

# Python | Sentiment Analysis using VADER

**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.

## 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.

Command to install **vaderSentiment** :

```
pip install vaderSentiment
```

## **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.

Below is the code:

- Python3

```
from vaderSentiment.vaderSentiment import  
SentimentIntensityAnalyzer  
  
def sentiment_scores(sentence):  
  
    sid_obj = SentimentIntensityAnalyzer()  
  
    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 = " ")

    if sentiment_dict['compound'] >= 0.05 :

        print("Positive")

    elif sentiment_dict['compound'] <= - 0.05 :

        print("Negative")

    else :

        print("Neutral")

if __name__ == "__main__" :

    print("\n1st statement :")

    sentence = "Geeks For Geeks is the best portal for \
                the computer science engineering students."

    sentiment_scores(sentence)

    print("\n2nd Statement :")

```

```
sentence = "study is going on as usual"

sentiment_scores(sentence)

print("\n3rd Statement :")

sentence = "I am very sad today."

sentiment_scores(sentence)
```

## Output :

```
1st statement :
Overall sentiment dictionary is : {'neg': 0.165, 'neu': 0.588, 'pos': 0.247, 'compound': 0.5267}
sentence was rated as 16.5 % Negative
sentence was rated as 58.8 % Neutral
sentence was rated as 24.7 % 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': 0.508, 'neu': 0.492, 'pos': 0.0, 'compound': -0.4767}
sentence was rated as 50.8 % Negative
sentence was rated as 49.2 % Neutral
sentence was rated as 0.0 % Positive
Sentence Overall Rated As Negative
```

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  $\geq 0.05$ )

neutral sentiment : (compound score  $> -0.05$ ) and  
(compound score  $< 0.05$ )

negative sentiment : (compound score  $\leq -0.05$ )

Looking to dive into the world of programming or sharpen

your Python skills? Our [Master Python: Complete Beginner to Advanced Course](#) is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!