CSE 312: Foundations of Computing II
Programming project
January 23
due: Friday, February 2, 10:00 p.m.

Attention! A previous version of the specification asked you to turn in a readme.txt. **This is incorrect.** Please **only** turn in the .java or .py source code file. We are extremely sorry about the confusion.

Correction: The due date is Friday, February 2 (not February 1).

Instructions:

You will submit a program source file via a Google Form (details below).

Use the Naive Bayes Classifier to implement a spam filter that learns word spam probabilities from our pre-labeled training data and then predicts the label (ham or spam) of a set of emails that it hasn't seen before. **You may use Java or Python 3.** We've provided starter code in Java and Python.

Download

`https://courses.cs.washington.edu/courses/cse312/18wi/312A/NaiveBayes.zip`

and unzip it. Inside the "data" folder, the emails are separated into "train" and "test" data. Each "train" email is already labeled as either *spam* or *ham*, and they should be used to train your model and word probabilities. The "test" data is not labeled, and they are the emails whose labels you will use your classifier to predict.

The emails we are using are a subset of the Enron Corpus, which is a set of real emails from employees at an energy company. The emails have a subject line and a body, both of which are "tokenized" so that each unique word or bit of punctuation is separated by a space or newline. The provided starter code takes a filename and returns a set of all of the distinct tokens in the file.

Some requirements and advice for the implementation:

- You will likely want to iterate over all the files in a directory. `os.listdir()` and `File.listFiles()` will be useful in Python and Java, respectively.
- Your program must accept the file path to your data directory as a command line argument, for example,

```
$ java NaiveBayes /path/to/your/data/directory
$
```

  The only paths hardcoded in your program should be to subdirectories of the directory given as the command line input and should be specified relative to that command line input ("./train/ham/", "./train/spam/", "./test/"). If you are using Windows, change all occurrences of "\" to "/" in these hardcoded paths before turning in your program.

- Read about how to avoid floating point underflow in the notes.
- Do not use integer division when generating your word probabilities.
- Think about the data structures you want to use to keep track of the word counts and probabilities.
- **If you use Eclipse, remove all package statements before you turn in your source code.**

For your output, for each file in the test data, print the filename, a space, and either the word ham or spam depending on how your program classified that file. **Print to stdout, and not to a file.** See `example_output.txt` for an example of the format. Note that we may run your code on test data you haven't seen yet.

After you've classified the 500 test emails, you can compare your results with the actual labels that we hid from you, by using the output checker here. It is not expected that Naive Bayes will classify every single test email correctly, but you should probably aim to get at least 90-95% of them classified correctly, and certainly do better than random chance. We are not grading you on whether you classify 100% of the examples accurately, but rather on general program correctness.

Needless to say, you should practice what you've learned in other courses: document your program, use good variable names, keep your code clean and straightforward, etc. Include comments outlining what your program does and how. We will not spend time trying to decipher obscure, contorted code.

What to turn in: Turn in your source file at `https://goo.gl/forms/fyPmLl8jGvtE4IgU2`. To do so, you must be logged into your uw.edu account in Google. Modify naive_bayes.py, or NaiveBayes.java, if you use the starter code. If you don't, please use the same file names to make our grading scripts happy. Do not turn in the training/test data that we provide you. **Please submit only a single source code file (naive_bayes.py or NaiveBayes.java). Do not submit a README file.** Remember that your program must accept a single command-line argument (the file path to your data directory). *Be careful not to turn in our unmodified starter code instead of your solution code by mistake.*