PR1: Text Classification

Description:

****
This is an individual assignment.
****

Overview and Assignment Goals:

The objectives of this assignment are the following:

- Choose appropriate techniques for modeling text.
- Implement a **k-nearest neighbor classifier** (cannot use libraries for the implementation – see details below).
- Use your version of the k-NN classifier to assign classes to short texts.
- Evaluate results using the F1 Scoring Metric (can use libraries for this part).

Detailed Description:
Develop predictive models that can determine, given short news item descriptions, which of 4 classes they fall in.

In the given dataset, abstracts from 4 different types of stories have been included. The goal of this competition is to allow you to develop predictive models that can determine, given a particular abstract, **which one of 4 classes it belongs to.** As such, the goal would be to develop the best classification model, with the restriction that you can only use your own implementation of the k-NN classifier. Given your implementation, there are many choices in text pre-processing and modeling, proximity measures to use, and classifier meta-parameters that will lead to many different solutions for the problem. Additionally, given the k retrieved neighbors, you must still decide on the way you aggregate their labels to make the final decision (e.g., majority count or weighted sum). Your goal is to find the best combination of pre-processing and meta-parameters.

As we have learned in class, there are many ways to represent text as sparse vectors. Feel free to use any of the code in activities and labs or write your own for the text processing step. Experiment with feature types, stemming, lemmatization, feature selection, etc. You may use external libraries for pre-processing, but not for the k-NN algorithm. You will loose 50% of the points if you use an external library there.

The performance scoring function we will use for this assignment is **F1-score**.

Caveats:

+ Use the data mining knowledge you have gained until now, wisely, to optimize your results. Actually look at your data and training/validation errors and figure out ways to deal with issues. Don’t cheat. You won’t learn anything if you do.

Data Description:

The training dataset consists of 102080 records and the test dataset consists of 25520 records. We provide you with the training class labels and the test labels are held out. The data are provided as text in train.dat and test.dat, which should be processed appropriately.

train.dat: Training set (class label, followed by a space and the text of the news story abstract, one sample per line).
test.dat: Testing set (text of news story abstracts in lines, no class label provided).
format.dat: A sample submission with 25520 entries randomly chosen to be 1 to 4.

Rules:

- This is an individual assignment. Discussion of broad level strategies are allowed but any copying of prediction files and source codes will result in an honor code violation.
- Some of your classmates may choose not to see the leaderboard status prior to the submission deadline. Please do not share leaderboard status information with others.
- The public leaderboard shows results for 50% of randomly chosen test instances only. This is a standard practice in data mining challenges to avoid gaming of the system. The private leaderboard will be released after the submission deadline, based on all the entries in the test set.
- In a given day (00:00:00 to 23:59:59), you are allowed to submit a prediction file only 5 times.
- The final ranking will always be based on the last submission or the submission you choose, not your best submission. Carefully decide what your chosen submission should be.
- Each time you submit a prediction file, you will also need to include the code that generated that prediction. Acceptable formats are: py, ipynb. *Your submission will not be valid unless it produces the output in the prediction file.*

Deliverables:

- Valid submissions to the Leader Board website: https://clp.engr.scu.edu (username is your SCU username and your password is your SCU password).
- **Canvas submission of report:**
  - Write a 2-page, single-spaced report describing details regarding the steps you followed for text processing and classifier model development. The report should be in PDF format and the file should be called **<SCU_ID>.pdf**. Be sure to include the following in the report:
  - Name and SCU ID.
  - Rank & F1-score for your submission (at the time of writing the report). If you chose not to see the leaderboard, state so.
  - Your approach.
  - Your methodology of choosing the approach and associated parameters.
  - Any special instructions for running your code.

Grading:

Grading for the Assignment will be split on your implementation (70%), report (30%). Extra credit (1% of final grade) will be awarded to the top-3 performing algorithms. Note that extra credit throughout the semester will be tallied outside of Canvas and will be added to the final grade at the end of the semester.

Files: In Canvas, you can find

- *Training Data:* train.dat
- *Test Data:* test.dat
- *Format File:* format.dat