AIM:Problems may be designed for the following topics so that students can get hands on experience in using python for natural language processing: • Part of Speech tagging • N-gram and smoothening • Chunking

```
import nltk
from nltk import word_tokenize
nltk.download('punkt')
nltk.download('average_perceptrom_tagger')
nltk.download('tagsets')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Error loading average_perceptrom_tagger: Package
[nltk_data]     'average_perceptrom_tagger' not found in index
[nltk_data] Downloading package tagsets to /root/nltk_data...
[nltk_data]   Unzipping help/tagsets.zip.
True
```

```
nltk.help.upenn_tagset()
```

```
RB: adverb
    occasionally unabatingly maddeningly adventurously professedly
    stirringly prominently technologically magisterially predominately
    swiftly fiscally pitilessly ...
RBR: adverb, comparative
    further gloomier grander graver greater grimmer harder harsher
    healthier heavier higher however larger later leaner lengthier less-
    perfectly lesser lonelier longer louder lower more ...
RBS: adverb, superlative
    best biggest bluntest earliest farthest first furthest hardest
    heartiest highest largest least less most nearest second tightest worst
RP: particle
    aboard about across along apart around aside at away back before behind
    by crop down ever fast for forth from go high i.e. in into just later
    low more off on open out over per pie raising start teeth that through
    under unto up up-pp upon whole with you
SYM: symbol
    % & ' '' ''. ) ). * + ,. < = > @ A[fj] U.S U.S.S.R * ** ***
TO: "to" as preposition or infinitive marker
    to
UH: interjection
    Goodbye Goody Gosh Wow Jeepers Jee-sus Hubba Hey Kee-reist Oops amen
    huh howdy uh dammit whammo shucks heck anyways whodunnit honey golly
    man baby diddle hush sonuvabitch ...
VB: verb, base form
    ask assemble assess assign assume atone attention avoid bake balkanize
    bank begin behold believe bend benefit bevel beware bless boil bomb
    boost brace break bring broil brush build ...
VBD: verb, past tense
    dipped pleaded swiped regummed soaked tidied convened halted registered
    cushioned exacted snubbed strode aimed adopted belied figgered
    speculated wore appreciated contemplated ...
```

```
    VBG: verb, present participle or gerund
        telegraphing stirring focusing angering judging stalling lactating
        hankerin' alleging veering capping approaching traveling besieging
        encrypting interrupting erasing wincing ...
    VBN: verb, past participle
        multihulled dilapidated aerosolized chaired languished panelized used
        experimented flourished imitated reunifed factored condensed sheared
        unsettled primed dubbed desired ...
    VBP: verb, present tense, not 3rd person singular
        predominate wrap resort sue twist spill cure lengthen brush terminate
        appear tend stray glisten obtain comprise detest tease attract
        emphasize mold postpone sever return wag ...
    VBZ: verb, present tense, 3rd person singular
        bases reconstructs marks mixes displeases seals carps weaves snatches
        slumps stretches authorizes smolders pictures emerges stockpiles
        seduces fizzes uses bolsters slaps speaks pleads ...
    WDT: WH-determiner
        that what whatever which whichever
    WP: WH-pronoun
        that what whatever whatsoever which who whom whosoever
    WP$: WH-pronoun, possessive
        whose
    WRB: Wh-adverb
        how however whence whenever where whereby whereever wherein whereof why
    ``: opening quotation mark
        ` ``
```

```
nltk.download('averaged_perceptron_tagger')
sentence=word_tokenize("Third wave of corona virus is here")
nltk.pos_tag(sentence)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[('Third', 'JJ'),
 ('wave', 'NN'),
 ('of', 'IN'),
 ('corona', 'NN'),
 ('virus', 'NN'),
 ('is', 'VBZ'),
 ('here', 'RB')]
```

```
#n grams in sentence analysis
import numpy as np
import pandas as pd
from sklearn import model_selection,naive_bayes,svm
```

```
df=pd.read_csv('/content/Reviews.csv',engine='python',error_bad_lines=False)
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: FutureWarn
```

```
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessD
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype  
---  ------                  --------------   -----  
 0   Id                      568454 non-null  int64  
 1   ProductId               568454 non-null  object 
 2   UserId                  568454 non-null  object 
 3   ProfileName             568438 non-null  object 
 4   HelpfulnessNumerator    568454 non-null  int64  
 5   HelpfulnessDenominator  568454 non-null  int64  
 6   Score                   568454 non-null  int64  
 7   Time                    568454 non-null  int64  
 8   Summary                 568427 non-null  object 
 9   Text                    568454 non-null  object 
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

```
df.shape
```

```
(568454, 10)
```

```
df.isnull().sum()
```

```
Id                      0
ProductId               0
UserId                  0
ProfileName            16
HelpfulnessNumerator    0
HelpfulnessDenominator  0
Score                   0
Time                    0
Summary                27
Text                    0
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df.isnull().sum()
```

```
Id                      0
ProductId               0
UserId                  0
ProfileName             0
HelpfulnessNumerator    0
HelpfulnessDenominator  0
Score                   0
Time                    0
Summary                 0
Text                    0
dtype: int64
```

```
df['Score'].value_counts()
```

```
5    363111
4     80655
1     52264
3     42638
2     29743
Name: Score, dtype: int64
```

```
df['positive ratings']=np.where(df['Score']>=3,1,0)
```

```
df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessD
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	

```
df['positive ratings'].value_counts()
```

```
1    486404
0     82007
Name: positive ratings, dtype: int64
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(df['Text'],df['positive ratings'],random_state
```

```
print(x_train)
```

```
430965    When I was a kid in Alabama, my dad used Dale'...
137224    As a huge fan of the Gears of War series this ...
497729    These are the most amazing lollipops I have ev...
453888    These were probably the best Oreo's I have eve...
526447    We're new to this brand, but not to healthy ea...
                              ...
385187    the flavor palet for this coffee is deep and r...
321525    What's not to love about this delicious chocol...
441668    My daughter seems to really like EB's organic ...
```

```
239517    I decided to try these cookies because of a wa...
103912    This product is great for the price. My MAJOR ...
Name: Text, Length: 426308, dtype: object
```

```
print(y_train)
```

```
430965    1
137224    1
497729    1
453888    1
526447    1
          ..
385187    1
321525    1
441668    1
239517    1
103912    1
Name: positive ratings, Length: 426308, dtype: int64
```

```
df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessD

```
from sklearn.feature_extraction.text import CountVectorizer

vector=CountVectorizer(min_df=5,ngram_range=(1,2)).fit(x_train)

vector.get_feature_names()
```

```
      '11 28',
      '11 29',
      '11 2oz',
      '11 30',
      '11 45',
      '11 50',
      '11 5oz',
      '11 64',
      '11 75',
      '11 82',
      '11 95',
      '11 99',
      '11 am',
      '11 and',
      '11 as',
      '11 at',
      '11 bags',
      '11 because',
      '11 boxes',
      '11 br',
      '11 bucks',
      '11 but',
      '11 cans',
      '11 carb',
      '11 cents',
      '11 coffee',
      '11 days',
      '11 despite',
      '11 different',
      '11 dogs',
      '11 dollars',
      '11 don',
      '11 edit',
      '11 feel',
      '11 fl',
      '11 fluid',
      '11 for',
      '11 gms',
      '11 grams',
      '11 have',
      '11 he',
      '11 in',
      '11 is',
      '11 it',
      '11 lb',
```

```
 '11 lbs',
 '11 less',
 '11 min',
 '11 minutes',
 '11 mo',
 '11 month',
 '11 months',
 '11 more',
 '11 mos',
 '11 net',
 ...]
```

```
len(vector.get_feature_names())
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: F
  warnings.warn(msg, category=FutureWarning)
563947
```

```
x_train_vectorized=vector.transform(x_train)
```

```
from sklearn.naive_bayes import MultinomialNB
model=MultinomialNB()
model.fit(x_train_vectorized,y_train)
```

```
MultinomialNB()
```

```
pred=model.predict(vector.transform(x_test))
```

```
pred
```

```
array([0, 1, 1, ..., 1, 1, 0])
```

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
[[ 17214   3331]
 [  6915 114643]]
              precision    recall  f1-score   support

           0       0.71      0.84      0.77     20545
           1       0.97      0.94      0.96    121558

    accuracy                           0.93    142103
   macro avg       0.84      0.89      0.86    142103
```

```
weighted avg       0.93      0.93      0.93    142103
```

```
#auc score
from sklearn.metrics import roc_auc_score
print('AUC score is:',roc_auc_score(y_test,pred))
```

```
AUC score is: 0.8904908349197139
```

✓ 0s completed at 9:09 PM