Cpts_575: Data Science

Assignment V

10/28/2019

Instructor: Assefaw Gebremedhin

Student: Yu-Chieh Wang 11641327

1. Logistic regression

$$\sigma(f(x)) = \frac{e^{f(x)}}{1 + e^{f(x)}}$$

$$f(x) = -7 + 0.1(x_1) + 1(x_2) - 0.04(x_3)$$

a. $f(x) = -7 + 3.2 + 3 - 0.48 = -1.28$

$$\sigma(-1.28) = \frac{e^{-1.28}}{1 + e^{-1.28}} = 0.21755$$

b. $e^{f(x)} = 0.5(1 + e^{f(x)})$

$e^{f(x)} = 1$

$f(x) = 0 = -7 + 0.1(x_1) + 3 - 0.48$

$x_1 = 44.8$

c. $e^{f(x)} = 0.5(1 + e^{f(x)})$

$e^{f(x)} = 1$

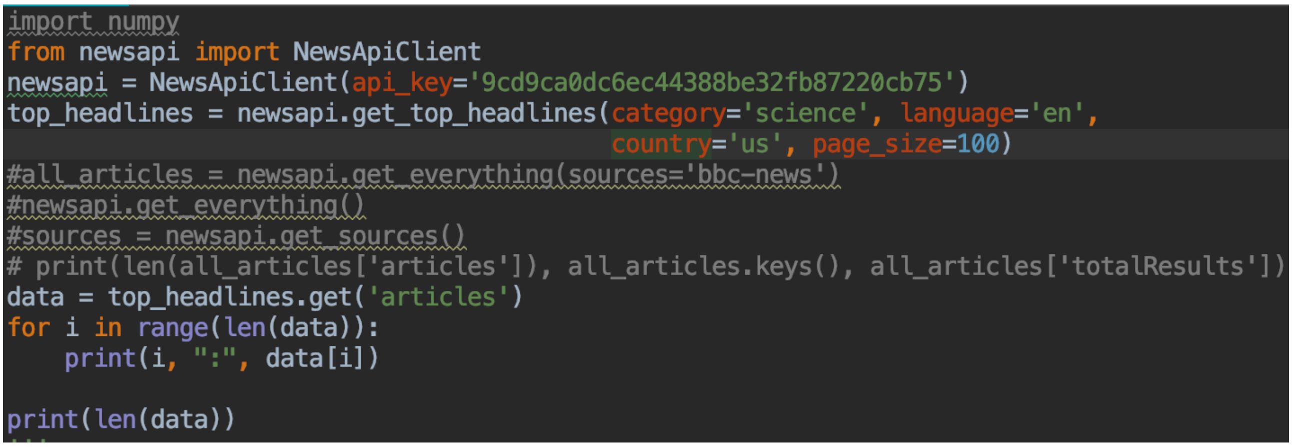
$f(x) = 0 = -7 + 0.1(x_1) + 3 - 0.12$

$x_1 = 41.2$

	X1	X2	X3	Pi(f(x))
a	32	3	12	0.21755
b	44.8	3	12	0.5
C	41.2	3	3	0.5

2. Naive Bayes Classification

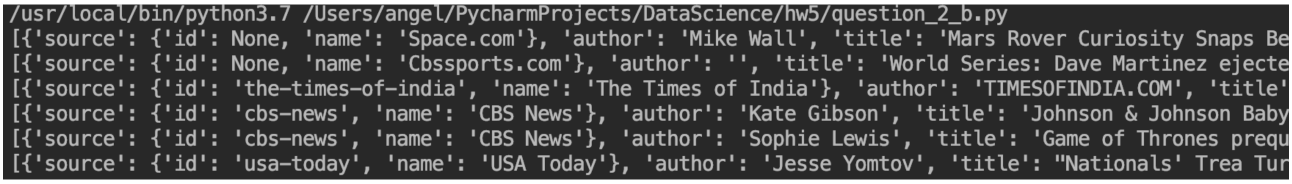
a. Data collection



```
import numpy
from newsapi import NewsApiClient
newsapi = NewsApiClient(api_key='9cd9ca0dc6ec44388be32fb87220cb75')
top_headlines = newsapi.get_top_headlines(category='science', language='en',
                                          country='us', page_size=100)
#all_articles = newsapi.get_everything(sources='bbc-news')
#newsapi.get_everything()
#sources = newsapi.get_sources()
# print(len(all_articles['articles']), all_articles.keys(), all_articles['totalResults'])
data = top_headlines.get('articles')
for i in range(len(data)):
    print(i, ":", data[i])

print(len(data))
```

-Use newsapi to get news from the datasets.

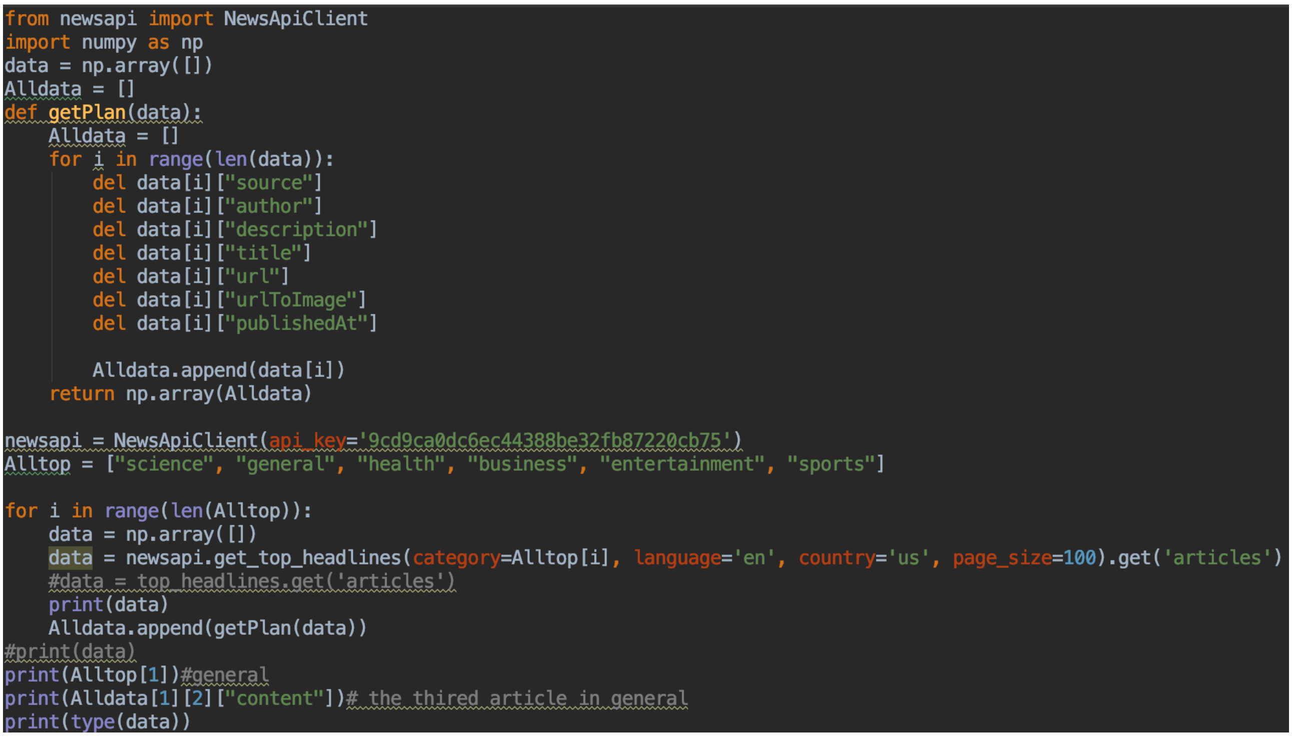
-The result shows as a dictionary:



```
/usr/local/bin/python3.7 /Users/angel/PycharmProjects/DataScience/hw5/question_2_b.py
[{'source': {'id': None, 'name': 'Space.com'}, 'author': 'Mike Wall', 'title': 'Mars Rover Curiosity Snaps Be
[{'source': {'id': None, 'name': 'Cbssports.com'}, 'author': '', 'title': 'World Series: Dave Martinez ejecte
[{'source': {'id': 'the-times-of-india', 'name': 'The Times of India'}, 'author': 'TIMESOFINDIA.COM', 'title'
[{'source': {'id': 'cbs-news', 'name': 'CBS News'}, 'author': 'Kate Gibson', 'title': 'Johnson & Johnson Baby
[{'source': {'id': 'cbs-news', 'name': 'CBS News'}, 'author': 'Sophie Lewis', 'title': 'Game of Thrones prequ
[{'source': {'id': 'usa-today', 'name': 'USA Today'}, 'author': 'Jesse Yomtov', 'title': "Nationals' Trea Tur
```

-The dictionary has many keys such as source, id, name, author, title, content, etc.

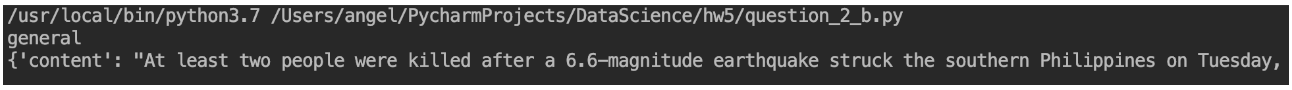
b. Data cleaning



```
from newsapi import NewsApiClient
import numpy as np
data = np.array([])
Alldata = []
def getPlan(data):
    Alldata = []
    for i in range(len(data)):
        del data[i]["source"]
        del data[i]["author"]
        del data[i]["description"]
        del data[i]["title"]
        del data[i]["url"]
        del data[i]["urlToImage"]
        del data[i]["publishedAt"]

        Alldata.append(data[i])
    return np.array(Alldata)

newsapi = NewsApiClient(api_key='9cd9ca0dc6ec44388be32fb87220cb75')
Alltop = ["science", "general", "health", "business", "entertainment", "sports"]

for i in range(len(Alltop)):
    data = np.array([])
    data = newsapi.get_top_headlines(category=Alltop[i], language='en', country='us', page_size=100).get('articles')
    #data = top_headlines.get('articles')
    print(data)
    Alldata.append(getPlan(data))
#print(data)
print(Alltop[1])#general
print(Alldata[1][2]["content"])# the thired article in general
print(type(data))
```

