rolando_lab6

February 24, 2023

1 Lab 6 - EDA with Clustering

1.0.1 Jackson Rolando

1.1 Part 1 - Load and Transform the Data

We'll load the data, split it into training and testing, transform it into a sparse bag of words matrix, excluding words appearing less than 10 times:

[1]:
```
import glob
import json
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
objects = []
for file in glob.glob('./email_json/*.json'):
    with open(file) as f:
        objects.append(json.load(f))

df = pd.DataFrame(objects)
df.head()
```

[2]:
```
  category                                         to_address  \
0      ham  BREAKINGNEWS Subscribers<BREAKINGNEWS-Subscrib…
1     spam                       <theorize@plg.uwaterloo.ca>
2     spam              "Theorize" <theorize@plg.uwaterloo.ca>
3     spam                 warwickktwarwic@speedy.uwaterloo.ca
4      ham                            R-help@stat.math.ethz.ch

                                        from_address  \
0           BREAKING NEWS<breakingnews@foxnews.com>
1                        "cschai" <cschai@syhmco.co.kr>
2  "Aegis Capital Group LLC" <Estela.Burch@smapxs…
3      "shar Nobis" <sharNobis@autotradebuyer.co.uk>
4                           jessica.gervais@tudor.lu

                                             subject  \
0                                          FNC Alert
```

```
1                                                  rtfmub
2  Invitation to fill in the vacant position of a…
3                             Terrific gains possible!
4                            [R] time serie generation

                                                     body
0  PELOSI, REID SIGN WAR-SPENDING BILL THAT INCLU…
1  \n\n\n\n\n\n\n\nwyat\nlnpmoqrkhapibcegd\n\n\n\…
2  \n\n\n\nDear    sirs,\nAegis      Capital  Gro…
3  http://s6.bilder-hosting.de/img/7LR4W.jpg\nImp…
4  \nDear all,\n\nI would like to generate a regu…
```

[3]:
```
df.category = df.category.astype('category')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63542 entries, 0 to 63541
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   category      63542 non-null  category
 1   to_address    63141 non-null  object
 2   from_address  63542 non-null  object
 3   subject       63410 non-null  object
 4   body          63542 non-null  object
dtypes: category(1), object(4)
memory usage: 2.0+ MB
```

[4]:
```
# from sklearn.model_selection import train_test_split

# data = df.drop(['category'], axis=1)
# labels = df['category']

# train, test, train_label, test_label = train_test_split(data, labels,␣
 ↪test_size=.15)


# train.to_json('./data/train_data.json')
# test.to_json('./data/test_data.json')
# train_label.to_json('./data/train_label.json')
# test_label.to_json('./data/test_label.json')

train = pd.read_json('./data/train_data.json')
test = pd.read_json('./data/test_data.json')
train_label = pd.read_json('./data/train_label.json', typ='series')
test_label = pd.read_json('./data/test_label.json', typ='series')
```

[5]:
```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(binary=True, min_df=10)
train_feat_mat = vectorizer.fit_transform(train.body)

test_feat_mat = vectorizer.transform(test.body)

print(train_feat_mat.shape)
print(test_feat_mat.shape)
```

```
(54010, 29326)
(9532, 29326)
```

1.2 Part 2 - Clustering the Emails

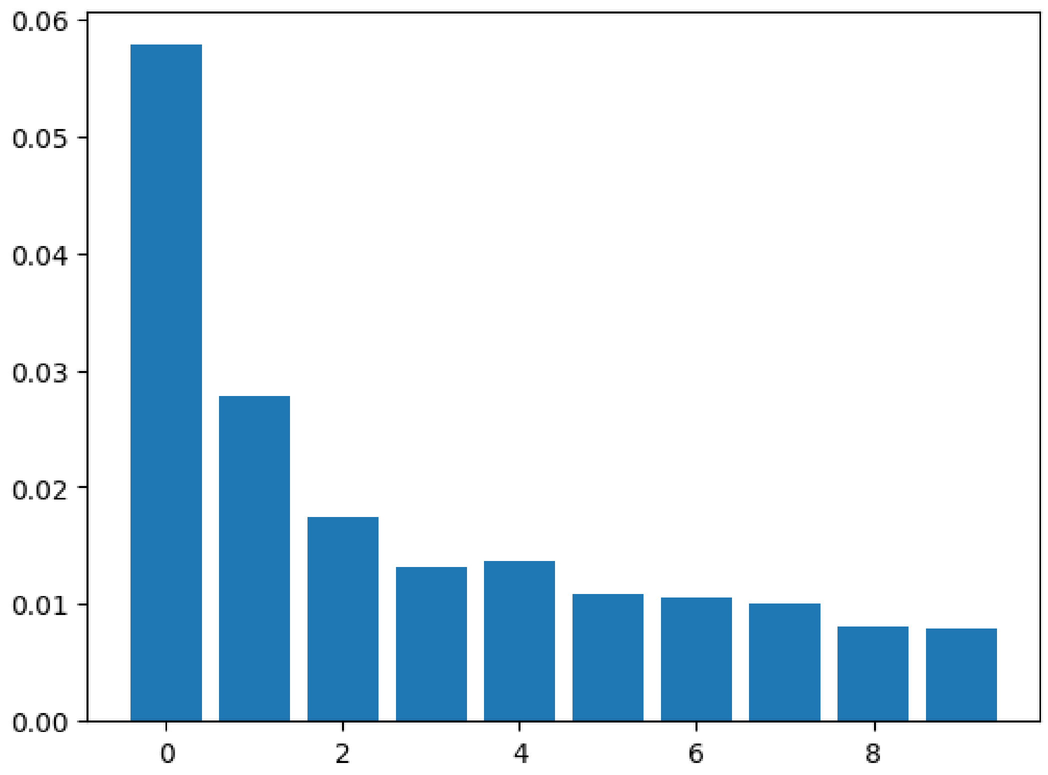
I chose KMeans, as the clusters should be linearly separable. Knowing how the data looks from the previous lab, the density of the data, as well as the mixture of ham vs spam, seems to change throughout the space, so I'll keep the number of clusters high.

