```
# Social media
from textblob import TextBlob
from os import path
ObjRead = open("/content/camden_SM.txt", "r")
txtContent = ObjRead.read();
ObjRead.close()
testimonial = TextBlob(txtContent)
testimonial.sentiment

```

```
Sentiment(polarity=0.09227621955505684, subjectivity=0.43873076984290216)
```

```
# Digital print
from textblob import TextBlob
from os import path
ObjRead = open("/content/camden_DP.txt", "r")
txtContent = ObjRead.read();
ObjRead.close()
testimonial = TextBlob(txtContent)
testimonial.sentiment

```

```
#Install package using pip
!pip install text2emotion
```

```
Collecting text2emotion
  Downloading https://files.pythonhosted.org/packages/fe/31/b190e37c1396ca68ab1b5c8ea1a23f2f7848df532ad69133e94853120aed/text2emotion-
     |████████████████████████████████| 61kB 3.2MB/s
Collecting emoji>=0.6.0
  Downloading https://files.pythonhosted.org/packages/ff/1c/1f1457fe52d0b30cbeebfd578483cedb3e3619108d2d5a21380dfecf8ffd/emoji-0.6.0.t
     |████████████████████████████████| 51kB 5.9MB/s
Requirement already satisfied: nltk in /usr/local/lib/python3.6/dist-packages (from text2emotion) (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from nltk->text2emotion) (1.15.0)
Building wheels for collected packages: emoji
  Building wheel for emoji (setup.py) ... done
  Created wheel for emoji: filename=emoji-0.6.0-cp36-none-any.whl size=49716 sha256=673a0e3773c1f9c5ed2a96f508336f582900785732ab255a81
  Stored in directory: /root/.cache/pip/wheels/46/2c/8b/9dcf5216ca68e14e0320e283692dce8ae321cdc01e73e17796
Successfully built emoji
Installing collected packages: emoji, text2emotion
Successfully installed emoji-0.6.0 text2emotion-0.0.5
```

```
#Import the modules
import text2emotion as te
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

```
from os import path
ObjRead = open("/content/camden_DP.txt", "r")
text = ObjRead.read();
ObjRead.close()
```

```
text1="""The Federal Aviation Administration is set to require a new environmental review for a scaled-down proposal to launch rock
AAC and Camden County plan to create a cooperative and collaborative operating approach that creates efficiency and effectiveness for
```

```
te.get_emotion(text1)
```

```
{'Angry': 0.04, 'Fear': 0.48, 'Happy': 0.1, 'Sad': 0.17, 'Surprise': 0.21}
```

```
text=str("""The Federal Aviation Administration is set to require a new environmental review for a scaled-down proposal to launch r
AAC and Camden County plan to create a cooperative and collaborative operating approach that creates efficiency and effectiveness for
```

```
te.get_emotion(text)
```

```
{'Angry': 0.04, 'Fear': 0.48, 'Happy': 0.1, 'Sad': 0.17, 'Surprise': 0.21}
```

Sentiment Analysis with xml file

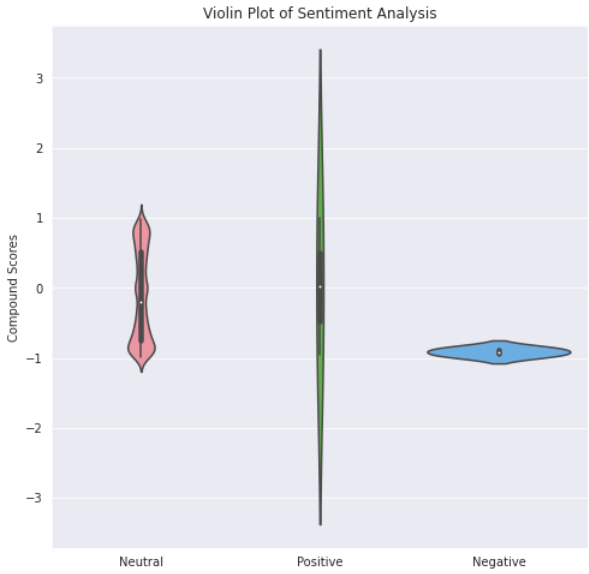
```
# Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt, seaborn as sb

import xml.etree.ElementTree as et

# Parsing the XML into a Pandas dataframe
file = et.parse('/content/nysk.xml')
root = file.getroot()
columns = ["DocID", "Source", "URL", "Title", "Summary", "Date"] # Setting the columns of the Dataframe

