Machine Problem 3

3 Credit Hours

Group Members:

Kartik Agrawal

Ajay Shekar

Extra Credit Completed

Part 1 Heatmap and ASCII map

Part 2.2 Completed

PART 1

DIGIT CLASSIFICATION

Description and Implementation Details:

- Training Phase
  - In the training phase, we build **priors** by calculating the number of occurrences of the class of digit and dividing it by the count of total number of digits in the training file.
  - The pixel value of '#' and '+' were treated the same as having a value of 1 and a pixel value of ' ' was considered to have a value of 0. We developed the **likelihood value** of each pixel of each class by dividing count of occurrences that pixel being 1 by the total number of occurrences of the class of digit.
- Laplace Smoothing Factor
  - In calculating posterior probability of a class, we took the log of each pixel value and added these values together; instead multiplying the probabilities together.
  - In doing so, we wanted to avoid taking the log of 0. This is where our Laplace smoothing factor played a major role. We experimented with different values of k but found that lower the value of k, the better overall precision we achieved.
  - We chose to take **k = 1** as we felt a lower value would give the posterior probability the appropriate value for that test class.
- Testing Phase
  - We performed a MAP for each test case for all classes and picked the correct answer as the one with the highest probability.
  - We achieved a success rate of **77.1%**
- Data Structures Used
  - Vectors served to be extremely useful in holding each picture and performing the required analysis.

<u>OVERALL ACCURACY</u>

Correct Percentage = 77.1%

InCorrect Percentage = 22.9%

Required Statistics and Outputs:

- Classification Rate for each digit:

Digit -----> Correct %
0 ------> 91.6
1 ------> 82.5
2 ------> 85.1
3 ------> 70.5
4 ------> 78.1
5 ------> 72.9
6 ------> 81.2
7 ------> 86.5
8 ------> 74.7
9 -------> 58.0

- Confusion Matrix:
  - Exported to excel in order to make it clearer to view.

	0	1	2	3	4	5	6	7	8	9
Predicted as 0	91.6	0	1.06	0	0.952	5.88	3.53	0	4.82	0
Predicted as 1	0	82.5	1.06	0	0	2.35	1.18	0	0	0
Predicted as 2	1.2	2.38	85.1	3.57	0.952	0	7.06	1.12	6.02	1.45
Predicted as 3	0	1.59	0	70.5	0	3.53	2.35	6.74	2.41	4.35
Predicted as 4	0	0.794	0	0	78.1	0	3.53	1.12	2.41	13
Predicted as 5	2.41	1.59	1.06	10.7	2.86	72.9	1.18	1.12	2.41	4.35
Predicted as 6	1.2	4.76	4.26	0	3.81	5.88	81.2	0	2.41	0
Predicted as 7	0	4.76	3.19	0	2.86	0	0	86.5	3.61	10.1
Predicted as 8	2.41	0.794	3.19	12.5	1.9	7.06	0	1.12	74.7	8.7
Predicted as 9	1.2	0.794	1.06	2.68	8.57	2.35	0	2.25	1.2	58

- Test examples of each class that have highest and lowest Posterior Priority:
  - Note: HighVal constitutes the test image with highest Posterior Priority and LowVal constitutes the test image with lowest Posterior Priority

HIGHVAL:

```
       +#++
     +#####+
    +########+
   +##########+
  +######+ +##+
 +#####+++ +###+
 ####+       +##+
+###+        ##+
####+        ##+
###+         +##
###+         ##+
###+         ##+
###+        +##+
###+       +##+
####+     +###+
+###++   +####+
 #####++#####+
 +##########+
  +########+
    ++++++
```

HIGHVAL:

```
     +#+
     +##+
     +##+
     ###+
     ###+
     ###
    +##+
    +##+
    +##+
    ###+
    ###
   +##+
   +##+
   ###+
  +###
  +##+
 +###+
 +####
 +###+
  +#+
```

HIGHVAL:

```
+###++
 ++++##+
     ++#+
      +##
       +#+
       +##
        ##
       +##
       +##
       +#+
      +##+
     ++##
     +##+
    +##+
   +##+
   +##
  +##+
  +##+
   +###++++
    ++++##+
```

