Experiment 4

NAME	**Shreya Shetty**
UID	**2019140059**
CLASS	**TE IT**
BATCH	**B**
SUBJECT	**NLP Lab**

AIM: Classification using suitable classification model (NB)

THEORY:

What is Text Classification?

Text clarification is the process of categorizing the text into a group of words. By using NLP, text classification can automatically analyze text and then assign a set of predefined tags or categories based on its context. NLP is used for sentiment analysis, topic detection, and language detection. There is mainly three text classification approach

1. Rule-based System
2. Machine System
3. Hybrid System.

What is Naive Bayes?

Naive Bayes is a family of algorithms based on applying Bayes theorem with a strong(naive) assumption, that every feature is independent of the others, in order to predict the category of a given sample. They are probabilistic classifiers, therefore will calculate the probability of each category using Bayes theorem, and the category with the highest probability will be output.

Why Naive Bayes classifiers?

We do have other alternatives when coping with NLP problems, such as Support Vector Machine (SVM) and neural networks. However, the simple design of Naive Bayes classifiers make them very attractive for such classifiers. Moreover, they have been demonstrated to be fast, reliable and accurate in a number of applications of NLP.

Practical Example -

Let’s say, we are given a sentence “ A very close game”, a training set of five sentences (as shown below), and their corresponding category (Sports or Not Sports). The goal is to build a Naive Bayes classifier that will tell us which category the sentence “ A very close game” belongs to.

We could try applying a Naive Bayes classifier, thus the strategy would be calculating the probability of both "A very close game is Sports", as well as it's Not Sports. The one with the higher probability will be the result.

Expressed formally, this is what we would like to calculate P(Sports | A very close game), i.e. the probability that the category of the sentence is Sports given that the sentence is "A very close game".

Calculating P(Sports | A very close game).

Bayes' Theorem is useful for dealing with conditional probabilities, since it provides a way for us to reverse them.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

In our case, the probability that we wish to calculate can be calculated as:

$$P(sports|a\,very\,close\,game) = \frac{P(a\,very\,close\,game|sports) \times P(sports)}{P(a\,very\,close\,game)}$$

IDE USED: Jupyter Notebook

DATASET USED: http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

This data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups

LIBRARIES USED:

Nltk:

The Natural Language Toolkit (NLTK) is a Python package for natural language processing. NLTK requires Python 3.7, 3.8, 3.9 or 3.10. It can be installed as:

pip install nltk

scikit-learn:

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines,Naïve Bayes etc. It can be installed as:

pip install scikit-learn

numpy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate. It can be installed as:

pip install numpy

matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It can be installed as:

