LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CU6051NA - Artificial Intelligence

Assignment

Naïve Bayes Classifier

Year and Semester

2018-19 Spring

Student Name: Rajat Shrestha

London Met ID: 17030954

College ID: np01cp4a170021

Table of Contents

Naïve Bayes Classifier Assignment Solution

We must calculate and classify the test data by using the labelled training data provided below using the Naïve Bayes Classifier for spam detection.

Training examples consist of text labeled as spam or not spam. Use the examples to build the vocabulary for the classifier. Then using the bag of words approach, transform the texts into feature vectors. Then following the algorithm for the Naïve Bayes Classifier, classify the 2 texts in the test data table as spam or not spam.

Training Data:

Text	Label
Congrats, You have won!! reply to our sms for a free nokia mobile + free camcorder.	spam
Congrats! 1 year special cinema pass for 2 is yours. reply to this sms to claim your prize.	spam
I am pleased to tell you that you are awarded with a 1500 Bonus Prize, reply to this sms to claim your prize.	spam
Dont worry. I guess he is busy.	not spam
Going for dinner. msg you later.	not spam
Ok, I will call you up when I get some cash.	not spam

Test Data:

Text	Label
I am busy. I will msg you later.	?
Congrats! You are awarded a free mobile.	?

Figure 1: Tables of data

To build a decision tree we must go through various process to find the nodes:

1. Building vocabulary:

First, we must fine the overall vocabulary used in both the mails and spams. By filtering out the special characters, punctuations, and numbers then converting all of the words to lowercase we get:

```
['ok', 'i', 'will', 'call', 'you', 'up', 'when', 'get', 'some', 'cash',
'going', 'for', 'dinner', 'msg', 'later', 'dont', 'worry', 'guess', 'he',
'is', 'busy', 'am', 'pleased', 'to', 'tell', 'that', 'are', 'awarded',
'with', 'a', 'bonus', 'prize', 'reply', 'congrats', 'year', 'special',
'cinema', 'pass', 'yours', 'this', 'sms', 'claim', 'your', 'have', 'won',
'our', 'free', 'nokia', 'mobile', 'camcorder']
```

This vocabulary can be used to

2. Bag of words approach to build feature vectors:

Using the created vocabulary and mapping them out with mails and then spams to calculate the frequency of their occurrence in a table. The first column represents the vocabulary words while the other columns represents each sentence and the tally of the occurrence of the respective word.

Table 1: Bag of words for mails

Words	1st	2nd	3rd
ok	1	0	0
i	2	0	1
will	1	0	0
call	1	0	0
you	1	1	0
up	1	0	0
when	1	0	0
get	1	0	0
some	1	0	0
cash	1	0	0
going	0	1	0
for	0	1	0
dinner	0	1	0
msg	0	1	0
later	0	1	0
dont	0	0	1
worry	0	0	1
guess	0	0	1
he	0	0	1
is	0	0	1
busy	0	0	1
am	0	0	0
pleased	0	0	0
to	0	0	0
tell	0	0	0
that	0	0	0
are	0	0	0
awarded	0	0	0
with	0	0	0
a	0	0	0
bonus	0	0	0
prize	0	0	0
reply	0	0	0
congrats	0	0	0
year	0	0	0
special	0	0	0
cinema	0	0	0
pass	0	0	0
yours	0	0	0
this	0	0	0
sms	0	0	0
claim	0	0	0
your	0	0	0
have	0	0	0
won	0	0	0
our	0	0	0
free	0	0	0
nokia	0	0	0
mobile	0	0	0
camcorder	0	0	0

Table 2: Bag of words for spams

Words	1st	2nd	3rd
ok	0	0	0
i	1	0	0
will	0	0	0
call	0	0	0
you	2	0	1
up	0	0	0
when	0	0	0
get	0	0	0
some	0	0	0
cash	0	0	0
going	0	0	0
for	0	1	1
dinner	0	0	0
msg	0	0	0
later	0	0	0
dont	0	0	0
worry	0	0	0
guess	0	0	0
he	0	0	0
is	0	1	0
busy	0	0	0
am	1	0	0
pleased	1	0	0
to	2	2	1
tell	1	0	0
that	1	0	0
are	1	0	0
awarded	1	0	0
with	1	0	0
a	1	0	1
bonus	1	0	0
prize	1	1	0
reply	1	1	1
congrats	0	1	1
year	0	1	0
special	0	1	0
cinema	0	1	0
pass	0	1	0
yours	0	1	0
this	0	1	0
sms	0	1	1
claim	0	1	0
your	0	1	0
have	0	0	1
won	0	0	1
our	0	0	1
free	0	0	2
nokia	0	0	1
mobile	0	0	1
camcorder	0	0	1

3. Calculating each word in both bags:

Calculating the weight for the words found in mails:

