

# MIS 515 HOMEWORK 4:

## CUSTOMER SERVICE ANALYTICS

Your assignment is to create a tool that allows the user to assess the linguistic properties of major companies' customer service tweets. Your program should first ask the user to specify the type of analysis they would like to perform. Your program should allow the user to perform an unlimited number of analyses.

The dataset is available at [https://dgoldberg.sdsu.edu/515/customer\\_service\\_tweets\\_full.json](https://dgoldberg.sdsu.edu/515/customer_service_tweets_full.json) and contains approximately 1,000 tweets for each of twelve major companies' customer service Twitter handles, listed alphabetically below:

@amazonhelp	@AppleSupport	@Ask_Spectrum	@AskPlayStation
@comcastcares	@hulu_support	@SpotifyCares	@sprintcare
@TMobileHelp	@Uber_Support	@UPSHelp	@XboxSupport

The data is formatted as a JSON array. An example of the formatting is below:

```
[  {"Company": "@sprintcare", "Text": "I understand. I would  
like to assist you. We would need to get you into a private  
secured link to further assist."},  
  ...  
  {"Company": "@UPSHelp", "Text": "Hello, please click the  
link to let us know how we can assist you. Click the link  
to DM us with your tracking and phone number. ^E.W  
https://t.co/wKJHDXWGRQ"}]
```

Your tool should support the following types of analyses:

- Polarity: for each company in the dataset, your tool should calculate the average sentiment polarity across that company's customer service tweets. Ensure that you calculate sentiment scores per-tweet rather than per-sentence. Average polarity values for each company should be printed and displayed visually in a bar graph.
- Subjectivity: for each company in the dataset, your tool should calculate the average sentiment subjectivity across that company's customer service tweets. Ensure that you calculate sentiment scores per-tweet rather than per-sentence. Average subjectivity values for each company should be printed and displayed visually in a bar graph.

- Readability: ask the user whether they would like to analyze by Flesch-Kincaid Grade Level or SMOG index. If the user fails to select one of these options, warn them that their choice was invalid.
  - Flesch-Kincaid Grade Level (FKGL): this metric is an attempt to estimate the grade level (e.g., third grade reading level) of text based upon its word choice and sentence structure. For each company in your dataset, your tool should calculate the average FKGL across that company's customer service tweets. Average FKGL values for each company should be printed and displayed visually on a bar graph. FKGL is computed as follows:

$$0.39 \left( \frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \left( \frac{\text{total syllables}}{\text{total words}} \right) - 15.59$$

To count total syllables, install the `syllables` module (in Colab, `!pip install syllables`), which can estimate the number of syllables in a word, sentence, or passage. Syllable counts can be estimated using the `syllables.estimate()` function. For example, using `syllables.estimate("coronavirus")` returns 5.

- SMOG index: this metric is an attempt to estimate the number of years of education required to understand text based on its word choice and sentence structure. For each company in your dataset, your tool should calculate the average SMOG across that company's customer service tweets. Average SMOG values for each company should be printed and displayed visually on a bar graph. SMOG is computed as follows, where a "polysyllable" refers to a word that is three or more syllables long:

$$1.043 \sqrt{\text{number of polysyllables} \times \frac{30}{\text{number of sentences}}} + 3.1291$$

- Search: ask the user which Twitter handle they would like to search. For that Twitter handle, compute and print the average polarity, average subjectivity, average Flesch-Kincaid Grade Level, and average SMOG index. Your code should handle the case that the user searches for a Twitter handle that is not in the dataset and print an appropriate warning if this occurs.

Write your code such that the entirety of your program is case insensitive (for example, the program should behave equivalently if the user enters "yes", "Yes", or "YES" or if they enter "@sprintcare" or "@SprintCare"). However, do not perform any spellchecking on this assignment (unfortunately, the spellchecker we have used corrects "polarity" to "popularity").

