Due Date Code: Sunday April 7 at 6AM, 2019 + Report: Friday April 12 at noon, 2019
Late Submissions 20%
Teams Students registered in COMP 472 can do the project in teams of at most 4.
Students registered in COMP 6721 can do the project in teams of at most 2.
Teams must submit only 1 copy of the project via the team leader.
Purpose In this project, you will implement a Naïve Bayes Classifier to use as a spam filter.

The project must be programmed in Python, and can only use the following libraries: `NumPy`, `math`, `re`, `sys` and `Matplotlib`.

1 Your Task

Download the email corpus available on Moodle. This dataset contains emails that are already classified into 2 classes: *ham* and *spam*. You can assume that the files can be processed using `latin-1` encoding.

The project will be divided into 3 tasks:

Task 1: Building the model.

Task 2: Evaluating and analyzing the classifier.

Task 3: Experimenting with the classifier.

1.1 Task 1: Building the Model

Write a Python program to build a probabilistic model from the training set. Your code will parse the files in the training set and build a vocabulary with all the words it contains. Then for each word, compute their frequencies and probabilities for each class (class *ham* and class *spam*).

To process the texts, fold them to lowercase, then tokenize them using `re.split('\[\^a-zA-Z\]',aString)` and use the set of resulting words as your vocabulary.

For each word w_i in the training set, save its frequency and its conditional probability for each class: $P(w_i|ham)$ and $P(w_i|spam)$. These probabilities must be smoothed using the *add* δ with $\delta = 0.5$.

Save your model in a text file called `model.txt`. The format of this file must be the following:

1. A line counter i, followed by 2 spaces.
2. The word w_i, followed by 2 spaces.
3. The frequency of w_i in the class *ham*, followed by 2 spaces.
4. The smoothed conditional probability of w_i in the class *ham* - $P(w_i|ham)$, followed by 2 spaces.
5. The frequency of w_i in the class *spam*, followed by 2 spaces.
6. The smoothed conditional probability of w_i in *spam* - $P(w_i|spam)$, followed by a carriage return.

Note that the file must be sorted alphabetically. For example your file `model.txt` could look like the following:

```
1  abc  3  0.003  40  0.4
2  airplane  3  0.003  40  0.4
3  password  40  0.4  50  0.03
4  zucchini  0.7  0.003  0  0.000001
```

1.2 Task 2: Building and Evaluating the Classifier

Once you have built your model (task 1), use it to implement and test a Naïve Bayes classifier to classify emails into their most likely class: *ham* or *spam*. To avoid arithmetic underflow, work in log_{10} space.

Run your classifier on the test set given and create a single file called `baseline-result.txt` with your classification results. For each test file, `baseline-result.txt` should contain:

1. a line counter, followed by 2 spaces
2. the name of the test file, followed by 2 spaces
3. the classification as given by your classifier (the label *spam* or *ham*), followed by 2 spaces
4. the score of the class *ham* as given by your classifier, followed by 2 spaces
5. the score of the class *spam* as given by your classifier, followed by 2 spaces
6. the correct classification of the file, followed by 2 spaces
7. the label *right* or *wrong* (depending on the case), followed by a carriage return.

For example your result file could look like the following :

```
1  test-ham-00001.txt  ham  0.004  0.001  ham  right
2  test-ham-00002.txt  spam  0.002  0.03  ham  wrong
```

1.3 Task 3: Experiment with your Classifier

Now the fun part! Tasks 1 and 2 above will constitute your experiment 1, or *baseline experiment*, and you will perform variations over this baseline to see if they improve the performance of your classifier.

1.3.1 Experiment 2: Stop-word Filtering

Download the list of stop words available on Moodle. Use the baseline experiment and redo tasks 1 and 2 but this time remove the stop words from your vocabulary. Generate the new model and result files that you will call `stopword-model.txt` and `stopword-result.txt`.

1.3.2 Experiment 3: Word Length Filtering

Use the baseline experiment and redo tasks 1 and 2 but this time remove all words with length ≤ 2 and all words with length ≥ 9. Generate the new model and result files that you will call `wordlength-model.txt` and `wordlength-result.txt`.

1.3.3 Experiment 4: Infrequent Word Filtering

Only COMP 6721 students need to do experiment 4. Use the baseline experiment, and gradually remove from the vocabulary words with frequency$= 1$, frequency ≤ 5, frequency ≤ 10, frequency ≤ 15 and frequency ≤ 20. Then gradually remove the top 5% most frequent words, the 10% most frequent words, 15%, 20% and 25% most frequent words. Plot the performance of the classifier against the number of words left in your vocabulary.