pip install matplotlib

CODE:

```
# Importing required Libraries
import os
import string
import numpy as np
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report

X = [] # text from file
Y = [] # category of the corresponding X element
count=0
row=0
for category in os.listdir('20_newsgroups'):
    count+=1
    # Dataset of 3 news group from dataset
    if count==4:
        break
    for document in os.listdir('20_newsgroups/'+category):
        row+=1
        with open('20_newsgroups/'+category+'/'+document, "r") as f:
            X.append((document,f.read()))
            Y.append(category)
    print("row: ",row)
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X, Y, test_size=0.25, random_state=0)
print(len(Y_test))
```

```python
# Displaying stop words
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
stopwords = set (stopwords.words('english'))
print("Stop Words List :\n",stopwords)
```

```python
# Building a vocabulary of words from dataset
vocab = {}
for i in range(len(X_train)):
    word_list = []
    for word in X_train[i][1].split():
        word_new  = word.strip(string.punctuation).lower()
        if (len(word_new)>2)  and (word_new not in stopwords):
            if word_new in vocab:
                vocab[word_new]+=1
            else:
                vocab[word_new]=1
```

```python
# Plotting a graph of no of words with a given frequency to decide cutoff drequency
num_words = [0 for i in range(max(vocab.values())+1)]
freq = [i for i in range(max(vocab.values())+1)]
for key in vocab:
    num_words[vocab[key]]+=1
plt.plot(freq,num_words)
plt.axis([1, 10, 0, 20000])
plt.xlabel("Frequency")
plt.ylabel("No of words")
plt.grid()
plt.show()
```

```python
cutoff_freq = 80
num_words_above_cutoff = len(vocab)-sum(num_words[0:cutoff_freq])
print("No. of words with frequency higher than cutoff frequency({}) :".format(cutoff_freq),num_words_above_
```

```python
# Words with frequency higher than cutoff frequency are chosen as features as words with low frequencies wou
features = []
for key in vocab:
    if vocab[key] >=cutoff_freq:
        features.append(key)

print("Features",features)
```

```python
# To represent training data as word vector counts
X_train_dataset = np.zeros((len(X_train),len(features)))
for i in range(len(X_train)):
    word_list = [ word.strip(string.punctuation).lower() for word in X_train[i][1].split()]
    for word in word_list:
        if word in features:
            X_train_dataset[i][features.index(word)] += 1

print("X_TRAIN :\n",X_train_dataset)
```

```
# To represent test data as word vector counts
X_test_dataset = np.zeros((len(X_test),len(features)))
for i in range(len(X_test)):
    # print(i) # Uncomment to see progress
    word_list = [ word.strip(string.punctuation).lower() for word in X_test[i][1].split()]
    for word in word_list:
        if word in features:
            X_test_dataset[i][features.index(word)] += 1

print("X_TEST :\n",X_test_dataset)
```

```
# Implementing Multinomial Naive Bayes from scratch
class MultinomialNaiveBayes:

    def __init__(self):
        # count stores several dictionaries corresponding to each news category
        # each value in the subdictionary represents the freq of the
        self.count = {}
        # classes represents the different news categories
        self.classes = None

    def fit(self,X_train,Y_train):
        self.classes = set(Y_train)
        for class_ in self.classes:
            self.count[class_] = {}
            for i in range(len(X_train[0])):
                self.count[class_][i] = 0
            self.count[class_]['total'] = 0
            self.count[class_]['total_points'] = 0
        self.count['total_points'] = len(X_train)

        for i in range(len(X_train)):
            for j in range(len(X_train[0])):
                self.count[Y_train[i]][j]+=X_train[i][j]
                self.count[Y_train[i]]['total']+=X_train[i][j]
```

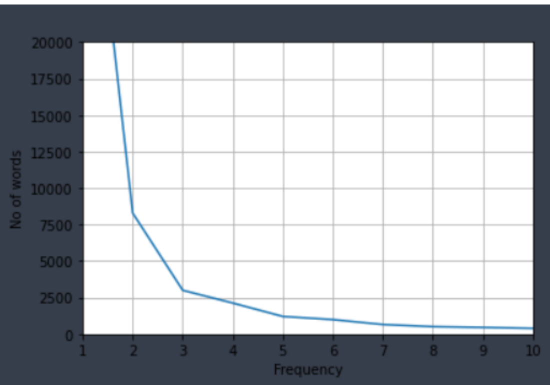

```
                self.count[Y_train[i]]['total']+=X_train[i][j]
```


```
            self.count[Y_train[i]]['total_points']+=1

    def __probability(self,test_point,class_):
        log_prob = np.log(self.count[class_]['total_points']) - np.log(self.count['total_points'])
        total_words = len(test_point)
        for i in range(len(test_point)):
            current_word_prob = test_point[i]*(np.log(self.count[class_][i]+1)-np.log(self.count[class_]['t
            log_prob += current_word_prob

        return log_prob


    def __predictSinglePoint(self,test_point):
        best_class = None
        best_prob = None
        first_run = True

        for class_ in self.classes:
            log_probability_current_class = self.__probability(test_point,class_)
            if (first_run) or (log_probability_current_class > best_prob) :
                best_class = class_
                best_prob = log_probability_current_class
                first_run = False
```

```
    def predict(self,X_test):
        # Function to predict categories
        Y_pred = []
        for i in range(len(X_test)):
            Y_pred.append( self.__predictSinglePoint(X_test[i]) )
        # print("Y_pred : ",Y_pred)
        return Y_pred

    def score(self,Y_pred,Y_true):
        # Function to find the mean accuracy
        count = 0
        for i in range(len(Y_pred)):
            if Y_pred[i] == Y_true[i]:
                count+=1
        return count/len(Y_pred)
```

```
nbModel = MultinomialNaiveBayes()
nbModel.fit(X_train_dataset,Y_train)
Y_test_pred = nbModel.predict(X_test_dataset)
print("Accuracy obtained :",nbModel.score(Y_test_pred,Y_test) )
print("\nClassification report for given Test Dataset :")
print(classification_report(Y_test, Y_test_pred))

print("Sample Text for prediction from Testing Set:\n",X_test[1])
print("Prediction : ",nbModel.predict([X_test_dataset[1]]))
print("Given prediction : ",Y_test[1])

print("\n\nSample Text 2 for prediction from Testing Set:\n",X_test[0])
print("Prediction : ",nbModel.predict([X_test_dataset[0]]))
print("Given prediction : ",Y_test[0])
```

