Unclassified Released Emails of HRC

STAT154 Final Project

Due 12/02/16

Submission Instructions: Each team must create a private github account. Using your .edu email you can sign up for a private account. Add solari AT berkeley.edu as a contributor to your STAT154-GroupXX, where XX is the number of your group, repository. All your progress must be reported on this account. Each repository must have a clear readme file explaining the structure of your project. As for the final submission within your repository there must be a folder with all .R/.py files.

You must produce a single .pdf file that is compiled from your .tex file. (You do not have to use tex for your report). You can use Word or any other application to produce the report. We need your hardcopy report however. This hard copy must be submitted in person to Professor's office by **2 pm December 2nd**).

Dataset Description: In 2015, controversies arose over a leak showing that HRC used her personal email accounts on private servers while she was the Secretary of State. In response to multiple FOI lawsuits, the State Department released thousands pages of redacted HRC emails. HRC *train.csv* contains a training set of 3505 emails from top 5 contacts among the full email dataset. In this project you are tasked to create a classifier that classifies emails according to their senders, class variable, using words or phrases, features, used in each email.

1. [20 points] Feature creation and filtering: Parse the dataset into a word feature matrix. Import HRC train.csv. Now try stop-words and stemming correction on the dataset. Try removing commonly occurring words or phrases from the emails. You may benefit from python nltk library. Also if you come up with "power features", that is, features that occur for a specific sender, e.g., you can add

those to the regular word features matrix as additional columns. (Optional: Produce a word cloud plot for each sender). In your write up, include your methods / steps for feature creation, filtering and dimension of final feature matrix.

Report the following metrics in this section for steps you took in feature creation and unsupervised feature filtering:

Step	Total # of features
Raw	
Remove Stop words	
Stemming	
Adding power features	

2. [30 points] Build your supervised classifier.

   - (15 points) Train an RF classifier with the final feature matrix your created in the step above; choose your best model using 5-fold CV. Report the training accuracy rate (see table below). Make plots and describe steps you took to justify choosing optimal tuning parameters. Report your top 10 important features.

     You may do supervised feature selection here as well (optional). This may create an opportunity to reduce # of features.

Report the following metrics in this section for steps you took in training your RF classifier in 5-fold CV:

Step	Total # of features used	Total Accuracy (xx%)	Accuracy per sender class xx%,yy%,zz%,aa%,bb%
RF			
RF with feature selection (optional)			

- (15 points) Train an SVM classifier with the final feature matrix your created in the step above; choose your best model using 5-fold CV. Report training accuracy rate (see table below). Make plots to justify choosing optimal tuning parameters. Report your top 10 important features.

  You may do supervised feature selection here as well (optional). This may create an opportunity to reduce # of features.

Report the following metrics in this section for steps you took in training your SVM classifier in 5-fold CV:

Step	Total # of features used	Total Accuracy (xx%)	Accuracy per sender class xx%,yy%,zz%,aa%,bb%
SVM			
SVM with feature selection			

(optional)			

3. [10 points] Build your unsupervised classifier on the training set. Use Kmeans clustering algorithm to cluster emails using your top 100 word features from RF classifier in step 2 above. Report your clustering accuracy.

4. [40 points] Validate your best supervised classifier on the test set. This means you must lock the final classifier model, feature matrix and threshold from step 2 above. Using a test set, HRC *test.csv*, which will be provided to you later, you will predict the classes of the test set and upload the predicted labels. Upload the predicted labels in a *predict.txt* in a single column in the same order of the emails appearing in the test set.

Report the following metrics in this section for the test set:

Step	Total # of features used	Total Accuracy (xx%)	Accuracy per sender class xx%,yy%,zz%,aa%,bb%
RF or SVM			