Marcy Held
SI 506 - Final Project README
13 Dec 2016

1. My project uses the iTunes API to take a movie director's name as input and the Liquor Control Board of Ontario (LCBO) API to output a variety of drink suggestions based on the film genre most frequently associated with that director's work. The dictionary of results from the iTunes API is used to create a Director object, which has the methods getGenreTypes(), getMostCommonGenre(), and getDrinkKeyWord(). The getDrinkKeyWord() method takes the result from the getMostCommonGenre() method in order to return a string, which is then used as the query term for the LCBO API request.

I have programmed the code to make a request to the iTunes API using the names of 4 different directors (Wes Anderson, Quentin Tarantino, Paul Thomas Anderson, and Alfred Hitchcock) who all have different film genres associated with their work. The program's output displays information about the director: the titles of the movies that are associated with their name, the genres of those movies, and the name of the genre most commonly associated with their work. The program also displays information about the type of drink associated with that director's work: the primary and secondary categorization of the drink, as well as a list of tuples that include the drink's name and price (in cents!) of some of the most expensive and least expensive drinks that one can buy that are of that those types.

2. In order to run this program, save Held_Final.py and cached_data.txt in the same directory. Then, from the terminal window, run the Held_Final.py file. The program's output text will appear in the terminal window.

3. 1) Held_Final.py (Python code file)
   2) Held_cached_data.txt (File containing all of the cached data from both the iTunes and LCBO APIs)
   3) Held_Final_README.txt (README file)

4. In order to run the Held_Final.py file, you will need the following packages/modules:
requests,
json,
and unittest

5. iTunes API
Link to documentation:

https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/#searching

LCBO API
Link to documentation: https://lcboapi.com/docs/v1/products

6. Line numbers in Python file to find the following mechanics requirements:

- Sorting with a key function: 33
- Use of list comprehension: 90 and 95
- Class definition beginning 1 (Director class): 9
- Class definition beginning 2 (DrinkRecs class): 74
- Creating instance of one class (Director): 178, 179, 180, 181
- Creating instance of second class (DrinkRecs): 186, 193, 200, 207
- Calling any method on any class instance:
    Director class:
        - getDrinkKeyWord method: 186-7, 193-4, 200-1, 207-8
        - getGenreTypes method called within getMostCommonGenre method
        - getMostCommonGenre method called within getDrinkKeyWord method and string method
        - string method calls: 184, 191, 198, 205

    DrinkRecs class:
        - getMostExpensiveDrinks method called within string method
        - getLeastExpensiveDrinks called within the string method
        - string method calls: 187, 194, 201, 208

- Function definitions outside of classes:
    getiTunesWithCaching: 115
    getLCBOWithCaching: 138
    canonical_order: 161
    requestURL: 168

- Code that handles data caching: 105 - 159, 161 - 171

- Test Cases: 212 - 232

7. I chose to do this project because I really enjoy movies, and I thought that it would be fun to create a program that would take a director's name as input and return a list of drink recommendations based on the genres associated with that director's work. Originally I wanted to search for the drink

recommendations based on the time length of the movies associated with that director, but I decided to use the genres associated with that director's work because it provided a more concrete way to connect the themes of each director's work to a particular type of drink.

In my cached data I chose to store information about directors whose work I really enjoy, and I was a little surprised by some of the genres that iTunes associates with each director. Specifically I was surprised to see 'Comedy' associated with Wes Anderson and 'Action & Adventure' associated with Quentin Tarantino. Although there are many funny parts in Wes Anderson's movies I would not consider them to mostly be 'comedies,' and when I think of 'Action & Adventure' movies I think of superhero movies and not Quentin Tarantino's work. However, I understand that when creating databases to categorize and store large amounts of data, some compromises have to be made so that the overall categorization structure fits mostly with most of the data - not everything can be totally specific!

I was also a little surprised by how expensive some of the drink recommendations were (e.g. some of the Rum recommendations for Alfred Hitchcock and some of the Vodka recommendations for Paul Thomas Anderson). I was also surprised to see that the LCBO API did not return any Lager beers over $50 for Wes Anderson's drink recommendations (the least expensive drink associated with Wes Anderson's work is 'Laker Ice' for $1.95)!