We'll run SVD dimension reduction to bring the columns down to 10, then cluster the data and make a map from cluster to label for predictions:

[6]:
```
from sklearn.decomposition import TruncatedSVD

condenser = TruncatedSVD(n_components=10)
train_condensed_mat = condenser.fit_transform(train_feat_mat)

test_condensed_mat = condenser.transform(test_feat_mat)

print(condenser.explained_variance_ratio_)
plt.bar(np.arange(condenser.explained_variance_ratio_.size), condenser.
  ↪explained_variance_ratio_)
```

```
[0.05780993 0.02786145 0.01750974 0.01319702 0.01363418 0.01094181
 0.01059009 0.01014235 0.00814024 0.0079912 ]
```

[6]:
```
<BarContainer object of 10 artists>
```



[104]:
```
from sklearn.cluster import KMeans

clusterer = KMeans(n_clusters=20)

train_clusters = clusterer.fit_predict(train_condensed_mat[:, 0:2])

test_clusters = clusterer.predict(test_condensed_mat[:, 0:2])
```

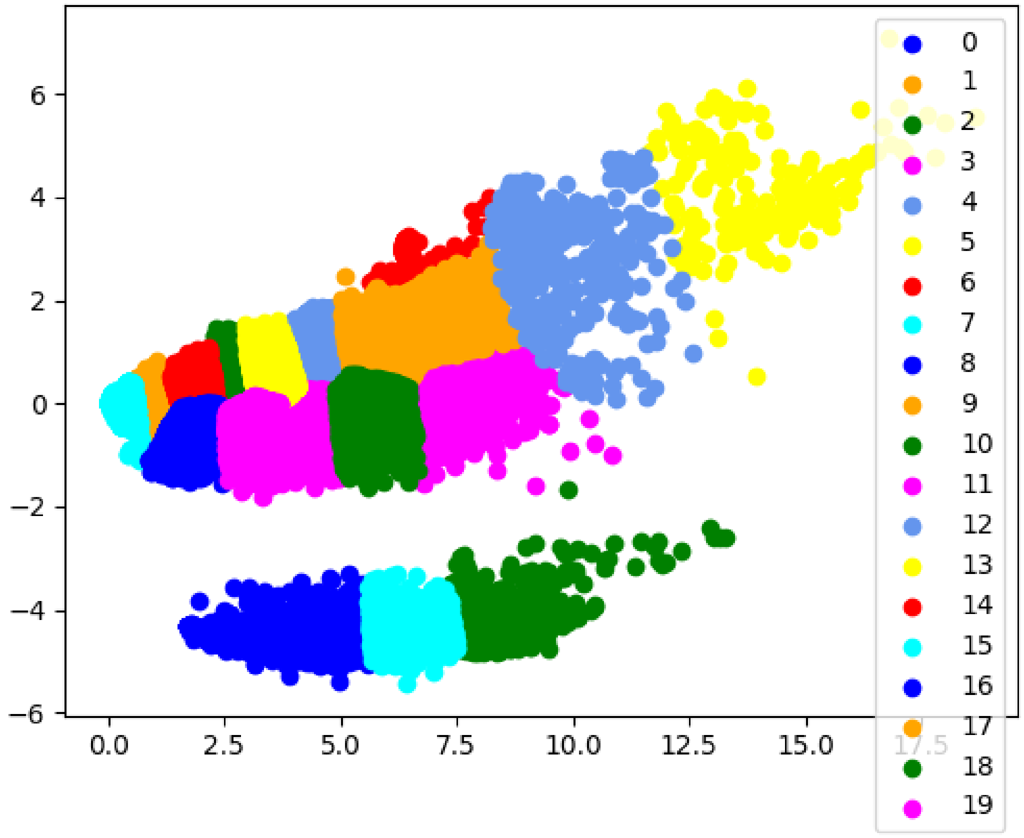
```
/home/rolo/.local/lib/python3.8/site-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

[105]:
```
np.unique(train_clusters)
```

[105]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19], dtype=int32)
```

[109]:
```
colors = {0: 'blue', 1: 'orange', 2: 'green', 3: 'magenta', 4:␣
 ↪'cornflowerblue', 5: 'yellow', 6: 'red', 7: 'aqua'}

fig, ax = plt.subplots()
```

```
for cluster_num in np.unique(train_clusters):
    indices = np.where(train_clusters == cluster_num)
    ax.scatter(train_condensed_mat[indices, 0], train_condensed_mat[indices,␣
 ↪1], label=cluster_num, c=colors[cluster_num % len(colors)])
ax.legend()
plt.show()
```



We'll check the counts of ham vs spam in each of the clusters:

