INSY 5378: Project 2: Pokemon Go! Analytics

DUE DATE: April 28th, Friday by 11:59 p.m.

0. General Instructions

1. This is the second group project. The total is 100 points. There are 20 extra points for extended works and 10 extra points for 5 teams with the best predictions.
2. Submit (1) your project report, (2) Python code(s), and (3) data files (CSV, Excel, JSON) in Blackboard. DO NOT email your project. Late submissions will not be accepted.
3. Please make proper references when you use others' codes.

1. Introduction

Pokemon Go! became a very famous augmented reality (AR) game in 2016 summer. In this project, we want to understand the success of the mobile app game. Specifically, the purposes of this project are (1) to do web scraping using `BeautifulSoup`, (2) to construct a `Pandas dataframe`, (3) to explore/visualize the numeric data using `matplotlib` or `seaborn`, and finally (4) to use `sklearn` to build machine learning models to predict the app's review counts. The 5 best teams will get extra credits. Finally, for more extra credits, you can (5) analyze the app's screenshot images using deep learning with `tensorflow`.

2. 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 October 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 (=24x6) HTML files for a given day and a given platform. You can download the zip file from the following link:

- http://diamond.mccombs.utexas.edu/insy5378/pokemon_5378.zip

Once you extract the ZIP file, you will see 103 date folders under "data" folder. Each date folder contains HTML files downloaded in the specified date. Each HTML file name is formatted as "`HH_MM_pokemon_PLATFORM.html`", 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.

3. Project Instructions

Please follow the following steps to parse, organize, explore, and predict.

3.1 Web Scraping [30 points]

The first step is to extract various values from the raw HTML files. You can use `BeautifulSoup` or other Python modules.

1. 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*); (ii) number of customer ratings in All Versions (*ios_all_ratings*); and (iii) file size in MB (*ios_file_size*). For example, the extracted values should be: 4688, 106508, 110 for “`2016-07-21/00_00_pokemon_ios.html`” file. Note that there are 3 values from iOS.
2. 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*); (iii) number of ratings for 1-5 stars (*android_ratings_1*, *android_ratings_2*, … , *android_ratings_5*); (iv) file size in MB (*android_file_size*). For example, the extracted numbers should be: 3.9, 1281802, 199974, 71512, 117754, 165956, 726597, 58 for the “`2016-07-21/00_00_pokemon_android.html`” file. Note that there are 8 values from Android.

3.2 Data Organization [30 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).

1. Using the extracted values from the previous step, create a 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, ‘ios_file_size’: 110, ‘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, ‘android_file_size’: 58}`
2. Convert the dictionary into a `Pandas dataframe` where the index is `datetime` and columns are names of the extracted 11 iOS/Android values.
3. Save the `dataframe` into three formats (JSON, CSV, Excel). The file names are `data.json`, `data.csv`, and `data.xlsx`.

3.3 Data Exploration [20 points]

Now that we have `Pandas dataframe` ready, we can start exploring the data.

1. Use `describe()` method to find the count/mean/std/min/25%/50%75%/max values for each 11 variables.
2. Use `scatter_matrix()` method to find pairs of variables with high correlations (either positive or negative).
3. For identified pairs, calculate the Pearon's correlation coefficients. You can use `corrcoef()` function in `numpy` module for this.
4. Use `matplotlib` or other tools to create time series graphs for each of the 11 variables.
    a. It is your decision either to put all time series in one graph or to have individual graphs for each time series. Also, use your judgment to combine Android and iOS data together.
    b. As the files are collected in every 10 minutes, there are multiple values for a given date. Thus, the X-axis should incorporate dates and times.

3.4 Prediction Model [20 points + 10 extra points]

At this point, I am sure you are familiar with the data. Now let's build a machine learning model on the success of Pokemon Go! app. People often use the number of ratings (`ios_all_ratings` and `android_total_ratings`) as a proxy of app success.

1. Build two best regression models (one for iOS and one for Android) using `sklearn` using cross validation. Try to add/remove variables among the 11. Of course, you can create your own variables if you want. Try various algorithms in the module: `LinearRegression`, `Ridge`, `Lasso`, etc.
    a. http://scikit-learn.org/stable/modules/linear_model.html
2. Submit your predicted values of `ios_all_ratings` and `android_total_ratings` for **2016/11/01 11:50 PM**.
3. 10 extra points will be given to 5 teams with the best predictions.

3.5 Deep Learning [20 extra points]

This is an optional task for extra points. We want to understand the screenshots of the app.

1. Identify all unique screenshots from iOS and Android pages. Note that you can use the URLs to distinguish different images. Also note that there are multiple images in each app page.
2. Download the screenshot images from iOS and Android webpages.
3. For each image, use `tensorflow` to extract the tags with the corresponding probabilities.

Project submission and report

1. Submit (1) your project report, (2) Python code(s), and (3) data files (CSV, Excel, JSON) in Blackboard.
2. Please make proper references when you use others' codes.
3. For the project report, I expect the followings:
    a. High-level description of your codes
    b. For 3.3 Data Exploration, present results, numbers, graphs, etc. with your interpretations
    c. For 3.4 Prediction Model, describe how you came up with your regression model. Also, report your two predicted values.
    d. For (optional) 3.5 Deep Learning, report the number of unique screenshots for iOS and Android and report the tags/probabilities for each image. Finally, please submit the downloaded images.
4. Good luck!