90-803 Machine Learning Foundations, Spring 2025

A2 Practicing Pandas with Git

Due: Friday, March 28 11:59 pm

Learning Objectives

This assignment has two intertwined objectives (i) to refresh your Pandas skills and (ii) practice working with Git and GitHub.

There are 5 questions below. When you finish answering each question in your Jupyter Notebook (NB), add, commit, and tag your work to your local Git repo. By the time you finish this assignment you should have done 5 commits and tags (one for each part). Tag each commit as `part-i`, `part-ii`, `parti-iii`, `part-iv`, and, `part-v`.

Analyzing scraped data

The file `itunes-raja.csv` was scraped from a web page that used to be at https://www.apple.com/itunes/charts/free-apps. For reference, an archived version of of that page can be found on the Wayback Machine at https://web.archive.org/web/20190124030104/https://www.apple.com/itunes/charts/free-apps/

Follow the instructions give in lab to download the Jupyter starter NB.

i. Accept the invitation to Classroom Git by clicking: https://classroom.github.com/a/S_7OOJun
ii. Click on **your name** from the list. You will now have a new GitHub created for you with assignment-2 starter code
iii. `clone` the remote repo and then start working on the NB locally.
iv. Follow the instructions below in terms of answering each question. Once done with each question, add, commit, tag, your version and then move to the next part.

Read in the csv file `itunes-raja.csv` and demonstrate the following analysis using Pandas

i. Clean the data. Convert entries in the column `num_ratings` of the form `"10.2K Ratings"` to 10200 and store it in a new column. Call it `num_ratings_clean`. Note that not all ratings have a 'K' or 'M' suffix. Some are just `"8 Ratings"`. Once you are done with this part add, commit, and tag your work. Create an *annotated tag* with the `-a` command line switch.

```
% git add .
% git commit -m "Solution to part-i"
% git tag -a part-i -m "Solution to part-i"
```

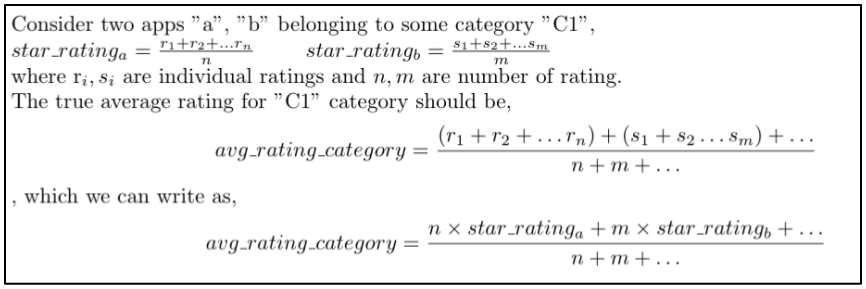
ii. List the names of the top apps sorted in descending order based on star rating and within those with the same star rating sort based on number of ratings in descending order. If the number of ratings are also the same, sort by app_name in ascending order. Your result should be a data frame with the `app_name, star_rating`, and `num_ratings_clean`. Again, once done, add, commit, and tag your work:

```
% git add .
% git commit -m "Solution to part-ii"
% git tag -a part-i -m "Solution to part-ii"
```

iii. For each category list the number of apps. Produce your answer as a series with the app categories as the index.

```
% git add .
% git commit -m "Solution to part-iii"
% git tag -a part-i -m "Solution to part-iii"
```

iv. For each category of app (game, music etc.) list the average rating of all apps in that category and sort in descending order by average rating. Your answer will be a series similar to your answer for iii.
Note the following observation that will guide your calculation in Pandas.



Consider two apps "a", "b" belonging to some category "C1",
$star\_rating_a = \frac{r_1+r_2+\ldots r_n}{n}$ $star\_rating_b = \frac{s_1+s_2+\ldots s_m}{m}$
where r$_i$, s_i are individual ratings and n, m are number of rating.
The true average rating for "C1" category should be,

$$avg\_rating\_category = \frac{(r_1+r_2+\ldots r_n)+(s_1+s_2\ldots s_m)+\ldots}{n+m+\ldots}$$

, which we can write as,

$$avg\_rating\_category = \frac{n \times star\_rating_a + m \times star\_rating_b + \ldots}{n+m+\ldots}$$

```
Finance            4.796536
Music              4.757942
Travel             4.715657
Photo & Video      4.704613
Health & Fitness   4.700000
```

First lines of the expected output are given above right.

```
% git add .
% git commit -m "Solution to part-iv"
% git tag -a part-i -m "Solution to part-iv"
```

v. For each category, list the app with the highest star rating. If there is a tie for apps with the highest star rating, list the one with the greatest number of ratings.

```
% git add .
% git commit -m "Solution to part-v"
% git tag -a part-i -m "Solution to part-v"
```

What to submit

Once done with all parts, push to your repository on GitHub classroom. [You can if you want to, but don't need to push to your personal repository on GitHub.]

```
% git push --tags
```

It is important to specify the command line flag --tags to have the tags also pushed to the remote repository.