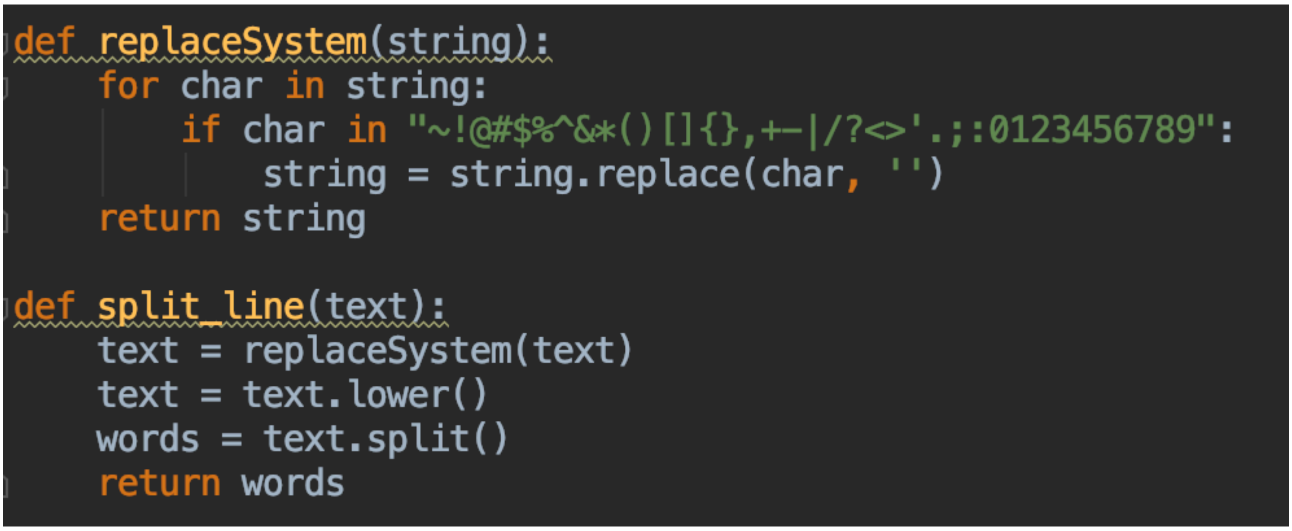
-Delete other keys which won't be used, and only leave the content.

-The output result is as following:



```
/usr/local/bin/python3.7 /Users/angel/PycharmProjects/DataScience/hw5/question_2_b.py
general
{'content': "At least two people were killed after a 6.6–magnitude earthquake struck the southern Philippines on Tuesday,
```

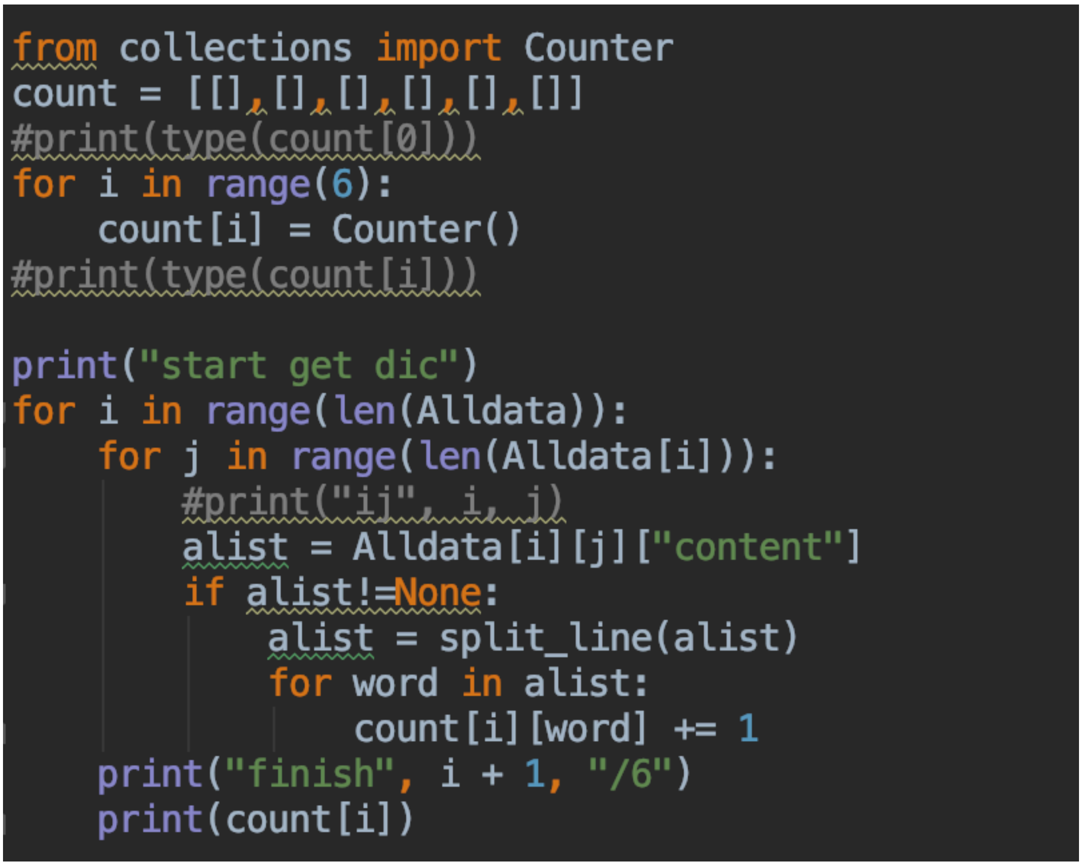
c. Tokenization



```
def replaceSystem(string):
    for char in string:
        if char in "~!@#$%^&*()[]{},+-|/?<>'.;:0123456789":
            string = string.replace(char, '')
    return string

def split_line(text):
    text = replaceSystem(text)
    text = text.lower()
    words = text.split()
    return words
```

