CSCI 5015 Assignment 3 Updated
Due Feb 23, 11:59 PM.
50 points

Objective

Create a Jupyter notebook, become familiar with Markdown, and perform some basic text analysis.

Background

Not all data is numeric. Not all data is nicely structured. One example of data that fits both of these statements is text, particularly unstructured text. There is a whole sub-area of data science devoted to unstructured text. Unstructured text would included data like emails, documents, and web pages. Another example are tweets from Twitter. In this assignment, we will do some basic processing of text data. You've been given a data file that is a collection of customer support tweets in CSV format. The original can be downloaded from Kaggle.

The file is fairly simple. It contains these fields:

- tweet_id - numeric
- author_id - string
- inbound - boolean
- created_at - string
- text - string
- response_tweet_id - string
- in_response_to_tweet_id - string

We are mostly concerned with the column **`text`** as that contains the text of the tweets.

Update: The file is not an ASCII file. It is actually a UTF-8 file. The call to **`open()`** needs to be changed in order to read the file without an error. When you open the file for reading, add an **`encoding`** argument like this:

```
open('twcs.csv', encoding='utf-8')
```

Coding instructions

Create a Jupyter notebook file named **assignment3.ipynb**. Unzip the file, **twcs.zip** and place the data file **twcs.csv** in the same folder as your notebook.

Make the first cell in your notebook a Markdown cell and create a header that says "Customer Support Analysis" and then on the next line, have a plain paragraph that says "Written by John Nicholson" (hopefully, it is obvious that you should replace my name with your name). It should look roughly like this after you render the Markdown:

> # Customer Support Analysis
>
> Written by John Nicholson

Use your notebook to answer the questions below. Your notebook should have both code cells and Markdown cells. The code cells should read and process the data, and the Markdown cell should explain what the associated code cell does and why you wrote the code the way you did. You can, and probably should, add extra cells to make your main code easier to read and use.

Requirements

- No single code cell should produce all the output for more than one question. At a minimum, you will need 8 code cells, one for each question.
- If you make additional code cells, they do not have to display data. For example, you may want to write a helper function in a code cell. You still need a Markdown cell describing the function and why you wrote the function in the first place.
- All code cells need a Markdown cell explaining what the code does and why you wrote it the way you did. Explanations should be complete sentences and paragraphs, use good grammar, and be spell-checked.
- Before turning in your notebook, ensure that all code cells that produce output have their output displayed. I should not have to run your notebook in order to see the output. You should all render all Markdown cells. That is, I should see the special Markdown characters like the #. I should see nicely rendered text.

Questions

1. How many tweets are in the data?
2. Some of the tweets contain languages other than English. Demonstrate this by printing the tweet with the **`tweet_id`** 269.
3. How many tweets contain at least 50% non-English characters? For the purpose of this question, "English" means ASCII characters. That will include letters, punctuation, numbers, and white space.
4. How many unique twitter names are used in the tweets? Users can be identified in a tweet with a word that starts with an **`@`** followed by a combination of alphanumeric characters and underscores. Be aware, some tweets contain 0 usernames while others can contain 1, 2 or more usernames.
5. What are the top 10 usernames that appear in the file? That is, count how many times each username is used in a tweet and determine the 10 usernames with the highest count.
6. How many unique hashtags are used in the tweets? Hashtags can be identified in a tweet with a word that starts with an **`#`** followed by a combination of alphanumeric characters and underscores.
7. What are the top 10 hashtags that appear in the file? That is, count how many times each hashtags is used in a tweet and determine the 10 hashtags with the highest count.
8. Excluding hashtags and usernames, what words are used most often in tweets? Display the top 20 words. You don't have to worry about punctuation or plural words. For our purposes, **`John`** and **`John's`** and **`John.`** will be considered three separate words. Uppercase and lowercase should not make a difference. For example, **`computer`**, **`Computer`**, **`COMPUTER`** are all the same word.

Tips and help

You may want to read Python's documentation on the **`string`** module, https://docs.python.org/3.1/library/string.html.

Dictionaries are useful things. Remember, their keys will be unique. Dictionaries are very useful for counting occurrences of things.

This is a big file, and it will take a while to process it. You may want to do short tests that only run on tweets with **`tweet_id`** less than some value, for example, 100, 300, or 1000.

When you are running code, you may not see results right away. You should look to the left of the code cell for the **`In [ ]:`** marker. If you see a number, for example, **`In [8]:`**, the code in the cell is not currently running. But if you see an asterisk, for example, **`In [*]:`**, the code in the cell is running on the kernel, and results may not be available yet. The asterisk should change to a number when the code is done.

Remember, you can stop a running cell by choosing the "stop" button in the menu. It is the one that looks like a black square.

You can always come see me in my office or send me email. If you send me email, you should send your code as an attachment. Don't copy/paste your code into the message because that will make it harder for me to debug your code. Send your file as a **`.ipynb`** file

Submission

When your program is correct, log into D2L and locate the Dropbox for assignment 3. Upload the file

- assignment3.ipynb

into the D2L dropbox.

Contact me if you have any problems.