**MSSP 607: Practical Programming for Data Science Homework 2**

Linya Liu

**Part 1: Yelp**

Source code: [homework2\_yelp.py](https://github.com/Liulinya/mssp607_homework2/blob/master/homework2_yelp.py)

In class, we made use of a subset of 10% of Pennsylvania takeout restaurants and their associated reviews, about 20,000 in total. Included with this homework are two new files, one of which contains business descriptions for each takeout restaurant in Pennsylvania, and one of which contains all reviews for those restaurants, over 210,000 in total. Import the two dataset files using Pandas and answer the following three questions about the data:

1. For each star count, 1-5, what percentage of restaurants receive that score on average? What is the average word count of reviews that give each star count?

|  |  |
| --- | --- |
| Coding Result | Further analysis |
| 0.50% of restaurants receive the star 1.0.  1.51% of restaurants receive the star 1.5.  4.48% of restaurants receive the star 2.0.  8.47% of restaurants receive the star 2.5.  16.85% of restaurants receive the star 3.0.  24.03% of restaurants receive the star 3.5.  24.22% of restaurants receive the star 4.0.  16.45% of restaurants receive the star 4.5.  3.49% of restaurants receive the star 5.0.  The average word count of reviews is 123.44 for the 1.0 star.  The average word count of reviews is 139.25 for the 2.0 star.  The average word count of reviews is 136.26 for the 3.0 star.  The average word count of reviews is 119.73 for the 4.0 star.  The average word count of reviews is 93.32 for the 5.0 star. | Nearly quarter of restaurants receive star either 3.5 or 4.0. Almost 85% restaurants receive above average star 2.5.  When the star is lower, average word count of reviews is usually higher except from star 2.0 to 1.0. Therefore, people generally intend to make more comment if they are not satisfied with the restaurants. |

1. Philadelphia ZIP codes begin with 19xxx, while Pittsburgh ZIP codes begin 15xxx. The rest of Pennsylvania ZIP codes begin with 16, 17, or 18. For each of the three regions, how many reviews are in our dataset? What is the mean score of reviews for each region?

|  |  |  |
| --- | --- | --- |
| Coding Result | Further analysis | |
| Philly has none number of reviews and the mean score of reviews is zero.  Pitts has 888561897 number of reviews and the mean score of reviews is 210211.0.  Rest of Pennsylvania has 210211 number of reviews and the mean score of reviews is 210211.0. | | More than 99% of the results live in Greater Pittsburgh Area, indicating a highly skewed demographic dataset. |

1. What are some common features of restaurants that receive higher-scoring reviews? This can be extracted either from attributes of the restaurant itself or from the review texts.

|  |  |
| --- | --- |
| Coding Result | Further analysis |
| sentiment Negative Positive Unknown  stars  1.0 53.666435 3.645922 16.786711  2.0 20.714781 6.790877 12.310424  3.0 10.143419 14.572091 13.437738  4.0 7.286607 33.591805 21.601572  5.0 8.188758 41.399304 35.863556  Review star 1.0 has the highest number of negative reviews, which occupies 53.67% among star 1-5.  Review star 5.0 has the highest number of positive reviews, which occupies 41.40% among star 1-5.  Review star 5.0 has the highest number of unknown reviews, which occupies 35.86% among star 1-5. | As we all known, higher-scoring reviews have more positive comments and lower-scoring reviews have more negative ones. To be specific, star 4 and 5 reviews have more than 70% positive comments overall and star 1 reviews has more than 50% negative comments. |

**Part 2: Wikipedia**

Source code: [homework2\_wiki.py](https://github.com/Liulinya/mssp607_homework2/blob/master/homework2_wiki.py)

This homework will require you to use HTTP requests to receive JSON objects, and clean real HTML data from the open web. In particular, we will be scraping content from Wikipedia. The English-language Wikipedia has over 5 million articles, and about 0.1% of those have been reviewed by the community as ​**Featured Articles**​. The full list of all featured articles is at the following URL:

https://en.wikipedia.org/wiki/Wikipedia:Featured\_articles

Wikipedia makes all public edits in the history of the wiki available through a public API, documented at ​https://www.mediawiki.org/wiki/API:Main\_page​.

**1. Individual Page Scraping**

Write a function ​**get\_featured\_biographies()** ​to scrape the contents of the list of featured articles and returns a list of names for all featured articles that are also biographies. Then, answer the following questions:

●  How did you determine which featured articles were biographies?

