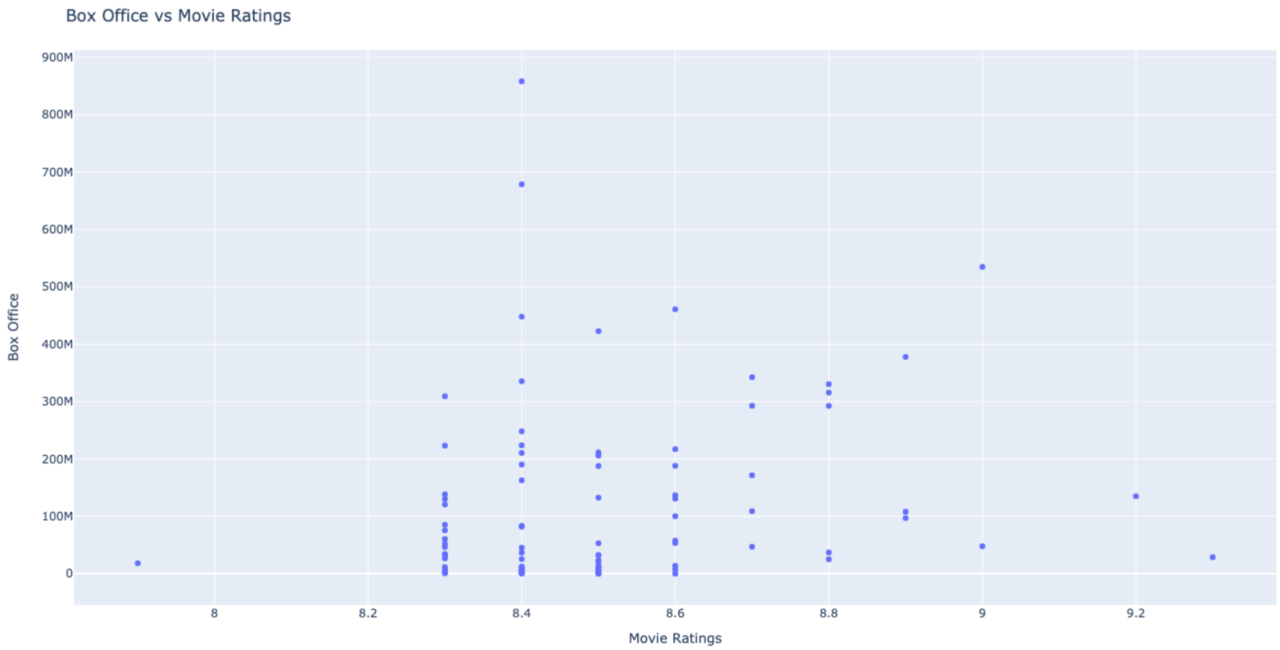
Locations of Restaurants in Ann Arbor






Reviews vs Movie Ratings
Number of Reviews
0
2k
4k
6k
8k
10k
8
8.2
8.4
8.6
8.8
9
9.2
Movie Ratings


Average Movie Ratings of Movies Available on Each Platform
Average Movie Rating
0
2
4
6
8
Prime Video
SHOWTIME
HBO Max
HBO
Netflix
AMC+
STARZ
FXNOW



Instructions for Running the Code

yelp.py: yelp.py collects data for the Locations (currently at 120 entries) and Restaurants (currently at 125 entries) Table in Database.db. You can directly run the code which will add more entries to both databases. When running the code, a bar plot titled "Average Ratings of Restaurants for Popular Streets in Ann Arbor" will pop up. You will need to drag out the plot to make it more readable. You can exit the bar plot, which will then prompt the code to bring you to our map visualization. To see the data on the map, hover your mouse over the dots to see the location and restaurant name.

imdb.py: imdb.py collects data for the number of reviews and platforms that stream each movie in the top 100 movies on the IMDb website. You can directly run the code which will add more entries to the "Movies" table. There will also be visualizations that are generated, including a scatter plot that compares the number of reviews a movie received and their rating (taken from omdbMovies table), and a bar graph showing the average rating of the movies available on each platform (Netflix, Prime Video etc). After viewing the scatter plot, you can exit it which will prompt the code to display the bar graph.

omdb.py: omdb.py collects data for the rating and box office amount for each of the top 100 movies on the IMDb website. You can directly run the code which will add more entries to the "omdbMovies" table. There will be a visualization produced (scatter plot), comparing the rating of movies to box office amount.

