Feature Engineering

```
from sklearn.feature_extraction import text
from os import path
ObjRead = open("/content/AST_A.txt", encoding='utf-8')
corpus = ObjRead.read();
ObjRead.close()
input=[corpus]
vectorizer = text.CountVectorizer(binary=True).fit(input)
vectorized_text = vectorizer.transform(input)
print(vectorized_text.todense())
```

```
[[1 1 1 ... 1 1 1]]
```

The TF-IDF transformation is a technique that, after counting how many times a token appears in a phrase, divides the value by the number of documents in which the token appears. Using this technique, the vectorizer deems a word less important, even if it appears many times in a text, when it also finds that word in other texts.

```
from sklearn.feature_extraction.text import TfidfTransformer
TfidF = text.TfidfTransformer(norm='l1')
tfidf = TfidF.fit_transform(vectorized_text)
phrase=0
total=2
for word in vectorizer.vocabulary_:
  pos = vectorizer.vocabulary_[word]
  value = list(tfidf.toarray()[phrase])[pos]
  if value !=0:
      print ("%10s: %0.3f" % (word, value))
      total += value
print ('\nSummed values of a phrase: %0.1f' % total)
```

```
     after: 0.001
    nearly: 0.001
      five: 0.001
     month: 0.001
     delay: 0.001
    caused: 0.001
        by: 0.001
   missing: 0.001
information: 0.001
 officials: 0.001
        at: 0.001
       the: 0.001
   federal: 0.001
  aviation: 0.001
administration: 0.001
      have: 0.001
     begun: 0.001
 reviewing: 0.001
   license: 0.001
application: 0.001
       for: 0.001
  proposed: 0.001
 spaceport: 0.001
    camden: 0.001
   kenneth: 0.001
      wong: 0.001
   manager: 0.001
        of: 0.001
       faa: 0.001
 licensing: 0.001
       and: 0.001
evaluation: 0.001
  division: 0.001
  informed: 0.001
    county: 0.001
commission: 0.001
  chairman: 0.001
        in: 0.001
    letter: 0.001
     dated: 0.001
      june: 0.001
        28: 0.001
      that: 0.001
   stalled: 0.001
    review: 0.001
       had: 0.001
      been: 0.001
 restarted: 0.001
        on: 0.001
```

```
          feb: 0.001
           12: 0.001
         2019: 0.001
           we: 0.001
   determined: 0.001
           to: 0.001
           be: 0.001
          not: 0.001
     complete: 0.001
       enough: 0.001
```

Using this new TF-IDF model rescales the values of important words and makes them comparable between each text in the corpus. To recover part of the ordering of the text before the BoW transformation, adding n-grams

Bi-Grams. The following example uses CountVectorizer to model n-grams in the range of (2, 2), that is, bigrams.

```
bigrams = text.CountVectorizer(ngram_range=(2,2))
print (bigrams.fit(input).vocabulary_)
```

```
{'after nearly': 159, 'nearly five': 2772, 'five month': 1564, 'month delay': 2704, 'delay caused': 1190, 'caused by': 881, 'by missing': 784, 'missing information': 26
```

Stemming and Wordstop Removal. The following example demonstrates how to perform stemming and remove stop words from a sentence. It begins by training an algorithm to perform the required analysis using a test sentence. Afterward, the example checks a second sentence for words that appear in the first. The first output shows the stemmed words. All the stop words are missing as well. For example, you don't see the words so, he, all, or the. The second output shows how many times each stemmed word appears in the test sentence.

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
# load data
filename = '/content/AST_A.txt'
file = open(filename, 'rt')
text = file.read()
file.close()
# split into words
from nltk.tokenize import word_tokenize
tokens = word_tokenize(text)
# convert to lower case
tokens = [w.lower() for w in tokens]
# remove punctuation from each word
import string
table = str.maketrans('', '', string.punctuation)
stripped = [w.translate(table) for w in tokens]
# remove remaining tokens that are not alphabetic
words = [word for word in stripped if word.isalpha()]
# filter out stop words
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
words = [w for w in words if not w in stop_words]
' '.join(words)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
'nearly fivemonth delay caused missing information officials federal aviation administration begun reviewing license application proposed spaceport camden k
enneth wong manager faa licensing evaluation division informed camden county commission chairman letter dated june stalled review restarted feb determine
d application complete enough begin review due four outstanding issues wong wrote letter requested information concerning environmental review mitigation
potential risk fire analysis individual risk ability account manage population might exposed risk overflight launch vehicle received additional information june ca
```

Frequency Distribution

