DAT405/DIT407 Introduction to Data Science and AI

2022-2023, Reading Period 4

Assignment 4: Spam classification using Naïve Bayes

The exercise takes place in this notebook environment. Hints: You can execute certain linux shell commands by prefixing the command with `!`. You can insert Markdown cells and code cells. The first you can use for documenting and explaining your results the second you can use writing code snippets that execute the tasks required.

In this assignment you will implement a Naïve Bayes classifier in Python that will classify emails into spam and non-spam ("ham") classes. Your program should be able to train on a given set of spam and "ham" datasets. You will work with the datasets available at https://spamassassin.apache.org/old/publiccorpus/. There are three types of files in this location:

- easy-ham: non-spam messages typically quite easy to differentiate from spam messages.
- hard-ham: non-spam messages more difficult to differentiate
- spam: spam messages

Execute the cell below to download and extract the data into the environment of the notebook -- it will take a few seconds. If you chose to use Jupyter notebooks you will have to run the commands in the cell below on your local computer, with Windows you can use 7zip (https://www.7-zip.org/download.html) to decompress the data.

What to submit: Convert the notebook to a pdf-file and submit it. Make sure all cells are executed so all your code and its results are included. Double check the pdf displays correctly before you submit it.

In [ ]:

```
#Download and extract data
!wget https://spamassassin.apache.org/old/publiccorpus/20021010_easy_ham.tar.bz
!wget https://spamassassin.apache.org/old/publiccorpus/20021010_hard_ham.tar.bz
!wget https://spamassassin.apache.org/old/publiccorpus/20021010_spam.tar.bz2
!tar -xjf 20021010_easy_ham.tar.bz2
!tar -xjf 20021010_hard_ham.tar.bz2
!tar -xjf 20021010_spam.tar.bz2
```

The data is now in the three folders `easy_ham`, `hard_ham`, and `spam`.

In [ ]:

```
!ls -lah
```

1. Preprocessing:

1.1 Look at a few emails from easy_ham, hard_ham and spam. Do you think you would be able to classify the emails just by inspection? How do you think a succesful model can learn the difference between the different classes of emails?

In [ ]:

```
# Write your code for here for looking a few emails
```

Answer 1.1:

1.2 Note that the email files contain a lot of extra information, besides the actual message. Ignore that for now and run on the entire text (in the optional part further down can experiment with filtering out the headers and footers). We don’t want to train and test on the same data (it might help to reflect on why if you don't recall). Split the spam and the ham datasets in a training set and a test set. (`hamtrain` , `spamtrain` , `hamtest` , and `spamtest`). Use only the easy_ham part as ham data for quesions 1 and 2.

In [ ]:

```
# Write your code for here for splitting the data
```

2.1 Write a Python program that:

1. Uses the four datasets from Question 1 (`hamtrain` , `spamtrain` , `hamtest` , and `spamtest`)
2. Trains a Naïve Bayes classifier (use the scikit-learn library) on `hamtrain` and `spamtrain` , that classifies the test sets and reports True Positive and False Negative rates on the `hamtest` and `spamtest` datasets. Use `CountVectorizer` (Documentation here) to transform the email texts into vectors. Please note that there are different types of Naïve Bayes Classifier in scikit-learn (Documentation here). Test two of these classifiers that are well suited for this problem:

- Multinomial Naive Bayes
- Bernoulli Naive Bayes.

Please inspect the documentation to ensure input to the classifiers is appropriate before you start coding.

In [ ]:

```
# Write your code here
```

2.2 Answer the following questions:

a) What does the CountVectorizer do?

Answer 2.2.a

b) What is the difference between Multinomial Naive Bayes and Bernoulli Naive Bayes

Answer 2.2.b

3.1 Run the two models:

Run (don't retrain) the two models from Question 2 on spam versus hard-ham. Does the performance differ compared to question 2 when the model was run on spam versus easy-ham? If so, why?

In [ ]:

```
# Write your code here
```

Answer 3.1:

3.2 Retrain

Retrain new Multinomial and Bernolli Naive Bayes classifers on the combined (easy+hard) ham and spam. Now evaluate on spam versus hard-ham as in 3.1. Also evaluate on spam versus easy-ham. Compare the performance with question 2 and 3.1. What do you observe?

In [ ]:

```
# Write your code here
```

Answer 3.2:

3.3 Further improvements

Do you have any suggestions for how performance could be further improved? You don't have to implement them, just present your ideas.

Answer 3.3: