Assignment 5: Classification

This assignment needs news_train.csv and news_test.csv. new_training.csv is for training and news_test.csv is for test. Both of them have samples in the following format:

label	**text**
Business	Fears for T N pension after talks. Unions repr...
Sci/Tech	The Race is On: Second Private Team Sets Launc...
Sci/Tech	The Race is On: Second Private Team Sets Launc...
Sci/Tech	Company Wins Grant to Study Peptides (AP)....

Q1 Classification

Write a function **classify** to conduct a classification experiements as follows:

1. Take the training and testing file name strings as inputs, e.g. classify(training_file, testing_file). I
2. Classify text samples in training file using **linear support vector machine** as follows:
    - a. First apply grid search with **6-fold cross validation** to find the best values for parameters **min_df**, **stop_words**, and **C (penality parameter of SVM)** that are used the modeling pipeline. Use f1-macro as the scoring metric to select the best parameter values. Potential values for these parameters are:
        - min_df' : [1,2,5]
        - stop_words' : [None,"english"]
        - C: [0.5,1,5]
    - b. Using the best parameter values, train a linear support vector machine classifier with all samples in news_train.csv
3. Test the linear support vector classifier created in Step 2.b using the testing file. Compare f1-macro score you obtain from the test dataset with the f1-macro of the best model from grid search, and comment if the model is overfitted or not. Save your comment into a pdf file
4. Your function "classify" t has no return. However, when this function is called, the best parameter values from grid search is printed and the testing precision, recall, and f1 score from Step 3 is printed.

Q2. How many samples are enough? Show the impact of sample size on classifier performance

Write a function "impact_of_sample_size" as follows:

- Take the full file name path strings for training and test datasets as inputs, e.g. impact_of_sample_size(train_file, test_file).
- Starting with 300 samples from the training file, **in each round you build a classifier with 300 more samples**. i.e. in round 1, you use samples from 0:300, and in round 2, you use samples from 0:600, …, until you use all samples.
- In each round, do the following:
  1. create tf-idf matrix using TfidfVectorizer with **stop words removed**
  2. train a classifier using **multinomial Naive Bayes** model
  3. train a classifier using **linear support vector machine** model
  4. for each classifier, test its performance using the testing file and collect the following metrics: macro precision, macro recall. Note, make sure you use the same model parameters for all iterations.
- Draw a line chart (two lines, one for each classifier) show **the relationship between sample size and precision**. Similarly, plot another line chart to show **the relationship between sample size and recall**
- Write your analysis on the following:
  - How sample size affects each classifier's performance?
  - How many samples do you think would be needed for each model for good performance?
  - How is performance of SVM classifier compared with Naïve Bayes classifier, as the sample size increases?
- There is no return for this function, but the charts should be plotted.

Q3 (Bonus). Sentiment Classification

You'll need amazon_review_500.csv for this assignment. This csv file has two columns as follows. The label column provides polarity sentiment, either positive or negative

label	text
2	I must admit that I'm addicted to "Version 2.0...
1	I think it's such a shame that an enormous tal...
2	The Sunsout No Room at The Inn Puzzle has oddl...
...	...

Write a function **detect_sentiment()** as follows:

- Take the filename string as an input
- Create a Multinomial Naive Bayes classifier with 5-fold cross validation as a **benchmarking model**. The average testing f1 macro is about 70%.
- Try your best to **improve average testing f1 macro by 5%** out of 5-fold cross validation. You can use any approach. Some possible directions:
  - Use different classification models, e.g. SVM
  - Tune model parameters
  - Add additional features, e.g. word POS or sentiment, phrases, negation words etc.
  - Combine more than one models (i.e. ensemble).
  - ...
- This function has no return. Print out benchmarking performance and the improved performance when it's called.
- Write a paragraph to describe your approach and explain why your approach would work.

Note, This question has no standard answer. Your objective is to improve the average testing f1 macro of 5-fold cross valuation by 5% by whateve means!

If you use additional resources, e.g. sentiment lexicon file, you need to submit these additional resource files to canvas and notify our TA.

In [ ]: