

In [1]:

```
import tweepy, codecs
import pandas as pd
import csv
from more_itertools import unique_everseen
import re
import nltk
from nltk.corpus import stopwords
from textblob import TextBlob
from textblob import Blobber
from newspaper import Article
import codecs
```

### Fetching the Tweets From Tweeter using News or Event as Keyword

In [165]:

```
# fill in your Twitter credentials
consumer_key = "/////////////////"
consumer_secret = "/////////////////"
access_token = "/////////////////"
access_token_secret = "/////////////////"

# let Tweepy set up an instance of the REST API
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Creating the API object while passing in auth information
api = tweepy.API(auth)

# The search term you want to find
query = "padmawat movie"
# Language code (follows ISO 639-1 standards)
language = "en"

# Calling the user_timeline function with our parameters
results = api.search(q=query, count=10000, since="2016-04-03", lang=language)
csvFile = open('F:/tweet csv/demo', 'a', newline='')
csvWriter = csv.writer(csvFile)

with open('F:/tweet csv/demo', 'a', newline='') as csvFile:
    fieldnames = ['Tweets']
    writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    writer.writeheader()
# foreach through all tweets pulled
for tweet in results:
    writer.writerow([{'Tweets': tweet.text.encode('utf-8')}])
```

### Reading Csv file using pandas

In [14]:

```
text = pd.read_csv('F:/tweet csv/demo')
```

In [15]:

```
text.Tweets
```

Out[15]:

```
0    b'RT @harshmadhusudan: How A Robust Tax Regime...
1    b'RT @muglikar_: Positive news for the day.\n\...
2    b'RT @muglikar_: Positive news for the day.\n\...
3    b'@Sanju_Verma_ @ArijitBanarji @RahulGandhi Sa...
4    b'RT @muglikar_: Positive news for the day.\n\...
5    b'RT @muglikar_: Positive news for the day.\n\...
6    b'RT @muglikar_: Positive news for the day.\n\...
7    b'@kiraniankalan @ArmchairDeen @India Policy
```

```

7      b'Ekranjankarai @aimchaitseph india_policy ...
8      b'RT @muglikar_: Positive news for the day.\n\...
9      b'RT @muglikar_: Positive news for the day.\n\...
10     b'RT @muglikar_: Positive news for the day.\n\...
11     b'RT @muglikar_: Positive news for the day.\n\...
12     b'RT @muglikar_: Positive news for the day.\n\...
13     b'@harsh8848 @SabinaBasha I was thinking of a ...
14     b'RT @muglikar_: Positive news for the day.\n\...
15     b'RT @muglikar_: Positive news for the day.\n\...
16     b'RT @muglikar_: Positive news for the day.\n\...
17     b'RT @muglikar_: Positive news for the day.\n\...
18     b'If the benefits of GST is transferred to com...
19     b'RT @muglikar_: Positive news for the day.\n\...
20     b"RT @shaikhmustkim93: #saveindia from fake @n...
21     b'RT @muglikar_: Positive news for the day.\n\...
22     b'RT @muglikar_: Positive news for the day.\n\...
23     b'RT @muglikar_: Positive news for the day.\n\...
24     b'RT @harshmadhusudan: How A Robust Tax Regime...
25     b'RT @muglikar_: Positive news for the day.\n\...
26     b'RT @muglikar_: Positive news for the day.\n\...
27     b'As India completes one year of #GST implemen...
28     b'RT @muglikar_: Positive news for the day.\n\...
29     b'RT @harshmadhusudan: How A Robust Tax Regime...
...
70     b'@DickDarryl @ncjain50 @radhacharandas @Times...
71     b'@divaaaaa_ash There are many things. Even I c...
72     b'RT @gst_station: Malaysian move means little...
73     b'RT @gst_station: GST on Bitcoin? Cryptocurre...
74     b'RT @harshmadhusudan: How A Robust Tax Regime...
75     b'RT @CensorReports: GST is the biggest failur...
76     b'RT @harshmadhusudan: How A Robust Tax Regime...
77     b'RT @jitu_vaghani: We think of putting \xe2\x...
78     b'RT @gst_station: If govt includes fuel under...
79     b'@subraca @sumanthraman @Sanju_Verma_ U r rig...
80     b'RT @arunjaitley: Rahul Gandhi has been advoc...
81     b'RT @ModiforPMOrg: More than 46.75 lakh new #...
82     b'RT @arunjaitley: Rahul Gandhi has been advoc...
83     b'We think of putting \xe2\x80\x98India First\...
84     b"RT @seriousfunnyguy: \xe0\xa4\xac\xe0\xa4\xa...
85     b'RT @Alisha2494: My report on how the govt ma...
86     b"RT @allaboutgaurav: Modi govt deserves credi...
87     b'RT @gst_station: GST: India flouting global ...
88     b'RT @jitu_vaghani: We think of putting \xe2\x...
89     b'RT @gst_station: GST: India\xe2\x80\x99s tex...
90     b'RT @arunjaitley: Rahul Gandhi has been advoc...
91     b'RT @IYCHimachal: Every Schemes which BJP lau...
92     b'RT @harshmadhusudan: How A Robust Tax Regime...
93     b'RT @narendramodi: I congratulate the people ...
94     b'RT @PiyushGoyal: Congratulate the Nation on ...
95     b'RT @dubeyamitabh: \xe2\x80\x99Reforms that h...
96     b'RT @CensorReports: GST is the biggest failur...
97     b'RT @arunjaitley: Rahul Gandhi has been advoc...
98     b'RT @harshmadhusudan: How A Robust Tax Regime...
99     b'RT @narendramodi: I congratulate the people ...
Name: Tweets, Length: 100, dtype: object