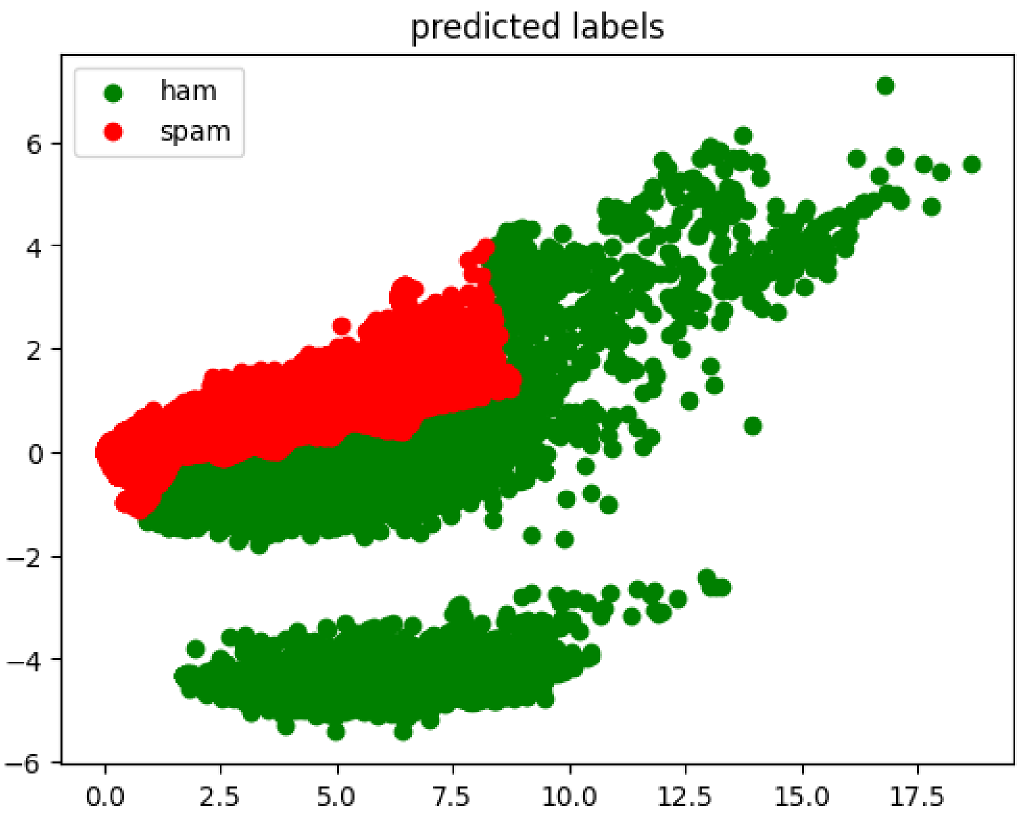
[110]:
```
total = train_label.shape[0]
ham_num = (train_label == 'ham').sum()
format_percent = '%.2f'%(100 * ham_num / total)
print(f'total: {total} - ham: {ham_num} - {format_percent}%\n')

cluster_hams = {}
for clust in np.unique(train_clusters):
    total = train_clusters[train_clusters == clust].shape[0]
    ham_num = train[(train_clusters == clust) & (train_label == 'ham')].shape[0]

    dec_ham = ham_num / total
    cluster_hams[clust] = dec_ham

    format_percent = '%.2f'%(100 * dec_ham)
    print(f'cluster {clust}: {total} - ham: {ham_num} - {format_percent}%')


print(f'\n{cluster_hams}')

clust_label_map = {}
for clust, percent_ham in cluster_hams.items():
    clust_label_map[clust] = 'ham' if percent_ham > 0.5 else 'spam'

print(f'\n{clust_label_map}')

def cluster_predict(condensed_mat):
    clusters = clusterer.predict(condensed_mat)

    mapped = np.vectorize(lambda cluster_num: clust_label_map.
 ↪get(cluster_num))(clusters)

    return mapped

train_predicted = cluster_predict(train_condensed_mat[:, 0:2])
train_predicted
```

