project[1] - spam,

due 10/31/2018, midnight

Purpose: *The purpose of this assignment is to provide a basic programming experience with an algorithm commonly-used in AI. You should be able to utilize probability in order to classify using a Naive Bayes Classifier.*

Skills demonstrated:

* using a github repository to download code
* c++ programming
  + working with an established dataset
  + file i/o
  + data transformations
* probability and example data
* natural language

Please write a program to parse an existing dataset on real-world SMS messages. *(note: since these data come from real-world interactions, these messages may use language which I would never use in class and that violate professional conversational norms. If that is likely to trigger a negative reactions, please do not read the messages themselves. However, I feel that it is important to work with real-world data wherever possible).*

The dataset can be found here: <https://www.cse.unr.edu/~dave/spam.csv>

You will need to write two programs:

training -i <spam.csv file> -os <output spam probability file> -oh <output ham probability file>

Trains dataset from .csv file and save to new file. Each line of the .csv file has at least two fields, separated by a comma:

1. <spam|ham> ham if it is a legitimate SMS, spam if not

2. "..." the SMS message

You will need to output two probability files (one for ham, one for spam):

<count of total number of words (n)>

m lines, one for each word

<word> <number of word occurrences>

classify -i <testing dataset .csv file> -is <spam probability file> -ih <ham probability file> -o <classification output filename>

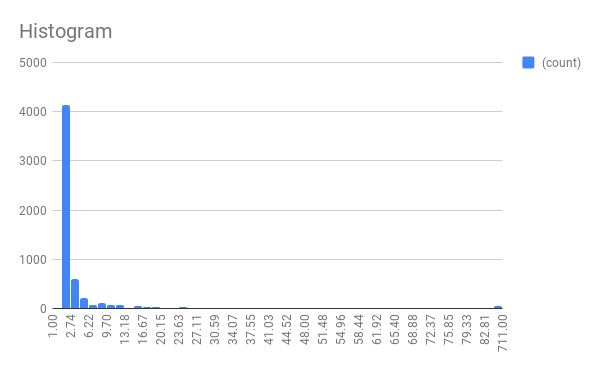
Classifies new data from training file and testing .csv file (same format as above, specified on the command-line)

You will need to output one classification file:

m lines, one for each SMS in the testing dataset (in the  
 same order as the testing set is in

<spam/ham> (the classification of the SMS)

# **!!! new note !!!:**

As you can see from the figure to the left, there are a lot of words that only appear once. This can result in a lot of bad classifications from low-occuring words (i.e., the word "0089my" probably should not affect your results as it only appears once in the SPAM training set). I wouldn't update your spam/ham probability unless it appears at least a few times in \*BOTH\* the spam and ham training sets, just to make sure you're not introducing bad data, otherwise, you're likely to get something like 25% classification accuracy. Play around with it a bit, especially how much a word should show up in the database before you use it for spam/ham probability.

**Graduate Student Extra Assignment:**

Please also write a program to add new data (in a .csv file) to the existing training database. This assignment will not be autograded.

**Code should be submitted using the submit script:**

(!!!do only once per computer!!!) first, download the submit script:

$ cd ~

$ wget http://www.cse.unr.edu/~newellz2/submit

$ chmod +x submit

then, from your project directory (1-tictactoe):

$ ~/submit

this will submit all files from the current directory

***(login: netid / password: passwd for your netid)***

**Your turned in directory should have the following files:**

.

+-- CMakeLists.txt

+-- src

+-- training.cc

+-- classify.cc (all other code files

needed to build your programs)

it is acceptable to turn in files that are not in this list, but these have to be turned in.

// must build executables named:

// training: train dataset from .csv file and save to new file (specified on command line)

// classify: classify new data from training file and testing .csv file (same format as above, specified on the command-line)