ABSTRACT

Sentiment analysis or Opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in the natural language processing and text mining in recent years. It's popularity is mainly because it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviors.

Sentiment classification is the way to analyze the subjective information in the text and then mine the opinion. Sentiment analysis is the procedure by which information is extracted from the opinions, appraisals and emotions of people in regards to entities, events and their attributes. The approaches of text sentiment analysis typically work at a particular level like phrase, sentence or document level. This project aims at analyzing a solution for sentiment classification using VADER (Valence Aware Dictionary and sEntiment Reasoner), which is a lexicon and rule-based sentiment analysis tool.

VADER has been found to be quite successful when dealing with social media texts, movie reviews, and product reviews. This is because VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is. It is fully open-sourced under the MIT License.

INDEX

- INTRODUCTION
  - Objective
  - Problem Definition
- METHODOLOGY (Design and Implementation)
  - System Design
  - Implementation of Proposed Solution
  - System requirements
- RESULTS AND DISCUSSION
- CONCLUSION AND FUTURE WORK
- REFERENCES
- SOURCE CODE

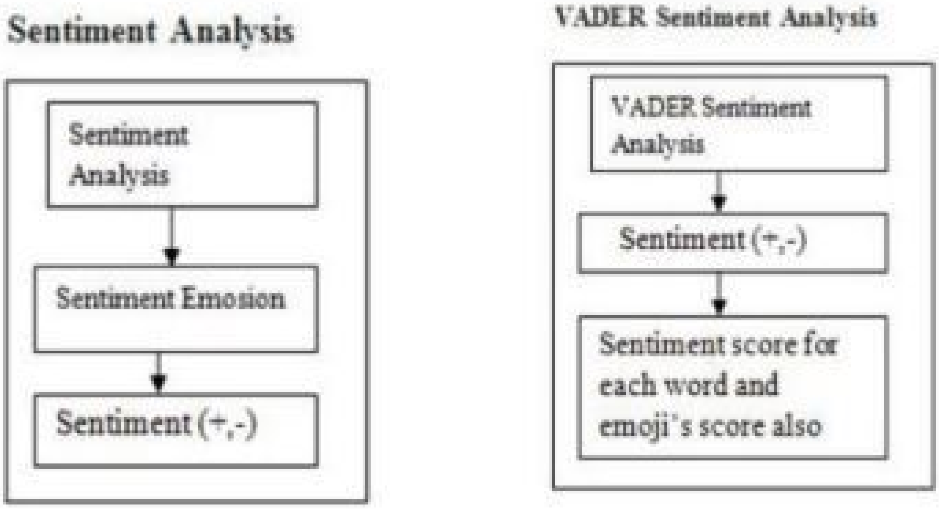
1.INTRODUCTION

1.1- OBJECTIVE

Sentiment classification is a way to analyse the subjective information in the text and then mine the opinion. Sentiment analysis is the procedure by which information is extracted from the opinions, appraisals and emotions of people in regards to entities, events and their attributes. In decision making, the opinions of others have a significant effect on customers ease, making choices with regards to online shopping, choosing events, products, entities.

1.2-PROBLEM STATEMENT

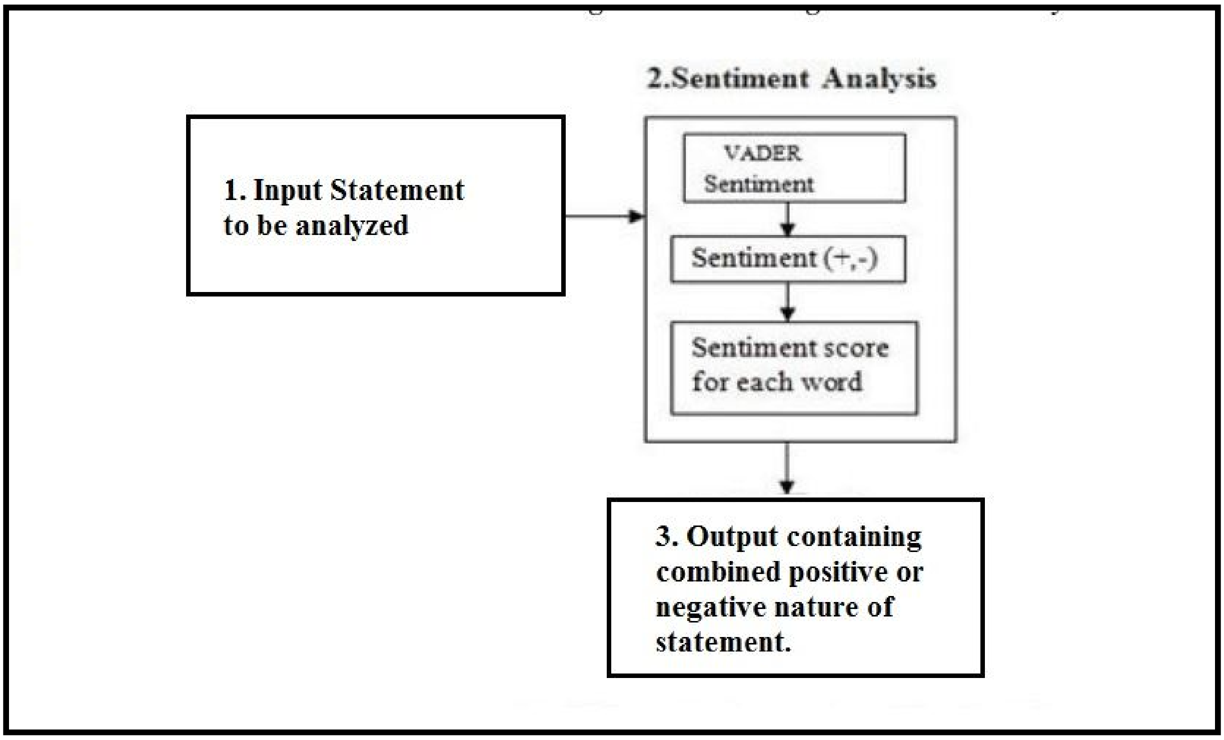
Sentiment Analysis or Opinion Mining is a study that attempts to identify and analyze emotions and subjective information from text. Since early 2001, the advancement of internet technology and machine learning techniques in information retrieval make Sentiment Analysis become popular among researchers. Sentiment Analysis or mining refers to the application of Natural Language Processing, Computational Linguistics, and Text Analytics to identify and extract subjective information in source materials. Sentiment Mining extracts the attitude of a writer in a document, writer’s judgement and evaluation towards the discussed issue. Product review data extraction is done using Python and it’s package rich library tools. VADER tool integrated with NLTK with it’s Sentiment Intensity Analyzer (SIA) has been used for unsupervised sentiment classification.



BASIC DIFFERENCE BETWEEN SENTIMENT ANALYSIS AND VADER SENTIMENT ANALYSIS

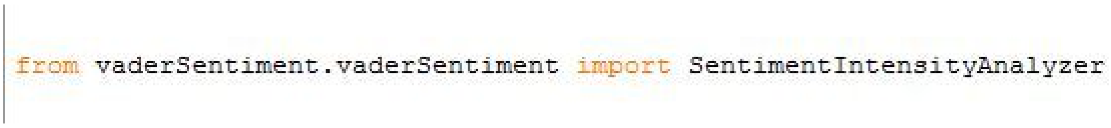
2.METHODOLOGY OF DESIGN AND IMPLEMENTATION

2.1-SYSTEM DESIGN


2.Sentiment Analysis
1. Input Statement to be analyzed
VADER Sentiment
Sentiment (+,-)
Sentiment score for each word
3. Output containing combined positive or negative nature of statement.

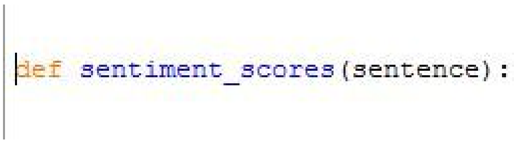
2.2- IMPLEMENTATION OF PROPOSED SYSTEM

1. Importing SentimentIntensityAnalyzer class from vaderSentiment.vaderSentiment module :



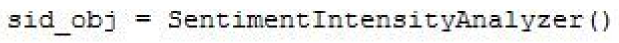
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

2. Function to print sentiments of the sentence:



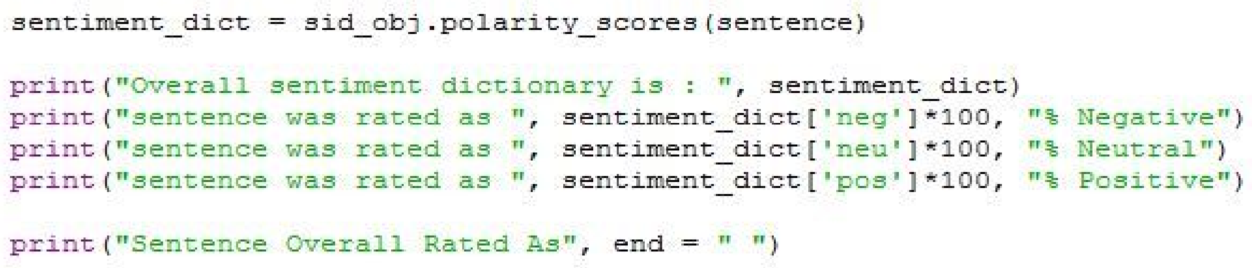
```
def sentiment_scores(sentence):
```

3. Creating a SentimentIntensityAnalyzer object:



```
sid_obj = SentimentIntensityAnalyzer()
```

4. Determining positive, negative, neutral and compound scores using polarity_scores method:

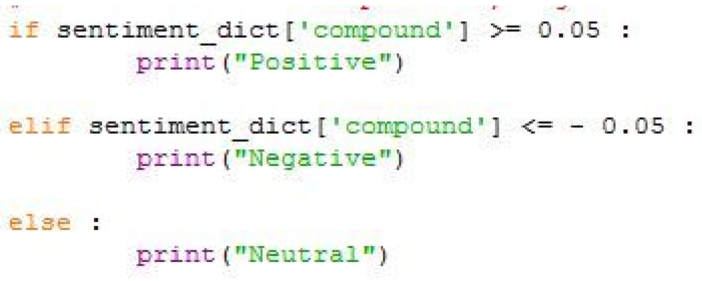


```
sentiment_dict = sid_obj.polarity_scores(sentence)

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 = " ")
```

5. Deciding on the positive, negative or neutral nature of the input:

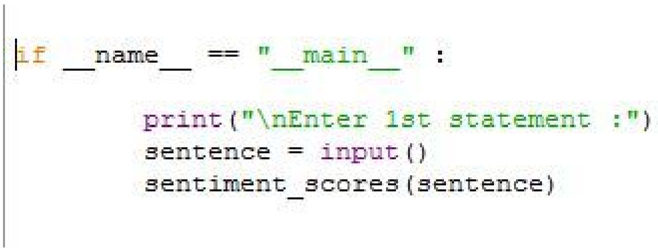


```
if sentiment_dict['compound'] >= 0.05 :
        print("Positive")

elif sentiment_dict['compound'] <= - 0.05 :
        print("Negative")

else :
        print("Neutral")
```

6. Driver code and function calling:



```
if __name__ == "__main__" :

        print("\nEnter 1st statement :")
        sentence = input()
        sentiment_scores(sentence)
```

2.3- SYSTEM REQUIREMENTS

2.3.1- HARDWARE REQUIREMENTS :

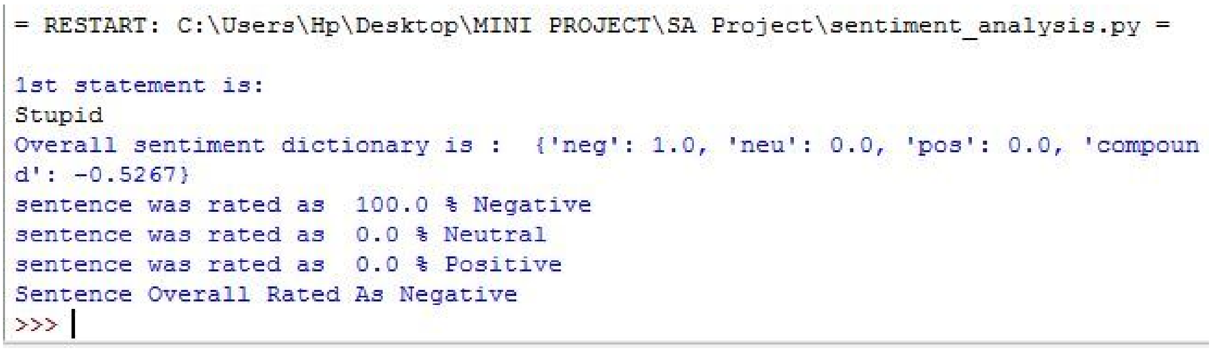
- A 32-bit or 64-bit processor.
- Sufficient RAM to run the program.

2.3.2- SOFTWARE REQUIREMENTS :

- Python 3 or above to be installed.
- Installation of the SentimentIntensityAnalyzer module in python.
- Installation on NLTK module in python.

