Untitled10

February 12, 2023

[1]:
```
#importing library
import pandas as pd
import requests
from bs4 import BeautifulSoup
import numpy as np
```

[2]:
```
#importing each extracted files
text=pd.read_csv('pagedata.csv')
```

[3]:
```
text
```

[3]:
```
     Unnamed: 0                                              title  \
0             0       ai-in-healthcare-to-improve-patient-outcomes
1             1    what-if-the-creation-is-taking-over-the-creator
2             2  what-jobs-will-robots-take-from-humans-in-the-…
3             3  will-machine-replace-the-human-in-the-future-o…
4             4                  will-ai-replace-us-or-work-with-us
..          …                                                   …
106         106                             blockchain-for-payments
107         107                             the-future-of-investing
108         108                   big-data-analytics-in-healthcare
109         109      business-analytics-in-the-healthcare-industry
110         110  challenges-and-opportunities-of-big-data-in-he…

                                                  text
0      Introduction  "If anything kills over 10 mil…
1      Human minds, a fascination in itself carryin…
2      Introduction  AI is rapidly evolving in the …
3      "Anything that could give rise to smarter-th…
4      "Machine intelligence is the last invention …
..                                                  …
106    Reconciling with the financial realities of …
107    What Is an Investment?  An investment is a r…
108    Quality and affordable healthcare is a visio…
109    Analytics is a statistical scientific proces…
110    Big Data  To begin with I shall first like t…
```

```
[111 rows x 3 columns]
```

[4]:
```
#information of data frame
text.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 111 entries, 0 to 110
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  111 non-null    int64 
 1   title       111 non-null    object
 2   text        111 non-null    object
dtypes: int64(1), object(2)
memory usage: 2.7+ KB
```

[5]:
```
#removing extra created colum
text.drop(1,axis=0,inplace=True)
```

[6]:
```
#converting type
text=text.astype(str)
```

[7]:
```
text['text']
```

[7]:
```
0          Introduction  "If anything kills over 10 mil…
2          Introduction  AI is rapidly evolving in the …
3          "Anything that could give rise to smarter-th…
4          "Machine intelligence is the last invention …
5          Introduction  Where is this disruptive techn…
                              …
106        Reconciling with the financial realities of …
107        What Is an Investment?  An investment is a r…
108        Quality and affordable healthcare is a visio…
109        Analytics is a statistical scientific proces…
110        Big Data  To begin with I shall first like t…
Name: text, Length: 110, dtype: object
```

[8]:
```
#converting text to sentence
import re
a=text['text'].str.split('([\.]\s)',expand=False)#splitting text on '.'
b=a.explode()#converting to rows
b=pd.DataFrame(b)#creating data frame
b.columns=['abc']
```

[9]:
```
b
```

[9]:
```
                                                   abc
0      Introduction  "If anything kills over 10 mil…
0                                                   .
0    Not missiles but microbes." Bill Gates's remar…
0                                                   .
0    When the new, unprecedented, invisible virus h…
..                                                  …
110  A good and efficient compensation strategy, co…
110                                                 .
110    In a nutshell, we can conclude that while bi…
110                                                 .
110   Blackcoffer Insights 10 | Subhasmita Dey, Xav…

[11274 rows x 1 columns]
```

[10]:
```
#removing . char from each rows
def abcd(x):
    nopunc =[char for char in x if char != '.']
    return ''.join(nopunc)
b['abc']=b['abc'].apply(abcd)
```

[11]:
```
#replacing emty space with null values
c=b.replace('',np.nan,regex=True)
c=c.mask(c==" ")
c=c.dropna()
c.reset_index(drop=True,inplace=True)
```

[12]:
```
c
```

[12]:
```
                                                   abc
0       Introduction  "If anything kills over 10 mil…
1     Not missiles but microbes" Bill Gates's remark…
2     When the new, unprecedented, invisible virus h…
3     This public health emergency demonstrated our …
4     For the past few years, artificial intelligenc…
…                                                   …
5681  Scarce resources like  data scientists are har…
5682            They are easily  poached by competitors
5683  A good and efficient compensation strategy, co…
5684    In a nutshell, we can conclude that while bi…
5685   Blackcoffer Insights 10 | Subhasmita Dey, Xav…

[5686 rows x 1 columns]
```

[13]:
```
!pip install nltk
```

```
Requirement already satisfied: nltk in /opt/conda/lib/python3.10/site-packages
```

```
(3.8.1)
Requirement already satisfied: joblib in /opt/conda/lib/python3.10/site-packages
(from nltk) (1.2.0)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-packages
(from nltk) (4.64.1)
Requirement already satisfied: regex>=2021.8.3 in
/opt/conda/lib/python3.10/site-packages (from nltk) (2022.10.31)
Requirement already satisfied: click in /opt/conda/lib/python3.10/site-packages
(from nltk) (8.1.3)
```

[14]:
```
#importing nltk library and stopwords
import nltk
import string
```

[15]:
```
punc=[punc for punc in string.punctuation]
```

[16]:
```
punc
```

[16]:
```
['!',
 '"',
 '#',
 '$',
 '%',
 '&',
 "'",
 '(',
 ')',
 '*',
 '+',
 ',',
 '-',
 '.',
 '/',
 ':',
 ';',
 '<',
 '=',
 '>',
 '?',
 '@',
 '[',
 '\\',
 ']',
 '^',
 '_',
 '`',
 '{',
```

```
'|',
'}',
'~']
```

[17]:
```
#importing stop words files that are provided
StopWords_Auditor= open('StopWords_Auditor.txt')
StopWords_Currencies= open('StopWords_Currencies.txt',encoding="ISO-8859-1")
StopWords_DatesandNumbers=open("StopWords_DatesandNumbers.txt")
StopWords_Generic=open("StopWords_Generic.txt")
StopWords_GenericLong=open("StopWords_GenericLong.txt")
StopWords_Geographic= open("StopWords_Geographic.txt")
StopWords_Names= open("StopWords_Names.txt")
```

[18]:
```
#creating func for removing stop words
def text_process(text):
    nopunc =[char for char in text if char not in punc or char not in [':
 ↪','.',',','(',')','’','?']]
    nopunc=''.join(nopunc)
    txt=' '.join([word for word in nopunc.split() if word.lower() not in␣
 ↪StopWords_Auditor])
    txt1=' '.join([word for word in txt.split() if word.lower() not in␣
 ↪StopWords_Currencies])
    txt2=' '.join([word for word in txt1.split() if word.lower() not in␣
 ↪StopWords_DatesandNumbers])
    txt3=' '.join([word for word in txt2.split() if word.lower() not in␣
 ↪StopWords_Generic])
    txt4=' '.join([word for word in txt3.split() if word.lower() not in␣
 ↪StopWords_GenericLong])
    txt5=' '.join([word for word in txt4.split() if word.lower() not in␣
 ↪StopWords_Geographic])
    return ' '.join([word for word in txt5.split() if word.lower() not in␣
 ↪StopWords_Names])
```

[19]:
```
#applying func for each row
c['abc']=c['abc'].apply(text_process)
```

[20]:
```
c
```

[20]:

	abc
0	Introduction "If anything kills over 10 millio…
1	Not missiles but microbes" Bill Gates’s remark…
2	When the new unprecedented invisible virus hit…
3	This public health emergency demonstrated our …
4	For the past few years artificial intelligence…
…	…
5681	Scarce resources like data scientists are hard…

```
5682              They are easily poached by competitors
5683  A good and efficient compensation strategy con…
5684  In a nutshell we can conclude that while big d…
5685  Blackcoffer Insights 10 | Subhasmita Dey Xavie…

[5686 rows x 1 columns]
```

[21]:
```
#importing master Dictionary
positive=pd.read_csv("positive-words.txt",header=None)
negative=pd.read_csv("negative-words.txt",header=None,encoding="ISO-8859-1" )
```

[22]:
```
positive.columns=['abc']
negative.columns=['abc']
positive['abc']=positive['abc'].astype(str)
negative['abc']=negative['abc'].astype(str)
```

[23]:
```
#positive and negative dictionary without stopwords
positive['abc']=positive['abc'].apply(text_process)
negative['abc']=negative['abc'].apply(text_process)
```

[24]:
```
#positive list
length=positive.shape[0]
post=[]
for i in range(0,length):
   nopunc =[char for char in positive.iloc[i] if char not in string.punctuation␣
 ↪or char != '+']
   nopunc=''.join(nopunc)

   post.append(nopunc)
```

[25]:
```
#negative list
length=negative.shape[0]
neg=[]
for i in range(0,length):
  nopunc =[char for char in negative.iloc[i] if char not in string.punctuation␣
 ↪or char != '+']
  nopunc=''.join(nopunc)
  neg.append(nopunc)
```

[26]:
```
#importing tokenize library
from nltk.tokenize import word_tokenize
```

[27]:
```
txt_list=[]
length=c.shape[0]
for i in range(0,length):
  txt=' '.join([word for word in c.iloc[i]])
```

```
  txt_list.append(txt)
```

[28]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /home/jovyan/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

[28]:
```
True
```

[29]:
```
#tokenization of text
tokenize_text=[]
for i in txt_list:

  tokenize_text+=( word_tokenize(i) )
```

[52]:
```
print(tokenize_text[0:1000])
```

```
['Introduction', '"', 'If', 'anything', 'kills', 'over', '10', 'million',
'people', 'in', 'the', 'next', 'few', 'decades', 'it', 'will', 'be', 'a',
'highly', 'infectious', 'virus', 'rather', 'than', 'a', 'war', 'Not',
'missiles', 'but', 'microbes', '"', 'Bill', 'Gates', '’', 's', 'remarks', 'at',
'a', 'TED', 'conference', 'in', '2014', 'right', 'after', 'the', 'world', 'had',
'avoided', 'the', 'Ebola', 'outbreak', 'When', 'the', 'new', 'unprecedented',
'invisible', 'virus', 'hit', 'us', 'it', 'met', 'an', 'overwhelmed', 'and',
'unprepared', 'healthcare', 'system', 'and', 'oblivious', 'population', 'This',
'public', 'health', 'emergency', 'demonstrated', 'our', 'lack', 'of',
'scientific', 'consideration', 'and', 'underlined', 'the', 'alarming', 'need',
'for', 'robust', 'innovations', 'in', 'our', 'health', 'and', 'medical',
'facilities', 'For', 'the', 'past', 'few', 'years', 'artificial',
'intelligence', 'has', 'proven', 'to', 'be', 'of', 'tangible', 'potential',
'in', 'the', 'healthcare', 'sectors', 'clinical', 'practices', 'translational',
'medical', 'and', 'biomedical', 'research', 'After', 'the', 'first', 'case',
'was', 'detected', 'in', 'China', 'on', 'December', '31st', '2019', 'it', 'was',
'an', 'AI', 'program', 'developed', 'by', 'BlueDot', 'that', 'alerted', 'the',
'world', 'about', 'the', 'pandemic', 'It', 'was', 'quick', 'to', 'realise',
'AI', '’', 's', 'ability', 'to', 'analyse', 'large', 'chunks', 'of', 'data',
'could', 'help', 'in', 'detecting', 'patterns', 'and', 'identifying', 'and',
'tracking', 'the', 'possible', 'carriers', 'of', 'the', 'virus', 'Many',
'tracing', 'apps', 'use', 'AI', 'to', 'keep', 'tabs', 'on', 'the', 'people',
'who', 'have', 'been', 'infected', 'and', 'prevent', 'the', 'risk', 'of',
'cross-infection', 'by', 'using', 'AI', 'algorithms', 'that', 'can', 'track',
'patterns', 'and', 'extract', 'some', 'features', 'to', 'classify', 'or',
'categorise', 'them', 'So', 'how', 'does', 'AI', 'do', 'that', 'IBM', 'Watson',
'a', 'sophisticated', 'AI', 'that', 'works', 'on', 'cloud', 'computing', 'and',
'natural', 'language', 'processing', 'has', 'prominently', 'contributed', 'to',
'the', 'healthcare', 'sector', 'on', 'a', 'global', 'level', 'Being', 'a',
'conversational', 'AI', 'since', '2013', 'Watson', 'has', 'helped', 'in',
```

'recommending', 'treatments', 'to', 'patients', 'suffering', 'from', 'cancer',
'to', 'ensure', 'that', 'they', 'get', 'the', 'best', 'treatment', 'at',
'optimum', 'costs', 'Researchers', 'at', 'Google', 'Inc', 'showed', 'that',
'an', 'AI', 'system', 'can', 'be', 'trained', 'on', 'thousands', 'of', 'images',
'to', 'achieve', 'physician-level', 'sensitivity', 'By', 'identifying', 'the',
'molecular', 'patterns', 'associated', 'with', 'disease', 'status', 'and',
'its', 'subtypes', 'gene', 'expression', 'and', 'protein', 'abundance',
'levels', 'machine', 'learning', 'methods', 'can', 'detect', 'fatal',
'diseases', 'like', 'cancer', 'at', 'an', 'early', 'stage', 'Machine',
'Learning', 'ML', 'techniques', 'focus', 'mainly', 'on', 'analyzing',
'structured', 'data', 'which', 'can', 'further', 'help', 'in', 'clustering',
'patients', '’', 'traits', 'and', 'infer', 'the', 'probability', 'of',
'disease', 'outcomes', 'Since', 'patient', 'traits', 'mainly', 'include',
'masses', 'of', 'data', 'relating', 'to', 'age', 'gender', 'disease', 'history',
'disease-specific', 'data', 'like', 'diagnostic', 'imaging', 'and', 'gene',
'expressions', 'etc', 'ML', 'can', 'extract', 'features', 'from', 'these',
'data', 'inputs', 'by', 'constructing', 'data', 'analytical', 'algorithms',
'ML', 'algorithms', 'are', 'either', 'supervised', 'or', 'unsupervised',
'Unsupervised', 'learning', 'helps', 'in', 'extracting', 'features', 'and',
'clustering', 'similar', 'features', 'together', 'that', 'further', 'leads',
'to', 'early', 'detection', 'of', 'diseases', 'Clustering', 'and', 'principal',
'component', 'analysis', 'enable', 'grouping', 'or', 'clustering', 'of',
'similar', 'traits', 'together', 'that', 'are', 'further', 'used', 'to',
'maximize', 'or', 'minimize', 'the', 'similarity', 'between', 'the', 'patients',
'within', 'or', 'between', 'the', 'clusters', 'Since', 'patient', 'traits',
'are', 'recorded', 'in', 'multiple', 'dimensions', 'such', 'as', 'genes',
'principal', 'component', 'analysisPCA', 'creates', 'the', 'apparatus', 'to',
'reduce', 'these', 'dimensions', 'which', 'humans', 'could', 'have', 'not',
'done', 'alone', 'Supervised', 'learning', 'considers', 'the', 'outcomes', 'of',
'the', 'subjects', 'together', 'with', 'the', 'traits', 'and', 'further',
'correlates', 'the', 'inputs', 'with', 'the', 'outputs', 'to', 'predict', 'the',
'probability', 'of', 'getting', 'a', 'particular', 'clinical', 'event',
'expected', 'value', 'of', 'a', 'disease', 'level', 'or', 'expected',
'survival', 'time', 'or', 'risk', 'of', 'Down', '’', 's', 'syndrome',
'Biomarker', 'panels', 'that', 'are', 'mostly', 'used', 'to', 'detect',
'ovarian', 'cancer', 'have', 'outperformed', 'the', 'conventional',
'statistical', 'methods', 'due', 'to', 'machine', 'learning', 'In', 'addition',
'to', 'this', 'the', 'use', 'of', 'EHRs', 'and', 'Bayesian', 'networks',
'which', 'are', 'a', 'part', 'of', 'supervised', 'machine', 'learning',
'algorithms', 'can', 'predict', 'clinical', 'outcomes', 'and', 'mortality',
'respectively', 'Unstructured', 'data', 'such', 'as', 'clinical', 'notes',
'and', 'texts', 'are', 'converted', 'into', 'machine-readable', 'structured',
'data', 'with', 'the', 'help', 'of', 'natural', 'language', 'processingNLP',
'NLP', 'works', 'with', 'two', 'components', 'text', 'processing', 'and',
'classification', 'Text', 'processing', 'helps', 'in', 'identifying', 'a',
'series', 'of', 'disease-relevant', 'keywords', 'in', 'clinical', 'notes',
'and', 'then', 'through', 'classification', 'are', 'further', 'categorized',
'into', 'normal', 'and', 'abnormal', 'cases', 'Chest', 'screening', 'through',

'ML', 'and', 'NLP', 'has', 'helped', 'find', 'abnormalities', 'in', 'the',
'lungs', 'and', 'provide', 'treatment', 'to', 'covid', 'patients', 'Healthcare',
'organizations', 'use', 'NLP-based', 'chatbots', 'to', 'increase',
'interactions', 'with', 'patients', 'keeping', 'their', 'mental', 'health',
'and', 'wellness', 'in', 'check', 'Deep', 'learning', 'is', 'a', 'modern',
'extension', 'of', 'the', 'classical', 'neural', 'network', 'techniques',
'which', 'helps', 'explore', 'more', 'complex', 'non-linear', 'patterns', 'in',
'data', 'using', 'algorithms', 'like', 'convolution', 'neural', 'network',
'recurrent', 'neural', 'network', 'deep', 'belief', 'network', 'and', 'deep',
'neural', 'network', 'which', 'enables', 'more', 'accurate', 'clinical',
'prediction', 'When', 'it', 'comes', 'to', 'genome', 'interpretation', 'deep',
'neural', 'networks', 'surpass', 'the', 'conventional', 'methods', 'of',
'logistics', 'regression', 'and', 'support', 'vector', 'machines', 'Sepsis',
'Watch', 'is', 'an', 'AI', 'system', 'trained', 'in', 'deep', 'learning',
'algorithms', 'that', 'holds', 'the', 'capability', 'to', 'analyze', 'over',
'32', 'million', 'data', 'points', 'to', 'create', 'a', 'patient', '’', 's',
'risk', 'score', 'and', 'identify', 'the', 'early', 'stages', 'of', 'sepsis',
'Another', 'method', 'known', 'as', 'the', 'Learning-based', 'Optimization',
'of', 'the', 'Under', 'Sampling', 'Pattern', 'LOUPE', 'is', 'based', 'on',
'integrating', 'full', 'resolution', 'MRI', 'scans', 'with', 'the',
'convolutional', 'neural', 'network', 'algorithm', 'which', 'helps', 'in',
'creating', 'more', 'accurate', 'reconstructions', 'Robotic', 'surgery', 'is',
'widely', 'considered', 'in', 'most', 'delicate', 'surgeries', 'like',
'gynaecology', 'and', 'prostate', 'surgery', 'Even', 'after', 'striking', 'the',
'right', 'balance', 'between', 'human', 'decisions', 'and', 'AI', 'precision',
'robotic', 'surgery', 'reduces', 'surgeon', 'efficiency', 'as', 'they', 'have',
'to', 'be', 'manually', 'operated', 'through', 'a', 'console', 'Thus',
'autonomous', 'robotic', 'surgery', 'is', 'on', 'the', 'rise', 'with',
'inventions', 'such', 'as', 'robotic', 'silicon', 'fingers', 'that', 'mimic',
'the', 'sense', 'of', 'touch', 'that', 'surgeons', 'need', 'to', 'identify',
'organs', 'cut', 'tissues', 'etc', 'or', 'robotic', 'catheters', 'that', 'can',
'navigate', 'whether', 'it', 'is', 'touching', 'blood', 'tissue', 'or', 'valve',
'Researchers', 'at', 'Children', '’', 's', 'National', 'Hospital', 'Washington',
'have', 'already', 'developed', 'an', 'AI', 'called', 'Smart', 'Tissue',
'Autonomous', 'Robot', 'STAR', 'which', 'performs', 'a', 'colon', 'anastomosis',
'on', 'its', 'own', 'with', 'the', 'help', 'of', 'an', 'ML-powered', 'suturing',
'tool', 'that', 'automatically', 'detects', 'the', 'patient', '’', 's',
'breathing', 'pattern', 'to', 'apply', 'suture', 'at', 'the', 'correct',
'point', 'An', 'image', 'of', 'STAR', 'during', 'surgery', 'Cloud', 'computing',
'in', 'healthcare', 'has', 'helped', 'in', 'retrieving', 'and', 'sharing',
'medical', 'records', 'safely', 'with', 'a', 'reduction', 'in', 'maintenance',
'costs', 'Through', 'this', 'technology', 'doctors', 'and', 'various',
'healthcare', 'workers', 'have', 'access', 'to', 'detailed', 'patient', 'data',
'that', 'helps', 'in', 'speeding', 'up', 'analysis', 'ultimately', 'leading',
'to', 'better', 'care', 'in', 'the', 'form', 'of', 'more', 'accurate',
'information', 'medications', 'and', 'therapies', 'How', 'can', 'It', 'help',
'in', 'Biomedical', 'research', 'Since', 'AI', 'can', 'analyze', 'literature',
'beyond', 'readability']

[31]:
```
len(tokenize_text)
```

[31]: 129949

positive Score

[32]:
```
positive_score=0
for i in tokenize_text:
  if(i.lower() in post):
    positive_score+=1
print('postive score=', positive_score)
```

```
postive score= 4108
```

[33]:
```
negative_score=0
for i in tokenize_text:
  if(i.lower() in neg):
    negative_score+=1
print('negative score=', negative_score)
```

```
negative score= 3490
```

[34]:
```
#Polarity Score = (Positive Score - Negative Score)/ ((Positive Score +␣
 ↪Negative Score) + 0.000001)
Polarity_Score=(positive_score-negative_score)/
 ↪((positive_score+negative_score)+0.000001)
print('polarity_score=', Polarity_Score)
```

```
polarity_score= 0.08133719398771555
```

[35]:
```
#Subjectivity Score = (Positive Score + Negative Score)/ ((Total Words after␣
 ↪cleaning) + 0.000001)
subjectiivity_score=(positive_score-negative_score)/((len(tokenize_text))+ 0.
 ↪000001)
print('subjectivity_score',subjectiivity_score)
```

```
subjectivity_score 0.00475571185615314
```

[36]:
```
length=c.shape[0]
avg_length=[]
for i in range(0,length):
  avg_length.append(len(c['abc'].iloc[i]))
avg_senetence_length=sum(avg_length)/len(avg_length)
print('avg sentence length=', avg_senetence_length)
```

```
avg sentence length= 134.6880056278579
```

[37]:
```
vowels=['a','e','i','o','u']
import re
count=0
complex_Word_Count=0
for i in tokenize_text:
  x=re.compile('[es|ed]$')
  if x.match(i.lower()):
   count+=0
  else:
    for j in i:
      if(j.lower() in vowels ):
        count+=1
  if(count>2):
   complex_Word_Count+=1
  count=0
```

[38]:
```
Percentage_of_Complex_words=complex_Word_Count/len(tokenize_text)
print('percentag of complex words= ',Percentage_of_Complex_words)
```

```
percentag of complex words=  0.25894004571024015
```

[39]:
```
#Fog Index = 0.4 * (Average Sentence Length + Percentage of Complex words)
Fog_Index = 0.4 * (avg_senetence_length + Percentage_of_Complex_words)
print('fog index= ',Fog_Index )
```

```
fog index=  53.97877826942726
```

