Assignment II – CS 8803 Data Analytics for Well-Being

Topic	Predictive Models of Well-Being
Grade	Max 60 points; 10% of overall grade (late policy applies)
Due	April 11, 2016, 2:05pm Eastern Time
What to hand in	Code and a report with answers to the different questions
Where to submit	T-Square
Useful resource	Python's nltk library (http://www.nltk.org/) Python's statsmodels library (http://statsmodels.sourceforge.net/) Python's scikit-learn library (http://scikit-learn.org/stable/) [You can also use your favorite other library, tool or software]

Tasks

This assignment tests your understanding and technical ability of predictive modeling, focusing on classifying emotional states in Twitter posts. Refer to the two enclosed files in this assignment: one each for two emotional states: "happy" and "sad". In each of the files (e.g., pos_examples_happy.txt), each line refers to a Twitter post containing an emotional state hashtag (e.g., #happy). Assume that this emotional state hashtag associated with a post is its (noisy) "ground truth" label, that is, the author of the post indicated their emotional state at the time of writing the post with this hashtag (similar to [1]). Based on this assumption, answer the following questions:

1) (**Total 20 points**) Extract different features from the posts in the two files, corresponding to the two classes, #happy and #sad. Note, since the emotional state hashtags #happy and #sad are used as ground truth, you cannot use these hashtags in ANY feature computation.
   a) (**8 points**) Using the LIWC categories from Assignment I (*positive affect, negative affect, anger, anxiety, sadness* and *swear*; files enclosed in this assignment as well), obtain the proportional counts of words in a post matching words in each category. (Like assignment I, the proportional count of a LIWC category in a post is defined as the number of LIWC words or stems that are present in the post, divided by the total number of whitespaced tokens in the post[1]). The feature vector of a post will thus be a vector of length six (there are six LIWC categories). Enclose tsv files of the feature vectors of each post in the #happy and #sad files (there will be two tsv files, one for #happy and the other for #sad posts). Each row/line in each tsv file should be a post and its LIWC values as tab-separated columns/fields.
   b) (**12 points**) Obtain *n*-gram (*n*=1, 2, and 3) tokens over all #happy and #sad posts[2]. Consider as features, those *n*-grams which appear more frequently in the entire dataset (#happy and #sad posts combined) than three different chosen thresholds *d*, say, 50, 100, or 500 times (you can choose your own preferred thresholds too). For a certain threshold *d*, the feature vector of a post will thus be a vector of the frequency of each *n*-gram in it, which occurs in the dataset *d* or more times. Enclose tsv files of the feature vectors of each post in the #happy and #sad files, based on each threshold *d* (there will six tsv files, three each for #happy and #sad posts; the three files for each class would correspond to the three values of *d*). Each row/line in each tsv file should be a post and its *n*-gram frequencies as tab-separated columns/fields.

2) (**Total 25 points**) Using each of the above feature representations, build different binary classifiers to distinguish between #happy posts and #sad posts.

[1] Ignore tokens that are "RT", Twitter usernames, or urls.
[2] Eliminate stopwords.

a) (**15 points**) Build the following classifiers: i) *k* Nearest Neighbor (*k*NN), ii) Naïve Bayes, and iii) Support Vector Machine (SVM). For *k*NN, build different classifiers with the following values of *k*: 1, 10, 100, 1000. Thus you will have six classifiers. Enclose your code for all of the classifiers.
b) (**10 points**) Evaluate the performance of all of the classifiers (*k*NN corresponding to the four values of *k*, Naïve Bayes and SVM – six classifiers in all) and all of the feature representations (LIWC, these sets of *n*-gram features, per the three chosen *n*-gram frequency thresholds *d* – four in all) based on the metrics, accuracy, precision, recall and F1 score[3]. Enclose a table for each of these metrics, where the columns are the various classifiers (six classifiers), the rows are the different feature representations (four feature types), and the values in the cells of the table are the corresponding accuracy, precision, recall, or F1 values for a classifier-feature pair.

3) (**Total 15 points**) Present a discussion of the performance of the different classifiers.
a) (7 points) Obtain the accuracy of a baseline "chance" model, which would be labeling everything to be either #happy or #sad based on the class with larger number of posts. For instance, if both classes are equal in size, then the accuracy of this chance model is 50%; if 60% of your entire dataset are #happy posts (hypothetically), then the chance model accuracy is 60%. How do your classification models fare compared to the chance model of this dataset?
b) (**4 points**) Why do you think some classifiers perform better than the others?
c) (**4 points**) How do the different feature choices impact classifier performance?

References

[1] De Choudhury, M., Gamon, M., & Counts, S. (2012). Happy, Nervous or Surprised? Classification of Human Affective States in Social Media. In ICWSM.
[2] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.
[3] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1, 12.

[3] Accuracy is the percentage of correct predictions made by your classification algorithm. Precision is defined as the number of true positives over the number of true positives plus the number of false positives. Recall is defined as the number of true positives over the number of true positives plus the number of false negatives. F1 score is the harmonic mean of precision and recall: 2 x ((precision x recall) / (precision + recall))