```
import nltk
# Tokenize text
from nltk.tokenize import sent_tokenize
from os import path
ObjRead = open("/content/AST_Clean.txt", encoding='utf-8')
text = ObjRead.read();
ObjRead.close()
tokenized_text=sent_tokenize(text)
print(tokenized_text)

# Tokenize words
from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

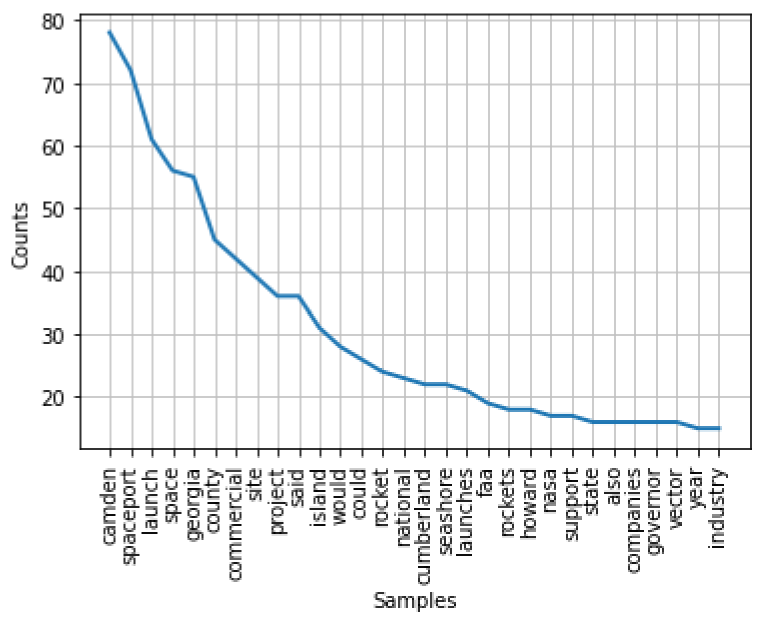
```
# Frequency Distribution
from nltk.probability import FreqDist
fdist = FreqDist(tokenized_word)
print(fdist)

# Ten most common words
fdist.most_common(10)

# Frequency Distribution Plots
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()

```

```
['nearly fivemonth delay caused missing information officials federal aviation administration begun reviewing license application proposed spaceport camden ke
['nearly', 'fivemonth', 'delay', 'caused', 'missing', 'information', 'officials', 'federal', 'aviation', 'administration', 'begun', 'reviewing', 'license', 'application', 'proposed'
<FreqDist with 1622 samples and 3949 outcomes>
```