```

## Eliminate the Duplicate Rows

In [16]:

```

from more_itertools import unique_everseen
with open('F:/tweet csv/demo', 'r') as f, open('F:/tweet csv/demo2', 'w', newline='') as out_file:
    out_file.writelines(unique_everseen(f))

```

In [17]:

```
text_without_duplicate = pd.read_csv('F:/tweet csv/demo2')
```

In [18]:

```
text_without_duplicate.Tweets
```

Out[18]:

```

0      b'RT @harshmadhusudan: How A Robust Tax Regime...
1      b'RT @muglikar_: Positive news for the day.\n\...
2      b'@Sanju_Verma_ @ArijitBanarji @RahulGandhi Sa...
3      b'@kiranjankalar @ArmchairPseph @India_Policy ...
4      b'@harsh8848 @SabinaBasha I was thinking of a ...
5      b'If the benefits of GST is transferred to com...
6      b'RT @shaikhmustkim93: #saveindia from fake @n...
7      b'As India completes one year of #GST implemen...
8      b'RT @AnthonySald: @DickDarryl @ncjain50 @radh...
9      b'Positive news for the day.\n\n"India adjudge...
10     b'Getting a country which is the size of India...
11     b'We think of putting \xe2\x80\x98India First\...
12     b'RT @dubeyamitabh: \xe2\x80\x99Reforms that h...
13     b'RT @seriousfunnyguy: \xe0\xa4\xac\xe0\xa4\xa...
14     b'RT @KoomarShah: #saveindia from fake @narend...
15     b'@Gurdeepgmd @NDTVRavish U saying GST is bigg...
16     b'A year on: How India\xe2\x80\x99s controvers...
17     b'Why #GST is still at risk? On its first birt...
18     b'It\xe2\x80\x99s been multi-year since the Go...
19     b'@NDTVRavish GST is by far the biggest and th...
20     b'RT @FinMinIndia: Govt of India celebrated th...
21     b'Art kosha silk sarees.\nPrice:- Rs-1485/-(...
22     b'#GST | Simplification of the tax regime has ...
23     b'@DickDarryl @ncjain50 @radhacharandas @Times...
24     b'@divaaaa_ash There are many things. Even I c...
25     b'RT @gst_station: Malaysian move means little...
26     b'RT @gst_station: GST on Bitcoin? Cryptocurre...
27     b'RT @CensorReports: GST is the biggest failur...
28     b'RT @jitu_vaghani: We think of putting \xe2\x...
29     b'RT @gst_station: If govt includes fuel under...
30     b'@subraca @sumanthraman @Sanju_Verma_ U r rig...
31     b'RT @arunjaitley: Rahul Gandhi has been advoc...
32     b'RT @ModiforPMOrg: More than 46.75 lakh new #...
33     b'We think of putting \xe2\x80\x98India First\...
34     b'RT @Alisha2494: My report on how the govt ma...
35     b'RT @allaboutgaurav: Modi govt deserves credi...
36     b'RT @gst_station: GST: India flouting global ...
37     b'RT @gst_station: GST: India\xe2\x80\x99s tex...
38     b'RT @IYCHimachal: Every Schemes which BJP lau...
39     b'RT @narendramodi: I congratulate the people ...
40     b'RT @PiyushGoyal: Congratulate the Nation on ...
Name: Tweets, dtype: object

```

Filtering – we remove URL links (e.g. <http://example.com>), Twitter user names (e.g. @alex – with symbol @ indicating a user name), Twitter special words (such as “RT”6), and emoticons

In [20]:

```

import re
list_of_text=[]
for each in (text_without_duplicate.Tweets):
    String_first =str(each)
    clean_tweets=' '.join(re.sub("(@[A-Za-z0-9]+)|([\^0-9A-Za-z \t])|(\w+:\/\/\S+)", "",String_first
).split())
    list_of_text.append(clean_tweets)
list_of_text

```

Out[20]:

```

['b RT How A Robust Tax Regime Can Drive India xe2 x80 x99s 10 Trillion Plus Growth Plan n n GST i
s a xe2 x80 xa6',
'b RT Positive news for the day n n India adjudged most improved jurisdiction in insolvency resol
ution this year award given by xe2 x80 xa6',
'b Verma Sanju Verma you seem confused The figures given for USA are incorrect xe2 x80 xa6',
'b Policy Or better you would prefer eati xe2 x80 xa6',
'b I was thinking of a federalized GST where every state would determine its own tax rate How xe2
x80 xa6',
'b If the benefits of GST is transferred to common ppls in India',
'b RT saveindia from fake jumla and you ll save India from n nUnemployment nBeef Lynching nGST Fi
asco nFalling Ru xe2 x80 xa6',
'b As India completes one year of GST implementation watch PwC experts and policy makers review t
he journey so far a xe2 x80 xa6',
'b RT k70 Sorry nPiyush as banker do u know xe2 x80 xa6',
'b Positive news for the day n n India adjudged most improved jurisdiction in insolvency resoluti
on this year award xe2 x80 xa6'

```

On this year award x2 x00 xa6',

'b Getting a country which is the size of India to do a complete transformation has been extraordinary in itself Desp x2 x80 xa6',

'b We think of putting x2 x80 x98India First x2 x80 x99 instead of politics first PM Narendra Modi via NaMo App',

'b RT x2 x80 x9cReforms that have been carried out in recent years have augmented India x2 x80 x99s potential growth in contrast to its diminution x2 x80 xa6',

'b RT x2 x80 x9cIndia for getting praise on GST It may have some flaws but it s improving We ll be a super power in a decade i x2 x80 xa6',

'b RT saveindia from fake and you ll save India from n nUnemployment nBeef Lynching nGST Fiasco n Falling Rupee nFall i x2 x80 xa6',

'b U saying GST is biggest n boldest step in independent India than why CM Gujarat now feku PM x2 x80 xa6',

'b A year on How India x2 x80 x99s controversial GST regime hit working capital The introduction of the unified Goods and Serv x2 x80 xa6',

'b Why GST is still at risk On its first birthday GST encounters a fundamental challenge Will it survive ensuing p x2 x80 xa6',

'b It x2 x80 x99s been multi year since the Goods and Services Tax GST touted as the greatest expense changes in India since x2 x80 xa6',

'b GST is by far the biggest and the boldest step in history of independent India it can b termed as 1991 x2 x80 xa6',

'b RT Govt of India celebrated the 1st year of implementation of the unprecedented reform of Indian taxation GST on 1st July 2 x2 x80 xa6',

'b Art kosha silk sarees nPrice Rs 1485 included gst nBlouse piece yes nColours Available Yes nTo book pleas x2 x80 xa6',

'b GST Simplification of the tax regime has been beneficial to the auto industry by way of reduce d compliance burd x2 x80 xa6',

'b k70 Sorry nPiyush as banker do x2 x80 xa6',

'b ash There are many things Even I calculate GST in it Lots of paper work is removed There is tr emendous x2 x80 xa6',

'b RT station Malaysian move means little for India x2 x80 x99s GST Source',

'b RT station GST on Bitcoin Cryptocurrency trade may get taxed 18 Report Source India',

'b RT GST is the biggest failure in Independence India n nRetweet If you agree',

'b RT vaghani We think of putting x2 x80 x98India First x2 x80 x99 instead of politics first PM x2 x81 xa6 x2 x81 xa9 NaMo App htt x2 x80 xa6',

'b RT station If govt includes fuel under highest GST slab petrol diesel prices would come down b y Rs 27 Rs 15',

'b Verma U r right GST in India is only for people thorough knowledge on TAXATION and not for xf0 x9f xa4 x94 xf0 x9f x98 x9c',

'b RT Rahul Gandhi has been advocating a single slab GST for India It is a flawed idea A single s lab GST can function only i x2 x80 xa6',

'b RT More than 46 75 lakh new GST registrations led to an increase in MSME credit uptake GST is contributing to India x2 x80 x99s grow x2 x80 xa6',

'b We think of putting x2 x80 x98India First x2 x80 x99 instead of politics first PM Narendra Modi via NaMo App',

'b RT My report on how the govt may be cutting a limb off its Make in India lion as GST burdens s mall businesses nvia ht x2 x80 xa6',

'b RT Modi govt deserves credit for pushing through India s biggest tax reform and rolling it out fairly well so far warts a x2 x80 xa6',

'b RT station GST India flouting global laws by taxing international tickets says IATA Source x2 x80 xa6',

'b RT station GST India x2 x80 x99s textile exporters expect easing of working capital Source ht tps x2 x80 xa6',

'b RT Every Schemes which BJP launched with pomp and show has turned the major failures whether i t is Swacch Bharat Make In Ind x2 x80 xa6',

'b RT I congratulate the people of India on the special occasion of GST completing 1 year n nA vi brant example of cooperative f x2 x80 xa6',

'b RT Congratulate the Nation on completion of one year of GST the most transformative tax reform in India ever One Nation O x2 x80 xa6']

## importing the Stopwords library set

In [22]:

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```

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

In [23]:

```
# Create a list of stopwords
stopwords = stopwords.words('english')
```

```
# stop_words = set(['a', 'an', 'the', 'in', 'b', 'and', 'or', 'Retweeted', 'x', 'xe2', 'x99', 'x9'])
```

importing text blob and Converting the each and every Tweet from list object to String and then string to Textblob for Removing Stop words and findout the polarity of each Tweets

In [24]:

```
from textblob import TextBlob
from textblob import Blobber

with open('F:/tweet csv/demo2_sentiment.csv', 'a', newline='') as csvFile:
    fieldnames = ['Tweets', 'Polarity', 'Sentiment']
    writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    writer.writeheader()

for each in (list_of_text):
    x=str(each)
    blob1 = TextBlob(x)
    blob1.tags          # [('The', 'DT'), ('titular', 'JJ'),
                        #  ('threat', 'NN'), ('of', 'IN'), ...]

    blob1.noun_phrases  # WordList(['titular threat', 'blob',
                                #    'ultimate movie monster',
                                #    'amoeba-like mass', ...])

    #blob.words
    current_word=blob1.correct()
    sub_stopwords = set(current_word.words) - stop_words
    final_string=' '

    for word in sub_stopwords:
        final_string=final_string + word + ' '
    #print(final_string)

    blob2= TextBlob(final_string)
    #WordList([u'Simple', u'is', u'better', u'than', u'complex'])

    polarity=blob2.sentiment.polarity
    #print(blob2.sentiment.polarity)

    if polarity==0:
        print("neutral")

        with open('F:/tweet csv/demo2_sentiment.csv', 'a', newline='') as csvFile:
            writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
            writer.writerow(['Tweets': blob2, 'Polarity':polarity, 'Sentiment':"neutral"])
        # printing the text stored inside the tweet object

    else:
        if polarity<0:
            print("negative")

            with open('F:/tweet csv/demo2_sentiment.csv', 'a', newline='') as csvFile:
                writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
                writer.writerow(['Tweets': blob2, 'Polarity':polarity, 'Sentiment':"negative"])
        else:
            print("positive")

            with open('F:/tweet csv/demo2_sentiment.csv', 'a', newline='') as csvFile:
                writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
                writer.writerow(['Tweets': blob2, 'Polarity':polarity, 'Sentiment':"positive"])
```

neutral  
positive  
negative  
positive  
positive  
negative  
neutral  
positive  
negative  
positive  
positive  
positive  
neutral

positive  
positive  
neutral  
positive  
positive  
positive  
positive  
positive  
negative  
neutral  
negative  
positive  
negative  
neutral  
negative  
positive  
negative  
positive  
negative  
positive  
positive  
negative  
positive  
neutral  
neutral  
positive  
positive  
positive

Importing the "newspaper" module to fetch News Article

In [26]:

```
from newspaper import Article
import codecs
```

Set Url of News Article to String Variable name "url"

In [118]:

```
url = 'https://theconversation.com/after-quebec-whats-the-future-for-keystone-xl-15971'
```

Downloading the news article

In [119]:

```
article = Article(url)
news_article = article.download()
```

parsing of downloaded Article using parse fuction

In [120]:

```
article.parse()
```

Extracting only text from Article using text object

In [121]:

```
article_text = article.text
```

Printing the text of Article

In [122]:

```
print(article_text)
```

The runaway train of 73 oil tankers that derailed and exploded in the small town of Lac-Mégantic in Quebec, Canada last week left 15 dead, around 50 missing, and shows how dangerous transporting oil can be.

An alternative to rail transport is the proposed, highly contested Keystone XL pipeline which would transport oil derived from Canadian tar sands in Alberta to refineries along the US Gulf Coast. The Obama administration has yet to decide whether to approve it.

That pipeline could reduce US dependence on foreign oil. But tar sands are a particularly carbon-intensive source of fuel, with the potential for leaks and spills from the pipeline along its proposed route through the heart of the US. Any decision will create winners and losers – either in the oil industry or environmental interests.

This whole debate would be unnecessary under a carbon tax or cap-and-trade policy that put a proper price on carbon and other greenhouse gas emissions. If oil companies had to pay the true social costs of producing gasoline, then they would have to charge a price high enough to cover not only production but also environmental costs. Then if consumers are willing to pay for that gasoline, they're welcome to it. Each ton of carbon dioxide emissions is estimated to impose about \$20 of costs on the rest of society, which would raise the cost of conventional gasoline by about twenty US cents per gallon.

The Keystone project to transport tar sands oil by pipeline means overcoming three problems. Because it is thick and viscous, the tar sand oil must first be diluted to a liquid before it can be transported. This requires additional energy and generates about 12% more CO<sub>2</sub> emissions than conventional petrol. Second, this process generates huge amounts of semi-solid waste for which disposal is uncertain. Additionally, spills from the pipeline might damage ecosystems along its route – with particular concerns about pollution of Nebraska's vast Ogallala Aquifer, an important water source.

Without paying these pollution costs, oil companies can make huge profits if the Keystone pipeline is built; environmentalists get a windfall if the Keystone is cancelled. It's "winner take all". But if oil companies did have to pay environmental costs, then Obama could just leave it up to oil companies: make them pay for all that pollution and then let them choose whether and how to use the tar sands. But without those payments to cover the environmental costs, we can't really know if it's a good idea or not.

Three recent events have raised the stakes. President Obama's "Climate Action Plan" speech includes imposing emissions limits on power plants. Mentioning Keystone XL by name, Obama linked the pipeline not just to the environmental problems on its route but also to an increase in greenhouse gas emissions. But while he stated the project would not go ahead if it could be shown that greenhouse gas emissions would rise, he was vague about the "burden of proof" required.

Proponents say Obama's requirement has already been met. Seeing, they argue, as tar sands will definitely be used one way or the other rather than left in the ground, pipeline transportation via Keystone will be the most carbon-effective method. Thus, according to them, the Keystone reduces emissions.

Just days after Obama's speech came a decision in British Columbia on Canada's west coast to reject the "Northern Gateway" pipeline, which would have provided an alternative to Keystone and brought tar sands oil from Alberta to the Pacific Ocean. That plan was scuttled by concerns about oil spills through the pristine British Columbian forests, the problems of building an oil tanker port on the beautiful Pacific coastline, and other subsequent problems.

The third event, of course, is the tragedy in Quebec. Railroad transport of oil hardly seems better than pipeline transport. Another alternative now drawing attention is to build a pipeline all the way from Alberta to the Atlantic. You have to give those Alberta oil interests some credit for persistence.

Ultimately, those tar sands can stay in the ground. After all, one policy to reduce global warming is to "sequester" atmospheric carbon by locking it into growing trees, reducing deforestation, or using carbon capture technology to store it deep underground. A shortcut route to sequestering carbon deep underground is to leave it there in the first place. Technology may advance fast enough to provide cleaner alternative fuels anyway, and the future introduction of a carbon tax or permit system might make tar sands too expensive to be viable.

Until we price the true cost of carbon and pollutants into our economy with a carbon price-per-ton, we prevent the market from doing what it does best. Without that, we allow gross profiteering in the oil and gas industry while leaving taxpayers to foot the bill.

If the Article Text is not in English Language, Convert it using following steps

In [ ]:

```
article_text_to_english =TextBlob(article_text)
english_lag=article_text_to_english.translate(to="en")
```

convert the Blob into String Format.

In [ ]:

```
article_text_to_english_to_string =str(english_lag)
```

Importing textblob module for Natural Language Processing

In [9]:

```
from textblob import TextBlob
```

The text is first converted into textblob object and then Part-of-speech tags can be accessed through the tags property and noun phrases are accessed through the noun\_phrases property. The sentiment property returns a namedtuple of the form Sentiment(polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0]. we are calculated only count of negative polarity(total negative phrases) ,total of all negative polarities and the average of negative polarities.