-First, replace numbers and punctuation with blanks.

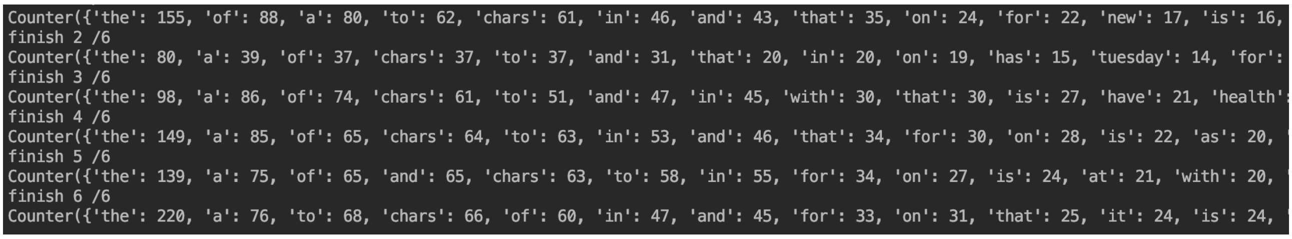
-Second, separate each word by blanks.



```
from collections import Counter
count = [[],[],[],[],[],[]]
#print(type(count[0]))
for i in range(6):
    count[i] = Counter()
#print(type(count[i]))

print("start get dic")
for i in range(len(Alldata)):
    for j in range(len(Alldata[i])):
        #print("ij", i, j)
        alist = Alldata[i][j]["content"]
        if alist!=None:
            alist = split_line(alist)
            for word in alist:
                count[i][word] += 1
    print("finish", i + 1, "/6")
    print(count[i])
```