Text Summarization

```
from os import path
ObjRead = open("/content/AST_A.txt")
txtContent = ObjRead.read();
ObjRead.close()
```

```
from gensim.summarization import summarize
print (summarize(txtContent))
```

```
After a nearly five-month delay caused by missing information, officials at the Federal Aviation Administration have begun reviewing a license application for th
Camden County is planning to build and operate a commercial spaceport where operators would launch liquid-fueled , small to medium-large vehicles .
We are now at T-minus 1 , a final decision by the FAA is the only outstanding item , Steve Howard , Spaceport Camden Project lead and Camden County admi
I have proudly supported Spaceport Camden from the first time I heard about this amazing project and all it can do for Georgia , said Gov.
The state of Georgia is firmly behind Camden County and we encourage the FAA to swiftly approve its launch site operator’s license application .
It’s too dangerous , Weinkle said , because the rockets would have to launch over Little Cumberland Island where the private owners and guests can’t be force
Camden’s rocket launching fantasy will only degrade these precious resources: polluting the salt marsh and tidal creeks , closing waters to commercial and rec
The Camden (Georgia) County Board of Commissioners sees the spaceport as an economic boon , one that could both attract commercial space flight compa
Unlike other communities along Georgia's 100-mile-long coast that have developed thriving tourist communities on their barrier islands , Camden County can't
With the Federal Aviation Administration poised to release a final environmental impact statement on the proposed launch facility by mid-December , with a go-
It's a scenario that paints a picture of as many as a dozen commercial launches a year that would arc across the northern end of Cumberland Island National S
While NASA long has handled space launches for the United States , recent years have seen the rise of private rocket companies that see a profit to be made
We have nothing against rocket ships unless they’re being launched over one of the largest maritime wildernesses on the East Coast , said Emily Jones , who
Spaceport Camden is committed to learning from NASA’s best practices for nighttime lighting and is already working with environmental groups and federal an
Kevin Lang , an Athens , Georgia , attorney working with a group opposed to the spaceport , pointed out during a phone call that the national seashore is visite
One of our biggest concerns is the fact that a lot of rocket failures do occur early , in the first stage of launch , and a rocket failure over the Cumberland Island N
Analysis of a proposed spaceport in Camden County shows the project has potential to create new jobs , increase tourism , boost the local economy and positi
McKay said the university study looked at the impacts of both construction of a spaceport and operations once launches are conducted at the site .
What makes Camden County such an attractive site when other commercial spaceports have struggled is the central location near other facilities including site
Bob Scaringe , a consultant with AVG Communications , said the commercial space industry continues to grow because private launch sites can launch rocket
Governor Casey each presented benefits Spaceport Camden will offer the State of Georgia and commercial space operators .
The Spaceport Camden project utilizes Georgia’s high-technology industry and cutting-edge research capabilities , bringing significant economic opportunities
Governor Casey Cagle echoed Governor Deals sentiments , noting Camden is the ideal region for space vehicle manufacturing due to its location … and also
Spaceport Camden , coupled with NASA’s presence in Alabama and Florida along with Georgia’s expertise in aerospace engineering and manufacturing has th
Other candidates for Georgia Governor , such as Secretary of State Brian Kemp and former Minority Leader Stacey Abrams , have also given support for Spac
Today , the Camden County Board of Commissioners formally submitted its application for a Launch Site Operator License LSOL to the Federal Aviation Admi
Commission Vice Chairman Gary Blount added , This launch site operator license application has been a strategic priority of the Board of County Commission
Camden will retain that title again , said Spaceport Camden project leader and County Administrator Steve Howard .
Spaceport Camden is a visionary project that will bring high-paying aerospace jobs to Southeast Georgia while supporting a new wave of STEM education and
Former Speaker of the House and member of the National Space Council Users Advisory Group , Newt Gingrich , joined Camden officials in celebrating the su
Spaceport Camden is an ideal location for the safe launch of the small satellites and rockets that are rapidly becoming the most important segment of this new
The Spaceport , local colleges and the nearby opportunities for the development , manufacture and test of satellites , ground hardware and rockets are valuabl
In June , both US Senators and the entire House Delegation from Georgia sent letters to the FAA supporting the Spaceport Camden draft environmental impac
After a two-year open comment process , a potential commercial space site on Georgia’s coast is moving forward with bipartisan political and business leaders
Should the site be approved , the project could almost immediately begin bringing some of the soon-to-be trillion-dollar commercial space market to the state .
```

```
But Camden County administrator and project lead Steve Howard says that’s a waste .
Additional support notes were submitted by former Speaker of the House and National Space Council Advisory Group member Newt Gingrich ; the private Con
Now , Camden County will submit an official launch site operator license application , after which the FAA has six months to issue a final decision on a launch s
Howard says the FAA will examine all facets of the project — economic impact , environmental factors and more — to determine whether the site will be approv
The support from elected officials , commercial space companies , and space policy advocates all reaches the same conclusion : The United States needs mor
Should Spaceport Camden become a reality , Howard says they have already had multiple commercial space companies reach out expressing interest in using
This launch represents both the first customer-funded launch operation for the new space commercial launch industry , as well as the first launch out of the hist
Our historic launch today is a testament to the hard work of the Vector team , as well as support from NASA and Spaceport Camden .
A primary objective of the launch was to demonstrate and evaluate a next-generation engine injector developed through collaborative research with the NASA
Georgia looks forward to working with commercial space companies , like Vector , as we begin the next chapter of space exploration and innovation , he said .
An ongoing environmental assessment conducted by the Federal Aviation Administration due to be completed later this year will likely determine if rockets are
Detractors say the county has already spent millions of dollars on a project that will never be approved because most of the launch trajectories are over Cumbe
Built through partnerships with private companies , the area could become the first exclusively commercial spaceport on the East Coast ; the others in Florida a
Spaceport Camden is strategically located on the coast , which would allow rockets to launch east over mostly open ocean , posing little risk to populated areas
So far , many commercial space companies have shown interest in the place , according to county officials .
Camden County , which the spaceport is named after , is currently working to get the site licensed for launches by the Federal Aviation Administration .
I think we have the opportunity to build the first exclusive non-federal range on the East Coast , Steve Howardm , the Camden County administrator and head o
Then , a little less than five years ago , a space company reached out to the state of Georgia , looking for a new location to launch its rockets .
That initial meeting prompted Camden County officials to meet with other private spaceflight companies to gauge interest in a Georgia spaceport where the boo
For a while , the Spaceport Camden initiative was just a promising idea , but now the state of Georgia has shown it’s serious about the project .
In August , a spaceflight startup named Vector launched one of its test rockets from the site — though the vehicle didn’t reach orbit .
Currently , 10 sites throughout the US hold FAA licenses to operate as commercial spaceports .
A report done by space consulting firm Astralytical found that Spaceport Camden could be a great home for businesses that not only launch from the site , but
```

```
print ('Summary:')
print (summarize(txtContent, word_count=600))
```

```
Summary:
We are now at T-minus 1 , a final decision by the FAA is the only outstanding item , Steve Howard , Spaceport Camden Project lead and Camden County admi
Camden’s rocket launching fantasy will only degrade these precious resources: polluting the salt marsh and tidal creeks , closing waters to commercial and rec
The Camden (Georgia) County Board of Commissioners sees the spaceport as an economic boon , one that could both attract commercial space flight compar
With the Federal Aviation Administration poised to release a final environmental impact statement on the proposed launch facility by mid-December , with a go-
While NASA long has handled space launches for the United States , recent years have seen the rise of private rocket companies that see a profit to be made
Analysis of a proposed spaceport in Camden County shows the project has potential to create new jobs , increase tourism , boost the local economy and positi
The Spaceport Camden project utilizes Georgia’s high-technology industry and cutting-edge research capabilities , bringing significant economic opportunities
Today , the Camden County Board of Commissioners formally submitted its application for a Launch Site Operator License LSOL to the Federal Aviation Admir
The Spaceport , local colleges and the nearby opportunities for the development , manufacture and test of satellites , ground hardware and rockets are valuabl
Additional support notes were submitted by former Speaker of the House and National Space Council Advisory Group member Newt Gingrich ; the private Con
Should Spaceport Camden become a reality , Howard says they have already had multiple commercial space companies reach out expressing interest in using
This launch represents both the first customer-funded launch operation for the new space commercial launch industry , as well as the first launch out of the hist
Camden County , which the spaceport is named after , is currently working to get the site licensed for launches by the Federal Aviation Administration .
Then , a little less than five years ago , a space company reached out to the state of Georgia , looking for a new location to launch its rockets .
```

Topic Modeling using NMF

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words='english')
tfidf = vectorizer.fit_transform(input)
from sklearn.decomposition import NMF
n_topics = 5
nmf = NMF(n_components=n_topics, random_state=101).fit(tfidf)
feature_names = vectorizer.get_feature_names()
n_top_words = 15
for topic_idx, topic in enumerate(nmf.components_):
  print ("Topic #%d:" % (topic_idx+1),)
  print (" ".join([feature_names[i] for i in
              topic.argsort()[:-n_top_words - 1:-1]]))
```

```
Topic #1:
camden spaceport launch space georgia county commercial site said project island rocket national cumberland launches
Topic #2:
casey support economic small near flight park smart mid location georgia long approved brian lead
Topic #3:
12 running canaveral knowledge ocean parking governor advance owned seashore bring residents 000 officials press
Topic #4:
rural operators central lie followers visited steven story mitigate asked discussed assessment examine offer betting
Topic #5:
vertical homes structures involves officials party sector loses 15 hold december lead attract wallops south
```

By using the argsort method, you can get the indexes of the top associations, whose high values indicate that they are the most representative words.

```
print (nmf.components_[0,:].argsort()[:-n_top_words-1:-1])
# Gets top words for topic 0
```

```
[ 222 1327  779 1323  606  340  289 1298 1235 1102  732 1218  926  353
  781]
```

Identify the representative words

```
print (vectorizer.get_feature_names()[222])
print (vectorizer.get_feature_names()[1327])
print (vectorizer.get_feature_names()[779])
print (vectorizer.get_feature_names()[1323])
print (vectorizer.get_feature_names()[606])
print (vectorizer.get_feature_names()[340])
```

```
camden
spaceport
launch
space
georgia
county
```

Text Classification with SVC

```
import numpy as np
import pandas as pd
from pandas import DataFrame
dataset = pd.read_csv('/content/MIA.csv')
data=pd.DataFrame(data=dataset, columns=['text','sentiment'])
data.head()
```

	text	sentiment
0	Few parts of the core of Central Florida are s...	0
1	“We are a busy airport,” Brown said.	0
2	“There are not too many places in metropolita...	0
3	Brown said his expectation is that flight path...	0
4	“It will narrow the tracks that aircraft will ...	0

