20_newspaper Dataset Text Classification

In [277…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
from sklearn.model_selection import train_test_split
import re
from nltk.corpus import stopwords
import operator
from os import listdir
from os.path import isfile, join
import string
import copy
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\siddh\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [278…
```
dir_path = "20_newsgroups"  # path to directory
# please Ensure that the this ipynb and this directory are in same directory
folders = [fo for fo in listdir(dir_path)]
# folders now contains all the folders present in the directory 20_newsgroup
```

In [279…
```
# We import Stopwords which are all the words which are commmon to general englishh and
# example - Pronouns, conjuctions, Modals etc.
stopWords=stopwords.words('english')
string.punctuation
```

Out[279…
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [280…
```
stopWords.extend(list(string.punctuation))
# Including Punctuation in our stopwords
```

In [281…
```
data = dict()
content = dict()    # This will contain all our Words which constitute the words we wil

for foldername in folders:

    data[foldername] = []

    for filename in listdir(join(dir_path, foldername)):
        with open(join(dir_path, foldername , filename), 'r' , errors='ignore') as f:
            text = f.read()
            # text = text.decode('utf16')
            text = text.lower()
            text = re.sub(r"[^a-z]", ' ', text) # Ignoring all the text which don't hav

            curr_wordList=text.split()  # goves us the list of all meaningful words in
            data[foldername].append(curr_wordList)
```

```
            for word in curr_wordList:
                if word in content:
                    content[word] += 1      #Adding to the Frequency of the given word

                elif word not in stopWords: # If this word is new so we will make a new
                    content[word] = 1


content=sorted(content.items(),key=operator.itemgetter(1),reverse=True) #Sorting the Co
```

In [282…

```
sampleSize = 500    # Feature Set will have these many Features, More the Features more
```

In [283…

```
type(content)
```

Out[283…

```
list
```

In [284…

```
feature_set=[]

for e in content:
    feature_set.append(e[0])

feature_set=feature_set[0:sampleSize]   #taking only the

feature_set
```