LOWVAL:

```
    ++++++#+++
   +##########+
 +#############
+##############+
########++  ++###+
#######+     ####
  +###+      +###+
  +###        +###+
  ####         ###+
  ####         ####
  ###+         ####
  ####         ####
  ###+         ####
  +###+        ####
   +####       ####
    +###++     ####
     +#####++++#####
      +############+
       +##########+
        +++#######
```

LOWVAL:

```
+##+
+##+
+##++
 +###+
 +###+
 +###+
  ####+
  +####+
  +####+
   +###+
   +####
   +####+
   +#####+
   +######
   +######
   +######
   +######
    +#####
     +###+
     ++#++
```

LOWVAL:

```
          +++++###+
     +++###########
   +###############+
 +#################+
+##########++#######
+#########+  ######+
####++++++  +######+
###+      ++######+
##+      +#######+
+#+     +#########+
++      ########
        ######++
        +######++
        +#########+
         +#########+
           +#######+
            ++#####
               +###
```

HIGHVAL:

```
    +#####++
    +#########+
       +++####+
           ++##
            +##+
           +##+
         ++##+
        +####+
       +####+
      +######+
      ###+###+
      ++   +##+
           +#+
           +##
           +##
          +##+
         ++##+
+++++++++###+
+###########+
 +++######++
```

HIGHVAL:

```
    +++
    +## +
   ###+ +++
 ++##+  +##+
 ###+   +##
+##+    +##
+##     ###
+###++++###+++
 ++##########+
   ++++#####++
      +###
      +##+
      +##+
      ###
     +##+
     +##+
     +##+
     +#+
     ##+
     +#+
```

HIGHVAL:

```
            ++#+
     ++++++#####+
     ##########+
     ###++++++
    +#++
    +#+
   +##
   +#++++
  +#######+
  +##+++##
   +    +#+
        +#+
        +#+
         ##
+#+      ##+
+#+     +##
##+     ###
+###+  +###+
 ##+++####+
    ###++
```

LOWVAL:

```
  ++#++++++
 +##########+
 +#++++######+
         ++##+
          +##+
        +####
 ++++++#####+
##############++
#################+
+++++++    ++####+
              +####+
                +##+
                 ###
  +#+            +##
  ###+         ++###
  +#####+++++######+
   +###############+
     ++++##+###++
            ++
```

LOWVAL:

```
 +#+
 +##
 +##     ++#+
 +##     +##+
+###     +##+         +
+###     +##+        +#+
+###     +##+        +#+
+###     +##+        +#+
+##+     +###       +##+
+##+      +##   ++###+
+##+      +##+#####+
+##+ ++++#######++
+##############++
+############++
  ++++++++###
          ###+
          +##+
          +##+
           +##
            +#
```

LOWVAL:

```
 ++##+
+######+
#######+
#######+
####++
+###+++
 ++++###++
     +++####+
        +#####+
          ++##+
             ##++
              +##+
               +##+
      ++++     ++##+
      +####+++++++###
       +#############
        ++###########
             +++++++
```

HIGHVAL:

```
        +#+
      +####+
     +####+
    +####+
    +####+
   +####+
  +####+
  +##+
  +##+
 +####+
 +##+
 +##+   ++##+
+####+ +#####+
+###++#######+
+##+ +#######+
+###+########+
+#########+
 +#######+
  +#####+
    +##+
```

LOWVAL:

```
    +++#+
  +#####+
 +##+++
 ##+
+##
+#+
+#+
##+
+#+
+#+
 ##
 ##+         +++
 +##       +#####+
  +#+     +###++##+
   ##+    ###   +##
   +##+  +##+    +#+
    ++##+ ##+    +#+
      ++#####+  +##
        +##########
           +++++++
```

HIGHVAL:

```
        +##+
      ++#####+
    ++#######+
   +#####++##+
  ++####+ +##+
  +##+    +##+
  ###       +#++
  ###     ++###
  +##++++###++
  +########++
   ######++
  +######
 +#######+
 ###++##+
+##+   ###+
+##+   ###+
+##+  +###
+########+
 +#######+
  ++##++
```

LOWVAL:

```
    +++++
  #######++
+##########+
###########+#+
###########+#+ +
+####+##+++#+++#+
+###++   ++#####+
 +####+++####++
 +############
  +##########+
   +#########++
    +###########++
    +#####+######+
    ++###+#+++#####+
      +#####++#####+
      +##############+
       ++############
         +###########
          +#+#######+
             +++++
```

HIGHVAL:

```
###+#+
++  +#++
     +#+
     +#+
     +#+
     ##+
    +##
   +##+
  +###+
++####
++++#+
   +#+
  +##
  +#+
  +#+
 +##
 +#+
 +#+
+##
+#+
```

LOWVAL:

```
 ####++##+
###########+
############+
+############+
 +++#########+
    ++++++####++
         ++#####+
          ++#####
           ++####
            ++###+
             +####+
             +#####
             +#####
             +####+
             +####+
             +####+
              +###
              +###+
              +####
              +#####
```

HIGHVAL:

```
    ++#++
  ++######+
 +#####++#+
+####+  +#
+#++   +##+
##    +####
#+    ####+
#+   +####+
##  +#####
+######+#+
 +###++#+
  +++ ##+
      ##+
      ##+
     +##+
     +##
     +##
     +##
     +##
      #+
```

LOWVAL:

```
           +##+++++
       ++###########+
     ++###############+
  ++#############+#####
 +########++#+++  ++##
+#####+            +#+
#####+             ##+
####+            ++##+
####+         +++####+
#####++++############+
+#####################
 +############++   +###
   +++++++++       +###
                   +###
                   +##+
                   +##+
                   +###
                   +##+
                   +##+
                   +##+
```

- Odds Ratio: The four pairs we chose to perform odds ratios on are as follows
    - 9 and 7
    - 4 and 9
    - 9 and 8
    - 1 and 8

Below are the feature maps of each digit and odds ratio in the form of heat map and ASCII values. Please note that a pixel was considered to be '#' if it had a feature probability of greater than 0.5.

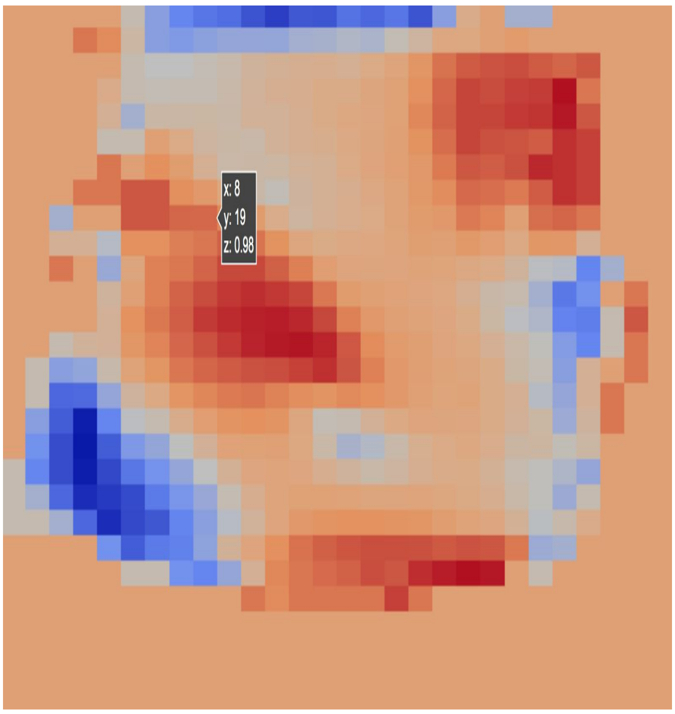
9 and 7:

```
      ###
   ########
  ##########
 ###########
 ####    ###
####    ####
###    #####
###   ######
############
###########
  #########
     #####
      ####
      ###
     ####
      ##
```

```
    ########
 #############
##############
##############
 ##      #####
         #####
         ####
        #####
        ####
       #####
       ####
      #####
      ####
     ####
     ####
     ###
     ##
```

```
      --         ++++
   ----
  -----
 -----
-----
----
 ---
 --
            +
        +++++          -
       +++++          --
        +++           --
                       -


                      +
                     ++
                      +
                    +++
                     ++
```



4 and 9:

```
            #
           ###
   ##     ####
  ###     ####
 ####     ###
 ####    ####
#####   #####
#############
#############
#############
  ##########
       ####
       ####
       ###
```

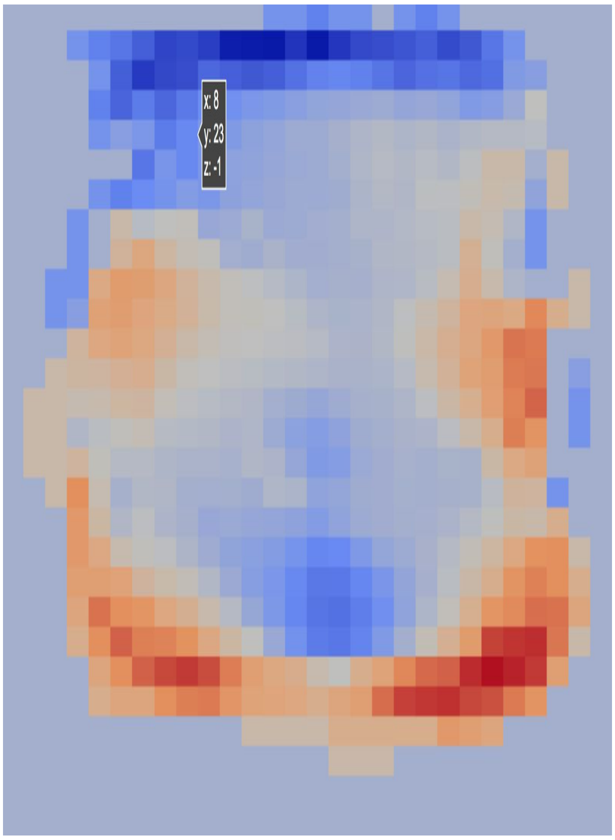
```
      ###
   ########
  ##########
 ###########
 ####     ###
####     ####
###     #####
###    ######
#############
############
  #########
     #####
      ####
      ###
     ####
      ##
```

```
                 ++
    ++++     +++++++++
  +++++++      +++++++++
 +++++            +++++
 +++        -      ++++
++                  +++
+                    ++
+
+

                    ++
                   +++
                   +++
                   +++
  +                  +
  ++



  - -
  ---------      -- --
   ----------------
```



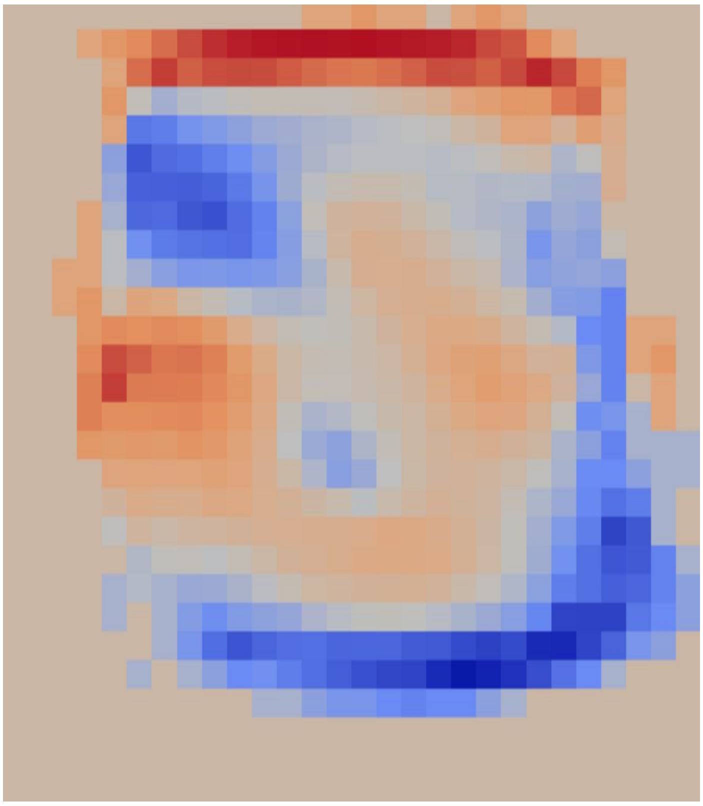
9 and 8:

```
     ######
   ##########
  ###########
  ############
 #####   #####
  ###    ####
  ####  #####
  ##########
   ########
   #######
   #######
  ########
 #########
 #########
 ###   ###
####  ####
##########
#########
 #######
    ##
```

```
      ###
   ########
  ##########
 ###########
 ####    ###
####    ####
###    #####
###   ######
############
###########
  #########
     #####
      ####
      ###
     ####
      ##
```

```
            ----
      ---------------
     -----------------
     -            ------
                   -----
                   -----
                    ---
                    ---
                    --
                     -
+                   -
+++++                -
 +++++               -
                    --
                     -

  ------
  ------
  -----
  -----
  ---
                   ++
  +++++++++++++++++++
   +++++++++++++++
```

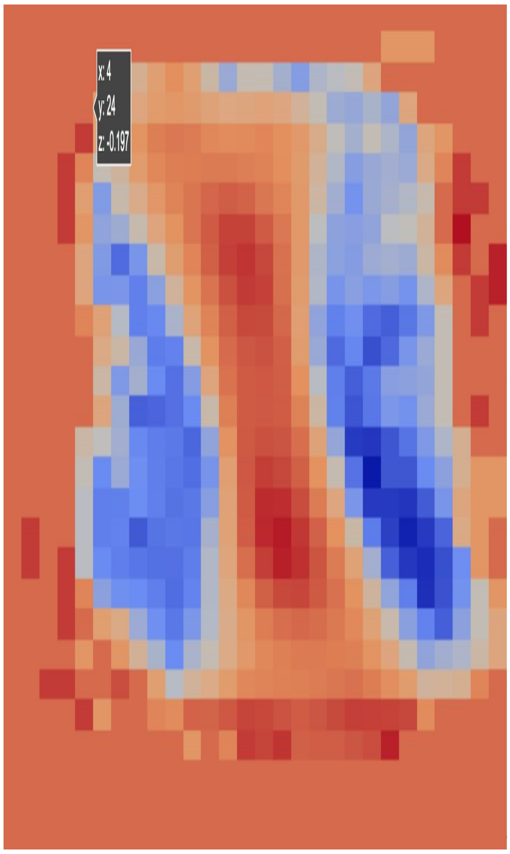


1 and 8:

```
    ###
   ####
   ####
   ####
   ####
  ####
  ####
  ####
 #####
 ####
 ####
 ####
####
####
####
###
###
```

```
     ######
   ##########
  ###########
  ############
 #####   #####
  ###    ####
  ####  #####
  ##########
   ########
   #######
   #######
  ########
 #########
 #########
 ###   ###
####  ####
##########
#########
 #######
    ##
```

```
     --            ---
   -----          -----
   -----         ------
 -------        -------
--------        ------
--------       ------
--------       ------
--------      --------
--------      --------
  ------     --------
 ------      --------
  -----      --------
  ----       --------
 -----       -------
 ----        -------
 ---         -------
 --          -------
 --          ------
              -----
  -         ------
   -   ---------
```



PART 2

TEXT DOCUMENT CLASSIFICATION

Description

- Training Phase
  - In the training phase, we first parse the training text files into separate hashmaps depending on the value of the label and the class that it corresponds to.
  - For each class, we create 2 hashmaps where one maps the word(Key) to the number of times(value) it occurs in the respective class. The second map maps a word(Key) to its respective probability which is calculated either using the Multinomial Naive Bayes Theorem or Bernoulli Naive Bayes, depending on whichever functionality is being carried out.
  - In each case, we also use an additional multimap to maintain an order of words that have highest probability of occurring in respective classes.
  - Below are the respective formulas used for the 2 probability models:
    - Multinomial Naive Bayes : $\frac{\textit{Number of Times the Word occurs in the class}}{\textit{Total number of words in tbe class}}$
    - Bernoulli Naive Bayes : $\frac{\textit{Number of Documents the word appears in}}{\textit{Total number of words in tbe class}}$
- Laplace Smoothing
  - Laplace smoothing factor plays a major role. We experimented with different values of k but found that lower the value of k, the better overall precision we achieved.
  - We chose to take **k = 1** as we felt a lower value would give the posterior probability the appropriate value for that test class.
- Testing
  - In the testing phase, the testing file is analyzed. For each document, a MAP value is created for each class that it may possibly belong to. The decision as to which class the document would be mapped to corresponded to the class with the highest MAP value.

Please note: Multinomial Naive Bayes probability and Bernoulli Naive Bayes probability outputs are placed in separate sections in the output category below.