```
X=data['text']
y=data['sentiment']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=.5, random_state=125)
```

```
from sklearn.feature_extraction import text
vectorizer = text.CountVectorizer(ngram_range=(1,2),
                stop_words='english').fit(X)
TfidF = text.TfidfTransformer()
X = TfidF.fit_transform(vectorizer.transform(X))
Xt = TfidF.transform(vectorizer.transform(X_test))
```

```
from sklearn.svm import LinearSVC
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [0.01, 0.1, 1.0, 10.0, 100.0]}
clf = GridSearchCV(LinearSVC(loss='hinge',
                random_state=101), param_grid)
clf = clf.fit(X, y)
y_pred=clf.predict(Xt)
print ("Best parameters: %s" % clf.best_params_)
```

```
Best parameters: {'C': 0.01}
```

```
from sklearn.metrics import accuracy_score
solution = clf.predict(Xt)
print("Achieved accuracy: %0.3f" %
     accuracy_score(y_test, solution))
```

```
Achieved accuracy: 0.600
```

```
print(X_test[y_test!=solution])
```

```
49   I don't give a damn about good windows, I live...
24   Gilderman said the current distribution of the...
10   The FAA plans to conduct its environmental stu...
14   Members of the Miami International Airport Noi...
45   NextGen’s promises to save on fuel and travel ...
33   Public input will be allowed during the enviro...
16   Noise complaints have spiked in communities ne...
19   The federal agency said NextGen will save $15....
28   Satellite-based routes have resulted in flight...
```

```
26    It’s going to be nonstop "He predicts concerne...
Name: text, dtype: object
```

Classification Report

```
from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.60      1.00      0.75        15
           1       0.00      0.00      0.00         5
           2       0.00      0.00      0.00         5

    accuracy                           0.60        25
   macro avg       0.20      0.33      0.25        25
weighted avg       0.36      0.60      0.45        25
```

Confusion Matrix

```
from sklearn.metrics import classification_report, confusion_matrix
cm=confusion_matrix(y_test, y_pred)
cm[::-1, ::-1]
```

```
array([[ 0,  0,  5],
       [ 0,  0,  5],
       [ 0,  0, 15]])
```

Text Classification

How to create a classifier

First create a training and test sample

```
train = [
...     ('I love this sandwich.', 'pos'),
...     ('this is an amazing place!', 'pos'),
...     ('I feel very good about these beers.', 'pos'),
...     ('this is my best work.', 'pos'),
...     ("what an awesome view", 'pos'),
...     ('I do not like this restaurant', 'neg'),
...     ('I am tired of this stuff.', 'neg'),
...     ("I can't deal with this", 'neg'),
...     ('he is my sworn enemy!', 'neg'),
...     ('my boss is horrible.', 'neg')
... ]
test = [
...     ('the beer was good.', 'pos'),
...     ('I do not enjoy my job', 'neg'),
...     ("I ain't feeling dandy today.", 'neg'),
...     ("I feel amazing!", 'pos'),
...     ('Gary is a friend of mine.', 'pos'),
...     ("I can't believe I'm doing this.", 'neg')
... ]
```

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
from textblob.classifiers import NaiveBayesClassifier
cl = NaiveBayesClassifier(train)
```

Classify a text

```
cl.classify("This is an amazing library!")
```

```
'pos'
```

```
prob_dist = cl.prob_classify("This one's a doozy.")
```

```
print('Overall Sentiment: ',prob_dist.max())
```

```
Overall Sentiment:  pos
```

Provide the probability distributions

```
prob_dist = cl.prob_classify("This one's a doozy.")
prob_dist.max() # classify probability of statement
```

```
'pos'
```

```
round(prob_dist.prob("pos"), 2) # probability of text being positive
```

```
0.63
```

```
round(prob_dist.prob("neg"), 2) # probability of text being negative
```

```
0.37
```

```
cl.classify("I am really upset!") # test on a sentence
```

```
'neg'
```