*HINT: It is possible to structure your code in a way that does not require excessive copying and pasting. One approach, discussed in class previously, is to take advantage of functions; for example, rather than having multiple sections of your code that calculate polarity, you might instead write one polarity function that you call multiple times. An additional strategy that may be helpful on this assignment is to take advantage of for loops. Rather than copying and pasting the same analysis for each company, an alternative might look as follows:*

```
companies = ["@amazonhelp", "@applesupport", "@ask_spectrum",
"@askplaystation", "@comcastcares", "@hulu_support",
"@spotifycares", "@sprintcare", "@tmobilehelp", "@uber_support",
"@upshelp", "@xboxsupport"]

for company in companies:
    # This loop selects one company at a time
    # First iteration: company = "@amazonhelp"
    # Second iteration: company = "@applesupport"
    # ... last iteration: company = "@xboxsupport"
    # Insert some analysis here to run once per company
```

Some considerations to note:

- Consider the possibility that, when loading the dataset, some connection issue occurs (that is, a status code other than 200). Ensure that your code handles this case and provides the user with a helpful printout if it does occur.
- There is a freely available module capable of computing readability measures called `textstat` ([find the details here](#)). Although this module is easy to use, its computations make some questionable assumptions. For example, it tokenizes sentences under the assumption that a period always marks the end of a sentence, which `textblob` and/or `nltk` handle much more carefully. Although it may be tempting, please *do not* use `textstat`.
- The examples on the following pages use rotated labels on the x-axis to improve the legibility of the bar graphs. To adjust this setting for the x-axis labels in `matplotlib`, `plt.xticks(rotation = 45, ha = "right")` was used. This is optional.
- Due to the size of the full dataset, some analyses may run rather slowly. For your testing purposes, you may use a smaller dataset that contains approximately 50 rather than 1,000 tweets per company. This small dataset uses the same format as the full dataset and is located at [https://dgoldberg.sdsu.edu/515/customer\\_service\\_tweets\\_small.json](https://dgoldberg.sdsu.edu/515/customer_service_tweets_small.json). However, ensure that your final submission is built to analyze the full dataset.

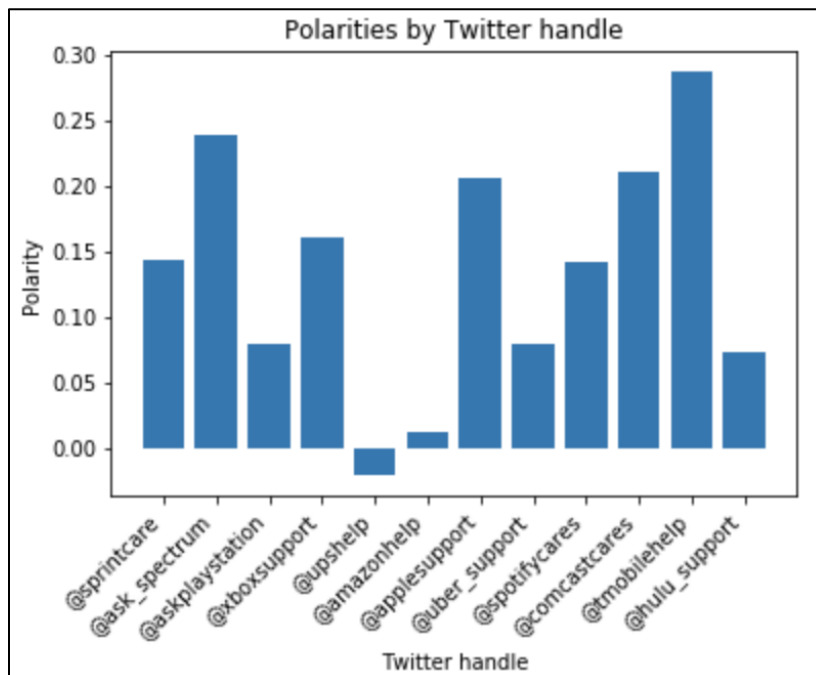
- Ensure that your prompts and output are crisp, professional, and well-formatted. For example, ensure that you have used spaces appropriately and checked your spelling. Ensure that graphs are appropriately titled and that axes are appropriately labeled.
- Adding comments in your code is encouraged. You may decide how best to comment your code. At minimum, please use a comment at the start of your code to describe its basic functionality.

Please use the following as a template for the tool's expected functionality (*full dataset*):

Welcome to the customer service linguistics analyzer!