1.3.4 Experiment 5: Smoothing

Only COMP 6721 students need to do experiment 5. Use the baseline experiment, and change the smoothing value gradually from $\delta = 0$ to $\delta = 1$ in steps of 0.1. Plot the performance of the classifier against the smoothing value.

1.4 Report

1.4.1 Report for COMP 472

In COMP 472, your report should be 3-5 pages (without references and appendix) and use the template provided on Moodle. The report should contain at least the following:

- $\frac{1}{2}$ *page:* Show and analyze the results of the baseline experiment (exp. #1). Give a table of results showing the accuracy, precision, recall and F_1-measure for each class, as well as a confusion matrix. Analyze and discuss these results.
- $\frac{1}{2}$ *page:* Do the same with the results of the stop-word filtering experiment (exp. #2).
- $\frac{1}{2}$ *page:* Do the same with the results of the word-length filtering experiment (exp. #3).
- *1 page:* Compare & discuss the results of the 3 experiments.
- $\frac{1}{2}$ *page:* Describe any difficulties that you have encountered and how you addressed them.
- $\frac{1}{2}$ *page:* If you were to continue working on this project, what do you feel would be interesting to investigate? Are there questions that you would like to investigate more, if you had the time and the energy?
- Your report should have a reference section (not included in the page count) that properly cites all relevant resources that you have consulted (books, Web sites ...), even if it was just to inspire you. Failure to properly cite your references constitutes plagiarism and will be reported.

Your report should:

□ follow the Word or LaTeX template provided on Moodle or an equivalent format

□ be submitted in PDF format

□ be called `472_P2_Report_StudentID1_StudentID2_...pdf`

1.4.2 Report for COMP 6721

In COMP 6721, your report should be 4-6 pages. In addition to the content of the COMP 472 report (see Section 1.4.1), your report should include:

- $\frac{1}{2}$ *page:* Results and analysis of the infrequent word filtering experiment (exp. #4).
- $\frac{1}{2}$ *page:* Results and analysis of the smoothing experiment (exp. #5)
- *1 page:* Compare & discuss the results of the 5 experiments

Your report should:

□ follow the Word or LaTeX template provided on Moodle or an equivalent format

□ be submitted in PDF format

□ be called `6721_P2_Report_StudentID1_StudentID2_...pdf`

1.5 Demos

The project will be demonstrated to the TAs in the lab. Regardless of the demo time, you will demo the program that was uploaded as the official submission on or before the due date. The schedule of the demos will be posted on Moodle. No special preparation is necessary for the demo (no slides or prepared speech). Your TA will ask you questions on your code, and you will have to answer him/her. Part of your marks will be based on your demos.

At the demo, the TA will give you a new dataset and you will need to generate new output files called `demo-model.txt` and `demo-result.txt`, that your team leader will submit right away on EAS as Project 6.

2 Evaluation Scheme

Students will be given individual grades that will be a function of the team grade and the peer-evaluation.

Team grade:

Code & Demo	40%
Report & Analysis	60%

Individual grade:

At the end of the project, all team members will fill-in a peer-evaluation form to evaluate the contribution of each team member. The grade of a student will be a function of his/her team grade and the peer evaluation received.

3 Submission

3.1 Code

The code is due Sunday April 7 at 6AM, 2019

- □ Create one zip file, containing:
  - □ all your code
  - □ a `README.txt` file which will contain specific and complete instructions on how to run your program. If the instructions in your readme file do not work or are incomplete, you will not be given the benefit of the doubt.
  - □ the files `baseline-model.txt`, `stopword-model.txt` and `wordlength-model.txt`
  - □ the files `stopword-result.txt`, `stopword-result.txt` and `wordlength-result.txt`
  - □ the signed expectation of originality form (available on Moodle; or at: `http://www.encs.concordia.ca/documents/expectations.pdf`)
- □ Name your zip file: `472_P2_studentID1_studentID2_....zip` or `6721_P2_studentID1_studentID2_....zip` where studentID1 is the team leader.
  For example, `6721_P2_12345678.zip` or `472_P2_12345678_87654321.zip`
- □ Have the Team Leader upload the zip file on EAS from his/her account as **Project 5**.

3.2 Demo

The demos will be during the lab periods on Monday April 8 and Thursday April 12, 2019.

- □ During your demo, you will also submit the files `demo-model.txt` and `demo-result.txt` with the dataset given by the TA. Submit these on EAS as **Project 6**.

3.3 Report

The report is due Friday April 12 at noon, 2019

- □ Print your report and submit a paper copy in the appropriate assignment box in EV 3.177.
- □ Follow the naming convention specified in Section 1.4, and have the Team Leader upload the PDF on EAS from his/her account as **Project 7**.

Have fun!