Sarah Jomaa
SI 206 Fall 2017

Project 4 Report

Goals:

My original goals were to use Spotify, iTunes, and YouTube to compare the number of views a song has on YouTube, the number of plays it has on iTunes, and how many plays it has on Spotify. I would also pull information like release date, artist, reviews and price on iTunes.
I also wanted to compare the price and popularity rating (ex. Rotten tomatoes) of movies on YouTube Red versus on iTunes. I'd listed IMDb as an alternative to iTunes in case they didn't have ratings. I also wanted to gather information about each movie like genre and release date. For visualization, I proposed using plotly and/or panda.

Which goals I achieved:

I did end up using iTunes and IMDb, and I did pull information on release date, artist, and price for songs, and genre, IMDb ratings, and release date for movies (I did end up using more data points: genre for songs, and description, actors, director, and rental and buying price, for movies). I also created a plotly visualization.

Problems I faced:

I wasn't able to get reviews for iTunes songs, as that wasn't an available option offered by the API. Neither was getting the number of plays a song has on iTunes and Spotify. For Spotify, I ran into authorization issues despite not utilizing private user info, and ultimately decided drop the API, and YouTube as well, for similar reasons. For the sake of time, I decided not to use panda visualizations.

When creating the OMDb and iTunes functions to retrieve movie info, I couldn't retrieve more than 1 movie because I ran into several issues trying to do that. One was that the search engines for iTunes and for IMDb didn't return the same movies when given the same keyword or movie title. Trying to pass a list of the returned movie titles from iTunes into IMDb as search terms didn't work either because some movie titles were written differently in iTunes than in IMDb, so they didn't show up in the IMDb search. This is also why I couldn't get 100 interactions for the IMDb API.

When creating the Database, I had an issue where the columns showed up fine, but the table itself didn't populate. I Googled the issue and found a solution, which will be further documented in the table below. I also ran into an error where the whole operation stopped if a movie on iTunes didn't have a trackRentalPrice or trackPrice. I solved it using try except: continue.

When creating the visualization in Plotly, I kept getting a syntax error. This held me up for a while before I realized the whole issue was because I was missing a comma.

Sarah Jomaa
SI 206 Fall 2017

Report:
Terminal Output
Preview:

User Instructions for running code:

- Required libraries: requests, json, sqlite3, plotly. Make sure these are installed on your computer before running the code.
- Run the code and follow directions when prompted on screen.

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
12/16/17	sqlite3.OperationalError: no such column when trying to create Database	Stack Overflow	Solved by putting a comma here: `(tup2[0],))` And switching out the `{}` in `('SELECT movie_name FROM Movies WHERE movie_name = {}'.format(tup2[0],))` With a `?`
12/17/17	Merging two dictionaries in Python3	Stack Overflow	Solution: `fullmovieInfo =`

Sarah Jomaa
SI 206 Fall 2017

			`{**itunesMovieDict, **omdbMovieDict}`

12/17/17	Syntax error trying to create a line in my line graph.	Plotly	Solution:
			Was missing this comma
			`name = 'Songs',`
			`line = dict(color = ('rgb(255, 187, 43)'), width = 2)`
			`)`

12/15/17	Wanted to possibly search iTunes by song genre	Stack Overflow	Not Solved. Stack Overflow solution was to use a different API.
12/14/17	Wanted to get track play count from Spotify	GitHub	Not Solved. The feature doesn't exist with the current Spotify API

Plotly Visualization (link also displayed in Terminal at the bottom when code is run)

Resources:

1. iTunes API

2. OMDb API

3. JSON editor online

4. iTunes constructing search results

5. iTunes understanding search results

6. Plotly getting started

7. Plotly line charts

Sarah Jomaa
SI 206 Fall 2017

8. Python3 documentation for basic python queries