• Your name(s)

Dennis Lee

• The data set you used (Sets A, B, or C)

Set B

• The tools you used in each part of the project.

1. C++17 and Visual Studio for dev environment
2. Windows File Explorer (Step 2)
3. utf8.h to handle locale and encoding (Step 3. (a))

• Any decisions specific to your spam filter, such as: words that you might have decided to

exclude, threshold for the probability of spam (especially if different than 0.5), etc.

1. Generic programming: will be likely to work with any type/name/number of the input files
   1. Use std::filesystem to iterate through given directories instead of looking for file names from the “cmd” file
2. Words, from most to less common, excluded due to emptiness, meaninglessness, general
   1. “” (empty)
   2. "your"
   3. "for"
   4. "the"
   5. "you"
   6. "a"
   7. "re"
   8. "for"
   9. "to"
   10. "on"
   11. "of"
   12. "in"
   13. "and"
   14. "with"
   15. "is"
   16. "from"
3. Threshold for the probability of spam = 0.8
   1. The lower the threshold, the more overall predicted spam which results in losing too many hams
   2. Therefore, the threshold of 0.8 is negotiated, because getting more correct hams is usually more important than filtering out correct spams for the most users

• The 5 most "spammiest" and the 5 most "hammiest" words from the training stage: answers

(c) and (d) from Step 3.

TOP 5 SPAM WORDS :

free: 0.0769231

adv: 0.0477454

get: 0.0450928

rates: 0.0424403

home: 0.0424403

TOP 5 HAM WORDS :

new: 0.0430993

ouch: 0.0251816

bliss: 0.0217918

selling: 0.0217918

spamassassin: 0.0217918

• The accuracy, precision and recall rates from the testing stage: answers (a), (b), and (c)

from Step 4.

ACCURACY RATE: 0.171182

PRECISION RATE: 0.0537459

RECALL RATE: 0.264

• Conclusions on the performance of your spam filter and possible steps you would take to

improve it.

* The performance is not ideal, but it is likely caused by the data set
  + Consulted by Dr. Vermesi, the instructor, that the calculation of scores is correct (10/23/20)
* Naïve Bayes is naïve that it treats all word orders the same
  + Ex: Score for Dear Friend = Score for Friend Dear
  + Ignores grammar rules and common phrases
  + Therefore, if there is another program or dictionary that allows to look up common grammar rules and phrases to take score from each, the algorithm will be improved.
* More, recent, and accurate training data sets will result better filtering performance
* Studying and researching how spammers to choose words to avoid getting filtered