Name: Sreejith Sathyadevan

Roll No: 23

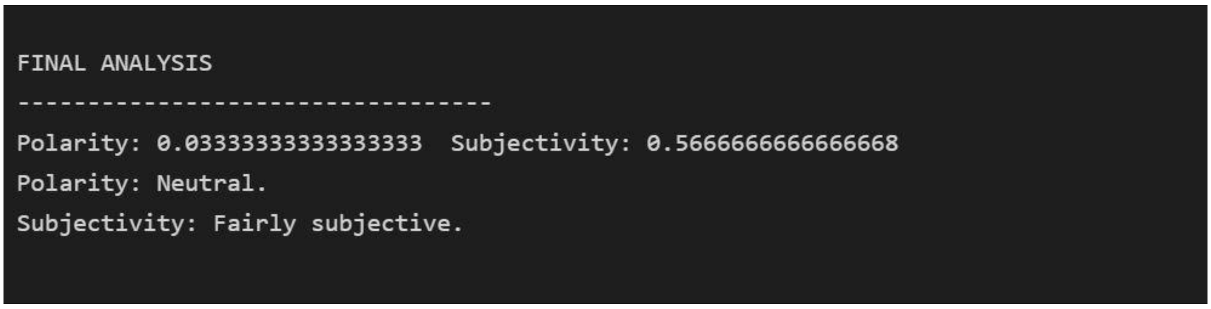
SNA Mini Project

Sentiment Analysis

Code:

```
from textblob import TextBlob
import nltk

news_summary=input("Enter the article:")
def find_sentiment(news_story):
    news = TextBlob(news_story)
    sentiments = []
    for sentence in news.sentences:
        sentiment = sentence.sentiment
        for metric in sentiment:
            sentiments.append(metric)

    # Every even index in the list corresponds to polarity and the rest relate to subjectivity.
    # Using this, the polarity_data and subjectivity_data lists are filled accurately.
    polarity_data = []
    subjectivity_data = []
    for i in range(len(sentiments)):
        if i % 2 == 0:
            polarity_data.append(sentiments[i])
        else:
            subjectivity_data.append(sentiments[i])

    # The averages of both sentiment lists are calculated.
    polarity_average = calculate_average(polarity_data)
    subjectivity_average = calculate_average(subjectivity_data)

    # Displays the sentiment that relates to the averages on the console.
    print()
    print("FINAL ANALYSIS")
    print("---------------------------------")
    print("Polarity:",polarity_average," Subjectivity:",subjectivity_average)
    print("Polarity: " + calculate_sentiment(polarity_average, "polarity"))
    print("Subjectivity: " + calculate_sentiment(subjectivity_average, "subjectivity"))


# Helper Methods (for the find_sentiment method)
# ---------------------------------------------------------

# Method Purpose: Given a list with numeric values, calculates and returns the average of all.
def calculate_average(list):
    return sum(list) / len(list)


# Method Purpose: Given an average polarity or subjectivity, uses intervals to calculate accurate
sentiments.
# Note: Polarity and Subjectivity in TextBlob fall in between -1 and 1, this method bases its intervals off of
that.
def calculate_sentiment(sentiment, type):
    sentiment_category = ""
    if type == "polarity":
        if sentiment > 0.75:
            sentiment_category = "Extremely positive."
        elif sentiment > 0.5:
            sentiment_category = "Significantly positive."
        elif sentiment > 0.3:
            sentiment_category = "Fairly positive."
        elif sentiment > 0.1:
            sentiment_category = "Slightly positive."
        elif sentiment < -0.1:
            sentiment_category = "Slightly negative."
        elif sentiment < -0.3:
            sentiment_category = "Fairly negative."
        elif sentiment < -0.5:
            sentiment_category = "Significantly negative."
        elif sentiment < -0.75:
            sentiment_category = "Extremely negative."
        else:
            sentiment_category = "Neutral."
        return sentiment_category
    elif type == "subjectivity":
        if sentiment > 0.75:
            sentiment_category = "Extremely subjective."
        elif sentiment > 0.5:
            sentiment_category = "Fairly subjective."
        elif sentiment > 0.3:
            sentiment_category = "Fairly objective."
        elif sentiment > 0.1:
            sentiment_category = "Extremely objective."
        return sentiment_category
    else:
        print("Invalid Input.")

find_sentiment(news_summary)
```

Output:



```
FINAL ANALYSIS
----------------------------------
Polarity: 0.03333333333333333  Subjectivity: 0.5666666666666668
Polarity: Neutral.
Subjectivity: Fairly subjective.
```