In [123]:

```
import csv

blob = TextBlob(article_text)
blob.tags          # [('The', 'DT'), ('titular', 'JJ'),
                    #  ('threat', 'NN'), ('of', 'IN'), ...]

blob.noun_phrases  # WordList(['titular threat', 'blob',
                              #  'ultimate movie monster',
                              #  'amoeba-like mass', ...])

total_number_of_sentences = 0
total_polarity_of_sentences=0.0
mean_total_polarity_of_sentences=0.0

total_negative_sentences=0
total_of_negative_polarity_sentences=0.0
mean_negative_polarity_sentences=0.0

percentage_of_negative_sentences=0

for sentence in blob.sentences:
    polarity_phrases = sentence.sentiment.polarity

    total_number_of_sentences = total_number_of_sentences + 1
    total_polarity_of_sentences = polarity_phrases + total_polarity_of_sentences
    mean_total_polarity_of_sentences = total_polarity_of_sentences/total_number_of_sentences

    if polarity_phrases <0:
        total_negative_sentences = total_negative_sentences + 1

        total_of_negative_polarity_sentences = polarity_phrases +
total_of_negative_polarity_sentences

        mean_negative_polarity_sentences = total_of_negative_polarity_sentences/total_negative_sentences

percentage_of_negative_sentences=((total_negative_sentences/total_number_of_sentences)*100)

print(total_number_of_sentences)
print(total_polarity_of_sentences)
print(mean_total_polarity_of_sentences)
print(total_negative_sentences)
print(total_of_negative_polarity_sentences)
print(mean_negative_polarity_sentences)
print(percentage_of_negative_sentences)
```



```

print(percentage_of_negative_sentences,
with open('F:/tweet csv/demo2_final_article.csv', 'a',newline='') as csvFile:
    fieldnames = ['News_or_Events','Article_Number','Total_Number_of_Sentences','Total_Polarity_of_Sentences','Mean_Total_Polarity_of_Sentences','Total_Negative_Sentences','Total_of_Negative_Polarity_Sentences','Mean_Negative_Polarity_Sentences','Percentage_of_Negative_Sentences']
    writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    #writer.writeheader()

    #writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    writer.writerows([{'News_or_Events': " ", 'Article_Number':'9',
'Total_Number_of_Sentences':total_number_of_sentences, 'Total_Polarity_of_Sentences':total_polarity_of_sentences, 'Mean_Total_Polarity_of_Sentences':mean_total_polarity_of_sentences, 'Total_Negative_Sentences':total_negative_sentences, 'Total_of_Negative_Polarity_Sentences':total_of_negative_polarity_sentences, 'Mean_Negative_Polarity_Sentences':mean_negative_polarity_sentences, 'Percentage_of_Negative_Sentences':percentage_of_negative_sentences}])

```

```

38
2.0828531746031747
0.054811925647451964
8
-1.3574404761904761
-0.16968005952380952
21.052631578947366

```

calculating totals of all attribute of all articles

In [93]:

```
Article_total=pd.read_csv('F:/tweet csv/demo2_final_article.csv')
```

In [94]:

```
Article_total
```

Out[94]:

	News_or_Events	Article_Number	Total_Number_of_Sentences	Total_Polarity_of_Sentences	Mean_Total_Polarity_of_S
0	spot settling in IPL	1	17	0.674167	0.039657
1		2	16	1.472063	0.092004
2		3	19	0.982468	0.051709
3		4	15	1.719502	0.114633

In [95]:

```

t_s = Article_total.Total_Number_of_Sentences.sum()
t_polarity_s = Article_total.Total_Polarity_of_Sentences.sum()
m_t_polarity_s = Article_total.Mean_Total_Polarity_of_Sentences.mean()
t_n_s = Article_total.Total_Negative_Sentences.sum()
t_n_polarity_s = Article_total.Total_of_Negative_Polarity_Sentences.sum()
m_n_polarity_s = Article_total.Mean_Negative_Polarity_Sentences.mean()
per_n_s = (t_n_s/t_s)*100

print(t_s)
print(t_polarity_s)
print(m_t_polarity_s)
print(t_n_s)
print(t_n_polarity_s)
print(m_n_polarity_s)
print(per_n_s)

with open('F:/tweet csv/demo2_final_article_total.csv', 'a',newline='') as csvFile:
    fieldnames =
['News_or_Events','Total_Number_of_Sentences','Total_Polarity_of_Sentences','Mean_Total_Polarity_of_Sentences','Total_Negative_Sentences','Total_of_Negative_Polarity_Sentences','Mean_Negative_Polarity_Sentences','Percentage_of_Negative_Sentences']
    writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    writer.writeheader()

```

```

writer.writerow(['News_or_Events': "spot settling in IPL" ,
'Total_Number_of_Sentences':t_s,'Total_Polarity_of_Sentences':t_polarity_s,'Mean_Total_Polarity_of_Sentences':m_t_polarity_s,'Total_Negative_Sentences':t_n_s,'Total_of_Negative_Polarity_Sentences':t_n_polarity_s,'Mean_Negative_Polarity_Sentences':m_n_polarity_s,'Percentage_of_Negative_Sentences':per_n_s}))

```

```

67
4.848199856
0.0745007815
14
-1.539545455
-0.10810200199999999
20.8955223880597

```

Consider the value of k as follow 0% > pnc <= 10% ----> value of 10%, 10% > pnc <= 20% ----> value of 20%, 20%> pnc <= 30% ----> value of 30%, 30% > pnc <= 40% ----> value of 40%, 40% > pnc <= 50% ----> value of 50%, 50% > pnc <= 60% ----> value of 60%, 60% > pnc <= 70% ----> value of 70%, 70% > pnc <= 80% ----> value of 80%, 80% > pnc <= 90% ----> value of 90%

In [96]:

```

k = 0
#per_n_s = 18.705035971223023
#m_n_polarity_s = -0.161176729
if ((per_n_s <10) or (per_n_s >90) or (m_n_polarity_s <(-1)) or (m_n_polarity_s >=(0)) ):
    print("this event or news is Non-Controversial ")
elif(m_n_polarity_s >= (-1) and m_n_polarity_s < (-0.9)):
    if per_n_s >= 10 and per_n_s <= 50:
        k=-2
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 50 and per_n_s <= 60:
            k=-1.33
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 60 and per_n_s <= 70:
                k=-0.85
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 70 and per_n_s <= 80:
                    k=-0.5
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    k=-0.22
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.9) and m_n_polarity_s < (-0.8)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-1.71
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-1.74
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-1.755
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-1.764
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-1.17

```

```

        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 65 and per_n_s <= 75:
            k=-0.7452
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        if per_n_s > 75 and per_n_s <= 85:
            k=-0.4275
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            k=-0.19
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.8) and m_n_polarity_s < (-0.7)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-1.44
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-1.49
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-1.52
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-1.536
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-1.013
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs_of_sentences)
                    else:
                        if per_n_s > 65 and per_n_s <= 75:
                            k=-0.64
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        if per_n_s > 75 and per_n_s <= 85:
                            k=-0.36
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        else:
                            k=-0.16
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.7) and m_n_polarity_s < (-0.6)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-1.19
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-1.26
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-1.295
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-1.316
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-0.863
                        cs of sentences = (per n s*k)/(m n polarity s*100)

```

```

        print(cs_of_sentences)
    else:
        if per_n_s > 65 and per_n_s <= 75:
            k=-0.54
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        if per_n_s > 75 and per_n_s <= 85:
            k=-0.2975
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            k=-0.132
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.6) and m_n_polarity_s < (-0.5)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-0.96
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-1.04
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-1.08
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-1.104
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-0.72
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs_of_sentences)
                    else:
                        if per_n_s > 65 and per_n_s <= 75:
                            k=-0.445
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        if per_n_s > 75 and per_n_s <= 85:
                            k=-0.24
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        else:
                            k=-0.106
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.5) and m_n_polarity_s < (-0.4)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-0.75
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-0.83
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-0.875
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-0.9
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-0.58
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs of sentences)

```

```

        else:
            if per_n_s > 65 and per_n_s <= 75:
                k=-0.3571
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            if per_n_s > 75 and per_n_s <= 85:
                k=-0.1875
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                k=-0.083
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.4) and m_n_polarity_s < (-0.3)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-0.56
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-0.64
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-0.68
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-0.704
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-0.45
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs_of_sentences)
                    else:
                        if per_n_s > 65 and per_n_s <= 75:
                            k=-0.274
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        if per_n_s > 75 and per_n_s <= 85:
                            k=-0.14
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        else:
                            k=-0.062
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.3) and m_n_polarity_s < (-0.2)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-0.39
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-0.46
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-0.495
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-0.516
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-0.33
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs_of_sentences)
                    else:

```

```

        if per_n_s > 65 and per_n_s <= 75:
            k=-0.197
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        if per_n_s > 75 and per_n_s <= 85:
            k=-0.097
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            k=-0.043
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.2) and m_n_polarity_s < (-0.1)):
    if per_n_s >= 10 and per_n_s <= 25:
        k=-0.24
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 25 and per_n_s <= 35:
            k=-0.29
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 35 and per_n_s <= 45:
                k=-0.32
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 45 and per_n_s <= 55:
                    k=-0.336
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 55 and per_n_s <= 65:
                        k=-0.213
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs_of_sentences)
                    else:
                        if per_n_s > 65 and per_n_s <= 75:
                            k=-0.125
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        if per_n_s > 75 and per_n_s <= 85:
                            k=-0.06
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
                        else:
                            k=-0.026
                            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                            print(cs_of_sentences)
elif(m_n_polarity_s >=(-0.1) and m_n_polarity_s < (0)):
    if per_n_s >= 10 and per_n_s <= 15:
        k=-0.1
        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
        print(cs_of_sentences)
    else:
        if per_n_s > 15 and per_n_s <= 25:
            k=-0.11
            cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
            print(cs_of_sentences)
        else:
            if per_n_s > 25 and per_n_s <= 35:
                k=-0.14
                cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                print(cs_of_sentences)
            else:
                if per_n_s > 35 and per_n_s <= 45:
                    k=-0.155
                    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                    print(cs_of_sentences)
                else:
                    if per_n_s > 45 and per_n_s <= 55:
                        k=-0.164
                        cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
                        print(cs_of_sentences)
                    else:
                        if per_n_s > 55 and per_n_s <= 65:

```

```

-- per_n_s < 65 and per_n_s > 65:
    k=-0.103
    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
    print(cs_of_sentences)
if per_n_s > 65 and per_n_s <= 75:
    k=-0.06
    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
    print(cs_of_sentences)
if per_n_s > 75 and per_n_s <= 85:
    k=-0.0275
    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
    print(cs_of_sentences)
else:
    k=-0.011
    cs_of_sentences = (per_n_s*k)/(m_n_polarity_s*100)
    print(cs_of_sentences)

```

0.4639067991668025

In [97]:

```
count = pd.read_csv("F:/tweet csv/demo2_sentiment.csv")
```

calculating total comments, negative comments, positive comments, neutral comments, total polarity of comments, mean of total polarity of comments, total of negative polarity of comments, mean of the negative polarity of the comments, percentage of negative comments.