Sentiment Analysis with TextBlob

```
from textblob import TextBlob
from os import path
ObjRead = open("/content/AST_A.txt")
txtContent = ObjRead.read();
ObjRead.close()
testimonial = TextBlob(txtContent)
testimonial.sentiment
```

```
Sentiment(polarity=0.08435739381391548, subjectivity=0.4159659832920702)
```

```
import nltk
nltk.download('movie_reviews')
nltk.download('punkt')
from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer
from os import path
ObjRead = open("/content/AST_A.txt")
txtContent = ObjRead.read();
ObjRead.close()
blob = TextBlob(txtContent, analyzer=NaiveBayesAnalyzer())
blob.sentiment
```

```
[nltk_data] Downloading package movie_reviews to /root/nltk_data...
[nltk_data]   Unzipping corpora/movie_reviews.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
Sentiment(classification='pos', p_pos=1.0, p_neg=6.99671245902884e-109)
```

Tokenize Text

```
from nltk.tokenize import word_tokenize
tokenizer = word_tokenize(txtContent)
print(tokenizer)
```

```
['After', 'a', 'nearly', 'five-month', 'delay', 'caused', 'by', 'missing', 'information', ',', 'officials', 'at', 'the', 'Federal', 'Aviation', 'Administration', 'have', 'begun', 'reviewin
```

Chunker

```
import nltk
nltk.download('conll2000')
from textblob import TextBlob
from textblob.np_extractors import ConllExtractor
extractor = ConllExtractor()
blob = TextBlob(txtContent, np_extractor=extractor)
blob.noun_phrases
```

```
[nltk_data] Downloading package conll2000 to /root/nltk_data...
```

```
[nltk_data]   Unzipping corpora/conll2000.zip.
WordList(['five-month delay', 'federal aviation administration', 'license application', 'proposed spaceport camden', 'kenneth wong', 'faa licensing', 'evaluation divi
```

Entity Recognition

```
from nltk import word_tokenize, pos_tag, ne_chunk
nltk.download('maxent_ne_chunker')
nltk.download('words')
my_string= 'Antonio Guttierez is the Secretary General of the United Nations in New York in the USA'
print(ne_chunk(pos_tag(word_tokenize(my_string))))
```

```
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping chunkers/maxent_ne_chunker.zip.
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Unzipping corpora/words.zip.
(S
  (PERSON Antonio/NNP)
  (PERSON Guttierez/NNP)
  is/VBZ
  the/DT
  Secretary/NNP
  General/NNP
  of/IN
  the/DT
  (ORGANIZATION United/NNP Nations/NNPS)
  in/IN
  (GPE New/NNP York/NNP)
  in/IN
  the/DT
  (ORGANIZATION USA/NNP))
```

POS Tagger. List of taggers: https://www.learntek.org/blog/categorizing-pos-tagging-nltk-python/

```
import nltk
nltk.download('averaged_perceptron_tagger')
from textblob import TextBlob
from textblob.taggers import NLTKTagger
nltk_tagger = NLTKTagger()
blob = TextBlob(txtContent, pos_tagger=nltk_tagger)
blob.pos_tags
```

```
 ('Georgians', 'NNPS'),
 ('have', 'VBP'),
 ('recognized', 'VBN'),
 ('that', 'IN'),
 ('our', 'PRP$'),
 ('thriving', 'VBG'),
 ('economy', 'NN'),
 ('and', 'CC'),
 ('fantastic', 'JJ'),
 ('quality', 'NN'),
 ('of', 'IN'),
 ('life', 'NN'),
 ('depend', 'NN'),
 ('on', 'IN'),
 ('the', 'DT'),
 ('clean', 'JJ'),
 ('water', 'NN'),
 ('productive', 'JJ'),
 ('fisheries', 'NNS'),
 ('and', 'CC'),
 ('spectacular', 'JJ'),
 ('scenery', 'NN'),
 ('of', 'IN'),
 ('the', 'DT'),
 ('salt', 'NN'),
 ('marshes', 'NNS'),
 ('barrier', 'JJR'),
 ('islands', 'NNS'),
 ('and', 'CC'),
 ('coastal', 'JJ'),
 ('waterways', 'NNS'),
 ('Camden', 'NNP'),
 ('"', 'NNP'),
 ('s', 'NN'),
 ('rocket', 'NN'),
 ('launching', 'VBG'),
 ('fantasy', 'NN'),
 ('will', 'MD'),
 ('only', 'RB'),
 ('degrade', 'VB'),
 ('these', 'DT'),
 ('precious', 'JJ'),
 ('resources', 'NNS'),
 ('polluting', 'VBG'),
 ('the', 'DT'),
 ('salt', 'NN'),
 ('marsh', 'NN'),
 ('and', 'CC'),
 ('tidal', 'JJ'),
 ('creeks', 'NN'),
 ('closing', 'VBG'),
 ('waters', 'NNS'),
 ('to', 'TO'),
 ('commercial', 'JJ'),
 ('and', 'CC'),
 ('recreational', 'JJ'),
 ('fishing', 'NN'),
 ('disrupting', 'VBG'),
 ('traffic', 'NN'),
 ('to', 'TO'),
```