●  What percentage of featured articles are biographies?

|  |  |
| --- | --- |
| Coding Result | Further analysis |
| Among 5760 number of featured articles, 24.29% are biographies. | If the section names include title like “biographies” or “Biographies” but not “autobiographies” (written by self), I will consider it as biographies (written by others). Almost a quarter of featured articles are biographies. |

**2. Scraping a dataset**

Next, write code that scrapes all of the individual pages for featured article biography titles in the list you created in part 1. Write a function ​**get\_first\_paragraph(page)**​that extracts the first paragraph of each biography.

These functions will probably not be able to cover 100% of pages in you dataset; because the data is messy and formatted differently from page to page, they will fail on some of them. With the code you wrote, what percentage of infoboxes and first paragraphs were you able to scrape? What are the characteristics of the pages that your code fails to scrape?

|  |  |
| --- | --- |
| Coding Result | Further analysis |
| Among 1399 number of biographies, 61.19% can be scraped as first paragraphs. | I only scraped biographies pages, which first paragraph begins with life span time. For example, “Bronwyn Bancroft (born 1958) …”, “Natalie Clifford Barney (October 31, 1876 – February 2, 1972)”, or “Richard Barre ([c.](https://en.wikipedia.org/wiki/Circa) 1130 – c. 1202)”. I failed to scrape any other format, such as paragraph name mismatching with title name like “Jacob Isaackszoon van Ruisdael”, or name not along with life span like “William Norman Birkett, 1st Baron Birkett”. |

**3. Extracting information from messy content**

Using regular expressions, write a new method ​**get\_pronouns(text)**​that determines the most common gender of pronouns in a given string of any length. Typically but not always, the three ways gender are marked in pronouns are:

●  Male: he/his/him

●  Female: she/her/hers

●  Plural, or singular non-binary: they/them/their

Answer the following questions about your calculations:

●  What are the drawbacks of your approach, and what types of content are excluded or missed because of the choices you made?

●  What percentage of biographies use he/his pronouns, she/her, or they/them pronouns?

●  What percentage of pages did your code fail to parse, or have unclear gender? Why?

|  |  |
| --- | --- |
| Coding Result | Further analysis |
| Among 856 number of first paragraphs, 9.11% of biographies use she/her;  33.29% of biographies use he/his;  0.82% of biographies use they/them;  56.78% of biographies fail to parse, or have unclear gender. | My approach failed to consider about first paragraphs that don’t include any pronouns listed above. For example, the following paragraph cannot be recognized by my function.  *“Bronwyn Bancroft (born 1958) is an*[*Australian*](https://en.wikipedia.org/wiki/Australian)*artist,*[*[notes 1]*](https://en.wikipedia.org/wiki/Bronwyn_Bancroft#cite_note-6)*notable for being amongst the first Australian fashion designers invited to show her work in Paris. Born in [Tenterfield](https://en.wikipedia.org/wiki/Tenterfield" \o "Tenterfield), New South Wales, and trained in*[*Canberra*](https://en.wikipedia.org/wiki/Canberra)*and*[*Sydney*](https://en.wikipedia.org/wiki/Sydney)*, Bancroft worked as a fashion designer, and is an artist, illustrator, and arts administrator.”* |

**4. Additional analysis**

Define and write a function that will extract one additional quantifiable feature of Wikipedia biographies based on the raw data you scraped. What question did you ask, and why is it interesting? Did you draw any new conclusions based on the feature you found and its distribution in your data? Share any statistics that support your analysis, and include those statistics in your final report.

|  |  |
| --- | --- |
| Coding Result | Further analysis |
|  |  |

**5. Preparing a dataset for sharing**

Then, either using Pandas or built-in data structures like dictionaries and lists, write a function **export\_dataset(df)** ​that will export the values that your pronouns function calculated to a CSV or JSON file with at least three columns for each biography: the page title, the most common pronoun used in the introduction of that page, and the additional variable that you defined in part 4.

To go along with the file, write technical documentation for how the file is constructed; your intended audience is classmates or peers that could work with you. The documentation should include:

●  Explanation of the meaning of each column in your CSV file.

●  Explanation of what data was not successfully scraped by your dataset-building process, and what the limitations are on any future analyses.

●  Instructions on how other data scientists could use your parsing code in their own work (you may include Python code samples if necessary).

●  Sample code that allows future users to load your file using Python and instructions on how to run a basic analysis to confirm that they successfully downloaded the dataset.

|  |  |
| --- | --- |
| Coding Result | Further analysis |
|  |  |