OUTPUTS

SPAM DETECTION

Multinomial Naive Bayes

```
CORRECT Detection of Normal/Spam Email = 94.6154 %
WRONG Detection of Normal/Spam Email = 5.38462 %
```

Confusion Matrix

```
100 10.8
0 89.2
```

Top 20 Words with Highest Likelihood

****

Words that have the highest probability(Multinomial Naive Bayes) of occurring in Normal Email dataset are:

****

address
abstract
research
http
edu
include
please
one
english
e
paper
email
workshop
conference
information
de
linguistic
s
university
language

****

Words that have the highest probability(Multinomial Naive Bayes) of occurring in Spam Email dataset are:

****

nt
com
work
d
one
business
name
receive
list
money
free

send
program
mail
address
our
report
order
s
email

Bernoulli Naive Bayes

```
CORRECT Detection of Normal/Spam Email = 95.3846 %
WRONG Detection of Normal/Spam Email = 4.61538 %
```

Confusion Matrix

```
100 9.23
0 90.8
```

Top 20 Words with Highest Likelihood

****

Words that have the highest probability(Bernoulli Naive Bayes) of occurring in Negative Review dataset are:

****

interest
address
word
research
www
call
one
english
include
follow
fax
e
please
email
http
linguistic
information
s
university
language

****

Words that have the highest probability(Bernoulli Naive Bayes) of occurring in Positive Review dataset are:

****

want
over
here
remove

information
day
send
us
receive
http
com
address
list
one
mail
email
please
free
s
our

MOVIE REVIEW CLASSIFICATION

Multinomial Naive Bayes

```
CORRECT Detection of Positive/Negative Review = 72 %
WRONG Detection of Positive/Negative Review = 28 %
```

Confusion Matrix

```
74.2 30.2
25.8 69.8
```

Top 20 Words with Highest Likelihood

****

Words that have the highest probability(Multinomial Naive Bayes) of occurring in Negative Review dataset are:

****

make
plot
makes
nothing
never
comedy
would
little
good
characters
even
time
much
story
bad
--
one
like
film

movie
****
Words that have the highest probability(Multinomial Naive Bayes) of occurring in Positive Review dataset are:
****
characters
life
makes
us
funny
make
performances
much
best
time
even
way
comedy
good
story
like
one
--
movie
film

Bernoulli Naive Bayes

```
CORRECT Detection of Positive/Negative Review = 71.8 %
WRONG Detection of Positive/Negative Review = 28.2 %
```

Confusion Matrix

74	30.4
26	69.6

Top 20 Words with Highest Likelihood

****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in Negative Review dataset are:
****
make
never
plot
makes
nothing
comedy
would
good
little
characters
even
time
bad

--
much
story
one
like
film
movie
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in Positive Review dataset are:
****
work
characters
life
make
makes
funny
performances
much
time
best
good
even
way
comedy
story
--
like
one
movie
film

EXTRA CREDIT (PART 2.2)
NEWSGROUP DATASETS

Multinomial Naive Bayes

```
CORRECT Detection of the right Newsgroup = 92.7757 %
WRONG Detection of the right Newsgroup = 7.22433 %
```

Confusion Matrix

	Class A	Class B	Class C	Class D	Class E	Class F	Class G	Class H
Predicted Cla	100	6.06	0	0	0	10	0	3.45
Predicted Cla	0	69.7	0	0	0	10	0	3.45
Predicted Cla	0	0	97.2	0	0	0	0	0
Predicted Cla	0	12.1	0	89.3	0	0	0	0
Predicted Cla	0	3.03	0	0	100	10	0	0
Predicted Cla	0	0	0	0	0	70	0	0
Predicted Cla	0	0	2.78	0	0	0	100	0
Predicted Cla	0	9.09	0	10.7	0	0	0	93.1

Top 20 Words with Highest Likelihood

****

Words that have the highest probability(Multinomial Naive Bayes) of occurring in Sci.Space dataset are:

****

mission
edu
orbit
time
first
data
could
writes
also
system
us
like
subject
earth
nasa
launch
one
would
nt
space

****

Words that have the highest probability(Multinomial Naive Bayes) of occurring in comp.sys.ibm.pc.hardware dataset are:

