MCIS6273 Data Mining (Prof. Maull) / Fall 2023 / HW0

Points Possible	Due Date	Time Commitment (estimated)
20	Tuesday September 26 @ Midnight	*up to* 4 hours

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by University or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

OBJECTIVES

- Familiarize yourself with Github and basic git
- Familiarize yourself with the JupyterLab environment, Markdown and Python
- Explore JupyterHub Linux console integrating what you learned in the prior parts of this homework
- Listen to the Talk Python To Me from July 7, 2023: How data scientists use Python
- Perfom basic data engineering in Python using Gutenberg.org text of Bertrand Russell's 1912 work *The Problems of Philosophy*
- Use structured data to develop basic statistical analyses

WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, make a directory in your Lab environment called `homework/hw0`. Put all of your files in that directory. Then zip that directory, rename it with your name as the first part of the filename (e.g. `maull_hw0_files.zip`), then download it to your local machine, then upload the `.zip` to Blackboard.

If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using zip in Linux.

If you choose not to use the provided notebook, you will still need to turn in a `.ipynb` Jupyter Notebook and corresponding files according to the instructions in this homework.

ASSIGNMENT TASKS

(0%) Familiarize yourself with Github and basic git

Github (https://github.com) is the *de facto* platform for open source software in the world based on the very popular git (https://git-scm.org) version control system. Git has a sophisticated set of tools for version control based on the concept of local repositories for fast commits and remote repositories only when collaboration and remote synchronization is necessary. Github enhances git by providing tools and online hosting of public and private repositories to encourage and promote sharing and collaboration. Github hosts some of the world's most widely used open source software.

If you are already familiar with git and Github, then this part will be very easy!

§ Task: Create a public Github repo named `"mcis6273-f23-datamining"` and place a readme.md file in it. Create your first file called `README.md` at the top level of the repository.

Please put your Zotero username in the file. Aside from that you can put whatever text you like in the file (If you like, use something like lorem ipsum to generate random sentences to place in the file.). Please include the link to **your** Github repository that now includes the minimal `README.md`. You don't have to have anything elaborate in that file or the repo.

§ Task: Fork the course repository:

- https://github.com/kmsaumcis/mcis6273_f23_datamining/

(10%) Familiarize yourself with the JupyterLab environment, Markdown and Python

As stated in the course announcement Jupyter (https://jupyter.org) is the core platform we will be using in this course and is a popular platform for data scientists around the world. We have a JupyterLab setup for this course so that we can operate in a cloud-hosted environment, free from some of the resource constraints of running Jupyter on your local machine (though you are free to set it up on your own and seek my advice if you desire).

You have been given the information about the Jupyter environment we have setup for our course, and the underlying Python environment will be using is the Anaconda (https://anaconda.com) distribution. It is not necessary for this assignment, but you are free to look at the multitude of packages installed with Anaconda, though we will not use the majority of them explicitly.

As you will soon find out, Notebooks are an incredibly effective way to mix code with narrative and you can create cells that are entirely code or entirely Markdown. Markdown (MD or `md`) is a highly readable text format that allows for easy documentation of text files, while allowing for HTML-based rendering of the text in a way that is style-independent.

We will be using Markdown frequently in this course, and you will learn that there are many different "flavors" or Markdown. We will only be using the basic flavor, but you will benefit from exploring the "Github flavored" Markdown, though you will not be responsible for using it in this course – only the "basic" flavor. Please refer to the original course announcement about Markdown.

§ Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

Play with and become familiar with the basic functions of the Lab environment given to you online in the course Blackboard.

§ Task: Please *create a markdown document* called `semester_goals.md` with 3 sentences/fragments that answer the following question:

- **What do you wish to accomplish this semester in Data Mining?**

Read the documentation for basic Markdown here. Turn in the text `.md` file *not* the processed `.html`. In whatever you turn in, you must show the use of *ALL* the following:

- headings (one level is fine),
- bullets,
- bold and italics

Again, the content of your document needs to address the question above and it should live in the top level directory of your assignment submission. This part will be graded but no points are awarded for your answer.

(0%) Explore JupyterHub Linux console integrating what you learned in the prior parts of this homework

The Linux console in JupyterLab is a great way to perform command-line tasks and is an essential tool for basic scripting that is part of a data scientist's toolkit. Open a console in the lab environment and familiarize yourself with your files and basic commands using git as indicated below.

1. In a new JupyterLab command line console, run the `git clone` command to clone the new repository you created in the prior part. You will want to read the documentation on this command (try here https://www.git-scm.com/docs/git-clone to get a good start).
2. Within the same console, modify your `README.md` file, check it in and push it back to your repository, using `git push`. Read the documentation about `git push`.
3. The commands `wget` and `curl` are useful for grabbing data and files from remote resources off the web. Read the documentation on each of these commands by typing `man wget` or `man curl` in the terminal. Make sure you pipe the output to a file or use the proper flags to do so.

§ Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

(30%) Perfom basic data engineering in Python using Gutenberg.org text of Bertrand Russell's 1912 work *The Problems of Philosophy*

You learned from the prior part that data science is one of Python's strengths.

In this part, you will interact directly with those strengths, but in a way that will allow you to see the challenges that you will face and confront as a real-world data scientist.

Data engineering as you have learned from the readings is about transforming data from one form to another so that it can be used in the appropriate analysis contexts.