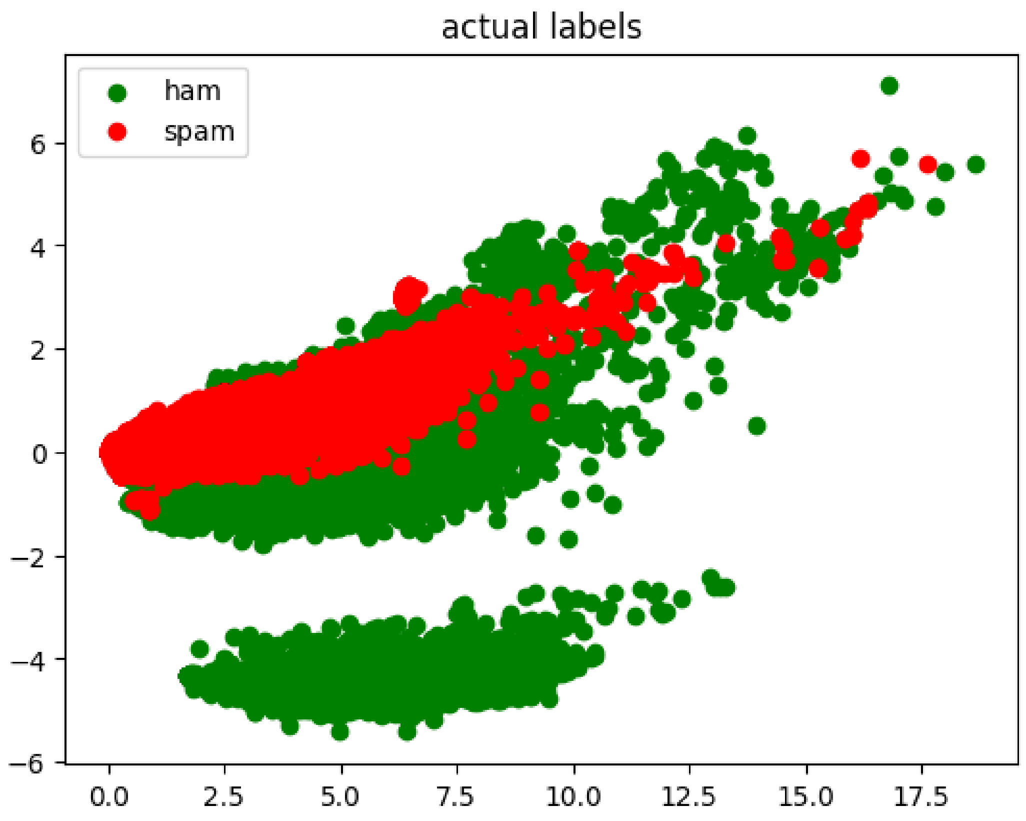
```
total: 54010 - ham: 19581 - 36.25%

cluster 0: 1857 - ham: 1857 - 100.00%
cluster 1: 5814 - ham: 986 - 16.96%
cluster 2: 5567 - ham: 826 - 14.84%
cluster 3: 706 - ham: 694 - 98.30%
cluster 4: 3673 - ham: 542 - 14.76%
cluster 5: 216 - ham: 186 - 86.11%
cluster 6: 753 - ham: 73 - 9.69%
cluster 7: 8343 - ham: 1423 - 17.06%
cluster 8: 1342 - ham: 1342 - 100.00%
cluster 9: 1670 - ham: 326 - 19.52%
cluster 10: 441 - ham: 441 - 100.00%
cluster 11: 2077 - ham: 1979 - 95.28%
cluster 12: 375 - ham: 297 - 79.20%
cluster 13: 5745 - ham: 756 - 13.16%
cluster 14: 5568 - ham: 516 - 9.27%
cluster 15: 1293 - ham: 1293 - 100.00%
cluster 16: 1993 - ham: 1840 - 92.32%
cluster 17: 2749 - ham: 605 - 22.01%
cluster 18: 1500 - ham: 1385 - 92.33%
cluster 19: 2328 - ham: 2214 - 95.10%
```

```
{0: 1.0, 1: 0.1695906432748538, 2: 0.14837434884138676, 3: 0.9830028328611898,
4: 0.14756329975496868, 5: 0.8611111111111112, 6: 0.09694555112881806, 7:
0.17056214790842622, 8: 1.0, 9: 0.19520958083832335, 10: 1.0, 11:
0.9528165623495426, 12: 0.792, 13: 0.13159268929503917, 14: 0.09267241379310345,
15: 1.0, 16: 0.9232313095835424, 17: 0.22008002910149146, 18:
0.9233333333333333, 19: 0.9510309278350515}

{0: 'ham', 1: 'spam', 2: 'spam', 3: 'ham', 4: 'spam', 5: 'ham', 6: 'spam', 7:
'spam', 8: 'ham', 9: 'spam', 10: 'ham', 11: 'ham', 12: 'ham', 13: 'spam', 14:
'spam', 15: 'ham', 16: 'ham', 17: 'spam', 18: 'ham', 19: 'ham'}
```

```
[110]: array(['spam', 'ham', 'spam', …, 'spam', 'spam', 'ham'], dtype='<U4')
```

```
[111]: colors = {'ham': 'green', 'spam': 'red'}

fig, ax = plt.subplots()
for label in np.unique(train_label):
    indices = np.where(train_predicted == label)
    ax.scatter(train_condensed_mat[indices, 0], train_condensed_mat[indices,␣
 ↪1], label=label, c=colors[label])
ax.set_title('predicted labels')
ax.legend()
plt.show()

fig, ax = plt.subplots()
for label in np.unique(train_label):
    indices = np.where(train_label == label)
    ax.scatter(train_condensed_mat[indices, 0], train_condensed_mat[indices,␣
 ↪1], label=label, c=colors[label])
ax.set_title('actual labels')
ax.legend()
plt.show()
```

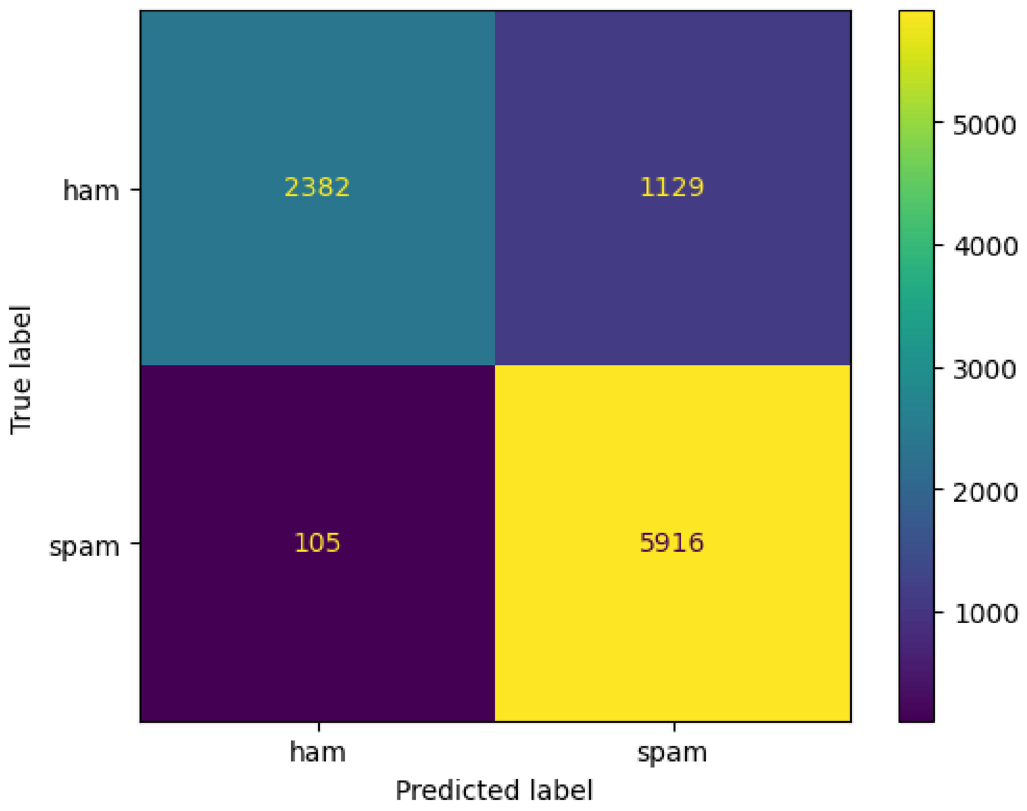

predicted labels
ham
spam
6
4
2
0
−2
−4
−6
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5



With 20 clusters, the algorithm was able to generalize the area in which most spam emails occur. Now we'll predict on the test data and make a confusion matrix based on its results.

