CS688 Assignment 3: Text Mining

Please follow the submission requirements at the end of the assignment!

Objectives: Demonstrate your ability to create corpus objects, pre-process text, create a DTM, and learn something useful from that DTM.

Make sure you complete both pages of questions.

(A note: This assignment uses the k Nearest Neighbors classifier as a tool. Documents in the train partition are used to train the classifier to then predict what the classification should be for documents in the test partition. Classification algorithms are useful tools. You can learn more about k Nearest Neighbors and many other classification algorithms in CS 699. This class is not intended to expose you to the details of classification.)

Similar to the classification example given in Module 3, process and classify the newsgroup document data. You can download this data from Blackboard and save it on your computer, for example in your "tm/text/" folder. Note that the data is separated into one test and one train folder, each containing 20 sub folders on different subjects. Choose these 2 subjects to analyze (**sci.space** and **rec.autos**) and 100 documents from each.

The merging/splitting technique used here requires some careful thought. It is used here to ensure that you get a DTM that contains the same columns (same column names and same ordering of these columns) for both the train and test partitions of the data. There may be other approaches to this that don't require merge/split, but this is an easy approach.

Your submission must accomplish the following tasks:

a) (15 points) Create five Corpus (or VCorpus) objects.
   - 100 "sci.space" documents for training from the correct train folder
   - 100 "sci.space" documents for testing from the correct test folder
   - 100 "rec.autos" documents for training from the correct train folder
   - 100 "rec.autos" documents for testing from the correct test folder
   - A merged corpus with 400 documents. Pay attention to the order in which you combine the four 100 document corpora because you will need to reverse this merge in part d below.

b) (15 points) Implement preprocessing (clearly indicate what you have used and make sure each preprocessing step works as expected, i.e., stop words really removed, punctuation really removed, etc.)

c) (10 points) Create the Document-Term Matrix. Use the "control" parameter to require
   - A minimum word length (if unsure, require words of at least 2 letters or longer)
   - A minimum number of occurrences each in the merged corpus (if unsure, require that each word in the DTM appears globally in the corpus at least 5 times)

d) (30 points) Classify and display results.
   - Classify text using the knn() function. This will require you to:
     - Split the Document-Term Matrix into a train DTM and a test DTM.

        - Make a vector of length 200 which indicates the correct classifications for the documents in the train partition. To simplify grading of this question please use "Sci" and "Rec" as the levels of this vector.
        - You may, if you wish, choose to run knn() multiple times in an attempt to improve classifier performance (e.g., get a better answer for part e). However, this is not required for this assignment.
    - Re-arrange the classification results into a data frame. It must contain 200 rows and at least the following four columns, which you must name appropriately:
        - Document number (from the test partition)
        - Predicted classification (either "Sci" or "Rec")
        - The probability, according to the classifier, that the predicted classification should be the right classification
        - For each document in the test partition, an assessment of whether the predicted classification is the correct classification. This should be either a TRUE or a FALSE for each document. This can be done by comparing the predicted classification to the "correct tags" for the test partition. (Note, if you have followed the instructions correctly, the "correct tags" for the test partition should be the same as the "correct tags" for the train partition.)
    - To save space in the report, please display only the first 6 and the last 6 rows of the results data frame.

e) (10 points) What percentage of documents in the test partition that have been predicted correctly?

f) (10 points) Explain the difference between the classification probability (from part d's results data frame) and the percentage of correct classifications (from part e).

g) (10 points) Based on your attempts to maximize the percentage of correct classifications (from part e), evaluate the effectiveness of this classification algorithm on this data set. Is this a good classification algorithm to use for this type of data? Discuss how you reached this conclusion.

SUBMISSION REQUIREMENTS:

- **Create a Word, PDF, or Rmd document. If you use Rmd you will need to make sure to save the output as a PDF.**
- **For each question, state the question you are answering. Then answer the question by explaining in sentences (in English, not in R or other languages) what you did to get to the answer. You may include screenshots and/or copy-paste of key lines of code and the corresponding output in your answer. (If you are using Rmd, this means you must generally use echo=FALSE and/or include=FALSE for the body of the document.)**
- **Full code should be included as an Appendix to your Word or PDF document. Coding must be in R. Do NOT include full code in the main part of your document.**
- **Please ensure that a Word or PDF file as the first file in your submission.**
- **You may also separately upload your R and/or Rmd code to Blackboard.**
- **If your facilitator tells you to submit the files differently than the above guidelines, you are expected to respect your facilitator's wishes starting on the next assignment.**
- **Facilitators can deduct up to 20% if you fail to follow these requirements (more if the questions are not actually answered).**

- **Facilitators can deduct 5% for each day the assignment is late. You may submit one (and only one) of the six assignments up to three days late with no penalty but all other assignments will be penalized.**
- **Unless your facilitator or the professor agrees, your assignment will not be graded if it is more than 3 days late (e.g., no credit will be given after Friday at 6 AM Boston time). The professor will usually ask the facilitator to make the decision but in rare cases (<1% of the time) has overridden a facilitator. Do not expect the professor to override in most cases.**