Parser

```python
from textblob import TextBlob
from textblob.parsers import PatternParser
blob = TextBlob(txtContent, parser=PatternParser())
blob.parse()
```

```
'After/IN/B-PP/B-PNP a/DT/B-NP/I-PNP nearly/RB/I-NP/I-PNP five-month/JJ/I-NP/I-PNP delay/NN/I-NP/I-PNP caused/VBN/B-VP/I-PNP by/IN/B-PP/B-PNP mi
ssing/VBG/B-VP/I-PNP information/NN/B-NP/I-PNP ,/,/O/O officials/NNS/B-NP/O at/IN/B-PP/B-PNP the/DT/B-NP/I-PNP Federal/NNP/I-NP/I-PNP Aviation/NN
P/I-NP/I-PNP Administration/NNP/I-NP/I-PNP have/VBP/B-VP/O begun/VBN/I-VP/O reviewing/VBG/I-VP/O a/DT/B-NP/O license/NN/I-NP/O application/NN/I-
NP/O for/IN/B-PP/O the/DT/O/O proposed/VBN/B-VP/O Spaceport/NNP/B-NP/O Camden/NNP/I-NP/O ./././O/O\nKenneth/NNP/B-NP/O Wong/NNP/I-NP/O ,/,/
O/O manager/NN/B-NP/O of/IN/B-PP/B-PNP the/DT/B-NP/I-PNP FAA/NNP/I-NP/I-PNP Licensing/NNP/I-NP/I-PNP and/CC/O/O Evaluation/NN/B-NP/O Divisi
```

Compare Taggers

```python
from textblob import Blobber
from textblob.taggers import NLTKTagger
tb = Blobber(pos_tagger=NLTKTagger())
blob1 = tb("This is a blob.")
blob2 = tb("This is another blob.")
blob1.pos_tagger is blob2.pos_tagger
```

```
True
```

Evaluate overall sentiment

```python
from textblob import TextBlob
blob = TextBlob("The beer is good. But the hangover is horrible.", classifier=cl)
blob.classify()
```

```
'pos'
```

```python
cl.show_informative_features(5)
```

```
Most Informative Features
          contains(my) = True              neg : pos    =      1.7 : 1.0
          contains(an) = False             neg : pos    =      1.6 : 1.0
           contains(I) = False             pos : neg    =      1.4 : 1.0
           contains(I) = True              neg : pos    =      1.4 : 1.0
          contains(my) = False             pos : neg    =      1.3 : 1.0
```

```python
blob = TextBlob("The beer is good. But the hangover is horrible.", classifier=cl)
for s in blob.sentences:
    print(s)
    print(s.classify())
```

```
The beer is good.
pos
But the hangover is horrible.
neg
```

```python
print('Classifier Accuracy: %.3f' % cl.accuracy(test))
```

```
Classifier Accuracy: 0.833
```

```python
print('Extract features: ', cl.extract_features(txtContent))
```

```
Extract features:  {'contains(an)': True, 'contains(best)': True, 'contains(sworn)': False, 'contains(like)': True, 'contains(horrible)': False, 'contains(my)': True, 'con
```

```
from textblob import TextBlob
from textblob.classifiers import MaxEntClassifier
print('Maximum Entropy Classifier: ', MaxEntClassifier(test))

Maximum Entropy Classifier:  <MaxEntClassifier trained on 6 instances>
```

```
new_data = [('She is my best friend.', 'pos'),
...             ("I'm happy to have a new friend.", 'pos'),
...             ("Stay thirsty, my friend.", 'pos'),
...             ("He ain't from around here.", 'neg')]
cl.update(new_data)
print('Accuracy with new data: ', cl.accuracy(test))

Accuracy with new data:  1.0
```

Did we update the data?

