Natural Language Processing and Machine Learning

There are five rules that apply to all projects so far:

a) Follow instructions precisely.
b) Do not use any functions or approaches to problems that we did not cover in this course.
c) All code must be scalable by sample size unless specifically noted otherwise.
d) Any code using magrittr should contain a max of one verb per line.
e) Always use the standard heading set up to this point in all files unless otherwise specified: **R Studio API Code, Libraries, Data Import and Cleaning, Analysis, Visualization**.

Part 1 – Set up a new R Studio Project with one R script called week13.R

Part 2 – Data Import and Cleaning

1. Create a tibble called *imported_tbl* containing the last 1000 posts from this page: https://www.facebook.com/groups/teachpsych/
2. Create a variable called *messages* containing the post text from *imported_tbl* by copy-pasting the following command (we'll discuss what this does in the project debrief):
   messages <- imported_tbl$message %>% iconv("UTF-8", "ASCII", sub="") %>% str_replace_all("^[:graph:]", "") %>% str_replace_na
3. Create a corpus called *facebook_cp*. When you do so, apply appropriate pre-processing algorithms except for stemming. Create a unigram DTM called *prestem_dtm* using this corpus.
4. Modify *facebook_cp* by converting text to stems, then create a new unigram DTM called *stemmed_dtm*.

Part 3 – Visualization

5. Generate a word cloud of up to the top 50 most frequent words in *prestem_dtm*.
6. Generate a horizontal bar chart of the top 20 unigrams in *stemmed_dtm,* ordered by most common on the top and least common on the bottom.

Part 4 – Analysis

7. We're going to run some machine learning algorithms on this dataset, but to reduce processing time, we're going to cut down the dataset a priori in a few ways. First, use the following function to create a new DTM containing only words that appear in at least 3% of cases:
   slimmed_dtm <- removeSparseTerms(stemmed_dtm, .97)
8. Next, create a new tibble called *slimmed_df* combining the variable counting likes from *imported_tbl* with the entire contents of *slimmed_df.*
9. Using two different appropriate machine learning algorithms, predict like counts from word usage using 10-fold cross validation. Ensure model performance can be compared across algorithms.
10. Conduct an appropriate analysis to determine which is likely to be more generalizable. Provide a visualization of this comparison.
11. Create a comment describing the predictive power of each model, explaining which you would recommend for use in a real-world environment and why.