Classification of Articles with document corpus and neural networks

**Approach**

For execution instructions jump directly to next section

Used R for the purpose of this challenge.

1. Extracted plain text from html files in positive and negative folders of training set.
2. Stored plain text as text documents in separate folders
3. Built separate document corpuses for positive and negative folders
4. Merged the corpuses for preprocessing and cleaning
5. Constructed document term and term document matrices
6. Removed the vast number of terms by filtering only documents which occur in greater than 10% and less than 60% of the documents. In the end, terms came down to 347.
7. Converted the corpus into a data frame and marked documents with corresponding labels positive and negative
8. Divided the data frame into training and test sets with 70:30 ratio for evaluation purposes. Model has no prior knowledge of the labels of test data
9. Constructed neural network model which predicts class variable positive or negative based on the term matrix which is of size 347 for the given set of documents
10. Executed the model to predict the data for test data.
11. Constructed confusion matrix and computed precision, recall and accuracy of the model
12. Note: I had two alternate models for this classification, the other approach is based on regression trees. It is included in the code, however its performance is poor compared to neural net solution

**Execution Instructions:**

1. System requirements: Operations are expensive as documents > 10000. Recommend running this script on system with at least 8 GB RAM and 4 core processor (I used 16 gig and 8 cores yet it was sluggish at times!)

2. Install R-studio from <https://cran.r-project.org/bin/windows/base/> here

3. Unzip 20nclassification.zip and copy files to any local folder.

4. Open ArticleClassification.R with any text editor and search the line (line 31) ***rootDirectory=* "C:/training”** , replace it with your custom folder path which contains ***positive*** and ***negative*** folders. Your custom folder should contain *only* these two folders. All files should have no file extension just like the files provided for this challenge, files with other formats like .txt or .html are not recognized by the script.

5. Copy the file, FileCleaner.ps1 to your positive and negative folders. Now in each of these folders, right click the file and select option ‘*run with powershell’.* This removes special or stray characters in your file names. R supports file names with a limited set of special characters, this script will remove all characters except dot and underscore

5. Open windows command prompt as administrator and navigate to C:\Program Files\R\R-3.2.1\bin. Use command ***pushd******C:\Program Files\R\R-3.2.1\bin for this purpose*** in our command window

6. Now run the command***Rscript ‘C:\users\ArticleClassification.R’*** replace this path with your custom folder path for the R file. This is an expensive operation can typically take up to 45 minutes to train and run the model on >10000 documents. You can test it with a much smaller subset to evaluate the code.

**Results:**

1. From the random test subset of documents, documents which are correctly classified as positive are summarized in **diseaseresults.csv** in your root folder.

2. Metrics like precision, recall and accuracy are written to **modelresults.txt** file in root folder

3. Owing to huge sample set for training the performance of neural net has been pretty high with a precision of .97