Artificial Intelligence Assignment

**Implementation of python on sentiment analysis using VADER**

INTRODUCTION

[Natural Language Processing](https://en.wikipedia.org/wiki/Natural_language_processing) (NLP) covers a broad range of techniques that apply computational analytical methods to textual content, which provide means of categorizing and quantifying text. These NLP approaches, which include sentiment analysis, can help researchers explore their textual data. In the words of Tukey, it can help the researcher to find “clues” about their texts and “indications” that something might be worth investigating further.

**Sentiment Analysis** is the process of ‘computationally’ determining whether a piece of writing is positive, negative or neutral. It’s also known as **opinion mining**, deriving the opinion or attitude of a speaker.

Some sentiment analysis tools can also factor in the emotional weight of other features of language such as punctuation or **[emojis](https://en.wikipedia.org/wiki/Emoji" \t "_blank).** Sentiment analysis tools generally process a unit of text (a sentence, paragraph, book, etc) and output quantitative scores or classifications to indicate whether the algorithm considers that text to convey **positive or negative emotion**. Some tools can also quantify the **degree of positivity** or **degree of negativity** within a text. Combined with other NLP methods like [**topic modeling**](https://programminghistorian.org/lessons/topic-modeling-and-mallet), sentiment analysis provides a means of characterizing the emotions expressed about different topics of conversation.

**Why sentiment analysis?**

Business: In marketing field companies use it to develop their strategies, to understand customers’ feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don’t buy some products.

Politics: In the political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well! .

Public Actions: Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

IMPLEMENTATION

**VADER Sentiment Analysis :**

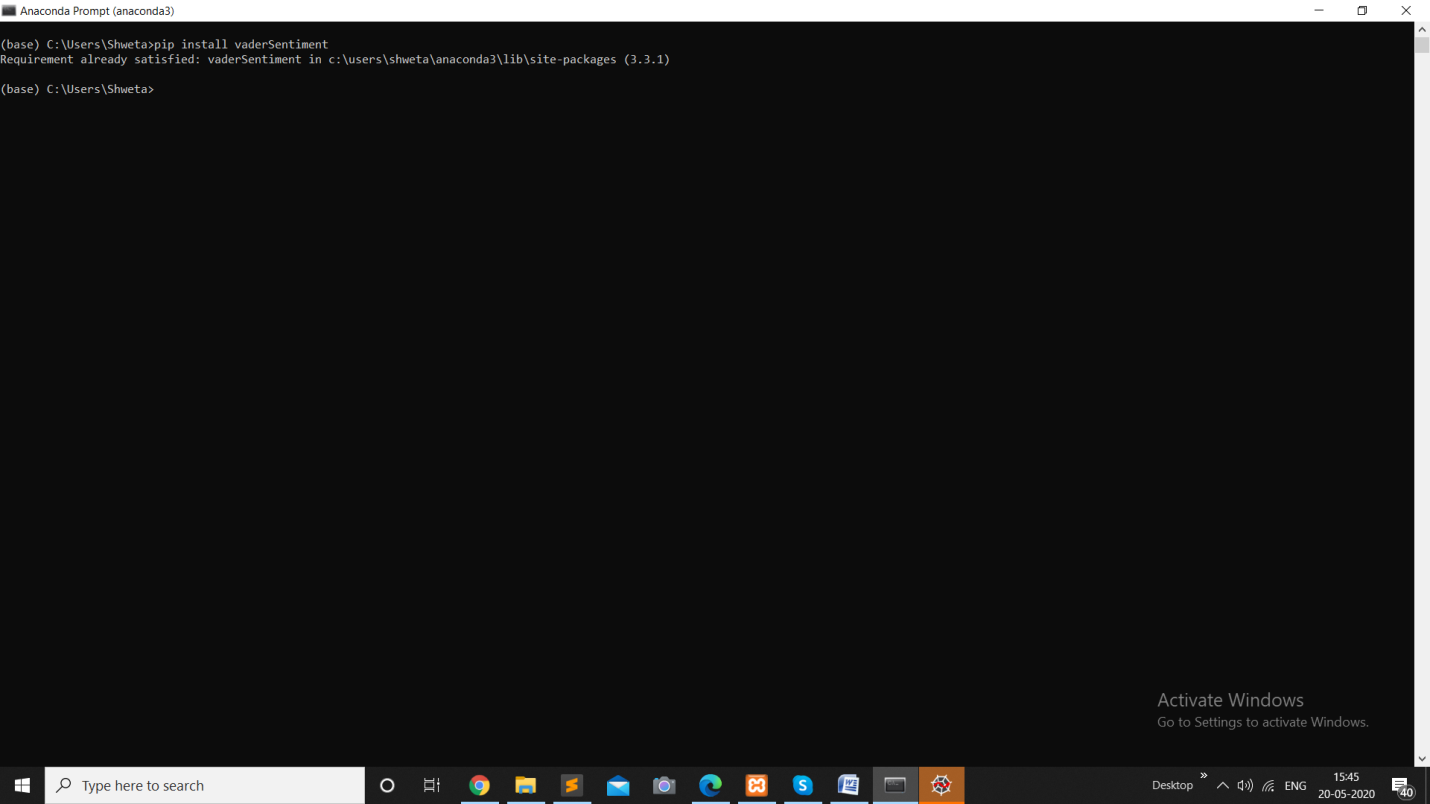
**VADER (Valence Aware Dictionary and sEntiment Reasoner)** is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

Unlike other techniques that require training on related text before use, VADER is ready to go for analysis without any special setup. VADER is unique in that it makes fine-tuned distinctions between varying degrees of positivity and negativity. For example, VADER scores “comfort” moderately positively and “euphoria” extremely positively. It also attempts to capture and score textual features common in informal online text such as capitalizations, exclamation points, and emoticons

The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

positive sentiment : (compound score >= 0.05)  
neutral sentiment : (compound score > -0.05) and (compound score < 0.05)  
negative sentiment : (compound score <= -0.05)

Command to install **vaderSentiment** : pip install vaderSentiment



PYTHON CODE

# import SentimentIntensityAnalyzer class

# from vaderSentiment.vaderSentiment module.

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# function to print sentiments

# of the sentence.

def sentiment\_scores(sentence):

# Create a SentimentIntensityAnalyzer object.

sid\_obj = SentimentIntensityAnalyzer()

# polarity\_scores method of SentimentIntensityAnalyzer

# object gives a sentiment dictionary.

# which contains pos, neg, neu, and compound scores

sentiment\_dict = sid\_obj.polarity\_scores(sentence)

print("Overall sentiment dictionary is : ", sentiment\_dict)

print("sentence was rated as ", sentiment\_dict['neg']\*100, "% Negative")

print("sentence was rated as ", sentiment\_dict['neu']\*100, "% Neutral")

print("sentence was rated as ", sentiment\_dict['pos']\*100, "% Positive")

print("Sentence Overall Rated As", end = " ")

# decide sentiment as positive, negative and neutral

if sentiment\_dict['compound'] >= 0.05 :

print("Positive")

elif sentiment\_dict['compound'] <= - 0.05 :

print("Negative")

else :

print("Neutral")

# Driver code

if \_\_name\_\_ == "\_\_main\_\_" :

print("\n1st statement :")

sentence = "health is wealth."

# function calling

sentiment\_scores(sentence)

print("\n2nd Statement :")

sentence = "study is going on as usual"

sentiment\_scores(sentence)

print("\n3rd Statement :")

sentence = "the girl sitting there is vey sad today."

sentiment\_scores(sentence)

OUTPUT

1st statement :

Overall sentiment dictionary is : {'neg': 0.0, 'neu': 0.385, 'pos': 0.615, 'compound': 0.4939}

sentence was rated as 0.0 % Negative

sentence was rated as 38.5 % Neutral

sentence was rated as 61.5 % Positive

Sentence Overall Rated As Positive

2nd Statement :

Overall sentiment dictionary is : {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

sentence was rated as 0.0 % Negative

sentence was rated as 100.0 % Neutral

sentence was rated as 0.0 % Positive

Sentence Overall Rated As Neutral

3rd Statement :

Overall sentiment dictionary is : {'neg': 0.307, 'neu': 0.693, 'pos': 0.0, 'compound': -0.4767}

sentence was rated as 30.7 % Negative

sentence was rated as 69.3 % Neutral

sentence was rated as 0.0 % Positive

Sentence Overall Rated As Negative