[121]:
```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

test_cluster_preds = cluster_predict(test_condensed_mat[:, 0:2])
print(test_cluster_preds.shape[0])
print((test_cluster_preds == 'ham').sum())

cm = confusion_matrix(test_label, test_cluster_preds)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['ham',␣
 ↪'spam'])

disp.plot()
plt.show()
```

```
9532
2487
```



```
[124]: print(f'{(2382 + 5916)} out of {(2382 + 5916 + 1129 + 105)} predicted␣
        ↪correctly')
```

```
8298 out of 9532 predicted correctly
```

It looks like the clustering performed well for the test set.

1.3 Part 3 - Calculating Document Frequencies of Words

```
[129]: ham_mat = train_feat_mat[train_predicted == 'ham']
       spam_mat = train_feat_mat[train_predicted == 'spam']

       from scipy.sparse import csc_matrix
       ham_mat = csc_matrix(ham_mat)
       spam_mat = csc_matrix(spam_mat)
```

```
[134]: ham_word_cnts = ham_mat.sum(axis=0)
       print(ham_word_cnts.shape)
       spam_word_cnts = spam_mat.sum(axis=0)
       print(spam_word_cnts.shape)
```

```
(1, 29326)
```

```
(1, 29326)
```

[149]:
```
def print_freqs(word):
    word_ind = vectorizer.vocabulary_[word]
    freq = ham_word_cnts[0, word_ind]
    total = ham_mat.shape[0]
    format_percent = '%.2f'%(100 * freq / total)
    print(f'{word}:\n  ham: {freq} - appears in {format_percent}% of␣
 ↪ham-predicted documents')

    freq = spam_word_cnts[0, word_ind]
    total = spam_mat.shape[0]
    format_percent = '%.2f'%(100 * freq / total)
    print(f'  spam: {freq} - appears in {format_percent}% of spam-predicted␣
 ↪documents\n')
```

[150]:
```
print_freqs('love')
print_freqs('works')
print_freqs('different')
```

```
love:
  ham: 355 - appears in 2.51% of ham-predicted documents
  spam: 1365 - appears in 3.42% of spam-predicted documents

works:
  ham: 1407 - appears in 9.96% of ham-predicted documents
  spam: 1179 - appears in 2.96% of spam-predicted documents

different:
  ham: 1497 - appears in 10.60% of ham-predicted documents
  spam: 936 - appears in 2.35% of spam-predicted documents
```

1.4 Part 4 - Find Enriched Words with Statistical Testing

[222]:
```
import scipy.stats as stats

enriched_in_0 = np.vectorize(lambda spam_word_cnt, ham_word_cnt: stats.
 ↪binom_test(ham_word_cnt, ham_mat.shape[0], (spam_word_cnt / spam_mat.
 ↪shape[0]), alternative='greater'))(spam_word_cnts, ham_word_cnts)
enriched_in_1 = np.vectorize(lambda spam_word_cnt, ham_word_cnt: stats.
 ↪binom_test(spam_word_cnt, spam_mat.shape[0], (ham_word_cnt / ham_mat.
 ↪shape[0]), alternative='greater'))(spam_word_cnts, ham_word_cnts)
```

[224]:
```
print(enriched_in_0.shape)
print(enriched_in_1.shape)
```

```
(1, 29326)
```

```
(1, 29326)
```

[228]:
```
word = 'works'
ind = vectorizer.vocabulary_[word]
print(f'{word}: \nin 0: {enriched_in_0[0, ind]}')
print(f'in 1: {enriched_in_1[0, ind]}')
print()

word = 'love'
ind = vectorizer.vocabulary_[word]
print(f'{word}: \nin 0: {enriched_in_0[0, ind]}')
print(f'in 1: {enriched_in_1[0, ind]}')
print()
```

```
works:
in 0: 0.0
in 1: 0.9999999999999999

love:
in 0: 0.9999999997999472
in 1: 1.8293020464383167e-28
```

Works is enriched in 0, and love is enriched in 1. So, works is more often in ham, and love is more often in spam.

[259]:
```
words_enriched_in_0 = []
words_enriched_in_1 = []
for word, index in vectorizer.vocabulary_.items():
    if(word.isalpha()):
        p_0 = enriched_in_0[0, index]
        p_1 = enriched_in_1[0, index]
        ham_cnt = ham_word_cnts[0, index]
        spam_cnt = spam_word_cnts[0, index]
        words_enriched_in_0.append((p_0, word, ham_cnt))
        words_enriched_in_1.append((p_1, word, spam_cnt))
```

[256]:
```
print('enriched in 0:')
for word in sorted(words_enriched_in_0, key=lambda x: x[0])[0:200]:
    print(word[1])
```

```
enriched in 0:
hello
new
and
http
www
with
```

```
need
for
any
this
to
try
be
me
in
the
kernel
module
so
can
use
my
dectalk
linux
running
that
from
am
there
would
have
some
something
help
just
files
ve
using
working
want
could
of
same
thanks
do
mail
speakup
mailing
list
braille
uwo
speech
mailman
listinfo
```

```
which
or
not
please
does
system
also
get
it
at
if
code
is
original
message
bounces
stat
math
ethz
ch
mailto
on
sent
subject
question
hi
context
statistics
other
example
as
mean
read
but
alternative
html
version
deleted
https
posting
guide
project
org
provide
commented
minimal
self
contained
```

```
reproducible
an
when
plot
format
way
wed
may
wrote
how
library
matrix
pairwise
dataset
output
each
function
following
like
problem
file
advance
university
brian
ripley
stats
ox
professor
applied
oxford
tel
fax
written
cc
cheers
lists
ibiblio
where
make
used
possible
root
rev
work
works
learn
what
error
```

```
unsubscribe
view
source
two
user
os
data
first
trouble
pl
prostoalex
writes
looks
instead
then
method
additional
methods
moskalyuk
binary
etc
seems
isn
archive
computerworld
least
maybe
end
command
viewarticlebasic
mises
functions
argument
tid
id
begin
since
latimes
done
signed
true
seem
add
skype
think
doesn
pcworld
gam
```

```
run
type
```

[257]:
```
print('enriched in 1:')
for word in sorted(words_enriched_in_1, key=lambda x: x[0])[0:200]:
    print(word[1])
```

```
enriched in 1:
visit
our
online
shipping
worldwide
lorena
your
loan
refinance
credit
ready
lenders
established
lowest
payments
thay
unhappy
lover
safest
pnis
anywhere
millions
men
enhan
ement
hes
atches
deliver
product
unreal
dis
ounts
pybal
hk
his
he
producttestpanel
speedy
uwaterloo
sternshirt
```

```
buy
high
quality
price
us
photoshop
canadian
sterling
thrive
industrious
fasten
delivery
customer
day
instructed
money
gram
instruct
literacy
sociability
expenses
chargers
traveling
vacancy
sydneycarcentre
vacancies
glad
yours
rcs
howstuffworks
prices
suite
click
younger
low
news
instant
premiere
safe
fast
alerts
chose
discounts
meds
oem
packing
booklets
macromedia
```

adobe
acrobat
corel
grafix
illustrator
ableton
gigabook
borland
gervasio
her
centrecar
viagra
erection
sex
life
suffer
difficulties
maintaining
erectile
dysfunction
lotteryagent
lotto
renown
powerballmegamillionseuromillions
lotterylotto
ennis
wakes
fiv
poo
shook
choky
mornin
arched
sheep
tent
lagging
commutin
aint
wouldnt
bein
mrs
solemnly
him
bled
danglars
thoughtfully
viscount
corporal

itch
strod
dine
caderousse
furnish
whine
uttered
tasteless
wove
repulsive
beauchamp
bovine
pasha
haide
noirtier
god
grieving
she
procession
strove
dodo
nursery
throng
pills
itemyour
soft
tabs
cialis
jelly
levitra
startet
firma
preis
wkn
isin
markt
sie
rallye
ist
alert
alerting
aqui
loss
blurted
macbain
drained
yourself
admirrante

```
fda
pharmacies
pharmacy
professional
soma
growth
hormone
meridia
tramadol
herbalking
herbal
penis
enlargement
tutorials
ineffective
pumps
```

1.5 Reflection Questions

1. There are many words in the English language that are necessary for basic sentence sturcture, like 'the', 'and', 'a'… that are sure to appear in most emails, regardless of if it's ham or spam. The separate cluster consists of words that were not seen in spam at all, potentially being terms directly related to specified work topics, maybe people's names, or just overall great vocabulary not usually seen in foreign-written spam emails.
2. In my clustering, there is both ham and spam in most of the clusters. I tried to achieve a tighter fitting clustering to the data. There were a few clusters which covered the bottom section of points, which only contained ham messages. The mapping could be changed to make sure that ONLY these clusters are mapped to ham.
3. The ham messages contained a lot of 'advertisey' language, trying to make whatever they're talking about seem too good to be true. The ham messages seem to be more topical, and sometimes contain typos.