0.02702702702702703	ok
0.05405405405405406	i
0.02702702702702703	will
0.02702702702702703	call
0.04054054054054054	you
0.02702702702702703	up
0.02702702702702703	when
0.02702702702702703	get
0.02702702702702703	some
0.02702702702702703	cash
0.02702702702702703	going
0.02702702702702703	for
0.02702702702702703	dinner
0.02702702702702703	msg
0.02702702702702703	later
0.02702702702702703	dont
0.02702702702702703	worry
0.02702702702702703	guess
0.02702702702702703	he
0.02702702702702703	is
0.02702702702702703	busy
0.013513513513513514	am
0.013513513513513514	pleased
0.013513513513513514	to
0.013513513513513514	tell
0.013513513513513514	that
0.013513513513513514	are
0.013513513513513514	awarded
0.013513513513513514	with
0.013513513513513514	a
0.013513513513513514	bonus
0.013513513513513514	prize
0.013513513513513514	reply
0.013513513513513514	congrats
0.013513513513513514	year
0.013513513513513514	special
0.013513513513513514	cinema
0.013513513513513514	pass
0.013513513513513514	yours
0.013513513513513514	this
0.013513513513513514	sms
0.013513513513513514	claim
0.013513513513513514	your
0.013513513513513514	have
0.013513513513513514	won
0.013513513513513514	our
0.013513513513513514	free
0.013513513513513514	nokia
0.013513513513513514	mobile
0.013513513513513514	camcorder

$$p(w_k|+) = \frac{n_k + 1}{n + |Vocabulary|}$$ $|Vocabulary|$ = total number of words in the vocabulary

Calculating the weight for the words found in spams:

```
0.010309278350515464          ok
0.020618556701030927          i
0.010309278350515464          will
0.010309278350515464          call
0.041237113402061855          you
0.010309278350515464          up
0.010309278350515464          when
0.010309278350515464          get
0.010309278350515464          some
0.010309278350515464          cash
0.010309278350515464          going
0.030927835051546393          for
0.010309278350515464          dinner
0.010309278350515464          msg
0.010309278350515464          later
0.010309278350515464          dont
0.010309278350515464          worry
0.010309278350515464          guess
0.010309278350515464          he
0.020618556701030927          is
0.010309278350515464          busy
0.020618556701030927          am
0.020618556701030927          pleased
0.061855670103092786          to
0.020618556701030927          tell
0.020618556701030927          that
0.020618556701030927          are
0.020618556701030927          awarded
0.020618556701030927          with
0.030927835051546393          a
0.020618556701030927          bonus
0.030927835051546393          prize
0.041237113402061855          reply
0.030927835051546393          congrats
0.020618556701030927          year
0.020618556701030927          special
0.020618556701030927          cinema
0.020618556701030927          pass
0.020618556701030927          yours
0.020618556701030927          this
0.030927835051546393          sms
0.020618556701030927          claim
0.020618556701030927          your
0.020618556701030927          have
0.020618556701030927          won
0.020618556701030927          our
0.030927835051546393          free
0.020618556701030927          nokia
0.020618556701030927          mobile
0.020618556701030927          camcorder
```

4. Mapping the words for given text

Text = 'I am busy. I will msg you later.'

First with weights calculated from mails

```
i 0.05405405405405406
am 0.013513513513513514
busy 0.02702702702702703
i 0.05405405405405406
will 0.02702702702702703
msg 0.02702702702702703
you 0.04054054054054054
later 0.02702702702702703
```

Then with weights calculated from spams

```
i 0.020618556701030927
am 0.020618556701030927
busy 0.010309278350515464
i 0.020618556701030927
will 0.010309278350515464
msg 0.010309278350515464
you 0.041237113402061855
later 0.010309278350515464
```

Text = 'Congrats! You are awarded a free mobile.'

First with weights calculated from mails

```
congrats 0.013513513513513514
you 0.04054054054054054
are 0.013513513513513514
awarded 0.013513513513513514
a 0.013513513513513514
free 0.013513513513513514
mobile 0.013513513513513514
```

Then with weights calculated from spams

```
congrats 0.030927835051546393
you 0.041237113402061855
are 0.020618556701030927
awarded 0.020618556701030927
a 0.030927835051546393
free 0.030927835051546393
mobile 0.020618556701030927
```

5. Final Calculation:

$$y_{new} = argmax_{y\epsilon(+,-)}\ p(y) \prod_{w \in words} p(w|y)$$

Figure 2: Formula for classifying

Multiplying the mapped value with their respective probabilities (using the above formula).

Text = 'I am busy. I will msg you later.'

Probability of being a mail P(M) = `4.2704876133169763e-13`

Probability of being spam P(S) = `2.041484601729748e-15`

As P(M) > P(S) it is a mail

Text = 'Congrats! You are awarded a free mobile.'

Probability of being a mail = `1.2344378257244386e-13`

Probability of being spam = `5.34664817193021e-12`

As P(M) < P(S) it is a spam

The link to the notebook used to calculate the values:

https://colab.research.google.com/drive/1uu4Cg6nLlRnKLM2MjalOZ0jpOoKBPrKo