DS-GA 1015, Text as Data
Prof. Andrew Halterman
Assignment date: March 28, 2022

Homework 2

This homework must be turned in on NYU Brightspace by **Monday, April 11, 2022, at 4pm**. Late work may be turned in up to 2 days late and will incur penalties of the equivalent of one half of a numeric grade per day late. (e.g., a 3 hour delay would move a 4/5 problem set to a 3.5/5, a 27 hour delay would move it to a 3/5).

It must be your own work, and your own work only—you must not copy anyone's work, or allow anyone to copy yours. This extends to writing code. You may consult with others, but when you write up, you must do so alone.

Your homework submission must be a PDF or HTML report, containing all written answers and code, generated from `RMarkdown`. **Raw `.R` or `.Rmd` files will not be accepted.**

Please remember the following:

- Each question part should be clearly labeled in your submission.
- Do not include written answers as code comments. We will not grade code comments.
- The code used to obtain the answer for each question part should accompany the written answer.
- **Your code must be included in full, such that your understanding of the problems can be assessed.**

  Please consult "RMarkdown Basics" on the course GitHub for help with RMarkdown. You can also use the "Sample RMarkdown HW" template on the course GitHub to get started. Using this template is not required.

Part 1

Part 1 of the problem set is an **optional** section and is meant to help you with progress toward your final project. It is worth **extra credit** toward the homework grade, with a maximum of 2 numeric points (e.g. 4.5/5 $\rightarrow$ 6.5/5). It should be turned in as a separate document on Brightspace.

Research Topic: State your general area of research here and why it's interesting. (2-3 sentences)

Research Question: Write your specific research question in 1 sentence.

Independent Variable: What is your independent variable? How will you define and measure it? (1-2 sentences)

Outcome variable: What is your outcome variable? How will you define and measure it? (1-2 sentences)

Text data: Where will your text come from? Do you already have it? What obstacles do you need to overcome in order to use it? (2-3 sentences)

Method/research design: Which text analysis method do you plan to use? Why is it appropriate? How will it enable you to answer your broader research question? (2-3 sentences)

Part 2

1. We would like you to perform some Naive Bayes classification **by hand** (that is, you may use math functions or DFM-creating functions, but not any built-in naive Bayes functions). Make sure to show your work!

   (a) Imagine a situation in which you are citizen of Georgia and you receive eight emails from Jon Ossoff (Democrat) and David Perdue (Republican) in anticipation of their 2021 Senate runoff election. The contents of those emails after all relevant preprocessing are displayed in Table 1. Using the standard Naive Bayes classifier without smoothing, estimate for each party the posterior probability (up to a proportionality constant: i.e., the prior multiplied by the likelihood) that the following email was sent by the respective candidate: "healthcare voter tax help jobs". Report these estimates. Based on these results, which party would you predict sent the mystery email? Explain whether you trust your findings and why.

email	content
Perdue1	immigration aliens criminals country loophole
Perdue2	voter economy tax jobs security
Ossoff1	immigration country diversity help security
Ossoff2	healthcare affordable expansion unfair help
Ossoff3	voter relief debt jobs help
Perdue3	healthcare cost socialism unfair help
Ossoff4	abortion choice right women help court

Table 1: Training set of Georgia senate candidate emails.

   (b) Now impose Laplace smoothing on the problem. That is add one to the numerator and the size of the vocabulary to the denominator, then re-estimate each candidate's respective posterior probability. Report your findings. Based on these new results, which candidate would you predict sent the mystery email? Beyond computational reasons (i.e. avoiding $log(0)$'s), can you explain any theoretical reasons that smoothing would make sense (hint: the above data is but a sample of each candidate's shared language)?

Part 3

For this exercise you will use a database of Trip Advisor reviews from Alam, M. H., Ryu, W. J., Lee, S., 2016. Joint multi-grain topic sentiment: modeling semantic aspects for online reviews. *Information Sciences* 339: 206–223. (paper here, data here). Each user left a star rating of 1-5 along with a written review. You'll be asked to use some of the supervised learning techniques we've discussed in class to analyze these texts.

The data are available in the file `tripadvisor.csv`.

Before we get started, be sure to actually read a few of the reviews, to get a feel for the language used, and any potential imperfections in the text created during the scraping process.

For each task (3) through (6), begin with the raw version of the text, and briefly explain which pre-processing steps are appropriate for that particular task.

2. Before we apply any classification algorithms to the Trip Advisor reviews, we will need a general classifier that tells us whether the review was positive or negative—the "true classification".

   (a) Divide the reviews at the empirical median star rating and assign each review a label as being "positive"—if the user rating was greater than or equal to the empirical median score—or "negative"—if the rating is less than the empirical median (you can use "1" and "0" as labels if you prefer, just be consistent as you do the exercises below). These are your *true class* labels. Report the proportion that are positive and negative, and the median star rating.

   (b) For some tasks, we will need "anchor" texts at the extreme of the distribution. Create a variable (name it "anchor") that has value "positive" if the user star rating given to a review is equal to 5, "neutral" if the user rating is less than 5 but greater than 2 and finally "negative" if the user rating is equal to 1 or 2. Report the proportion of reviews that are anchor positive, neutral and negative.

3. The first method we'll use to classify reviews as being positive or negative will be dictionary based. To do so, you will use the dictionaries of positive and negative words discussed in Hu & Liu (2004)—provided to you as "negative-words.txt" and "positive-words.txt". You **must** use the dictionaries provided and may not use any substitutes from R packages.

   (a) Briefly explain which pre-processing steps are appropriate for this particular task. Then, generate a sentiment score for each review based on the number of positive words minus the number of negative words. Create a histogram to visualize the distribution of the continuous sentiment score.

(b) Create a vector of dichotomous variables, of equal length to the number of reviews, in which texts that have a positive sentiment score (from part (a)) are labeled "positive," while those with a negative score are labeled "negative"; if the sentiment score is equal to 0, score them as *negative*. Report the percent of reviews in each category, and discuss the results.

(c) Evaluate the performance of your model at identifying positive or negative reviews by creating a confusion matrix with the positive and negative values assigned by the sentiment score (created in 3(a)) on the vertical dimension and the binary "true" classifications (created in 2(a)) on the horizontal dimension. Use this confusion matrix to compute the accuracy, precision, recall, specificity, and F1 score of the sentiment classifier. Report these findings along with the confusion matrix. In terms of accuracy, how would you evaluate the performance of this classifier? (Hint: is there a baseline we can compare it to?)

4. Next, we'll train a Naive Bayes classifier to predict if a review is positive or negative.

(a) Briefly explain which pre-processing steps are appropriate for this particular task.

(b) Use the "textmodel" function in quanteda to train a *smoothed* Naive Bayes classifier with uniform priors, using 20% of the reviews in the training set and 80% in the test set (Note: features in the test set should match the set of features in the training set). Report the accuracy, precision, recall and F1 score of your predictions. Include the confusion matrix in your answer.

(c) Were you to change the priors from "uniform" to "docfreq," would you expect this to change the performance of Naive Bayes predictions? Why? Re-estimate Naive Bayes with the "docfreq" prior and +1 smoothing. Report the accuracy, precision, recall and F1 score of these new results. Include the confusion matrix in your answer. In terms of accuracy, how would you evaluate the performance of this classifier?

(d) Fit the model without smoothing and a uniform prior. Report the accuracy, precision, recall and F1 score of your predictions. Include the confusion matrix in your answer. How does the accuracy compare to the previous models? Why might this be?

(e) In the above exercise we only used words as features. Can you think of other features beyond words that may help classify the sentiment of a document?

5. Next, we will estimate the latent ideological space of some party manifestos using the "wordscores" function in quanteda.

(a) Create a vector of `wordscores` for the Labor Party 1945 Manifesto (anchor labor) and for the Conservative Party 1983 Manifesto (anchor conservative) using the technique described in Laver, Benoit & Garry (2003). That is, you should fit a `wordscores` model to the anchor texts. What are the 10 most extreme words in either direction (i.e. the 10 lowest and 10 highest wordscores)? Report your findings. *Note: Briefly state your choice regarding smoothing for this model.* You should assign the Conservative and Labour anchor texts scores of -1 and 1, respectively.

(b) Apply your wordscores model to the two non-anchor documents. This should generate a wordscores estimate for each document. Report your results. Did you correctly predict the party? What does the warning message suggests? What can you tell about the shrinkage of the scores? Can this be problematic? Why?

(c) Now, fit the model with the standard LBG (Laver, Benoit & Garry) rescaling (`rescaling = 'lbg'`). How does your results compare to the previous specification without rescaling?

6. Now we'll do the classification task using a Support Vector Machine (SVM) from the `caret` package. Since SVM functions are computationally intensive, restrict your analysis to the first 1000 Trip Advisor reviews using the original ordering of the review data.

(a) Briefly explain which pre-processing steps are appropriate for this particular task. Describe an advantage offered by SVM or Naive Bayes relative to the dictionary approach or wordscores in classifying positive and negative reviews.

(b) In this step, you will train SVM models with a linear kernel. Your goal is to maximize out-of-sample accuracy by fitting models with 5-fold cross-validation. Optimize over the relative size of the training and test sets, try each value from 10 to 90 (by 10s). The remaining data is the validation set. For example, the last model in this sequence uses 90% of the data for the 5-fold CV, and 10% is saved as the validation set. Report which model has the highest accuracy for out-of-sample predictions made on the validation set. In terms of accuracy, how would you evaluate the performance of this classifier?

(c) To the same restricted dataset used in question 3(b), apply a logistic regression using caret (`method = 'glm', family='binomial'`). Report your findings. Because the algorithm does not converge, results show where the algorithm ended with the maximum iteration. What do you think would help make the algorithm converge? Discuss how this accuracy measure differs from the highest accuracy produced by the SVM model. *Note: Set the chunk options* `cache=TRUE, cache.lazy=FALSE`*, otherwise you won't be able to knit your homework.*

7. Finally, let's try out a Random Forest classifier. For this question use the first 500 Trip Advisor reviews in the dataset.

(a) Split the dataset into a training (80%) and a test set (20%) and construct a document feature matrix for each (Note: features in the test set should match the set of features in the training set).

(b) Using the `randomForest` package fit a random forest model to the training set using the package's default values for `ntree` and `mtry` (set the `importance=TRUE`). After fitting the model, extract the *mean decrease in Gini index* (hint: see `$importance`) for the feature set and order from most important to least important. What are the top 10 most important features according to this measure?

(c) Using the fitted model, predict the sentiment values for the test set and report the confusion matrix along with accuracy, precision, recall and F1 score.

(d) Now you will do some tuning of a model parameter. The package's default value for the argument `mtry` is `sqrt(# of features)`. Estimate two more models, one for each of these values of `mtry`: `0.5*sqrt(# of features)` and `1.5*sqrt(# of features)`. As you did above, use each of the fitted models to predict the sentiment values for the test set. Report the respective accuracy scores. Which value of `mtry` yielded the best accuracy?