[261]:
```
train[train_predicted == 'ham'][0:25]
```

[261]:
```
                                              to_address  \
42791  "Speakup is a. screen review system for Linux…
34315  "Y G" <gatemaze@gmail.com>, <r-help@stat.math…
54248       Felicity Jones <felicity.jones@stanford.edu>
35193                         cc-community@lists.ibiblio.org
37793                            avcooper@flax9.uwaterloo.ca
32395                               r-help@stat.math.ethz.ch
56651         "Christopher R. Hertel" <crh@ubiqx.mn.org>
30652                               perl6-internals@perl.org
49556            Chabot Denis <chabotd@globetrotter.net>
434             Christoph Buser <buser@stat.math.ethz.ch>
61015  "Speakup is a screen review system for Linux."…
53029            "H. Paul Benton" <hpbenton@scripps.edu>
1066                   Ben Edwards <funkytwig@gmail.com>
```

```
44225                  James Peach <jpeach@samba.org>
21039                        r-help@stat.math.ethz.ch
62338                          msn-list@te.verweg.com
2380       "Sundar Dorai-Raj" <sundar.dorai-raj@pdf.com>
33977                             samba-cvs@samba.org
4286   "Speakup is a screen review system for Linux."…
28368  ted.harding@manchester.ac.uk, r-help@stat.math…
25997            bugs-bitbucket@netlabs.develooper.com
13420                        r-help@stat.math.ethz.ch
44333          "Darren Duncan" <darren@darrenduncan.net>
62197                  cc-community@lists.ibiblio.org
21450                 samba-technical@lists.samba.org

                                          from_address  \
42791           Radical NetSurfer <radsurfer@yahoo.com>
34315  "Leeds, Mark \(IED\)" <Mark.Leeds@morganstanle…
54248          Prof Brian Ripley <ripley@stats.ox.ac.uk>
35193             Andy Kaplan-Myrth <techlaw@uottawa.ca>
37793                             slashdot@slashdot.org
32395       (Ted Harding) <ted.harding@nessie.mcc.ac.uk>
56651                     Jeremy Allison <jra@samba.org>
30652                        allison@cvs.develooper.com
49556          Prof Brian Ripley <ripley@stats.ox.ac.uk>
434       "Daniel Tahin" <e0226781@student.tuwien.ac.at>
61015                     Gaijin <gaijin@clearwire.net>
53029                 "jim holtman" <jholtman@gmail.com>
1066                 Srinivas <srini@geekcrossing.net>
44225          "Gerald (Jerry) Carter" <jerry@samba.org>
21039                  Leonardo Lami <lami@faunalia.it>
62338                    msn-list-request@te.verweg.com
2380       "Deepayan Sarkar" <deepayan.sarkar@gmail.com>
33977                               abartlet@samba.org
4286         "Littlefield, tyler" <compgeek13@gmail.com>
28368              Stephen Tucker <brown_emu@yahoo.com>
25997  Steve Peters (via RT) <parrotbug-followup@parr…
13420                 francogrex <francogrex@mail.com>
44333             "Jonathan Lang" <dataweaver@gmail.com>
62197       Terry Hancock <hancock@anansispaceworks.com>
21450                     Jeremy Allison <jra@samba.org>

                                               subject  \
42791                     Help with dtpc.o for Centos 5
34315           Re: [R] general question about use of list
54248              Re: [R] sizing and saving graphics in R
35193   [cc-community] Podcasting Legal Guide for Canada
37793                    [Slashdot] Stories for 2007-05-23
32395                         Re: [R] to draw a smooth arc
```

```
56651                   Re: Preserving NTFS permissions.
30652          [svn:parrot-pdd] r18213 - trunk/docs/pdds
49556  Re: [R] Reducing the size of pdf graphics file…
434        Re: [R] Preconditions for a variance analysis
61015                Re: gentoo dropping speakup support
53029                  Re: [R] data type for block data?
1066        Re: Passing multiple mixed arguments to subs
44225  Re: svn commit: samba r23689 - in branches/SAM…
21039                                     [R] select row
62338                   MSN-list Digest, Vol 7, Issue 90
2380                      Re: [R] Positioning in xyplot
33977  svn commit: samba r23695 - in branches/SAMBA_4…
4286                   Re: Slackware 11 aliases, anyone?
28368      Re: [R] Tools For Preparing Data For Analysis
25997             [perl #43033] [PATCH] Silence warning
13420             [R] Comparing MCMClogit, glm and BRUGS
44333  Re: [svn:perl6-synopsis] r14401 - doc/trunk/de…
62197  Re: [cc-community] Strayform - where artists a…
21450  Re: svn commit: samba r23691 - in branches:\n\…

                                                   body
42791  One reason I rejoined this group was to try to…
34315  it's also not unbiased.\n\n\n\n-----Original M…
54248  Why not plot directly to a bitmapped format, u…
35193  When Creative Commons published their Podcasti…
37793  ===============================================…
32395  This thread prompts me to ask about something …
56651  On Wed, Jun 20, 2007 at 11:44:21AM -0500, Chri…
30652  Author: allison\nDate: Sat Apr 14 17:06:35 200…
49556  >From the help page\n\n      'pdf' writes unco…
434    Thanx for your answer. I don't have the book, …
61015  Spie Sutherland wrote:\n> There seems to be se…
53029  This will create a list of the matrix subsets:…
1066   Hi Ben,\n\nYou can use shift for this.\n\nsub …
44225  -----BEGIN PGP SIGNED MESSAGE-----\nHash: SHA1…
21039  Hi all,\nI have a little problem selecting som…
62338  Send MSN-list mailing list submissions to\n\tm…
2380   On 4/11/07, Sundar Dorai-Raj  wrote:\n\n> Seem…
33977  Author: abartlet\nDate: 2007-07-04 03:25:44 +0…
4286   what does sendmail have to do with the aliases…
28368  Embarrasingly, I don't know awk or sed but R's…
25997  # New Ticket Created by  Steve Peters \n# Plea…
13420  \nHello,\nI have two "related" questions, one …
44333  Darren Duncan wrote:\n> Jonathan Lang wrote:\n…
62197  drew Roberts wrote:\n> On Friday 08 June 2007 …
21450  On Tue, Jul 03, 2007 at 11:34:02PM +0000, idra…
```

```
[262]: train[train_predicted == 'spam'][0:25]
```

```
[262]:                                             to_address  \
       7375                         theorize@plg.uwaterloo.ca
