STAT 154
FINAL PROJECT
BOOK MINING

This is the final project for the class. Form teams of 4 and let the GSI and me know who your team members are by 11/11(Wed). The team will receive the same score for the project. Total points: 100 (30% of the course grade). Keep all descriptions and summaries complete but succinct and precise. Provide tables and charts whenever appropriate. Section competition on Friday12/4. **Final write up due by 12/7 (2 pm or earlier). No late project write-ups will be accepted.**

Data: The data will be available to you on BOX. Note these are the actual text data (not a data matrix) from "Project Gutenberg", including four categories: Science, Child, Religion, History. Each observation is part of the content of a book. You should not try to collect books data by yourself, since the training and testing data are preprocessed.

1. *Description:* Write one paragraph describing the data. (Hint: How many observations are there? Distribution between categories. Look at a couple of the observations in each categories. Is it easy for human to classify these books from different categories?)

2. Feature Creation: Define a feature as a unique word in the text. That is a continuous string of alphabet characters without white space.

    a. Use Python to parse out the word features from the texts.

    b. Exclude the common word features (known as stop words) listed in http://www.textfixer.com/resources/common-english-words.txt

    c. Replace the non-alphabet characters (",", "\n", etc) in the content with a space character. Convert uppercase to lower case characters. Remove all common words of Project Gutenberg, i.e. Project, Gutenberg, eBook, Title, Author, Release, Chapter, etc. Derive a dictionary of words and total number of their appearances through out the whole dataset.

d. Derive a word feature matrix where rows= # of observations and cols=# of word features, select 1000~2000 words, through out the whole dataset. Stopwords and rare words (e.g. words that does not appear more than 10 or 20 times) should be excluded here. At the same time create the target vector of having tags “Science”, “Child”, “Religion”, “History”. Report the dimensions of your feature matrix.

Describe the process you undertook to derive the word feature matrix. How many features did you end up with? Did you encounter programming challenges?

3. Unsupervised Feature Filtering: Make a histogram of the # of times word features appear in a text. (Hint: For each column in your data matrix – count the number of times that feature is non-zero). You can filter words that are too rare and too common this way. What minimum and maximum threshold will you use? Apply the filters. How many features do you come up with? Give dimensions. Store this as your word feature matrix.

4. Power Feature Extraction: Derive Power features to be used for classification in addition to the Word features. You should create anywhere between 1 and 25 power features. Power features are aspects of data that help discriminate between categories – and not included in the word matrix. Describe the features and code them. Create a matrix of power features with rows= # of observations and columns = # of power features. Store this as your power feature matrix.

5. Word and Power Feature Combination: Create a combined feature matrix. Describe the dimensions of the data. Store this as your combined feature matrix.

6. Classification on the filtered Word Feature Matrix: Use the two classification algorithms (SVM and Random Forest) to classify the data; note, this is a multi-class problem**. Use V-fold cross validation with V=10 to produce the predictions and to tune your model parameters.** Compute the accuracies in each loop. Report the overall accuracy rates and accuracy rates per class. (Note: it might be useful to produce ROC curves for your classification (you should have at least one curve for

word, one curve for power, and one curve for combined) for pairwise comparison between classes). Report the dimension of your feature matrix.

7. VERIFICATION: Submit the word feature matrix to your BOX directory by 12/3(Thu) midnight. This will serve as verification that you have a final output at the end of step 3. Save your final Random Forest model in BOX as well – ready for prediction on an unseen test set.

   In-class competition: *In section on Friday 12/04, you will be given a competition test data set of books with no tags. You will need to produce a word feature matrix from this. Next you will use Random Forest classifier saved above on the training set (word feature matrix) and predict on the competition test set (word matrix you just produced). Output the predicted class labels and submit to GSI and upload to BOX before leaving. Total time for processing raw data, and predicting class label for the competition test set should take no more than 5 minutes per team. The accuracy on this test set will be graded. Note: You will need a program to deal with raw text files, create the data matrix and return results (prediction labels) in a CSV file (one row per book). You will upload this program in BOX ahead of the competition (We'll give you a small practice data set prior to the competition).*

8. Repeat Step 6 on the power feature matrix. Write and describe your output. Compare the results (**overall accuracy % and accuracy % per class** (or tag) either from cross-validation or taking a small validation set from training set) with results obtained from using only the word feature matrix.

9. Repeat step 8 on the combined feature matrix. Write and describe your output. Compare these results to the results from steps 6 & 8. Did your results improve? Provide graphs and charts.

10. *Validation set:* Submit the prediction of your best model (SVM, or Random Forest, or any model that you have learned in this class) built with best feature set to Kaggle, where there will be a large test set for you. The Kaggle competition will be open during the week of 11/30, and will end at the same time as the deadline of project. Continue to refine your model and feature set to improve your Kaggle performance. **Write down your accuracy (overall and per class) from the Kaggle competition leaderboard.** Comment on your final results, as well as your learning process. Number of times you can submit your prediction is limited. Report your final or highest score in your write up.