Thanks, I used the new notebook and all the code runs without errors.

SPARSE

Code: findFreqTerms(frequencies, lowfreq = 10)

• Results: 'care'

• 'clinic'

• 'clinician'

• 'inform'

• 'organ'

• 'practic'

• 'present'

• 'provid'

• 'topic'

This has nothing to do with your code but is a question about the impact of stemming on the results. When stemming from plural to singular, as an example, “provides” (#8) the result is “provid” and not “provide”. The latter (provide) is the correct spelling. I have run into this before, the stemming package removes “es” regardless of the word. If, for example, the word is “plus”, and the plural is “pluses” then removing the “es” results in the word spelled correctly (plus). I would like your opinion on stemming in this case. If the word “provide” was in the data and the word “provides” was in the data then stemming the plural results in two different words, “provide” and “provid”. My belief is that these words are not combined and are treated as two different words. In a few recent projects, I manually recoded plural words to ensure that words like “provides” are combined with their non-plural word (e.g., “provide”). Is my understanding correct and is this a shortcoming of the package?

ANSWER : That is just how the Porter Stemmer works. The reason for this is that it allows fairly simple rules to create the stems without having to store a large English vocabulary. For example, I think that you would not like that both change and changing go to chang. It seems more natural that they should both stem to change. So would you make a rule that if you take ing off the end of a word, you should add back e to get the stem? Then what would happen with clang and clanging? The Porter Stemmer gives clang. Adding e would give the non-word clange. Either you use simple processing rules that sometimes create stems that are not words, or you must include a large vocabulary and have more complex rules that depend on what the words are. The Porter Stemmer uses the simple rules method.

Reference link: https://stackoverflow.com/questions/42908720/word-stemming-in-r

SUPERVISD ANALYSIS WITH COMBINED UNSORTED AND OUTPUT DATASETS

result\_df <- rbind(result, output\_df) – combines the unsorted data (result) with the sorted data (output\_df) and then further processing with to create “output\_sparse”

ANSWER : YES

SPLIT DATA INTO TRAINING AND TESTING

Code: split <- sample.split(output\_sparse$clusters, SplitRatio = 0.7).

My interpretation is that “output\_sparse” contains both the sorted and the unsorted data and that the variable $clusters are the cluster numbers from my analysis of the sorted data and the prior analysis of unsorted data in the code. As a result, “split”, the train and test datasets contain a mix of sorted and unsorted data.

ANSWER: YES, CORRECT.

predictRF <- predict(outputRF, newdata = testSparse)

table(testSparse$clusters, predictRF)

Accuracy= 0.350649350649351

I am having difficulty understanding this section. The model is trying to predict cluster membership of the combined (sorted and unsorted) dataset first on training and then applying to test data. My confusion is that each term/word has a cluster number, but the numbers do not have the same meaning. For example, cluster #4 from the sorted data is simply the fourth cluster and the number are an arbitrary assignment to categorize content. Cluster #4 from the analysis of the unsorted data is likewise an arbitrary label. In the “result\_df” it seems these clusters values are treated as equivalent but there is no reason for the content of cluster 4 in the sorted data to have any resemblance to the content of cluster 4 in the unsorted data. If my understanding is correct, then it seems that accuracy of prediction would have to be low because the cluster values (1-5) do not have a consistent meaning. Am I misunderstanding what is happening here?

ANSWER: The model doesn’t care about the meaning of clusters. It just take the all labels and fits the data points first in training set. And predicts the labels in test set.