Documentation for Each Function

File Name	Function Name	Input	Output
imdb.py	get_title Using beautiful soup (BS), it fetches returns the top 100 movie titles on imdb.com based off of imdb movie rating.	soup (beautiful soup of https://www.imdb.com/list/ls091520106/)	list (list containing names of top 100 movies)
	get_link Returns a list after fetching the links to each of the top 100 movie pages, using BS.	soup (beautiful soup of https://www.imdb.com/list/ls091520106/)	movie_links (list containing the webpage links of the top 100 movies)
	get_movie_reviews Returns a list of the number of reviews using BS for each of the top 100 movies	soup, links (list containing the webpage links of the top 100 movies)	movie_reviews (list containing the number of movie reviews for all the top 100 movies)
	where_stream Returns a list of the platform where each of the top 100 movies can be watched	soup, links (list containing the webpage links of the top 100 movies)	stream_list(list containing the platform for all the top 100 movies)
	setUpDatabase Creates the table “Movies” and assigns the database cursor and database connection object	db_name (name of database)	cur (database cursor), conn (database connection object)
	addEntriesToDatabase Inserts all the data (rating, platform, number of reviews) corresponding to each movie into the database and gives it a unique number id.	cur, conn, soup, links	None

	RatingVsReviews Creates visualization (scatter plot) comparing the ratings and the reviews received of each movie	cur, conn	None
	PlatformVsRating Creates visualization (bar graph) comparing the average movie rating for movies available on each of the platforms.	cur, conn	None
omdb.py	create_request_url Creates the url for requesting later based on the movie title passed in	title (Title of Movie)	request_url (string containing specific url for movie)
	get_title_and_rating Fetches the movie title and rating from the OMDb API	cur (database cursor), conn (database connection object)	title_and_rating (list containing tuples of movie titles with their corresponding rating)
	get_box_office Fetches the box office amount from the OMDb API	cur (database cursor), conn (database connection object)	box_office (list containing the box office dollar amount of all the movies)
	create_database Sets up the file path and the assigns the database cursor and database connection object	db_name (database file name (database.db))	cur (database cursor), conn (database connection object)
	setUpMoviesTable Creates “omdbMovies” table in database and assigns the rating and box office amount to each movie and also a unique id number.	cur (database cursor), conn (database connection object)	None
	RatingVsBoxOfficePlot	cur (database	None

	Creates the visualization (scatter plot) for comparing the rating of movies to their box office amount	cursor), conn (database connection object)	
yelp.py	get_url Grabs the URL for the search results from Yelp using the location parameter	cur, conn, location	url
	request_data Uses the URL from get_url to pull data from YELP	url	r.text
	setUpDatabase Creates the table “Restaurants” and “Locations” and assigns the database cursor and database connection object	db_name	cur, conn
	addEntriesToDatabase Inserts all the data corresponding to each restaurant and into the database tables “Restaurants” and “Locations”.	cur, conn, data, location	None
	RatingVsPrice Calculates the average rating of restaurants of each price level and writes that to a txt file.	cur, conn	None
	StreetVsRating Calculates the average rating of restaurants every street has and writes that to a txt file and	cur, conn	None

	creates a bar graph of the first ten streets		
	MapPlot Creates a visualization of a map with all the restaurants marked	cur, conn	None

All Resources Used

Date	Issue	Description	Location of Resource	Result (did it solve the issue?)
4/16	Using Yelp API	Had trouble accessing the URL and needed to figure out how to use API/APIKEY	https://www.yelp.com/developers/documentation/v3/business_search	Greatly helped in understanding the API for Yelp
4/23	Creating Map Visualization	Couldn't understand how to use plotly mapbox	https://plotly.com/python/scattermapbox/	Solved how to display the map on a browser and gave us an API key
4/23	Issues Importing Numpy	VSCode kept giving us an error where numpy was not an imported module	https://stackoverflow.com/questions/7818811/import-error-no-module-named-numpy	Did not solve the issue, and we ended up typing stuff into the terminal until it worked.
4/23	Creating Bar Graph	Needed to find extra parameters for bar graphs	https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html#matplotlib.pyplot.bar	Helped us create more sophisticated bar graphs
4/25	Creating Bar Graphs, Scatter plots	We had to find a way to generate our visualizations in our	https://plotly.com/python/line-an	After exploring the different plots available and figuring out the

		desired format, such as scatter plots and bar graphs.	d-scatter/	syntax, we were able to generate our visualizations using the resource.