OUTPUT:




```
No. of words with frequency higher than cutoff frequency(80) : 701
```

```
Features ['xref', 'cantaloupe.srv.cs.cmu.edu', 'newsgroups', 'path', 'subject', 'jesus', 'message-id', 'sender', 'usenet', 'news', 'organi
zation', 'university', 'engineering', 'computer', 'network', 'references', 'distribution', 'date', 'tue', 'apr', '1993', 'gmt', 'lines',
'note', 'whatever', 'says', 'latest', 'thing', 'like', 'good', 'even', 'better', 'comp.graphics', 'tiff', 'anything', 'college', 'program
s', 'ibm', 'works', 'view', 'read', 'basic', 'jpeg', 'gif', 'bmp', 'etc', 'thanks', 'post', 'looks', 'know', 'nntp-posting-host', 'reply-t
o', 'school', 'wed', 'article', 'writes', 'according', 'version', 'number', 'bytes', 'last', 'actually', "i'm", 'sure', 'every', 'time',
'part', 'anyway', 'think', 'find', 'choice', 'file', 'use', 'support', 'word', 'either', "i've", 'found', 'many', 'matter', 'problems', 'a
lso', 'important', 'help', 'define', 'little', 'bit', 'image', 'format', 'chris', 'get', 'opinions', 'mean', 'anybody', 'else', 'frank@d01
2s658.uucp', 'frank', "o'dwyer", 'alt.atheism', 'behavior', 'mike', 'making', 'wrong', 'basis', 'come', 'question', 'idea', 'heard', 'on
e', 'morality', 'necessarily', 'stuff', 'deal', 'necessary', 'keep', 'society', 'going', 'often', 'different', 'includes', 'example', 'peo
ple', 'though', 'may', 'point', 'answer', 'make', "that's", 'yet', 'another', 'claim', 'well', 'things', 'system', 'least', 'comes', 'cert
ain', 'see', 'must', 'moral', 'position', 'put', 'sometimes', 'hard', 'would', 'right', 'i.e', 'various', 'sort', 'possible', 'take', 'awa
y', 'object', 'way', 'problem', 'much', 'seem', 'agree', 'level', 'need', 'guess', 'thus', 'law', 'allows', 'believe', 'understand', 'stat
ement', 'seen', 'let', 'benedikt', 'islamic', 'women', 'technical', 'germany', 'bill', "can't", 'anyone', 'god', 'makes', 'world', 'standa
rd', 'argument', 'based', 'quite', 'work', 'reason', 'known', 'advance', 'say', 'something', 'defined', 'cannot', 'exist', 'really', 'exis
tence', 'course', 'posting', 'group', 'instance', 'reality', 'unless', 'information', 'definition', 'however', 'said', 'got', 'looking',
'belief', "i'd", 'rest', 'evidence', 'easy', 'gods', 'comp.os.ms-windows.misc', 'windows', '3.1', 'please', 'access', 'usa', 'eric', 'fr
i', 'never', 'package', 'seems', 'day', 'ram', 'win', 'reference', 'tell', 'email', 'back', 'memory', 'run', 'data', 'due', 'real', 'erro
r', 'christian', 'simple', 'logic', 'arguments', 'bible', 'made', 'tried', 'following', 'simply', 'indeed', 'opinion', 'dave', 'newsgrou
p', 'probably', 'couple', 'printer', 'uses', 'drivers', 'mail', 'department', 'tel', 'fax', 'user', 'michael', 'learn', 'lot', 'graphics',
'reading', 'code', 'hardware', 'groups', 'result', 'per', 'kind', 'big', 'feel', 'already', 'current', 'amiga', 'environment', 'internet',
'address', 'since', 'asked', 'maybe', 'talking', 'higher', 'values', 'upon', 'man', 'free', 'type', 'software', 'platforms', 'keywords',
'unit', 'quality', 'mac', 'able', 'include', 'hold', 'author', 'project', 'computing', 'driver', 'center', 'mon', 'microsoft', 'bbs', 'pri
nt', 'fonts', 'available', 'research', 'australia', 'sun', 'first', 'sorry', 'dos', 'try', 'write', 'small', 'program', 'display', 'valu
e', 'command', 'instead', 'move', 'using', 'function', 'worth', 'utility', 'copy', 'order', 'deleted', 'ms-windows', 'directory', 'alway