One area of intense work is in transforming unstructured data, like a book or text, into structured data. More importantly, producing statistical analyses of these unstructured data is often difficult, because one must convert that unstructured data to something that a machine can process algorithmically.

In this part of the homework you will take a text from the Project Gutenberg https://gutenberg.org and convert it to something more structured. In fact, you will convert it to multiple structured forms.

For this part we will be working with Betrand Russell's 1912 work *The Problems of Philosphy* which is located at the Project Gutenberg's website https://gutenberg.org. The `.txt` file you will want to work with is here:

- https://www.gutenberg.org/cache/epub/5827/pg5827.txt

If you are not familiar with Betrand Russell, you may want to be. He is widely regarded as an important and influential 20th century western logician, mathematician and philosopher who made prolific, deep and crucial contributions to the philosophy of mathematics, logic, set theory, computer science, artificial intelligence, epistemology and metaphysics.

Additionally, if you are unfamiliar with Project Gutenberg, you can learn more about it here: https://gutenberg.org/about/background/. It is an essential repository of many classic books and texts which are now out of copyright, but more importantly it's founder, Michael Hart, invented eBooks in 1971, before probably all of us were born, and certainly before the widespread ubiquity of the public Internet as we know it. It is a fascinating history that you should know a little about.

For our purposes, though, what makes Gutenberg most interesting is that we can directly obtain the `.txt` version of the texts allowing us to use the power of Python to computationally process this unstructured data and convert it to something more useful to our machines and algorithms.

Your code must be implemented in Jupyter as a notebook – you will be required to turn in a `.ipynb` file.

§ Task: Use Python to parse and tokenize the text file.

You will produce a `.csv` file which will have all the full words lowercase and with all punctuation removed *unless* it is part of the word. For example, if you have a token "`world.`", you will drop the ending period, however, if you have a word "`can't`", you will retain the apostrophe "`'`".

Your output `.csv` file will contain all the words in alphabetical order with their frequency counts.

Here is an example of some lines in such a `.csv` file:

```
...

the,112
there,62
thing,3
this,200

...
```

NOTE: Only the words (first column) are sorted, the counts do not need to be sorted.

Please name your file `all_words.csv`.

§ Task: Now that we have all the words, let's go back to the drawing board and **get all *capitalized* (uppercase) words**.

To do this, you will tokenize as before, but you will retain only those words that are capitalized.

Also, as before, you will remove punctuation except when it is part of the word, such as an example of a possessive proper noun like "`Carl's`".

You will also include the frequency counts of these capitalized words in *sorted* order by word.

Please name your file `all_uppercase_words.csv`

§ Task: Answer the following questions:

1. Which were the 5 most frequent words in `all_words.csv` were most frequent?

2. Which were the 5 most frequent words in `all_uppercase_words.csv`.
3. Compare and contrast these top 5. Explain in 2-3 sentences what you observe about the similariries and differences.
4. In your own words, what were the most surprising parts of each list?

(30%) Use structured data to develop basic statistical analyses

Now that we have a sense of taking this text and producing some output files that are quite a bit more interesting, we are going to go further into some statistical analyses.

Of course, one thing that we are concerned about in unstructured data, are elements that do not add much to our understanding or conversion of that data.

One such area in the English language, at least (and most other languages), are words that do not increase the information of the sentence at an *essential* level.

For example, the word `'the'` is not a very useful word when analyzing text, and especially the words that add to the meaning of a sentence. It is usually the *nouns* and *verbs* that get us to the useful parts, and then the *pronouns*, *adjectives*, *adverbs*, etc. Critically, the less common a word is, the more likely that word is important to understanding a text.

We are going to delve into a basic and rudimentary statistical analysis of the text.

When we are done, we should be able to answer a question like *How likely is it to see a sentence with the words `car`, `plant`, `simple`?* We will also continue some basic data engineering along the way.

§ Task: Remove the stopwords from your `all_words.csv` and put the remaining non-stopwords in a file `all_ns_words.csv`. Please retain the frequency column as before.

A good list of stopwords to start with can be found here:

- https://raw.githubusercontent.com/stopwords-iso/stopwords-en/master/stopwords-en.txt

Furthermore, you can learn what a *stopword* is from the excellent text Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. https://nlp.stanford.edu/IR-book/. :

- here is primary source information on stopwords https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html

§ Task: Add a new column to your `all_ns_words.csv` that contains the probability of that word.

To do this, use the denomator of the sum of stopwords **not** all words. Alternatively, do not include stopword counts in your sum.

Thus, W are all words and if w is a non-stopword, $w \in W$, let C_w be the frequency (count) of word w. Thus,

$$\Pr(w \in W) = \frac{C_w}{\sum_{w' \in W} C_{w'}}.$$

Concretely, if "`righteous`" appears 200 times, and the sum of frequencies of all non-stopwords is 10000, then $\Pr(w = righteous) = \frac{200}{10000} = 0.02$.

Your new file will look something like:

```
...
friend, 112, .003
fruit, 67, .00014
grand, 88, .01763
...
```

§ Task: Answer the following questions using your analysis and results from the text:

1. How many unique non-stop words are in the text?
2. Which is the least probable word? (if there is a tie, please state the tie words)
3. What observation can you make about the probabilities?

4. Which sentence is more likely:

    a. *If a belief is true, it can be deduced it is universal.*
    b. *Criticism of knowledge is counter to scientific results.*

    You will use the sum of the probabilities of each non-stop word to answer the question. You will need to give numeric rationale for your answer. Show your work in Python!