

Readme

□ bayes.py

This is the main python file which contains the program for naïve bayes classifier. If the dictionaries are not already present in the working directory, then the program executes the training for the model. Once the training is complete, a message will appear to rerun the program. After rerunning the program, the classifier will classify each file as positive, negative, or neutral.

To overcome the problem of underflow the probability of each word is calculated and then its log is calculated and finally all the logs of each word in the sentence is added to get the final probability. Whichever probability is greater (positive, negative or neutral) is the output class. Smoothing is used to overcome the problem of zero probability. If a word is not in dictionary then that word is ignored for probability calculation to avoid getting probabilities as zero.

Steps to run on tux:

- 1) Create 'training' directory in work directory and add training files to that folder.
- 2) Create 'testing' directory in work directory and add testing files to the folder.
- 3) Now run evaluate.py using command "python3 evaluate.py" on tux.
- 4) The output showing performance of the classifier will be

generated. Result:

```
movies-5-20568.txt: positive
movies-5-20630.txt: positive
actors-1-935.txt: negative
actors-1-10189.txt: positive
movies-5-20657.txt: positive
movies-5-20533.txt: positive
actors-1-1449.txt: positive
actors-1-1168.txt: positive
movies-5-20704.txt: positive
movies-5-20581.txt: positive
actors-1-827.txt: positive
sports-5-8253.txt: positive
actors-1-924.txt: positive
actors-1-1389.txt: positive

Results Summary:
negative: 94
neutral: 5
positive: 344
Total files passed to the model: 443
Correctly predicted positive: 274
Correctly predicted negative: 70
ak4229@tux4:~/IntrotoAI/sentiment$
```

□ bayesbest.py

This is the python file which contains improved naïve bayes classifier. Some features modified in this model is that each word is converted in lowercase, then if punctuation is present then it is removed, and any whitespaces are also removed. Addition of these features resulted in better performance from normal naïve bayes classifier i.e bayes.py. The execution of this file is same as bayes.py.

Steps to run:

- 1) Run 'python3 evaluatebest.py' on tux.

Result:

```
sports-5-8234.txt: positive
actors-1-1851.txt: neutral
movies-5-20635.txt: positive
actors-1-10054.txt: positive
movies-5-20742.txt: positive
actors-1-1191.txt: negative
movies-5-20491.txt: positive
sports-5-8067.txt: positive
actors-1-2142.txt: positive
actors-1-1342.txt: positive
movies-5-20568.txt: positive
movies-5-20630.txt: positive
actors-1-935.txt: negative
actors-1-10189.txt: positive
movies-5-20657.txt: positive
movies-5-20533.txt: positive
actors-1-1449.txt: positive
actors-1-1168.txt: positive
movies-5-20704.txt: positive
movies-5-20581.txt: positive
actors-1-827.txt: positive
sports-5-8253.txt: positive
actors-1-924.txt: positive
actors-1-1389.txt: positive

Results Summary:
negative: 43
neutral: 9
positive: 391
Total files passed to the model: 443
Correctly predicted positive: 297
Correctly predicted negative: 44
```

□ singleline.py

This python file takes one sentence as a input from user and outputs if the sentence is positive or negative.

Steps to run:

- 1) Simply run this file on tux with "python3 singleline.py"
- 2) You have an option to get the result from bayes.py or bayesbest.py.
- 3) "exec(open("bayes.py").read())" , this code in line 3 can be changed to "exec(open("bayesbest.py").read())" if you want to run it with bayesbest.py.

Result:

```
ak4229@tux4:~/IntrotoAI/sentiment$ python3 singleline.py
Enter the sentence:
I loved this food and will eat it again someday
positive
ak4229@tux4:~/IntrotoAI/sentiment$ python3 singleline.py
Enter the sentence:
I do not want to eat this food
negative
ak4229@tux4:~/IntrotoAI/sentiment$
```

□ evaluate.py

This file is to run bayes.py and bayesbest.py. I have added some code to the already provided code of evaluate.py to get number of correct positive and negative outputs given by the classifier.

USE THIS FILE TO RUN THE CLASSIFIERS.

Remember making 'testing' and 'training' directories in working directory to run this file.

Steps to run:

- 1) Run this file on tux using "python3 evaluate.py"
- 2) 'testFile = "bayes.py"', this code in line 5 can be changed to 'testFile = "bayesbest.py"' if you want to run bayesbest.py

NOTE: USE evaluatebest.py to run bayesbest.py

□ Evaluation.pdf

This pdf document contains calculation of precision, recall and f-measure for bayes.py and bayesbest.py. The calculation is based on the results of testing from 400 files.