s', 'wanted', 'x-newsreader', 'tin', '1.1', 'window', 'technology', 'inc', 'video', 'card', 'info', 'keith', 'schneider', 'california', 'i
nstitute', 'look', 'religious', 'enough', 'similar', 'could', 'name', 'everyone', 'books', 'history', 'someone', 'whether', 'david', 'file
s', 'net', 'size', 'bits', 'pixel', 'color', 'faq', 'wrote', 'without', 'true', 'power', 'perhaps', 'imagine', 'canada', 'systems', 'phon
e', 'next', 'still', 'saying', 'two', 'years', 'buy', 'users', 'therefore', 'death', 'book', 'exists', 'old', 'new', 'around', 'second',
'complete', 'church', 'truth', 'within', 'given', 'text', 'original', 'written', 'yes', 'call', 'mark', 'john', 'peter', 'james', 'long',
'jim', 'conversion', 'e-mail', 'site', 'appreciated', 'personal', 'scott', 'certainly', 'case', 'person', 'running', 'amount', 'expect',
'sense', 'local', 'supports', "i'll", 'ask', 'ryan', 'dept', 'ftp', 'means', 'versions', 'create', 'whole', 'interface', 'source', 'gettin
g', 'several', 'used', 'space', 'points', 'great', 'server', 'ago', 'left', 'list', 'objects', 'called', 'bob', 'issue', 'atheism', 'speci
fic', 'knowledge', 'national', 'hope', 'takes', 'thomas', 'public', 'unix', 'mode', 'images', 'science', 'faith', 'ray', 'fact', 'give',

 Accuracy obtained : 0.9533333333333334

 Classification report for given Test Dataset :
                         precision    recall  f1-score   support

            alt.atheism       0.98      1.00      0.99       280
          comp.graphics       0.96      0.89      0.93       237
comp.os.ms-windows.misc       0.91      0.97      0.94       233

               accuracy                           0.95       750
              macro avg       0.95      0.95      0.95       750
           weighted avg       0.95      0.95      0.95       750



Sample Text for prediction from Testing Set:
 ('37937', 'Path: cantaloupe.srv.cs.cmu.edu!crabapple.srv.cs.cmu.edu!fs7.ece.cmu.edu!europa.eng.gtefsd.com!howland.reston.ans.net!usc!ente
rpoop.mit.edu!eru.mt.luth.se!lunic!sunic!news.chalmers.se!pc5_b109.et.hj.se!d91-fad\nFrom: d91-fad@tekn.hj.se (DANIEL FALK)\nNewsgroups: c
omp.graphics\nSubject: RE: VESA on the Speedstar 24\nMessage-ID: <d91-fad.23.734094936@tekn.hj.se>\nDate: 6 Apr 93 11:15:36 GMT\nSender: n
ews@news.chalmers.se\nOrganization: H|gskolan i J|nk|ping\nLines: 39\nNntp-Posting-Host: pc5_b109.et.hj.se\n\n>>>kjb/MGL/uvesa32.zip\n>>>
\n>>>This is a universal VESA driver.  It supports most video\n>>>boards/chipsets (include the Speedstar-24 and -24X) up to\n>>>24 bit col
or.\n>>>\n>>>Terry\n>>>\n>>>P.S.  I\'ve tried it on a Speedstar-24 and -24X and it works. :)\n\n>>Not with all software. :( For instance i
t doesn\'t work at all with\n>>Animator Pro from Autodesk. It can\'t detect ANY SVGA modes when \n>>running UniVESA. This is really a prob
lem as we need a VESA driver\n>>for both AA Pro and some hi-color stuff. :(\n\n>Just out of curiosity... Are you using the latest version
(3.2)?  Versions\n>previous to this did not fill in all of the capabilities bits and other\n>information correctly.  I had problems with a
lot of software until I got\n>this version.  (I don\'t think the author got around to posting an \n>announcementof it (or at least I misse
d it), but 3.2 was available in the \n>directory indicated as of 3/29.)\n\nI sure did use version 3.2. It works fine with most software bu
