Information Retrieval – Assignment 13

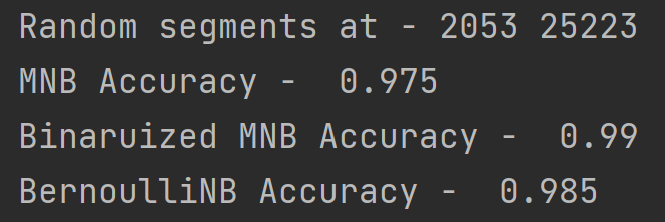
# IR14A.py

## Code

# IR14A.py CS5154/6054 cheng 2022  
# Comparing MNB (multinomialNB) with binarized MNB and BernoulliNB  
# on two random segments of bible.txt from the first third and last third  
# 100 test documents are at the center of 1000 training documents  
# Usage: python IR14A.py  
  
import numpy as np  
import random  
from sklearn.feature\_extraction.text import CountVectorizer  
from sklearn.naive\_bayes import MultinomialNB  
from sklearn.naive\_bayes import BernoulliNB  
from sklearn.metrics import accuracy\_score  
  
f = open("bible.txt", "r")  
docs = f.readlines()  
f.close()  
N =len(docs)  
N1 = N // 3 - 1100  
c0 = random.randrange(N1)  
c1 = N - 1100 - random.randrange(N1)  
print('Random segments at -', c0, c1)  
  
trainX = np.concatenate([docs[c0:c0+500], docs[c0+600:c0+1100],  
docs[c1:c1+500], docs[c1+600:c1+1100]])  
y = np.concatenate([np.zeros(1000, dtype=np.int16), np.ones(1000, dtype=np.int16)])  
testX = np.concatenate([docs[c0+500:c0+600], docs[c1+500:c1+600]])  
testY = np.concatenate([np.zeros(100, dtype=np.int16), np.ones(100, dtype=np.int16)])  
  
cv = CountVectorizer(max\_df=0.4, min\_df=4)  
X = cv.fit\_transform(trainX).toarray()  
T = cv.transform(testX).toarray()  
  
model = MultinomialNB()  
model.fit(X, y)  
pred = model.predict(T)  
print ('MNB Accuracy - ', accuracy\_score(testY, pred))  
  
cv = CountVectorizer(binary=True, max\_df=0.4, min\_df=4)  
X = cv.fit\_transform(trainX).toarray()  
T = cv.transform(testX).toarray()  
  
model = MultinomialNB()  
model.fit(X, y)  
pred = model.predict(T)  
print ('Binaruized MNB Accuracy - ', accuracy\_score(testY, pred))  
  
model = BernoulliNB()  
model.fit(X, y)  
pred = model.predict(T)  
print ('BernoulliNB Accuracy - ', accuracy\_score(testY, pred))

## Results

### Run 1

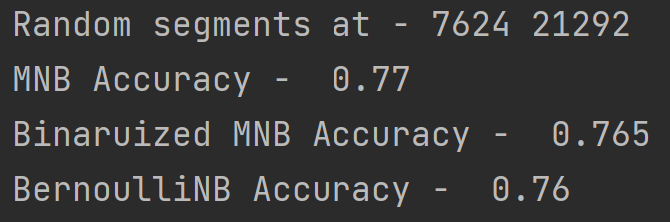


### Run 2

A picture containing text, sign, close

Description automatically generated

### Run 3



### Run 4

A picture containing text, outdoor, sign

Description automatically generated

### Run 5

Text

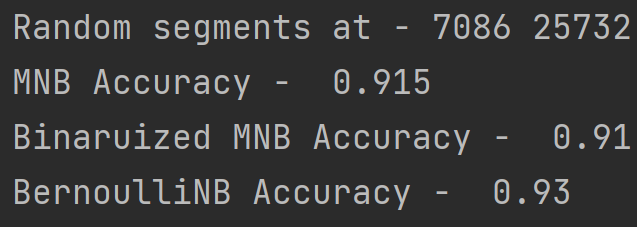
Description automatically generated

### Run 6

Text

Description automatically generated

### Run 7



### Run 8

A picture containing text, sign

Description automatically generated

### Run 9

A picture containing text, sign

Description automatically generated

### Run 10

A picture containing text, sign

Description automatically generated

# Times MNB was the best = 1

# Times Binaurized MNB was the best = 2

# BernoulliNB was the best = 7

# IR14C.py

## Code

# IR14C.py CS5154/6054 cheng 2022  
# NB top features by parameter weights  
# Usage: python IR14C.py  
  
import numpy as np  
from sklearn.feature\_extraction.text import CountVectorizer  
from sklearn.naive\_bayes import MultinomialNB  
  
f = open("bible.txt", "r")  
docs = f.readlines()  
f.close()  
N =len(docs)  
trainX = np.concatenate([docs[0:1000], docs[N-1000:N]])  
y = np.concatenate([np.zeros(1000, dtype=np.int16), np.ones(1000, dtype=np.int16)])  
  
cv = CountVectorizer(binary=True, max\_df=0.4, min\_df=4)  
X = cv.fit\_transform(trainX).toarray()  
print(X.shape)  
voc = cv.get\_feature\_names()  
  
model = MultinomialNB()  
model.fit(X, y)  
logprob = model.feature\_log\_prob\_  
logodds = logprob[0] - logprob[1]  
vocpos = logodds.argsort()  
for i in vocpos[:10]:  
 print(voc[i], logodds[i])  
for i in vocpos[-10:]:  
 print(voc[i], logodds[i])

## Results

Text

Description automatically generated