In [98]:

```

total_comments=0
total_polarity_of_comments=0.0
mean_of_total_polarity_of_comments=0.0

total_negative_comments=0
total_of_negative_polarity_comments=0.0
mean_of_negative_polarity_comments=0.0

percentage_of_negative_comments=0

for each in (count.Polarity):
    total_comments=total_comments + 1
    total_polarity_of_comments=each + total_polarity_of_comments
    mean_of_total_polarity_of_comments = total_polarity_of_comments/total_comments

    if each <0:
        total_negative_comments = total_negative_comments + 1

        total_of_negative_polarity_comments = each + total_of_negative_polarity_comments

        mean_of_negative_polarity_comments = total_of_negative_polarity_comments/total_negative_comments

percentage_of_negative_comments=((total_negative_comments/total_comments)*100)

print(total_comments)
print(total_polarity_of_comments)
print(total_negative_comments)
print(total_of_negative_polarity_comments)
print(percentages_of_negative_comments)

print(mean_of_total_polarity_of_comments)
print(mean_of_negative_polarity_comments)

with open('F:/tweet csv/demo2_final_comments_total.csv', 'a') as csvFile:
    fieldnames = ['News_or_Events', 'Total_Comments', 'Total_Polarity_of_Comments', 'Total_Negative_Comments', 'Total_of_Negative_Polarity_Comments', 'Percentage_of_Negative_Comments', 'Mean_of_Total_Polarity_of_Comments', 'Mean_of_Negative_Polarity_Comments']
    writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    writer.writeheader()

```

```

        #writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
        writer.writerow(['News_or_Events': "spot settling in IPL" , 'Total_Comments':total_comments, 'Total_Polarity_of_Comments':total_polarity_of_comments, 'Total_Negative_Comments':total_negative_comments, 'Total_of_Negative_Polarity_Comments':total_of_negative_polarity_comments, 'Percentage_of_Negative_Comments':percentage_of_negative_comments, 'Mean_of_Total_Polarity_of_Comments':mean_of_total_polarity_of_comments, 'Mean_of_Negative_Polarity_Comments':mean_of_negative_polarity_comments])
    })

```

```

15443
1712.66589015
2602
-690.106261054
16.84905782555203
0.110902408221
-0.265221468506

```

Finding the Controversy Score of comments : Dividing Percentage of Negative Comments by Mean of the polarity of the negative comments multiply by proportionality Constant k

In [99]:

```

k = 0
#percentage_of_negative_comments = 44.03669724770643
#mean_of_negative_polarity_comments = -0.116333912037
if ((percentage_of_negative_comments < 10) or (percentage_of_negative_comments > 90) or (mean_of_negative_polarity_comments < (-1)) or (mean_of_negative_polarity_comments >= (0))):
    print("this event or news is Non-Controversial ")
elif (mean_of_negative_polarity_comments >= (-1) and mean_of_negative_polarity_comments < (-0.9)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 55:
        k=-2
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 55 and percentage_of_negative_comments <= 65:
            k=-1.33
            cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 65 and percentage_of_negative_comments <= 75:
                k=-0.85
                cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 75 and percentage_of_negative_comments <= 85:
                    k=-0.5
                    cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:
                    k=-0.22
                    cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
elif (mean_of_negative_polarity_comments >= (-0.9) and mean_of_negative_polarity_comments < (-0.8)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
        k=-1.71
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
            k=-1.74
            cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                k=-1.755
                cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:

```



```

else:
    if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
        k=-1.764
        cs_of_comments =
        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
            k=-1.17
            cs_of_comments =
            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                k=-0.7452
                cs_of_comments =
                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                k=-0.4275
                cs_of_comments =
                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                k=-0.19
                cs_of_comments =
                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
elif (mean_of_negative_polarity_comments >=(-0.8) and mean_of_negative_polarity_comments < (-0.7)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
        k=-1.44
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*10
0)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
            k=-1.49
            cs_of_comments =
            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                k=-1.52
                cs_of_comments =
                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                    k=-1.536
                    cs_of_comments =
                    (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:
                    if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
                        k=-1.013
                        cs_of_comments =
                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    else:
                        if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                            k=-0.64
                            cs_of_comments =
                            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
                        if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                            k=-0.36
                            cs_of_comments =
                            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
                        else:
                            k=-0.16
                            cs_of_comments =

```

```

cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
print(cs_of_comments)
elif(mean_of_negative_polarity_comments >=(-0.7) and mean_of_negative_polarity_comments < (-0.6)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
        k=-1.19
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
    print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
            k=-1.26
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                    k=-1.295
                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                    else:
                        if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                            k=-1.316
                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
                            else:
                                if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
                                    k=-0.863
                                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                    print(cs_of_comments)
                                    else:
                                        if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                                            k=-0.54
                                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                            print(cs_of_comments)
                                            if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                                                k=-0.2975
                                                cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                                print(cs_of_comments)
                                                else:
                                                    k=-0.132
                                                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                                    print(cs_of_comments)
                                                elif(mean_of_negative_polarity_comments >=(-0.6) and mean_of_negative_polarity_comments < (-0.5)):
                                                    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
                                                        k=-0.96
                                                        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                                    print(cs_of_comments)
                                                    else:
                                                        if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
                                                            k=-1.04
                                                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                                            print(cs_of_comments)
                                                            else:
                                                                if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                                                                    k=-1.08
                                                                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                                                    print(cs_of_comments)
                                                                    else:
                                                                        if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                                                                            k=-1.104
                                                                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                                                            print(cs_of_comments)
                                                                            else:
                                                                                if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=