[40]:
```
length=c.shape[0]
avg_length=[]
for i in range(0,length):
  a=[word.split( ) for word in c.iloc[i]]
  avg_length.append(len(a[0]))
  a=0
#avg
avg_no_of_words_per_sentence=sum(avg_length)/length
print("avg no of words per sentence= ",avg_no_of_words_per_sentence)
```

```
avg no of words per sentence=  22.307949349278932
```

[41]:
```
#complex word count
vowels=['a','e','i','o','u']
import re
count=0
complex_Word_Count=0
for i in tokenize_text:
  x=re.compile('[es|ed]$')
```

```
  if x.match(i.lower()):
   count+=0
  else:
    for j in i:
      if(j.lower() in vowels ):
        count+=1
  if(count>2):
   complex_Word_Count+=1
  count=0
print('complex words count=',  complex_Word_Count)
```

```
complex words count= 33649
```

[42]:
```
#WORD COUNT
word_count=len(tokenize_text)
print('word count= ', word_count)
```

```
word count=  129949
```

[43]:
```
#SYLLABLE PER WORD
vowels=['a','e','i','o','u']
import re
count=0
for i in tokenize_text:
  x=re.compile('[es|ed]$')
  if x.match(i.lower()):
   count+=0
  else:
    for j in i:
      if(j.lower() in vowels ):
        count+=1
syllable_count=count
print('syllable_per_word= ',syllable_count)
```

```
syllable_per_word=  245158
```

[44]:
```
# PERSONAL PRONOUNS
pronouns=['i','we','my','ours','us' ]
import re
count=0
for i in tokenize_text:
  if i.lower() in pronouns:
   count+=1
personal_pronouns=count
print('personal pronouns= ',personal_pronouns )
```

```
personal pronouns=  794
```

[45]:
```
#Average Word Length
count=0
for i in tokenize_text:
  for j in i:
    count+=1
avg_word_length=count/len(tokenize_text)
print('avg word= ', avg_word_length)
```

```
avg word=  4.960969303342081
```

[46]:
```
data={'positive_score':positive_score,'negative_score':
 ↪negative_score,'Polarity_Score':Polarity_Score,'subjectiivity_score':
 ↪subjectiivity_score,'avg_senetence_length':
 ↪avg_senetence_length,'Percentage_of_Complex_words':
 ↪Percentage_of_Complex_words,'Fog_Index':
 ↪Fog_Index,'avg_no_of_words_per_sentence':
 ↪avg_no_of_words_per_sentence,'complex_Word_Count':
 ↪complex_Word_Count,'word_count':word_count,'syllable_count':
 ↪syllable_count,'personal_pronouns':personal_pronouns,'avg_word_length':
 ↪avg_word_length}
```

[47]:
```
output=pd.DataFrame()
output=output.append(data,ignore_index=True)

output.
 ↪columns=['positive_score','negative_score','Polarity_Score','subjectiivity_score','avg_sene
output
```

```
/tmp/ipykernel_279/1695427803.py:2: FutureWarning: The frame.append method is
deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  output=output.append(data,ignore_index=True)
```

[47]:

	positive_score	negative_score	Polarity_Score	subjectiivity_score	\
0	4108.0	3490.0	0.081337	0.004756	

	avg_senetence_length	Percentage_of_Complex_words	Fog_Index	\
0	134.688006	0.25894	53.978778	

	avg_no_of_words_per_sentence	complex_Word_Count	word_count	\
0	22.307949	33649.0	129949.0	

	syllable_count	personal_pronouns	avg_word_length
0	245158.0	794.0	4.960969

[48]:
```
data
```

[48]:
```
{'positive_score': 4108,
 'negative_score': 3490,
 'Polarity_Score': 0.08133719398771555,
 'subjectiivity_score': 0.00475571185615314,
 'avg_senetence_length': 134.6880056278579,
 'Percentage_of_Complex_words': 0.25894004571024015,
 'Fog_Index': 53.97877826942726,
 'avg_no_of_words_per_sentence': 22.307949349278932,
 'complex_Word_Count': 33649,
 'word_count': 129949,
 'syllable_count': 245158,
 'personal_pronouns': 794,
 'avg_word_length': 4.960969303342081}
```

[ ]:

[ ]: