DSA 8640 Final Term Project: Pokemon Go! Analytics

DUE DATE: Dec. 11th Sunday11:59 p.m.

GENERAL INSTRUCTION:

- Submit (1) Python code(s) and (2) data files (CSV, Excel, JSON) on Canvas. DO NOT email your project.
- Late submission will not be accepted.
- Please make proper references when you use others' codes. You may use instructor-provided codes without permissions.

INTRODUCTION:

Pokemon Go! became a very famous augmented reality (AR) game in summer of 2016. In this project, we want to analyze the mobile game app, Pokemon Go!. Specifically, the purposes of this project are (1) to do web scraping using `BeautifulSoup` and (2) to construct a `Pandas dataframe`.

DATA DESCRIPTION:

For this project, I have downloaded app pages of Pokemon Go! from Google Play Store and Apple App Store from July 21 2016 to July 31 2016:

- `https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=en`
- `https://itunes.apple.com/us/app/pok%C3%A9mon-go/id1094591345?mt=8`

The webpages were downloaded every ten minutes. This means that there are 144 (=24 x 6) HTML files for a given day and a given platform.

Once you extract the ZIP file, you will see 11 date folders. Each date folder contains HTML files downloaded in the specified date. Each HTML file name is formatted as "`HH_MM_pokemon_PLATFORM.htlm`", where HH is hour, MM is minute, and PLATFORM is either "android" or "ios". Note that due to intermittent connection errors, some HTML files may not be properly downloaded.

PROJECT INSTRUCTIONS:

Please follow the following steps to parse and organize.

1) **[Web Scrapping: 50 points]** The first step is to extract various values from the raw HTML files. You can use `BecutifulSoup` or other Python modules.
    a. From all the iOS pages (ending with "`_ios.html`"), extract (i) number of customer ratings in the Current Version (let's call it *ios_current_ratings*); and (ii) number of customer ratings in All Versions (*ios_all_ratings*). For example, the extracted values should be: 4688, 106508 for the "`2016-07-21/00_00_pokemon_ios.html`" file. There are 2 values from iOS pages.
    b. From all the Android pages (ending with "`_android.html`"), extract (i) average rating (in the scale between 1.0 and 5.0) (*android_avg_rating*); (ii) number of total ratings (*android_total_ratings*); and (iii) number of ratings for 1-5 stars (*android_ratings_1, android_ratings_2, …, android_ratings_5*). For example, the extracted values should be: 3.9, 1281802, 199974, 71512, 117754, 165956, 726597 for the "`2016-07-21/00_00_pokemon_android.html`" file. There are 7 values from Android pages.

2) **[Data Organization: 50 points]** The next step is to organize the extracted values, so that we can do some data exploration. As we have time series data, we will organize the data by `datetime` (note that `datetime` is a Python data type).
    a. Using the extracted values from the previous step, create a Python dictionary, where the key is `datetime` object and the value is a dictionary with extracted values from iOS and Android HTML files. For example, for the case of "`2016-07-21-00_00_pokemon_android.html`" file and "`2016-07-21/00_00_pokemon_ios.html`" file, the key should be `datetime(2016, 7, 21, 0, 0, 0)` and the value should be:
    ```
    {'ios_current_ratings': 4688, 'ios_all_ratings': 106508, 'android_avg_rating': 3.9, 'android_total_ratings': 1281802, 'android_rating_1': 199974, 'android_rating_2': 71512, 'android_rating_3': 117754, 'android_rating_4': 165956, 'android_rating_5': 726597}
    ```
    b. Convert the dictionary into a `Pandas dataframe, pokemon_db,` where the index is `datetime` and columns are names of the extracted 9 iOS/Android values.
    c. Save the `dataframe` into two formats (CSV and Excel). The file names should be `pokemon.csv` and `pokemon.xlsx`.

PROJECT SUBMISSION AND REPORT:

- Submit the following in Canvas:
    a. Python code(s) (if multiple files, please zip them)
    b. Constructed data files (CSV, Excel)
- Please make proper references when you use others' codes.