Out[284…

```
['edu',
 'ax',
 'cmu',
 'com',
 'cs',
 'apr',
 'news',
 'srv',
 'cantaloupe',
 'net',
 'c',
 'message',
 'subject',
 'x',
 'lines',
 'id',
 'date',
 'newsgroups',
 'path',
 'organization',
 'state',
 'gmt',
 'would',
 'ohio',
 'one',
 'r',
 'writes',
 'article',
 'comp',
 'w',
```

```
 'references',
 'b',
 'talk',
 'u',
 'misc',
 'p',
 'ca',
 'q',
 'f',
 'ans',
 'sender',
 'g',
 'e',
 'howland',
 'reston',
 'v',
 'people',
 'like',
 'university',
 'posting',
 'nntp',
 'cc',
 'mps',
 'know',
 'may',
 'host',
 'sci',
 'get',
 'n',
 'k',
 'think',
 'usenet',
 'zaphod',
 'l',
 'alt',
 'politics',
 'windows',
 'also',
 'h',
 'time',
 'j',
 'rutgers',
 'use',
 'harvard',
 'eng',
 'near',
 'new',
 'z',
 'us',
 'rec',
 'good',
 'system',
 'crabapple',
 'noc',
 'sei',
 'well',
 'soc',
 'could',
 'das',
 'xref',
```

```
                'cis',
                'even',
                'see',
                'europa',
                'gtefsd',
                'ece',
                'god',
                'uunet',
                'world',
                'religion',
                'fs',
                'way',
                'uk',
                'org',
                'two',
                'say',
                'make',
                'many',
                'christian',
                'much',
                'first',
                'gov',
                'max',
                'right',
                'de',
                'rochester',
                'distribution',
                'hp',
                'sun',
                'club',
                'ac',
                'space',
                'mail',
                'andrew',
                'want',
                'gatech',
                'anyone',
                'magnesium',
                'ibm',
                'go',
                'said',
                'udel',
                'reply',
                'utexas',
                'used',
                'sys',
                'culture',
                'need',
                'uiuc',
                'work',
                'really',
                'nasa',
                'something',
                'please',
                'problem',
                'os',
                'computer',
                'ms',
                'sura',
                'believe',
```

```
                'since',
                'graphics',
                'hardware',
                'mac',
                'still',
                'back',
                'pc',
                'netcom',
                'usa',
                'years',
                'going',
                'government',
                'information',
                'help',
                'find',
                'point',
                'take',
                'file',
                'might',
                'better',
                'year',
                'using',
                'question',
                'never',
                'au',
                'last',
                'things',
                'read',
                'software',
                'thanks',
                'david',
                'mit',
                'must',
                'etc',
                'bit',
                'without',
                'cso',
                'uucp',
                'sure',
                'usc',
                'another',
                'made',
                'bb',
                'number',
                'someone',
                'access',
                'data',
                'case',
                'forsale',
                'look',
                'part',
                'let',
                'got',
                'thing',
                'cwru',
                'program',
                'drive',
                'long',
                'fact',
                'co',
```

```
                'version',
                'available',
                'law',
                'apple',
                'guns',
                'day',
                'come',
                'science',
                'atheism',
                'john',
                'anything',
                'power',
                'little',
                'columbia',
                'around',
                'however',
                'darwin',
                'give',
                'tue',
                'sport',
                'best',
                'every',
                'public',
                'true',
                'different',
                'seems',
                'fri',
                'key',
                'car',
                'probably',
                'dos',
                'group',
                'least',
                'game',
                'set',
                'line',
                'life',
                'research',
                'att',
                'put',
                'enough',
                'actually',
                'try',
                'course',
                'tell',
                'real',
                'mr',
                'support',
                'name',
                'great',
                'run',
                'lot',
                'free',
                'list',
                'inc',
                'washington',
                'next',
                'high',
                'post',
                'jesus',
```

```
                'man',
                'systems',
                'says',
                'either',
                'though',
                'bill',
                'far',
                'nothing',
                'hard',
                'mean',
                'image',
                'called',
                'old',
                'second',
                'pl',
                'hockey',
                'else',
                'possible',
                'end',
                'internet',
                'call',
                'st',
                'rather',
                'card',
                'wrong',
                'non',
                'pitt',
                'stanford',
                'ins',
                'mon',
                'reason',
                'acs',
                'turkish',
                'network',
                'thu',
                'yes',
                'athos',
                'autos',
                'mark',
                'unix',
                'general',
                'uwm',
                'others',
                'gun',
                'person',
                'baseball',
                'jewish',
                'order',
                'keep',
                'based',
                'looking',
                'found',
                'problems',
                'agate',
                'email',
                'maybe',
                'technology',
                'done',
                'able',
                'security',
```

```
'human',
'umd',
'files',
'team',
'seen',
'example',
'bad',
'place',
'center',
'info',
'ever',
'med',
'american',
'yet',
'mb',
'always',
'three',
'israel',
'quite',
'purdue',
'control',
'send',
'following',
'mideast',
'thought',
'wrote',
'heard',
'sgi',
'evidence',
'michael',
'chip',
'means',
'electronics',
'crypt',
'less',
'virginia',
'trying',
'jews',
'book',
'whether',
'phone',
'left',
'window',
'colorado',
'children',
'opinions',
'wed',
'win',
'start',
'big',
'away',
'given',
'national',
'today',
'idea',
'times',
'several',
'rights',
'user',
'home',
```

```
 'se',
 'word',
 'berkeley',
 'geneva',
 'ftp',
 'kind',
 'getting',
 'questions',
 'keywords',
 'paul',
 'jpl',
 'president',
 'seem',
 'note',
 'steve',
 'fi',
 'remember',
 'service',
 'mike',
 'games',
 'standard',
 'change',
 'scsi',
 'makes',
 'history',
 'followup',
 'code',
 'saying',
 'clipper',
 'local',
 'source',
 'toronto',
 'open',
 'server',
 'wanted',
 'price',
 'full',
 'wupost',
 'department',
 'perhaps',
 'caltech',
 'abortion',
 'men',
 'cannot',
 'jim',
 'frank',
 'fbi',
 'already',
 'ask',
 'money',
 'large',
 'ucs',
 'bogus',
 'institute',
 'stuff',
 'small',
 'fax',
 'disk',
 'digex',
 'motorcycles',
```

```
'sdd',
'answer',
'indiana',
'war',
'address',
'running',
'ed',
'th',
'speed',
'clinton',
'magnus',
'matter',
'issue',
'days',
'interested',
'mil',
'buy',
'bible',
'original',
'ago',
'agree',
'approved',
'states',
'sale',
'show',
'whole',
'came',
'play',
'video',
'works',
'type',
'pretty',
'told',
'current',
'box',
'color',
'legal',
'fire',
'care',
'sol',
'hope',
'cb',
'machine',
'canada',
'claim',
'earth',
'robert',
'stratus',
'live',
'feel']
```

In [285…

```
def to_dataframe(cols):

    Data = pd.DataFrame(columns = cols)
    label = []
    for foldername in folders:
        for filename in data[foldername]:

            label.append(foldername)
```

```
            Data.loc[len(Data)] = np.zeros(len(cols))

            for text in filename:
                for word in text.split():
                    if word in cols:
                        Data.loc[len(Data)-1,word] = Data.loc[len(Data)-1,word] + 1

    Data['label']=label
    return Data
```

In [286…
```
Data = to_dataframe(feature_set)
```

In [287…
```
Data.head()
```

Out[287…

	edu	ax	cmu	com	cs	apr	news	srv	cantaloupe	net	...	cb	machine	canada	claim	earth
0	10.0	0.0	5.0	1.0	3.0	1.0	4.0	3.0	2.0	2.0	...	0.0	0.0	0.0	0.0	2.0
1	5.0	0.0	4.0	1.0	3.0	1.0	3.0	3.0	2.0	1.0	...	0.0	0.0	0.0	1.0	0.0
2	7.0	0.0	3.0	3.0	4.0	1.0	2.0	2.0	1.0	1.0	...	0.0	0.0	0.0	1.0	0.0
3	6.0	0.0	4.0	1.0	3.0	1.0	1.0	3.0	2.0	1.0	...	0.0	0.0	0.0	0.0	0.0
4	4.0	0.0	4.0	10.0	3.0	4.0	0.0	3.0	2.0	1.0	...	1.0	0.0	0.0	0.0	0.0

5 rows × 501 columns

In [288…
```
nonlabelData = Data.drop(['label'],axis=1).values   #Seperate L=Non-label Data
labels = Data['label'].values   # seperate labels

X_train,X_test,Y_train,Y_test=train_test_split(nonlabelData,labels,random_state=0,test_
```

In [289…
```
Data.to_csv('data.csv') # Just in case code takes extra time we can use up this data fr
```

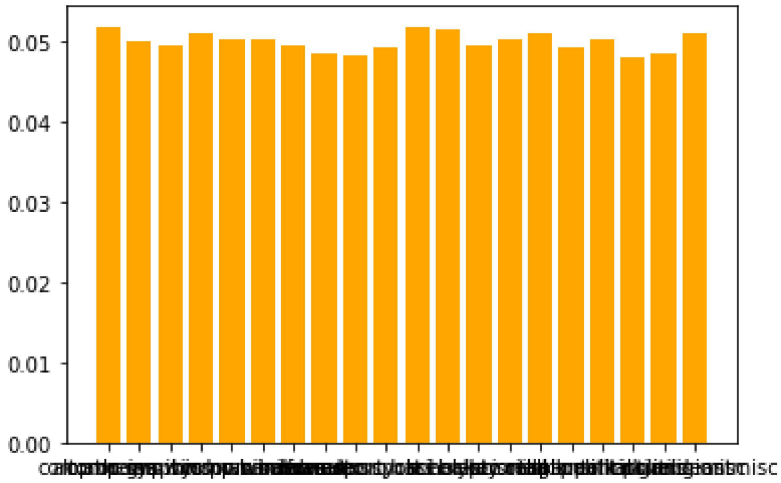
Getting Priors

In [290…
```
priors = dict() # Priors of all the newspapers will be stored along theor names.

for foldername in folders:
    val = len(Y_train[Y_train == foldername])
    priors[foldername] = val / len(Y_train)

plt.bar(priors.keys(),priors.values(),color='orange')
```

Out[290…
```
<BarContainer object of 20 artists>
```



In [291…
```
priors.values()
```

Out[291…
```
dict_values([0.051868257483746516, 0.049939272701293135, 0.04943916553547189, 0.05115381
867543045, 0.05022504822461956, 0.05036793598628277, 0.0495820532971351, 0.0485818389654
92605, 0.04829606344216618, 0.04936772165464028, 0.051868257483746516, 0.051582481960420
09, 0.049510609416303496, 0.05022504822461956, 0.05093948703293563, 0.04922483389297707,
0.05029649210545117, 0.04793884403800815, 0.04865328284632421, 0.05093948703293563])
```

Getting Class Conditionals

In [292…
```
#Calculating the Class Conditionals

classConditional = pd.DataFrame(columns=feature_set)

for i,foldername in enumerate(folders):

    classConditional.loc[len(classConditional)] = np.zeros(len(feature_set))
    for feature in feature_set:

        num = Data[feature][Data['label']==foldername].sum()
        denom = len(Data[Data['label']==foldername])

        classConditional[feature][i] = ( num + 1) / ( denom + len(feature_set))   #LaPL
# Each Column needs to be divided by its net sum to normalize it
classConditional = classConditional.div(classConditional.sum( axis = 0), axis = 1)
# This ensures that now each feature has a net sum of 1
```

In [293…
```
classConditional
```

Out[293…

	edu	ax	cmu	com	cs	apr	news	srv	cantaloupe	net
0	0.056935	0.000016	0.052200	0.064111	0.053778	0.049854	0.055730	0.050114	0.048599	0.055040
1	0.045603	0.000048	0.048634	0.035303	0.051706	0.039820	0.049375	0.049898	0.050747	0.053668
2	0.046913	0.998690	0.046765	0.040668	0.048554	0.036178	0.048771	0.049404	0.050632	0.052413
3	0.042579	0.000096	0.047813	0.052014	0.047276	0.045078	0.041553	0.048725	0.044149	0.051159

	edu	ax	cmu	com	cs	apr	news	srv	cantaloupe	net
4	0.050040	0.000032	0.050846	0.030848	0.044433	0.043149	0.048397	0.048601	0.043536	0.047631
5	0.039866	0.000192	0.047642	0.049756	0.049348	0.033887	0.050238	0.047860	0.049213	0.053864
6	0.048730	0.000064	0.048691	0.028967	0.045623	0.037939	0.042761	0.050917	0.052895	0.043436
7	0.048468	0.000048	0.045677	0.061003	0.041083	0.060514	0.052308	0.044217	0.045147	0.045749
8	0.038903	0.000080	0.040890	0.073754	0.039827	0.059501	0.053545	0.041191	0.039892	0.049395
9	0.053820	0.000032	0.034539	0.043539	0.048356	0.062516	0.059958	0.036991	0.039086	0.054491
10	0.045426	0.000016	0.047566	0.027561	0.044830	0.058392	0.051906	0.038690	0.038664	0.047395
11	0.045261	0.000192	0.053821	0.077179	0.054990	0.060635	0.052538	0.054406	0.055772	0.054256
12	0.046127	0.000048	0.050064	0.041698	0.047012	0.050698	0.045090	0.049373	0.046451	0.042378
13	0.051009	0.000144	0.049721	0.053122	0.054197	0.052193	0.045723	0.049373	0.048407	0.043475
14	0.050820	0.000064	0.054012	0.036590	0.053491	0.056438	0.053574	0.050886	0.052473	0.067624
15	0.061019	0.000016	0.037513	0.019978	0.041121	0.029411	0.022677	0.045574	0.049119	0.054247
16	0.060178	0.000144	0.062442	0.074684	0.053646	0.054581	0.054810	0.055796	0.051936	0.042182
17	0.054313	0.000032	0.054451	0.039065	0.059552	0.052579	0.062431	0.058019	0.059723	0.047866
18	0.056849	0.000032	0.059619	0.074189	0.059068	0.064349	0.054034	0.064812	0.068084	0.046102
19	0.057142	0.000016	0.067095	0.075971	0.062109	0.052290	0.054580	0.065152	0.065476	0.047631

20 rows × 500 columns

In [294…
```
classConditional.to_csv('classConditionals.csv')
```

In [296…
```
def posterior(str): #This is the posterior a.k.a q_i of our model

    p = dict()

    for foldername in range(len(folders)):
        p[folders[foldername]] = np.log(priors[folders[foldername]])
        for i in range(len(str)):
            p[folders[foldername]] += np.log(classConditional.values[foldername][i]) *

    return max(p.items(), key = operator.itemgetter(1))[0]
```

In [297…
```
def classifier(test_data):
    pred = list()
    for i in range(len(test_data)):
        pred.append(posterior(test_data[i]))
    return pred
```

In [305…
```
y_pred=classifier(X_test)
```

In [299…

```
def accuracy(predicted,actual):
    matchs = sum(predicted == actual)
    total = len(actual)
    return matchs/total
```

Accuracy of our Model

In [301…

```
accuracy(y_pred,Y_test)
```

Out[301… 0.664

In [ ]: