Homework 1: Stance Classification (100 points)

Kathleen McKeown, Fall 2019
COMS W4705: Natural Language Processing

Due 9/25/19 at 11:59pm

Please post all clarification questions about this homework on the class Piazza under the "hw1" folder. You may post your question privately to the instructors if you wish. If your question includes code or a partial solution, please post it privately to the instructors ONLY.

This is an individual assignment. Although you may discuss the questions with other students, you should not discuss the answers with other students. All code and written answers must be entirely your own.

Late policy: You may use late days on this assignment. However, you **MUST** include at the top of your submission how many late days you are using. If you don't tell us or have used all your late days, 10% per late day will be deducted from the homework grade. You may choose to save your late days for harder assignments later in the class.

1 General Instructions

The main goal of Homework 1 is to build a classifier that will predict whether a person is "pro" or "con" a given topic, depending on a piece of text. Your job is to see how accurate a classifier you can build, depending on the machine learning approach and on the various features you will be asked to implement.

The data you will be given is a set of posts from the online debate forum "CreateDebate". The posts are taken from debates on the following 4 controversial topics: abortion, gay rights, Obama, and marijuana. **You will use only the topics 'abortion' and 'gay rights' for this assignment**. You will be given a set of posts, each with a topic and a label of "pro" or "con". This data was collected and labeled by [1].
Warning: *this data was collected from public forums and is unfiltered. Therefore, in some cases the text may be disturbing or you may not agree with it.*

The data file provided to you is in csv format and are formatted with one post per line. We provide, along with the post and information about it, 6 LIWC[1] features, and counts of 3 part-

[1] LIWC (Linguistic Inquiry and Word Count) features capture a variety of psychometric and linguistic properties. See [2] for more details.

of-speech tags for the sentence (see section 1.1.3 for details). The dataset is available on the website. You will use cross-validation to evaluate your models.

1.1 Programming portion

You must write all code yourself. Do not use publicly available code. Please refer to the Academic Integrity policy if you have questions in this regard.
You should use 5-fold cross-validation to perform model selection, choosing the best features and model type for each topic. You should report the average accuracy and F1-measure from cross-validation.
You should then report the top 20 features for each topic.
Your submission must include a **README** file as specified in Section 2.2 below. See details on deliverables in Section 2.1.

1.1.1 Data Format

The data is provided in csv format, you may want to use pandas to read the data file. The file has the following columns:

- post_text: the actual post, tokenized.
- topic: the topic of the post. The topics are: "abortion", "gay rights", "obama", and "marijuana". You only need to worry about the data for the topics **"abortion" and "gay rights"**, but the data file will have data for ALL topics.
- label: the stance label the post takes on the topic. This will either be "pro" or "con".
- author: the author of the post.
- id: an id number for the post.

The remaining columns contain features that we provide for you. They are specified in section 1.1.3.

1.1.2 Models

You will use scikit-learn to do the homework. See the scikit-learn documentation for many useful functions available. You are to experiment with SVMs and Naive Bayes.

Input: features you have selected from a post and its corresponding topic.
Output: a label $y \in \{0, 1\}$, indicating the stance of the post on the topic: "pro" (1) or "con" (0).

In this assignment you will build, **for each** of your 2 topics, **2** models using features as specified below (total of **4** models):

1. Ngrams: this model should use a combination of unigrams, bigrams and trigrams as features. The combination you use is up to you and you should experiment to find the one that performs best.
2. Other features: this model can use any combination of features (see section 1.1.2). You must experiment with at least **3** feature types other than ngrams for this model, though you do not need to include all of them in the final model. See **Grading** for more details.

For each feature set above, you should select the model (SVM or Naive Bayes) that gives the **highest accuracy**. If this model has tunable parameters, you should tune them to perform well without overfitting. You may tune parameters you find in the scikit-learn documentation. Some examples are "kernel" and "C" for SVM.

1.1.3 Feature Selection

As part of this assignment you will experiment with a variety of features to find the best performing features for your models. For the second of the two models you make for each topic, you can use any of the features listed below or use additional features you come up with yourself.

Feature types
NOTE: each of these counts as ONE feature type in the second model setting described in 1.1.2.