t NOT\nwith Animator Pro and that one is quite important to me. Pretty\nuseless program without that thing working IMHO.\nSo I hope the au
thor can fix that.\n\n/Daniel...\n\n\n\n\n=============================================================================\n!!      Daniel Fa
lk          \\\\  " Don\'t quote me! No comments! "          !! \n!!      ^^^^^^ ^^^^           \\\\               Ebenezum the Great Wiza
rd   !! \n!!      d91-fad@tekn.hj.se     \\\\                                  !!\n!!      d91fad@hjds90.hj.se    //  Also known a
s the mega-famous musician  !!\n!!      Jkpg, Sweeeeeden...    \\\\         Leinad of The Yellow Ones        !!\n========================
====================================================\n')
Prediction :  ['comp.graphics']
Given prediction :  comp.graphics
```

```
Sample Text 2 for prediction from Testing Set:
 ('53160', "Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!noc.near.net!howland.reston.ans.net!news.cac.psu.edu!psuvm!sec108\nOrgani
zation: Penn State University\nDate: Thu, 15 Apr 1993 20:53:12 EDT\nFrom: <SEC108@psuvm.psu.edu>\nMessage-ID: <93105.205312SEC108@psuvm.ps
u.edu>\nNewsgroups: alt.atheism\nSubject: Why the bible?\nLines: 38\n\n      One thing I think is interesting about alt.athiesm is the fac
t that\nwithout bible-thumpers and their ilk this would be a much duller newsgroup.\nIt almost needs the deluded masses to write silly thi
ngs for athiests to\ntear apart. Oh well, that little tidbit aside here is what I really wanted\nwrite about.\n\n      How can anyone beli
eve in such a sorry document as the bible? If you\nwant to be religious aren't there more plausable books out there? Seriously,\nthe bible
was written by multiple authors who repeatedly contradict each\nother. One minute it tells you to kill your kid if he talks back and the n
ext\nit says not to kill at all. I think that if xtians really want to follow a\ndeity they should pick one that can be consistent, unlike
the last one they\ninvented.\n\n      For people who say Jesus was the son of god, didn't god say not to\nEVER put ANYONE else before him?
Looks like you did just that. Didn't god\nsay not to make any symbols or idols? What are crosses then? Don't you think\nthat if you do in
fact believe in the bible that you are rather far off track?\n\nWas Jesus illiterate? Why didn't he write anything? Anyone know?\n\n
I honestly hope that people who believe in the bible understand that\nit is just one of the religious texts out there and that it is one o
f the\npoorer quality ones to boot. The only reason xtianity escaped the middle east\nis because a certain roman who's wine was poisoned w
ith lead made all of rome\nxtian after a bad dream.\n\n      If this posting keeps one person, just ONE person, from standing on a\nstreet
corner and telling people they are going to hell I will be happy.\n\n\n\n\n\n*** Only hatred and snap judgements can guide your robots thr
ough life. ***\n***                                    Dr. Clayton Forester             ***\n***                                        Mad
Scientist                 ***\n\n")
Prediction :  ['alt.atheism']
Given prediction :  alt.atheism
```

REFERENCES:

1. https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf
2. https://www.analyticsvidhya.com/blog/2020/12/understanding-text-classification-in-nlp-with-movie-review-example-example/
3. https://www.youtube.com/watch?v=60pqgfT5tZM
4. https://www.youtube.com/watch?v=temQ8mHpe3