CSE 398-498 BIG DATA ANALYTICS

Fall 2021 • 2:05 pm – 3:20 pm MW • Mountaintop Building C 115 and Hybrid

Professor Daniel Lopresti (dal9@lehigh.edu)

Homework #7

The topic of this homework assignment is sentiment analysis of Twitter text data using Weka Knowledge Flow. You should follow the instructions I have provided here carefully otherwise you will not receive full credit for the assignment.

While you are in the process of debugging your Knowledge Flow pipeline, keep in mind that you can usually output the dataSet or text from a module to a TextViewer Visualizaion module to see the intermediate data that is produced at each stage in the pipeline.

Part 1

1. Download the "text_emotion" dataset which is in CSV format to your own computer from CourseSite. Examine the dataset using a spreadsheet program so that you can understand what it contains, and how it is formatted.
2. Follow the general procedure of Reference #1 below to using Weka Knowledge Flow, with the following important notes which update some of those instructions.
3. You must configure CSVLoader to specify which attributes are nominal, and which are strings.
4. Ignore the suggestion from Reference #1 about reducing the size of the dasetset – that should not be necessary. It is also unnecessary to save the CSV data to a new file.
5. You should follow the CSVLoader module with a Remove module to eliminate the "tweet_id" and "author" attributes because we should not use these for sentiment classification. After you have done this, there should only be two attributes remaining: "sentiment" and "content".
6. Take the dataSet output from Remove and use it as input to the ClassAssigner module. Specify the first attribute (sentiment) as the classIndex.
7. Take the dataset output from ClassAssigner and input it to CrossValidationFoldMaker. This has as its default 10-fold cross-validation. Keep the default for all of your experiments in this part.
8. Take the trainingSet and testSet output from CrossValidationFoldMaker and input it to the NaiveBayesMultinomialText classifier. Note that this is different from the classifier used in Reference #1 below; I have found that one does not work. You should, however, use the same configuration settings as below for lowercaseTokens, the stemmer, stopwordsHandler, and tokenizer.
9. Take the batchClassifier output from NaiveBayesMultinomialText and input it to a ClassifierPerformanceEvaluation module.
10. Take the text output from the ClassifierPerformanceEvaluation module and input it to a TextViewer module.
11. Run the entire pipeline and report the results you obtain.

12. Provide a screen snapshot of your pipeline as drawn in Weka Knowledge Flow when you submit your writeup.
13. I strongly suggest that you "Save" your Weka Knowledge Flow pipeline so you can easily reload it later without having to redraw it again.

After you have reported and discussed the basic results, including the accuracy, examine the confusion matrix. Where are some of the most common confusions; be specific? Do they make sense to you?

How does the ZeroR classifier perform for comparison purposes?

Part 2

1. There are a total of 13 different sentiment tags represented in the original dataset. Create your own small dataset consisting of 13 different Twitter "tweets" that you have written (they do not have to be real), one for each sentiment tag. This will form the test set for the second part of this assignment. Be sure to include this in your writeup.
2. Draw a new Weka Knowledge Flow pipeline that makes use of the entire original file ("text_emotion") as the training set for the NaiveBayesMultinomialText classifier, and your own small tweet file as the test set. For this process you will want to use the Training SetMaker and the Test SetMaker modules in Weka. You will not use the CrossValidationFoldMaker module.
3. Run the entire pipeline and report the results you obtain. Analyze the errors that arise in particular, and explain why you think they may be happening.
4. Provide a screen snapshot of your pipeline as drawn in Weka Knowledge Flow when you submit your writeup.
5. As for Part 1, I strongly suggest that you "Save" your Weka Knowledge Flow pipeline so you can easily reload it later without having to redraw it again.

Note 1: even though you are making up the test dataset and have complete control over it, you should not expect 100% accuracy of this classifier. You might want to play around a bit with your simulated tweets to see if you can get better performance, but do not bother to waste too much time on this. On the other hand, if you get extremely low accuracy then you are doing something wrong.

Note 2: it is important that your test dataset file, in CSV format, be formatted in exactly the same way as the original input file, otherwise you will get cryptic error messages from Weka. All CSV files are not created equal; there can be some differences. It might be necessary for you to view and edit the file using a simple text editor (like WordPad on Windows, or TextEdit on a Mac) to get it exactly right, since MS Excel will mask some of the low-level details of the file format.

Below are some references that will be useful for you to review, but note that there are important caveats in each case as they do not exactly match our assignment. If you follow the instructions below and not my instructions above, you will likely get stuck and not do well.

Reference #1: "NLP using WEKA"

https://medium.com/analytics-vidhya/nlp-sentiment-analysis-using-weka-hands-on-648af0859797

This article provides a good overview of most of the basic steps we will be using in this assignment. It also provides the pointer to the dataset I want you to use. I have also placed a copy of this data on the CSE Department Sunlab network.

Here is some more information about the data:

https://data.world/crowdflower/sentiment-analysis-in-text

"In a variation on the popular task of sentiment analysis, this dataset contains labels for the emotional content (such as happiness, sadness, and anger) of texts. Hundreds to thousands of examples across 13 labels. A subset of this data is used in an experiment we uploaded to Microsoft's Cortana Intelligence Gallery."

However, there are some important differences you must pay attention to. The biggest difference I have found is that the Naïve Bayes classifier as described in the article performs very badly. By now you should be able to recognize this. Do not turn in an assignment that uses this approach; follow my instructions above.

Reference #2: "Sentiment Analysis of Tweets using Multinomial Naive Bayes"

https://towardsdatascience.com/sentiment-analysis-of-tweets-using-multinomial-naive-bayes-1009ed24276b

This article provides a brief overview of the Multinomial Naive Bayes technique you will be using for the assignment. This will give you some useful background. Note, however, that it programs the solution using Python using Jupyter Notebook and it uses a completely different dataset that is labeled only in terms of "positive" and "negative" sentiment. You must use Weka Knowledge Flow and the dataset I have provided for this assignment.

Reference #3: "Performing Sentiment Analysis on Movie Reviews"

https://towardsdatascience.com/imdb-reviews-or-8143fe57c825

This is yet another blog post about performing sentiment analysis on movie reviews, and uses yet another classification technique: logistic regression.

Reference #4" "50 free Machine Learning datasets: Sentiment Analysis"

https://blog.cambridgespark.com/50-free-machine-learning-datasets-sentiment-analysis-b9388f79c124

This blog post provides pointers to a number of different datasets that are appropriate for sentiment analysis. It also gives a link to a video presentation by a professor from the UK about analyzing social media discussions surrounding Brexit which you might find interesting.