- Ngrams. Unigrams, bigrams, and trigrams.
- Cue words. The initial unigram, bigram, and trigram in a post.
- Repeated punctuation. Features to capture punctuation such as ??, ??????, !!!, or ?!.
- Part-of-speech tags. Features to count the number of nouns, verbs and adjectives in the text (count_noun, count_verb, and count_adj).
- LIWC features. We give you a number of LIWC features in the data files. You may use any combination of these as features.The measures we give and their descriptions are:
  - word_count: the number of words in the post.
  - words_pronom: the percent of words in the text that are pronominal.
  - words_per_sen: the average number of words per sentence in the text.
  - words_over_6: the percent of words in the text that are over 6 letters long.
  - pos_emo: the number of positive emotion words in the text.
  - neg_emo: the number of negative emotion words in the text.
- Any other features you come up with.

1.2 Written portion

1. For each topic, provide a written description of your best model and a justification of the features that you used or did not use. Why do you think they are helpful or not? You must explicitly list the features you tried here.
2. For each topic, do an error analysis. For your best performing model, select **3** interesting examples that did not work. Indicate why you think they were classified incorrectly. If you could add additional features, what features do you think would help improve performance? For each example, you do not need to include the entire post text if it is long, only the relevant portions to your discussion and the ID.
3. Compare and contrast the features you used for each topic. Why do you think some features worked better or worse for different topics? Or did you use the same features for all topics and if so, why do you think that worked well?

2 Grading

You will be graded on the following elements:

2.1 Classifier (60 points total)

Your deliverables are the following (50 points):

- A **classify.py** file that takes the train file and topic as command line arguments (i.e., can be run as 'python classify.py train-data.csv abortion'), runs cross-validation to do model selection, prints the model's average accuracy and F1-measure and prints the top 20 features for that topic. (60 points).
- Your **README**, described below (2.3)

Accuracy (10 points)
Your system will receive up to 10 points depending on whether the best models produced accuracy above or below our thresholds.
These will be posted on Piazza and the course website by Friday Sept. 13.

2.2 Written Answers (30 points)

Justification of best performing model for each topic: the descriptions are complete and provide information about why each model (and the features it uses or doesn't use) was the best. (20 points)
Error analysis: Selected posts should be different and the explanation should clearly explain why the features failed to predict the correct class. (10 points)

Your deliverables are the following:

- A **hw1-written.pdf** file containing your name, email address, the homework number, and your answers for the written portion.

2.3 Software Engineering (includes documentation) (10 points)

Your README file must include the following:

- Your name and email address.
- Homework number.
- Information on how to train and test your classifier.
- A description of the special features (or limitations) of your classifier.

Within Code Documentation:

- Code should be documented in a meaningful way. This can mean expressive function/variable names as well as commenting.
- Informative method/function/variable names.
- Efficient implementation.

3 Submission instructions

You should submit the following on Courseworks:

- A zip file named <**YOUR-UNI**>**-hw1.zip**. This should have exactly the files listed in the deliverables in section 2.1. You do not need to include the data files in the zip.

You should submit the following on Gradescope:

- The **hw1-written.pdf** as described in section 2.2.

NOTES:

1. We **WILL NOT** grade code that does not run in Python 3.6.9, INCLUDING because of Python 2 print errors.
2. Handwritten solutions **WILL NOT** be graded. This includes pictures of handwritten answers inside the pdf. If you have concerns about typesetting, please talk to the TAs or post on Piazza.

4 Academic integrity

Copying or paraphrasing someone's work (code included), or permitting your own work to be copied or paraphrased, even if only in part, is not allowed, and will result in an automatic grade of 0 for the entire assignment or exam in which the copying or paraphrasing was done. Your grade should reflect your own work. If you believe you are going to have trouble completing an assignment, please talk to the instructor or TA in advance of the due date.

References

[1] Kazi Saidul Hasan and Vincent Ng. 2013. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. *IJCNLP*.

[2] James W. Pennebaker, Boyd, R.L., Jordan, K., and Blackburn, K. The Development and Psychometric Properties of LIWC2015. `http://liwc.wpengine.com/wp-content/uploads/2015/11/LIWC2015_LanguageManual.pdf`