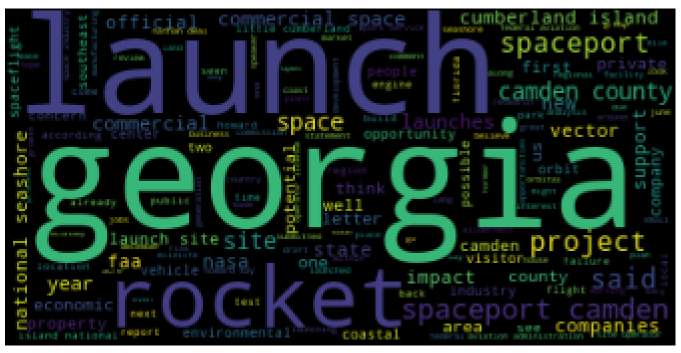
```
cl.update(new_data)

True
```

WordCloud

```
import matplotlib.pyplot as plt
%matplotlib inline
import nltk
nltk.download('punkt')
nltk.download('stopwords')
# load data
filename = '/content/AST_A.txt'
file = open(filename, 'rt')
text = file.read()
file.close()
# split into words
from nltk.tokenize import word_tokenize
word_tokens = word_tokenize(text)
# convert to lower case
word_tokens = [w.lower() for w in word_tokens]
# remove punctuation from each word
import string
table = str.maketrans('', '', string.punctuation)
stripped = [w.translate(table) for w in word_tokens]
# remove remaining tokens that are not alphabetic
words = [word for word in stripped if word.isalpha()]
# filter out stop words
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
words = [w for w in words if not w in stop_words]
' '.join(words)
from wordcloud import WordCloud
document_file_path = '/content/AST_A.txt'
text_from_file = open ( '/content/AST_A.txt') . read ()
stop_words = set ( nltk . corpus . stopwords . words ( 'english' ))
word_tokenize = word_tokenize( text_from_file )
filtered_sentence = [ w for w in word_tokens if not w in stop_words ]
wl_space_split = " " . join ( filtered_sentence )
my_wordcloud = WordCloud () . generate ( wl_space_split )
plt . imshow ( my_wordcloud )
plt . axis ( "off" )
plt . show ()

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```



Finding Synonyms

```
import nltk
nltk.download('wordnet')
from nltk.corpus import wordnet
syn=wordnet.synsets("automobile")
print(syn)
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
[Synset('car.n.01'), Synset('automobile.v.01')]
```

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

Cosine Similarity

```
# importing all necessary modules
from nltk.tokenize import sent_tokenize, word_tokenize
import warnings
import nltk

warnings.filterwarnings(action = 'ignore')

import gensim
from gensim.models import Word2Vec

#  Reads ‘the text’ file
sample = open("/content/AST_Clean.txt", "r")
s = sample.read()

# Replaces escape character with space
f = s.replace("\n", " ")

data = []

# iterate through each sentence in the file
for i in sent_tokenize(f):
    temp = []

    # tokenize the sentence into words
    for j in word_tokenize(i):
        temp.append(j.lower())

    data.append(temp)

#Create CBOW model
model1 = gensim.models.Word2Vec(data, min_count = 1, size = 100, window = 5)

#Print results
print("Cosine similarity between 'spaceport' " +
               "and 'environmental' - CBOW : ",
    model1.similarity('spaceport', 'environmental'))
#Print results
print("Cosine similarity between 'spaceport' " +
               "and 'booming' - CBOW : ",
    model1.similarity('spaceport', 'booming'))
```

```
Cosine similarity between 'spaceport' and 'environmental' - CBOW :  0.09163346
Cosine similarity between 'spaceport' and 'booming' - CBOW :  -0.28629297
```

Jaccard Index

```
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
True
```

```
from nltk import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
lemmatizer = WordNetLemmatizer()
pair1 = ["the agency renewed launch certification ","launches will resume soon"]
pair2 = ["leaders come together ", "georgia has the support of local leaders"]
pair3 = ["the area is in the wilderness", "wild life is significant"]
def extract_text_similarity_jaccard (text1, text2):
    words_text1 = [lemmatizer.lemmatize(word.lower()) for word in word_tokenize(text1)]
    words_text2 = [lemmatizer.lemmatize(word.lower()) for word in word_tokenize(text2)]
    nr = len(set(words_text1).intersection(set(words_text2)))
    dr = len(set(words_text1).union(set(words_text2)))
    jaccard_sim = nr/dr
    return jaccard_sim
print("Text Similarity: \n", extract_text_similarity_jaccard(pair1[0],pair1[1]))
```

```
Text Similarity: 
 0.125
```

Similarity across pairs

```
tfidf_model = TfidfVectorizer()
corpus = [pair1[0], pair1[1], pair2[0], pair2[1], pair3[0], pair3[1]]
tfidf_results = tfidf_model.fit_transform(corpus).todense()
cosine_similarity(tfidf_results[0],tfidf_results[1])
```

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