PROJECT 1 README

PART 1: DATA PREPARATION

in order to extract the data from the training and test folders and store them into csv files, you have to run 'cleaner.py'.in the terminal, use the following command:

```
python cleaner.py
```

The program will then ask which dataset you would like to pull from

Enter either a 1, 2, or 4

the program will then produce 4 csv files. Each one should follow the form:

```
enron#_type_(train/test).csv
```

If you want to generate all 12 csv files, then all you have to do is run the program 3 times and select a unique number each time. I have them already made, but if you don't trust these csv files, then go ahead.

One important thing to note is that in the original zip file given to us, the directories for the datasets are structured differently. by that i mean to access the ham emails in enron1_test, it followed the form:

```
dataset\enron1_test\test\ham
```

however, in order to access the ham files in enron2_test, it followed the form:

```
dataset\enron2_test\ham
```

The differing structures made it very annoying to program, so I altered the structures of enron1 and enron4 so that they followed the design of enron2. This is not really that important, but it is worthy to note that if you use my code on the original zip file given to us, it wouldn't work.

PART 2: MULTINOMIAL NAIVE BAYES

in order to report the metrics of multinomial naive bayes, run the following line of code:

```
python multinomial.py
```

Similar to the data preparation, it will prompt you to select which dataset you want to choose from. Enter 1, 2, or 4, and it will print out the metrics for that dataset

PART 3: BERNOULLI NAIVE BAYES

in order to report accuracy of bernoulli naive bayes, run the following line of code:

```
python bernoulli.py
```

Again, this uses the same method as before. The program will prompt you to select which dataset to choose from. Enter either a 1, 2, or 4, and the program will report the metrics generated from this model.

IMPORTANT NOTE:

Should you decide to delete the CSV files that I have made, you will have to recreate them using the instructions labled in part 1.

PART 4: LOGISTIC REGRESSION

To run the Logistic Regression model, simply run the following command:

```
python logistic.py
```

Again, it will ask you which dataset to choose from. Select either 1,2, or 4. In addition, this one will ask you to select which representation you would like to use, as this model can work with both the bag-of-words as well as the bernoulli representation.

The logistic regression model uses gradient ascent, which takes forever. Please be patient. I have the model print out which iteration it is on so you can see the progress the algorithm is making.

The logistic regression model first trains a number of models, each one differs by what the regularization constant is. The program then selects the model which scores the highest accuracy on the validation set, then tests this model on the validation set. Metrics for each model's performance on the validation set are reported as well as the metrics for the best model's performance on the test set.

Have a nice day!