Northeastern University
College *of* Professional Studies

ALY 6980: Final Individual Coding Challenge

This challenge was inspired by a real data challenge from an AI company in which they asked the candidate to create their own "AI" and then construct a pipeline for analyzing the outputs.

You may turn in either an R Markdown document or a Jupyter Notebook. The due date is Saturday, EOD, March 28th. Please upload both an HTML version and a raw RMD/ipynb and ensure that everything is reproducible so that I can play with your code if needed.

Assignment

The assignment has three components:

1) Create a mock "persona", which is a function that takes an arbitrary string of text as input (a "message") and outputs a number between 0 and 1 (an "interest score"). The persona should translate the message into an interest score with the following high-level steps:

1. Accept an arbitrary message as input
2. *Tokenize* by splitting the message into individual words
3. *Vectorize* by replacing each word with a number. The numbers should be reproducible (i.e., if you run the same word, you should get the same number every time) and should somehow reflect the "personality" of your persona.
4. *Summarize* the vector of numbers with a single number (e.g., average them)

2) Design a "message optimization" pipeline to evaluate "lift" in the interest score associated with each word. The pipeline should do the following high-level steps:

1. Accept message as input
2. Compute the baseline interest score using your persona
3. Iteratively delete words from the message and compute the change in interest score
4. Output the "lift" associated with each word. Positive values mean the interest score improved, negative values mean it hurts the interest score, and 0 means it has no effect

3) Present the output to users in a clear and compelling way that you think will be most helpful for them in optimizing their message.

At minimum, you can simple show the table with all of the "lift" scores for each word in the message. More advanced approaches could involve word clouds or curated and sorted lists of the "best" and "worst" words.

Rubric

Grading an assignment like this is inherently subjective, so I am keeping the rubric on a very high level. Below are the three major categories of "success" you can aim for:

1. Accepted criteria for sharing with Steve

This is based on my subjective determination of what Steve will find interesting or valuable. I may choose anywhere from 0 to 30 (but probably closer to half a dozen). It is up to you to argue that your work is valuable and that it makes sense.

2. "A" on the assignment

- The document is clean, organized, and has enough narrative text and comments to make sense of what the code is doing.
- The persona has a real "personality" in that it can take any input and systematically outputs preferences that have some predictability to them.
- The pipeline correctly determines the "lift" for dropping each word from the message.
- The results are presented in an easy-to-understand format for non-technical users.

3. "MVP" (minimum viable product / passing grade)

- The document is clean, organized, and has enough narrative text and comments to make sense of what the code is doing.
- The persona reproducibly outputs values from 0 to 1, but they do not map on to any sort of "personality" (so this can be an incredibly simple function)
- The pipeline correctly determines the "lift" for dropping each word from the message.
- Results are presented in an accurate, but not visually appealing way. For example, printing the table with the word/lift data would satisfy the assignment requirements.

Cheating

This is a programming assignment, so it is essential that students write their own code. This project is complex enough that nobody should have identical code. If I find improbable similarities between people's code, expect a severe reduction in your assignment grade.

General advice

- Any decisions that are not clearly specified are up to you to decide
- Brainstorm with others, but work independently
- Ask me questions if anything is unclear or you get stuck
- This is a proof-of-concept exercise; it does not need to be perfect!

Tips on giving your persona a “personality”

Your persona must have some kind of internal logic to decide how much the persona is interested in each word and ultimately the overall message. It should not output random nonsense. This is probably the trickiest part of the assignment, so here are some ideas on how to get started:

The simplest thing I can imagine is use the length of the word. You could say this gives the persona a personality of “likes big words”, which is pretty silly and simple and will not earn you the highest grade, but it would satisfy the basic requirements of the assignment.

A more complex approach could involve sentiment analysis that codes each word as a particular sentiment, and you can create your persona’s “personality” by designing it to respond to particular sentiments (e.g., a persona that likes disgusting things or angry things).

The most advanced approach I can think of would use word embeddings (e.g., Word2Vec, BERT) to translate each word into an embedding and then compute its similarity relative to a set of words your persona is “interested” in (e.g., using cosine or Euclidean distance between the 2 vector embeddings to compute the score).