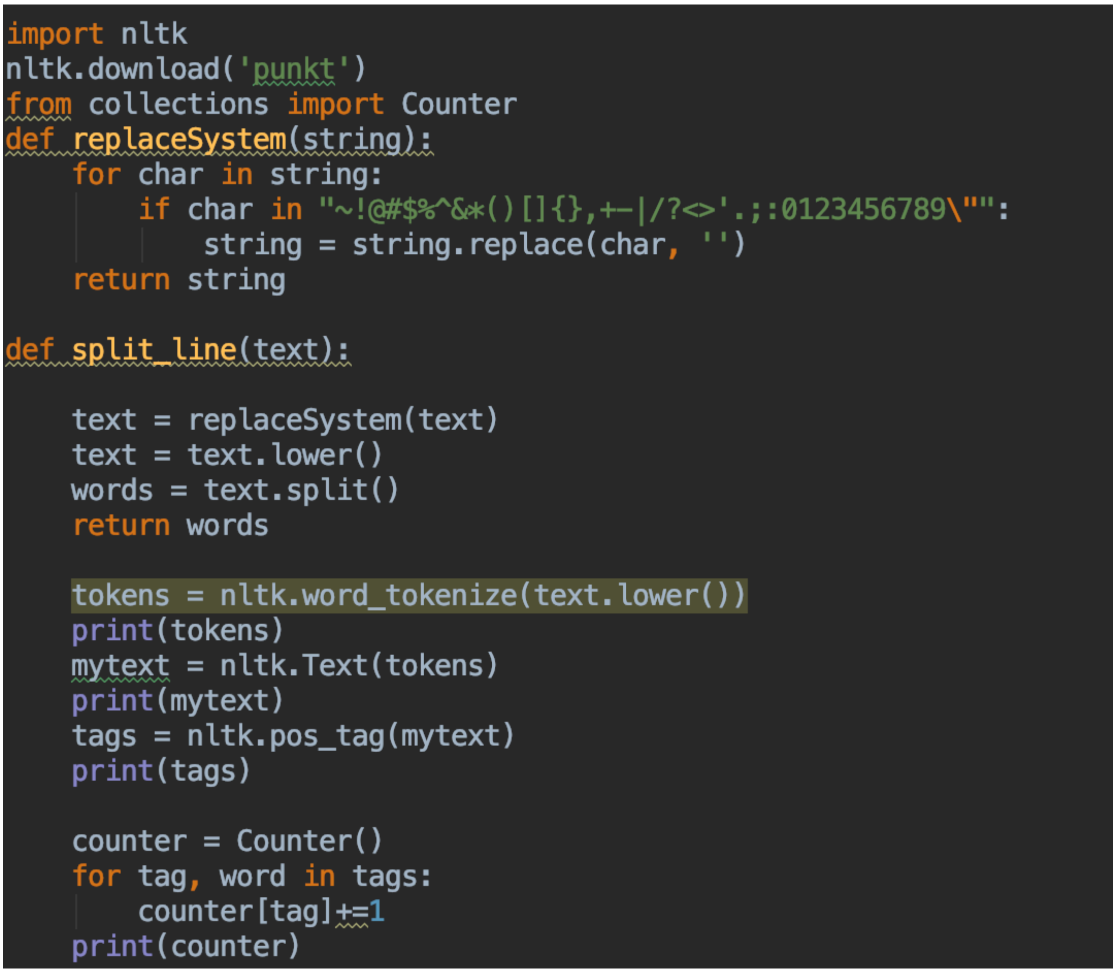
-Next, use collections.Counter function to make each word become a dictionary key.

-The result is as following:



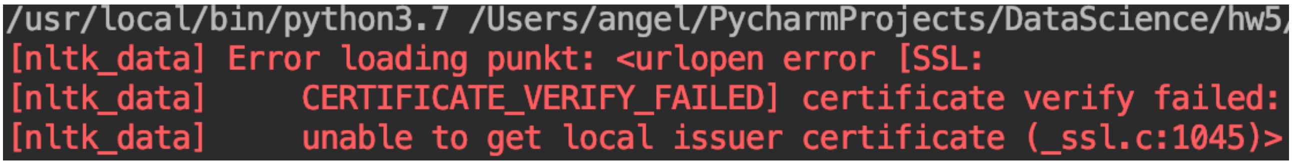
```
Counter({'the': 155, 'of': 88, 'a': 80, 'to': 62, 'chars': 61, 'in': 46, 'and': 43, 'that': 35, 'on': 24, 'for': 22, 'new': 17, 'is': 16,
finish 2 /6
Counter({'the': 80, 'a': 39, 'of': 37, 'chars': 37, 'to': 37, 'and': 31, 'that': 20, 'in': 20, 'on': 19, 'has': 15, 'tuesday': 14, 'for':
finish 3 /6
Counter({'the': 98, 'a': 86, 'of': 74, 'chars': 61, 'to': 51, 'and': 47, 'in': 45, 'with': 30, 'that': 30, 'is': 27, 'have': 21, 'health'
finish 4 /6
Counter({'the': 149, 'a': 85, 'of': 65, 'chars': 64, 'to': 63, 'in': 53, 'and': 46, 'that': 34, 'for': 30, 'on': 28, 'is': 22, 'as': 20,
finish 5 /6
Counter({'the': 139, 'a': 75, 'of': 65, 'and': 65, 'chars': 63, 'to': 58, 'in': 55, 'for': 34, 'on': 27, 'is': 24, 'at': 21, 'with': 20,
finish 6 /6
Counter({'the': 220, 'a': 76, 'to': 68, 'chars': 66, 'of': 60, 'in': 47, 'and': 45, 'for': 33, 'on': 31, 'that': 25, 'it': 24, 'is': 24,
```