****

also
data
m
get
bus

hard
edu
would
use
subject
system
disk
controller
drives
card
one
ide
nt
scsi
drive
****
Words that have the highest probability(Multinomial Naive Bayes) of occurring in rec.sport.baseball dataset are:
****
get
well
better
games
baseball
like
players
think
article
last
subject
team
good
one
game
writes
edu
year
would
nt
****
Words that have the highest probability(Multinomial Naive Bayes) of occurring in comp.windows.x dataset are:
****
windows
m
one
c
sun
program
system
version
motif
edu
get
available
also

server
file
subject
nt
use
window
x
****
Words that have the highest probability(Multinomial Naive Bayes) of occurring in talk.politics.misc dataset are:
****
right
get
going
subject
like
edu
us
know
stephanopoulos
government
article
writes
president
think
mr
one
q
people
would
nt
****
Words that have the highest probability(Multinomial Naive Bayes) of occurring in misc.forsale dataset are:
****
system
vs
good
hulk
nt
drive
comics
one
list
price
cover
shipping
wolverine
subject
art
appears
sale
dos
edu
new
****

Words that have the highest probability(Multinomial Naive Bayes) of occurring in rec.sport.hockey dataset are:
****
like
edu
la
get
players
think
year
first
one
games
nhl
season
period
subject
play
would
hockey
team
game
nt
****
Words that have the highest probability(Multinomial Naive Bayes) of occurring in comp.graphics dataset are:
****
would
system
version
get
format
files
program
one
use
available
software
graphics
also
data
images
file
nt
edu
jpeg
image

Bernoulli Naive Bayes

```
CORRECT Detection of the right Newsgroup = 93.1559 %
WRONG Detection of the right Newsgroup = 6.84411 %
```

Confusion Matrix

	Class A	Class B	Class C	Class D	Class E	Class F	Class G	Class H
Predicted Class A	100	6.06	0	0	0	10	0	3.45
Predicted Class B	0	72.7	0	0	0	10	0	3.45
Predicted Class C	0	0	97.2	0	0	0	0	0
Predicted Class D	0	9.09	0	89.3	0	0	0	0
Predicted Class E	0	3.03	0	0	100	10	0	0
Predicted Class F	0	0	0	0	0	70	0	0
Predicted Class G	0	0	2.78	0	0	0	100	0
Predicted Class H	0	9.09	0	10.7	0	0	0	93.1

Top 20 Words with Highest Likelihood

****

Words that have the highest probability(Bernoulli Naive Bayes) of occurring in Sci.Space dataset are:

****

see
edu
much
way
m
new
time
us
think
get
also
could
like
one
article
writes
space
nt
would
subject

****

Words that have the highest probability(Bernoulli Naive Bayes) of occurring in comp.sys.ibm.pc.hardware dataset are:

****

think
problem
work
drive
edu
system
two
m
also
like
card
article
get
know
use
writes

would
one
nt
subject
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in rec.sport.baseball dataset are:
****
first
team
game
know
time
m
get
think
good
baseball
like
year
last
one
would
edu
article
writes
nt
subject
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in comp.windows.x dataset are:
****
help
email
set
m
code
know
problem
also
would
like
one
using
article
window
get
writes
use
nt
x
subject
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in talk.politics.misc dataset are:
****
much

time
make
know
could
government
get
m
even
us
think
edu
like
one
would
people
article
writes
nt
subject
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in misc.forsale dataset are:
****
sell
know
use
good
used
want
condition
like
list
get
one
nt
price
email
please
new
shipping
edu
sale
subject
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in rec.sport.hockey dataset are:
****
last
time
games
year
nhl
get
go
think
first

play
like
article
one
would
writes
hockey
game
team
nt
subject
****
Words that have the highest probability(Bernoulli Naive Bayes) of occurring in comp.graphics dataset are:
****
two
m
program
think
could
need
computer
know
get
graphics
edu
use
article
also
like
would
writes
one
nt
subject

__GROUP CONTRIBUTIONS__

- Part 1 Digit Classification - Ajay Shekar
- Part 2 Text Document Classification - Kartik Agrawal & Ajay Shekar
- Extra Credit 2.2 - Kartik Agrawal
- Report Creation - Kartik Agrawal & Ajay Shekar