       22780                        theorize@plg.uwaterloo.ca
       63457             "theorize" <theorize@plg.uwaterloo.ca>
       26257  "Subscriber" <producttestpanel@speedy.uwaterlo…
       58300                     the00@plg2.math.uwaterloo.ca
       41544                producttestpanel@flax9.uwaterloo.ca
       1519            "Gnitpick" <gnitpick@flax9.uwaterloo.ca>
       32192                     <ktwarwic@flax9.uwaterloo.ca>
       7250                        <theorize@plg.uwaterloo.ca>
       13006                     <gnitpick@flax9.uwaterloo.ca>
       8258                       <manager@flax9.uwaterloo.ca>
       42460             "The00" <the00@plg2.math.uwaterloo.ca>
       1046          "Subscriber" <cruiseca@flax9.uwaterloo.ca>
       5802                        <mail@speedy.uwaterloo.ca>
       59057                     <smiles@speedy.uwaterloo.ca>
       59295                                  1@bellsouth.net
       24453         "Kami Crawford" <mail@speedy.uwaterloo.ca>
       31894                        theorize@plg.uwaterloo.ca
       29012                          mail@flax9.uwaterloo.ca
       6031                          mail@speedy.uwaterloo.ca
       23167                      ktwarwic@flax9.uwaterloo.ca
       10491                  <debian-legal@lists.debian.org>
       6222                       ktwarwic@flax9.uwaterloo.ca
       1981                     <the00@plg2.math.uwaterloo.ca>
       493                        gnitpick@flax9.uwaterloo.ca

                                                 from_address  \
       7375              "Darla Blanco" <p4L4MLJ@fanciers.com>
       22780          "Most languages" <stzezxh@ehotelier.com>
       63457           "Fabian Jewell" <tiqannada@hostune.com>
       26257  "Shipment Notice #HL-225825" <Jenny@strongimpr…
       58300    "Kimberley Mora" <the00@plg2.math.uwaterloo.ca>
       41544  CollegePlanner <CollegePlanner@massiveservices…
       1519             "Sydney Car Centre" <qaubm@altern.org>
       32192     "HowStuffWorks" <newsletter@howstuffworks.com>
       7250            "Guadalupe Rhodes" <ifuuow@boardiq.com>
       13006  "Zachary Adams" <Tristan.Richardson@adultactio…
       8258            "Alexander" <richard@prostateforum.biz>
       42460                        "SCC" <jjnvi@tenchiclub.com>
       1046   "Healthcare Degrees Online" <qq5ci9zmi@daisyha…
       5802              "Janine Lockett" <jpil@boston-ed.com>
       59057  "Zachary Brown" <automacdev.com@lottozubotto.com>
       59295          UK ONLINE PROMO <r_hoaglund@bellsouth.net>
       24453     "Selene Harper" <macbrydejtfhi@163data.com.cn>
```

```
31894                   fought by <elawsuit@cccfs.org>
29012                     Federico <ter53@icqmail.com>
6031             "gambling" <pmqigufgumq@verizon.net>
23167     "CNNMoney.com Alerts" <cnnalerts@mail.cnn.com>
10491                        <boletim@ecolatina.com.br>
6222         "therein Dolan" <largesseLeach@ac9hold.com>
1981       "Trevor Wood" <qsolo.com@usedcompanycars.com>
493     "Roe, Darrin" <Scruggs9S@yourpositiveenergy.com>

                                               subject  \
7375                    diagnoses malawi commonality__
22780                     Loan for a low month payment
63457                          you afton go imlaystown
26257  Brand Samples shipping to your area =?UTF8?Q?=…
58300             Domingo Buy your loved one Glashutte 
41544  adf, 10k scholarship giveaways - next drawing …
1519   Account managers vacant position in the Sydney…
32192    HowStuffWorks Lifestyle Newsletter May 10, 2007
7250            Global sale before new project opening!
13006                          Zachary Adams   get it!
8258                          Need medicine? All here!
42460    job - just for you. [letter id: KU32860911070A]
1046                    Find the right program for you
5802                          Janine - Viagra for you!
59057   Lotto Tickets from 50 countries around the world
59295       NOTIFICATION…MAIL
24453                                       Be careful
31894                                       Be no pest
29012                          look-out the price list
6031                          DIE RALLYE IS GESTARTET!
23167            Lockheed employs D.C.'s last honest man
10491  =?iso-8859-1?Q?=5BNS=5D_E-mail_Ecolatina_-_Tra…
6222                    Go out with a new babe tonight
1981                              Beware of fake pills
493              solid stainless steel replicas PCXAW68

                                                  body
7375   Hello,\n\nVisit our new online store and save…
22780  \n\n\n\n\n\nThank you for your loan request, w…
63457  Does Size Matter'?\n_____\n\n60% of WOMEN said…
26257  ttp://www.lynxtrack.com/afclick.php?o=4343&b=d…
58300  \nHighest qualities Replic Watches now !\n\n\n…
41544  \n\n251839499     FAIR USE of Philadelphia Cit…
1519   \n\n\n\nWhile       we     may  have    high …
32192  \nhowstuffworks® | lifestyle May 10, 2007\nRSS…
7250   \n\n\n\n\n\n\n\nDear valued member.\n\nExtra d…
13006  OEM software means no CD/DVD, no packing case,…
```

```
8258                                    \n\n\n\n\n\n\n\n
42460  \n\n\n\nWhile we  may   have       high   expe…
1046   \n\n\n <!--U.S. military officials said they h…
5802   \n\n\n\n\n\n\nVIAGRA\nIf you have a problem ge…
59057  \n\n\n\n\n\n\nLotteryagent (TM) is the only on…
59295  \nREFERENCE NUMBER:UK/839030X2/14\nThis Email …
24453  \n\n\n\n\n\n\n\n\n\nWe were cystic working num…
31894  \n\n\n\n\n\n\nlooking anxiously about her.  Oh…
29012  \n\n\n\nhttp://carepicture.hk\n\n\nViagra\n10 …
6031   \n\n\n\n\n\nAN ALLE FINANZINVESTOREN!\nDIESE A…
23167  \n\n\n\n\n\n\n\nThe Internet home of:\n\n\n\n\…
10491  \n\n\n\nUntitled Document\n\n\n\n\n\n\n\n\n\n\…
6222
1981   \nHi\n\n\n\n\n\n\nIm very sleepy, she blurted ou…
493    \n\nRolex Replica order! Tracking #EI59 - 3481…
```

4. and
5. It looks like the R newsletter was more the in the ham category, and the college emails were much more spammy. We can tell these apart by the email addresses, containing ‘uwaterloo’ in very many of the spam emails.