-I was trying to use the package to detect each part of speech, but every package I used has SSL problem, and I don't know how to solve it.



```python
import nltk
nltk.download('punkt')
from collections import Counter
def replaceSystem(string):
    for char in string:
        if char in "~!@#$%^&*()[]{},+-|/?<>'.;:0123456789\"":
            string = string.replace(char, '')
    return string

def split_line(text):

    text = replaceSystem(text)
    text = text.lower()
    words = text.split()
    return words

    tokens = nltk.word_tokenize(text.lower())
    print(tokens)
    mytext = nltk.Text(tokens)
    print(mytext)
    tags = nltk.pos_tag(mytext)
    print(tags)

    counter = Counter()
    for tag, word in tags:
        counter[tag]+=1
    print(counter)
```

-The Error result is as following:

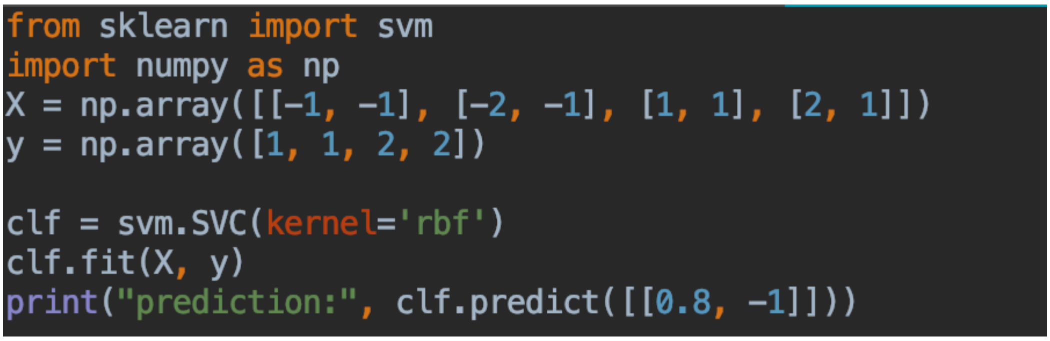


```
/usr/local/bin/python3.7 /Users/angel/PycharmProjects/DataScience/hw5/
[nltk_data] Error loading punkt: <urlopen error [SSL:
[nltk_data]     CERTIFICATE_VERIFY_FAILED] certificate verify failed:
[nltk_data]     unable to get local issuer certificate (_ssl.c:1045)>
```

d. Classification

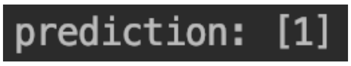
-I didn’t finish this part because it already runs out of time. So far, I make a matrix of each article, but I cannot combine these matrixes. As long as I finish combine them, I can use sklearn.SCV to do the machine learning, which uses Naive Bayes when its parameter ‘kernel’ equal to ‘rdf’. To compare with other algorithms , it’s a good function for predicting multi-class labels and easy for operating.

-The code should look like the following:



```
from sklearn import svm
import numpy as np
X = np.array([[-1, -1], [-2, -1], [1, 1], [2, 1]])
y = np.array([1, 1, 2, 2])

clf = svm.SVC(kernel='rbf')
clf.fit(X, y)
print("prediction:", clf.predict([[0.8, -1]]))
```

-The output result is as following:



```
prediction: [1]
```