```

```

        if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
            k=-0.72
            cs_of_comments =
            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                k=-0.445
                cs_of_comments =
                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
                if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                    k=-0.24
                    cs_of_comments =
                    (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:
                    k=-0.106
                    cs_of_comments =
                    (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
            elif(mean_of_negative_polarity_comments >=(-0.5) and mean_of_negative_polarity_comments < (-0.4)):
                if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
                    k=-0.75
                    cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*10
0)
                    print(cs_of_comments)
                else:
                    if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
                        k=-0.83
                        cs_of_comments =
                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    else:
                        if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                            k=-0.875
                            cs_of_comments =
                            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
                        else:
                            if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                                k=-0.9
                                cs_of_comments =
                                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                print(cs_of_comments)
                            else:
                                if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
                                    k=-0.58
                                    cs_of_comments =
                                    (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                    print(cs_of_comments)
                                else:
                                    if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                                        k=-0.3571
                                        cs_of_comments =
                                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                        print(cs_of_comments)
                                    if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                                        k=-0.1875
                                        cs_of_comments =
                                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                        print(cs_of_comments)
                                    else:
                                        k=-0.083
                                        cs_of_comments =
                                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                                        print(cs_of_comments)
                                elif(mean_of_negative_polarity_comments >=(-0.4) and mean_of_negative_polarity_comments < (-0.3)):
                                    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
                                        k=-0.56
                                        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*10
0)

```

```

0)
    print(cs_of_comments)
else:
    if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
        k=-0.64
        cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
            k=-0.68
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                k=-0.704
                cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
                    k=-0.45
                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:
                    if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                        k=-0.274
                        cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                        k=-0.14
                        cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    else:
                        k=-0.062
                        cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
elif (mean_of_negative_polarity_comments >=(-0.3) and mean_of_negative_polarity_comments < (-0.2)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
        k=-0.39
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*10
0)
    print(cs_of_comments)
else:
    if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
        k=-0.46
        cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
            k=-0.495
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                k=-0.516
                cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
                    k=-0.33
                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:

```

```

        if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
            k=-0.197
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
            k=-0.097
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            k=-0.043
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
elif(mean_of_negative_polarity_comments >=(-0.2) and mean_of_negative_polarity_comments < (-0.1)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 25:
        k=-0.24
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*10
0)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
            k=-0.29
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                k=-0.2
                cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 45 and percentage_of_negative_comments <= 55:
                    k=-0.336
                    cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:
                    if percentage_of_negative_comments > 55 and percentage_of_negative_comments <=
65:
                        k=-0.213
                        cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    else:
                        if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                            k=-0.125
                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
                        if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                            k=-0.06
                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
                        else:
                            k=-0.026
                            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                            print(cs_of_comments)
elif(mean_of_negative_polarity_comments >=(-0.1) and mean_of_negative_polarity_comments < (0)):
    if percentage_of_negative_comments >= 10 and percentage_of_negative_comments <= 15:
        k=-0.1
        cs_of_comments = (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*10
0)
        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 15 and percentage_of_negative_comments <= 25:
            k=-0.11
            cs_of_comments =
(percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)

```

```

        print(cs_of_comments)
    else:
        if percentage_of_negative_comments > 25 and percentage_of_negative_comments <= 35:
            k=-0.14
            cs_of_comments =
            (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
            print(cs_of_comments)
        else:
            if percentage_of_negative_comments > 35 and percentage_of_negative_comments <= 45:
                k=-0.155
                cs_of_comments =
                (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                print(cs_of_comments)
            else:
                if percentage_of_negative_comments > 45 and percentage_of_negative_comments <=
55:
                    k=-0.164
                    cs_of_comments =
                    (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                    print(cs_of_comments)
                else:
                    if percentage_of_negative_comments > 55 and
percentage_of_negative_comments <= 65:
                        k=-0.103
                        cs_of_comments =
                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    if percentage_of_negative_comments > 65 and
percentage_of_negative_comments <= 75:
                        k=-0.06
                        cs_of_comments =
                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    if percentage_of_negative_comments > 75 and
percentage_of_negative_comments <= 85:
                        k=-0.0275
                        cs_of_comments =
                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)
                    else:
                        k=-0.011
                        cs_of_comments =
                        (percentage_of_negative_comments*k)/(mean_of_negative_polarity_comments*100)
                        print(cs_of_comments)

```

0.247760205423

In [100]:

```
print(cs_of_comments)
```

0.247760205423

Finding the Controversy Score of articles : Diving Percentange of Negative Sentences by Mean of the polarity of the negative Sentences

In [101]:

```
cs_of_sentences
```

Out[101]:

0.4639067991668025

Final Controversy of the news or event : its mean of controversy score of the comments and controversy score of sentences of article.

In [102]:

```
cs_of_news_or_event = ((cs_of_comments + cs_of_sentences)/2)
```

In [103]:

```
print(cs_of_news_or_event)
```

0.355833502295

In [104]:

```
with open('F:/tweet csv/demo2_Final_CS_Result_Table.csv', 'a', newline='') as csvFile:
    fieldnames =
    ['News_or_Events', 'Percentage_of_Negative_Comments', 'Mean_of_Negative_Polarity_Comments', 'Percentage_of_Negative_Sentences', 'Mean_Negative_Polarity_Sentences', 'Controversy_Score_Comments', 'Controversy_Score_Sentences', 'Final_Controversy_Score']
    writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    #writer.writeheader()

    #writer = csv.DictWriter(csvFile, fieldnames=fieldnames)
    writer.writerow(['News_or_Events': "spot settling in IPL"
, 'Percentage_of_Negative_Comments':percentage_of_negative_comments, 'Mean_of_Negative_Polarity_Comments':mean_of_negative_polarity_comments, 'Percentage_of_Negative_Sentences':per_n_s, 'Mean_Negative_Polarity_Sentences':m_n_polarity_s, 'Controversy_Score_Comments':cs_of_comments, 'Controversy_Score_Sentences':cs_of_sentences, 'Final_Controversy_Score':cs_of_news_or_event])
```