CS 535: Assignment #8

Due on 11:59pm April 8, 2018

Alexander C. Nwala 16:20

David Sinclair

April 5, 2018

Contents

1 Problem 1

Support your answer: include all relevant discussion, assumptions, examples, etc.(10 points)

1.1 Question 1

(Spam classification using Naive Bayes classifier)
1. Create two datasets; the first called Testing, the second called Training.
The Training dataset should:
a. consist of 10 text documents for email messages you consider spam (from your spam folder)
b. consist of 10 text documents for email messages you consider not spam (from your inbox)

The Testing dataset should:
a. consist of 10 text documents for email messages you consider spam (from your spam folder)
b. consist of 10 text documents for email messages you consider not spam (from your inbox)

Upload your datasets on github

1.2 Answer 1

To start with I attempted to save 20 emails from odu email accounts. I found that I did not have many spam emails in my odu account. I then went to my yahoo account and found a lot of spam there. I was unable to save the emails as a plan txt document. I was able to print the emails to a pdf file.

I then printed the 40 emails to files. I then attempted to look at them as plain text. This failed, I needed to convert them from pdf format to text format. I used a program called pdfminer which I got from the following website:

https://github.com/pdfminer/pdfminer.six

This created an output file of spam#.txt or nonspam#.txt which I saved in folder pdfprogram directory. In looking at this data it was not usable for my needs. I then decided to just open the orginial file and do a select all, copy and save to a vim document called spam#.txt or nonspam#.txt.

After saving all the files as text. I then modified the docclass.py code. I created a program called 01_getwords.py for all the training emails.

```
for file in os.listdir(path):
     if file.endswith(".txt"):
          print(file,file=open('01names.txt','a'))
with open('01names.txt') as f:
    text = f.read().splitlines()
for name in text:
    filename=path+name
    with open(filename) as f:
         doc = f.read()
         print(doc)
         splitter=re.compile('\\W*')
         # Split the words by non-alpha characters
         #words=[s.lower() for s in splitter.split(doc) if len(s)>2 and len(s)<20]
         words=[s.lower() for s in splitter.split(doc) if len(s)>2 and len(s)<20]
         # Return the unique set of words only
         print(words,file=open('01words.txt','a'))
```

The program created 2 files. File 1 was the list of the file names in the training directory. File 2 created a list of all the words from each file called 01words.txt.

I took the 01words.txt file and using a web alphabetizer https://www.textfixer.com/tools/alphabetical-order.php. I removed duplicates and alphabetized the words. I then took the words and added them to the docclass.py code.

Below is a example of the words added to the docclass.py:

```
  cl.train('07675 502 2018 73407 alright amber amprint april bloomberg blunt built bunny buzz click
      colbert com creepy date detailspark diy dress dsincl1999 easter edt eletter emily finds folders
      fox from here https idea introduction late lele like little mail makeover mary messages must news
      nhl nice old onlinevideo plaza pons poppins print prom quite request save scholl send show sports
      state stephen subject sunday tank tappan the unsubscribe wheelchair window with yahoo you your',
      'spam')
  cl.train('16th 2017 2018 10004 73406 ability able accounting adverse advertisement advice all amount
      amprint and approximate april are assume assurance attorney available away bankruptcy based before
      broadway calls can circumstances claims clients com company complete consequences consult consumer
      contact content copyright credit creditors date debt debts depending discuss dsincl1999 edt
      eligible enrolled enrollment estimate estimates fees floor folders for free from funds get
      guarantee how https impact including information legal llc local lowered mail make materials may
      messages monitored monthly months more national new not note obligation our over
      panoramicdirection payments percentage period please potential print prior professional program
      provide purposes quality quickly rating read ready realize reasons recommend recorded relief
      repair reserved results rights save savings see services settled settlement specific state states
      stay subject sufficient sunday tax that the their this time training understand unsubscribe
      upfront usa various vary which who will window with within yahoo york you your', 'spam')
```

After added the words I verified that the test.py document would work. The program did not work the way I expected. I then modified the program and called it train.py.

```
import docclass
from subprocess import check_output
import os, os.path

path ='/home/david/Documents/cs532/assignment8_draft/train/'

with open('01names.txt') as f:
#with open('/home/david/Documents/cs532/assignment8_draft/train/nonspam1.txt') as f:
    text = f.read().splitlines()
for name in text:
    filename=path+name
    with open(filename) as f:
       doc = f.read()
#doc = docclass.naivebayes(docclass.getwords)
       cl = docclass.naivebayes(docclass.getwords)
#remove previous db file
       check_output(['rm', 'spam.db'])
       cl.setdb('spam.db')
       docclass.spamTrain(cl)
       #classify text: "the banking dinner" as spam or not spam
       print(filename, cl.classify(doc))
```

Here are the results using the following program.

/home/david/Documents/cs532/assignment8_draft/train/nonspam1.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam2.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam3.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam4.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam5.txt not spam

/home/david/Documents/cs532/assignment8_draft/train/nonspam6.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam7.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam8.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam9.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/nonspam10.txt not spam
/home/david/Documents/cs532/assignment8_draft/train/spam1.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam2.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam3.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam4.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam5.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam6.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam7.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam8.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam9.txt spam
/home/david/Documents/cs532/assignment8_draft/train/spam10.txt spam

2 Problem 2

2.1 Question 2

2. Using the PCI book modified docclass.py code and test.py (see Slack assignment-8 channel)
Use your Training dataset to train the Naive Bayes classifier (e.g.,docclass.spamTrain())
Use your Testing dataset to test (test.py) the Naive Bayes classifier and report the classification results.

2.2 Answer 2

I had to modify the test.py code in the following way.

```
import docclass
from subprocess import check_output
import os, os.path

path ='/home/david/Documents/cs532/assignment8_draft/test/'

with open('02test.txt') as f:
#with open('/home/david/Documents/cs532/assignment8_draft/train/nonspam1.txt') as f:
    text = f.read().splitlines()
for name in text:
    filename=path+name
    with open(filename) as f:
       doc = f.read()
#doc = docclass.naivebayes(docclass.getwords)
       cl = docclass.naivebayes(docclass.getwords)
#remove previous db file
       check_output(['rm', 'spam.db'])
       cl.setdb('spam.db')
       docclass.spamTrain(cl)
       #classify text: "the banking dinner" as spam or not spam
       print(filename, cl.classify(doc))
```

Here are the results using the following program.

/home/david/Documents/cs532/assignment8_draft/test/nonspam11.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam12.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam13.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam14.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam15.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam16.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam17.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam18.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam19.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/nonspam20.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/spam11.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam12.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam13.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam14.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam15.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/spam16.txt not spam
/home/david/Documents/cs532/assignment8_draft/test/spam17.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam18.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam19.txt spam
/home/david/Documents/cs532/assignment8_draft/test/spam20.txt spam

3 Problem 3

==========
========Each question below is for 3 points extra credit==========
==========

3.1 Question 3

3. Draw a confusion matrix for your classification results
(see: https://en.wikipedia.org/wiki/Confusion_matrix)

3.2 Answer 3

Here is my confusion matrix

Confusion Matrix Title	Not Spam	Spam
Predicted Condition Positive	10	0
Predicted Condition Negative	2	8

4 Problem 4

4.1 Question 4

4. Report the precision and accuracy scores of your classification results (see: https://en.wikipedia.org/wiki/Precision_and_recall)

4.2 Answer 4

To do precision $tp/(tp + fp)$
$10/(10 + 0) = 1$
To do accuracy $(tp + tn)/(tp + tn + fp + fn)$
$(10 + 8)/(10 + 8 + 0 + 2) = 18/20 = .9$
So the precision is 1 and the accuracy is .9