Which analysis would you like to perform  
(polarity/subjectivity/readability/search)? Polarity

```
@sprintcare: 0.1441766155603655
@ask_spectrum: 0.2387153826960076
@askplaystation: 0.0799678848003847
@xboxsupport: 0.16134886649230387
@upshelp: -0.020923390993824426
@amazonhelp: 0.012632791606541599
@applesupport: 0.20552694173881667
@uber_support: 0.07891686147186143
@spotifycares: 0.14145763313406176
@comcastcares: 0.21032808897121366
@tmobilehelp: 0.2874421763768641
@hulu_support: 0.07364242688792685
```

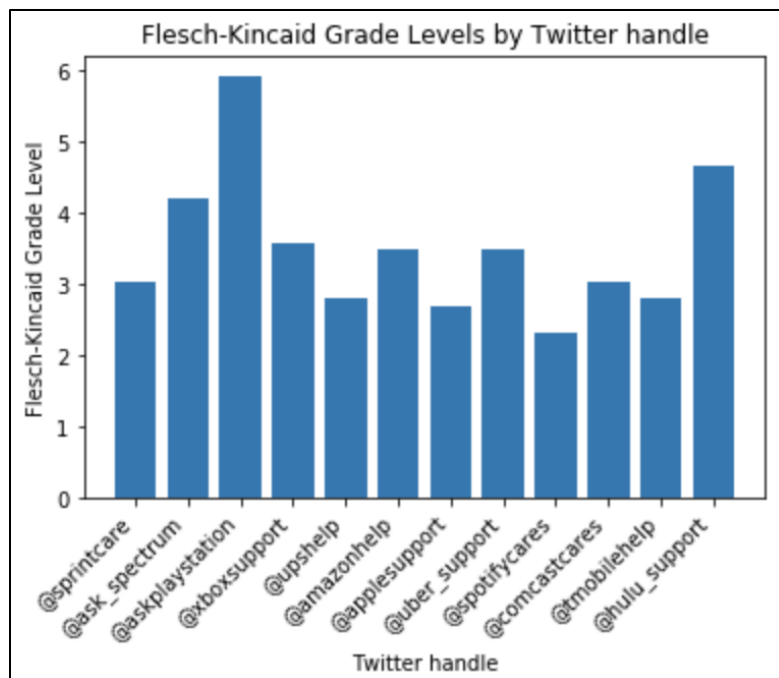


Would you like to run another analysis (yes/no)? yes  
Which analysis would you like to perform  
(polarity/subjectivity/readability/search)? search  
Which Twitter handle would you like to search? @upshelp

Average polarity: -0.020923390993824426  
Average subjectivity: 0.4081721997854607  
Average Flesch-Kincaid Grade Level: 2.7999036537954005  
Average SMOG index: 7.315399000283902

Would you like to run another analysis (yes/no)? YES  
Which analysis would you like to perform  
(polarity/subjectivity/readability/search)? profit  
Sorry, that type of analysis is not supported. Please try again.  
Would you like to run another analysis (yes/no)? Yes  
Which analysis would you like to perform  
(polarity/subjectivity/readability/search)? readability  
Would you like to analyze FKGL or SMOG? ABCD  
Sorry, that type of analysis is not supported. Please try again.  
Would you like to run another analysis (yes/no)? yes  
Which analysis would you like to perform  
(polarity/subjectivity/readability/search)? READABILITY  
Would you like to analyze FKGL or SMOG? FKGL

@sprintcare: 3.0329275523159693  
@ask\_spectrum: 4.220037957756203  
@askplaystation: 5.916860592588254  
@xboxsupport: 3.5803500408234434  
@upshelp: 2.7999036537954005  
@amazonhelp: 3.480966816998824  
@applesupport: 2.7059316297371927  
@uber\_support: 3.4990759106428193  
@spotifycares: 2.310499516709366  
@comcastcares: 3.0421282080843373  
@tmobilehelp: 2.8096163639219127  
@hulu\_support: 4.664881752457174



Would you like to run another analysis (yes/no)? YES  
Which analysis would you like to perform  
(polarity/subjectivity/readability/search)? search  
Which Twitter handle would you like to search? @cocacola  
Sorry, that Twitter handle was not found. Please try again.  
Would you like to run another analysis (yes/no)? no