summary_list = []
dataset = pd.DataFrame(columns = columns)
```

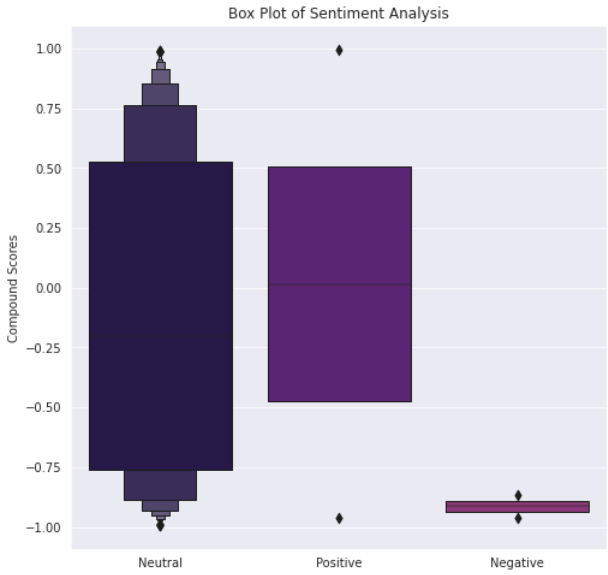
```
for each_node in root:
    doc = each_node.find("docid").text
    source = each_node.find("source").text
    url = each_node.find("url").text
    title = each_node.find("title").text
    summary = each_node.find("summary").text
    date = each_node.find("date").text

    dataset = dataset.append(pd.Series([doc, source, url, title, summary, date],
                                       index = columns), ignore_index = True)
```

```
# Sentiment Analysis with Vader
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk
nltk.download('vader_lexicon')

analyzer = SentimentIntensityAnalyzer()

sentiment_dict = dict()

for sentence in dataset["Summary"]:
    sent = analyzer.polarity_scores(sentence)

    # Creating a dictionary of sentiment scores and their values
    # Setting up keys
    sentiment_dict.setdefault('Negative_Score', [])
    sentiment_dict.setdefault('Neutral_Score', [])
    sentiment_dict.setdefault('Positive_Score', [])
    sentiment_dict.setdefault('Compound_Score', [])

    # Appending values to the respective keys
    sentiment_dict['Negative_Score'].append(sent.get('neg'))
    sentiment_dict['Neutral_Score'].append(sent.get('neu'))
    sentiment_dict['Positive_Score'].append(sent.get('pos'))
    sentiment_dict['Compound_Score'].append(sent.get('compound'))

sentiment_df = pd.DataFrame.from_dict(sentiment_dict, orient = "columns")
sentiment_df.insert(loc = 0, column = "Article", value = dataset["Summary"])

# Bonus Dataset creation
class_list = list()

for m in range(len(sentiment_df)):
    if sentiment_df['Neutral_Score'][m] > 0.5:
        class_list.append("Neutral")
    elif sentiment_df['Negative_Score'][m] > 0.5:
        class_list.append('Negative')
    else:
        class_list.append('Positive')

# Adding the Class Label as the last column to the sentiment dataframe
sentiment_df.insert(loc = 5, column = "Sentiment", value = class_list)

# Storing the sentiment dataframe as a CSV for easy perusal later on
sentiment_df.to_csv('sentiment_analysis.csv', index = False, header = True)
```

```
/usr/local/lib/python3.6/dist-packages/nltk/twitter/__init__.py:20: UserWarning: The twython library has not been installed. Some funct
  warnings.warn("The twython library has not been installed. "
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

```
# Seaborn plot visualizations
import matplotlib.pyplot as plt
import seaborn as sb

# 1) Violin Plot
plt.figure(figsize=(8,8))
sb.set_style('darkgrid')
plt.title("Violin Plot of Sentiment Analysis")
sb.violinplot(x = sentiment_df.iloc[:, -1].values, y = sentiment_df.iloc[:, -2].values,
            data = sentiment_df, palette = sb.set_palette('magma', n_colors = 1))
plt.ylabel("Compound Scores")
plt.show()

# 2) Box Plot
plt.figure(figsize=(8,8))
sb.set_style('darkgrid')
plt.title("Box Plot of Sentiment Analysis")
sb.boxenplot(x = sentiment_df.iloc[:, -1].values, y = sentiment_df.iloc[:, -2].values,
            data = sentiment_df, palette = sb.set_palette('magma', n_colors = 6))
plt.ylabel("Compound Scores")
plt.show()
```


Violin Plot of Sentiment Analysis
Compound Scores
3
2
1
0
−1
−2
−3
Neutral
Positive
Negative


Box Plot of Sentiment Analysis
Compound Scores
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Neutral
Positive
Negative