DS5110 HW 6 - Due Apr. 5

Kylie Ariel Bemis

10/31/2018

Instructions

Create a directory with the following structure:

- `hw6-your-name/hw6-your-name.Rmd`
- `hw6-your-name/hw6-your-name.pdf`

where `hw6-your-name.Rmd` is an R Markdown file that compiles to create `hw6-your-name.pdf`.

Do not include data in the directory. Compress the directory as `.zip`.

Your solution should include all of the code necessary to answer the problems. All of your code should run (assuming the data is available). All plots should be generated using `ggplot2`. Missing values and overplotting should be handled appropriately. Axes should be labeled clearly and accurately.

To submit your solution, create a new private post of type "Note" on Piazza, select "Individual Student(s) / Instructor(s)" and type "Instructors", select the folder "hw6", go to Insert->Insert file in the Rich Text Editor, upload your `.zip` homework solution. Title your note "[hw6 solutions] - your name" and post the private note to Piazza. **Be sure to post it only to instructors**

Part A

Problem 1

Write a function that performs cross-validation for a linear model (fit using `lm`) and returns the average root-mean-square-error across all folds. The function should take as arguments (1) a formula used to fit the model, (2) a dataset, and (3) the number of folds to use for cross-validation. The function should partition the dataset, fit a model on each training partition, make predictions on each test partition, and return the average root-mean-square-error (RMSE).

(Do NOT use a pre-existing function such as `caret::train()` that already performs cross-validation.)

Problem 2

Using 5-fold cross-validation, report the cross-validated RMSE of the model you fit in Homework 5, Problem 5, for predicting `crim` from the `BostonHousing` dataset in the `mlbench` package. Can you find a model with a better cross-valided RMSE? (You do not have to try all possible models.)

Part B

Problems 3–5 use the text of 56 major speeches by Donald Trump from June 2015 through November 2016. Download the data from https://github.com/PedramNavid/trump_speeches. (Use either the "full_speech.txt" file or the "speech_##.txt" files but not both, as the former contains all of the text of the latter. If you use

the "speech_##.txt" files, you should skip the first line of each file.) Use the `read_lines()` function from the `readr` package to import the data into R.

Problem 3

Import the text from all 56 Donald Trump speeches into R. Tokenize the data into a tidy text data frame, using *bigrams* as tokens. Filter the data, removing bigrams where either word is a stop word or the word "applause", and removing bigrams where the first word is a negation word such as "never", "no", "not", or "without". Then plot the top 15 most common bigrams in Trump's speeches.

Problem 4

We would like to see the most commonly negated words in Donald Trump's speeches, and how they're negated. Filter the bigrams, keeping only bigrams where the first word is any of "not", "no", "never", or "without", and removing those where the second word is a stop word or "applause". Then visualize the most common (top ~5) words preceded (separately) by each of "never", "no", "not", and "without".

Problem 5

We would like to do a sentiment analysis of Donald Trump's speeches. In order to make sure sentiments are assigned to appropriate contexts, first tokenize the speeches into bigrams, and filter out all bigrams where the first word is any of the words "not", "no", "never", or "without".

Now consider only the second word of each bigram. Filter out all bigrams where the second word is a stop word or "applause". Then visualize the most common words (top ~5) in Trump's speeches that are associated with each of the 6 sentiments in the "loughran" lexicon.