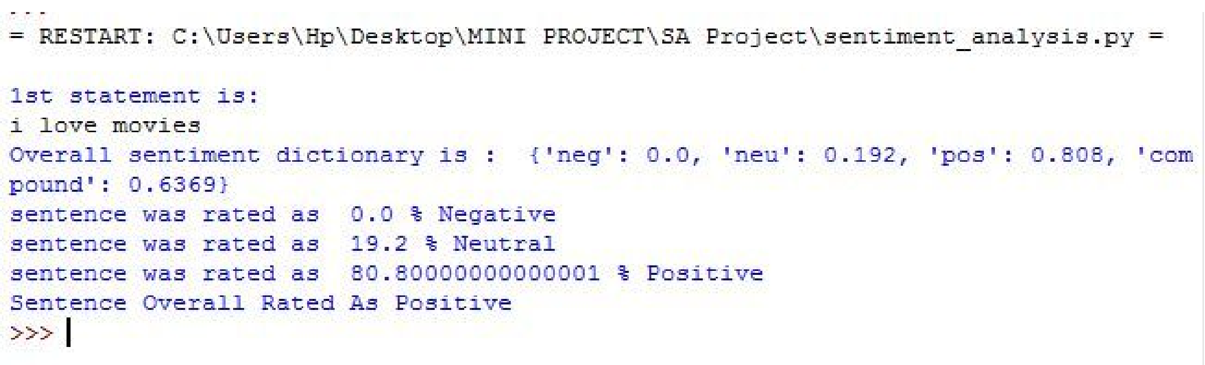
3. RESULTS AND OUTPUT

Following are a few outputs obtained on running the program :



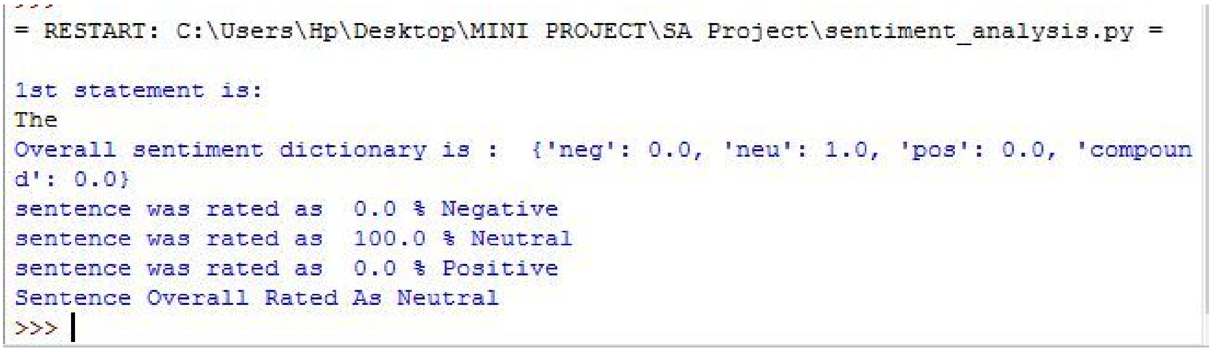
```
= RESTART: C:\Users\Hp\Desktop\MINI PROJECT\SA Project\sentiment_analysis.py =

1st statement is:
Stupid
Overall sentiment dictionary is :  {'neg': 1.0, 'neu': 0.0, 'pos': 0.0, 'compoun
d': -0.5267}
sentence was rated as  100.0 % Negative
sentence was rated as  0.0 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Negative
>>> 
```



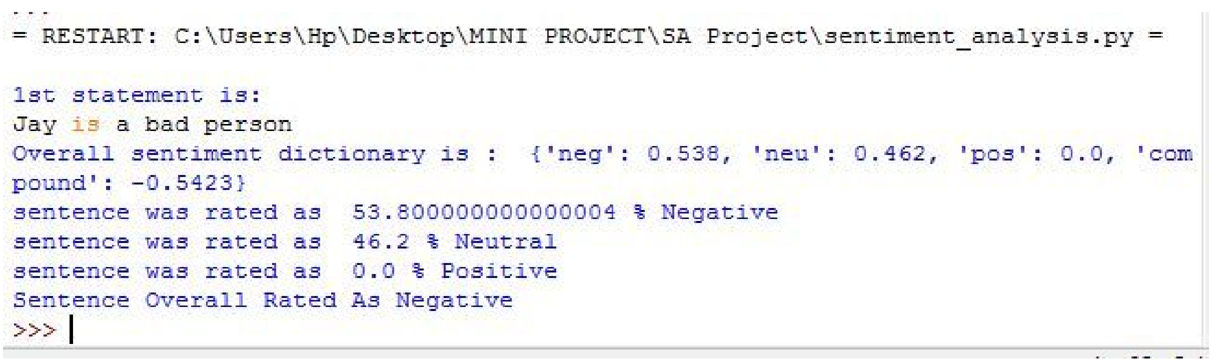
```
...
= RESTART: C:\Users\Hp\Desktop\MINI PROJECT\SA Project\sentiment_analysis.py =

1st statement is:
i love movies
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.192, 'pos': 0.808, 'com
pound': 0.6369}
sentence was rated as  0.0 % Negative
sentence was rated as  19.2 % Neutral
sentence was rated as  80.80000000000001 % Positive
Sentence Overall Rated As Positive
>>> 
```



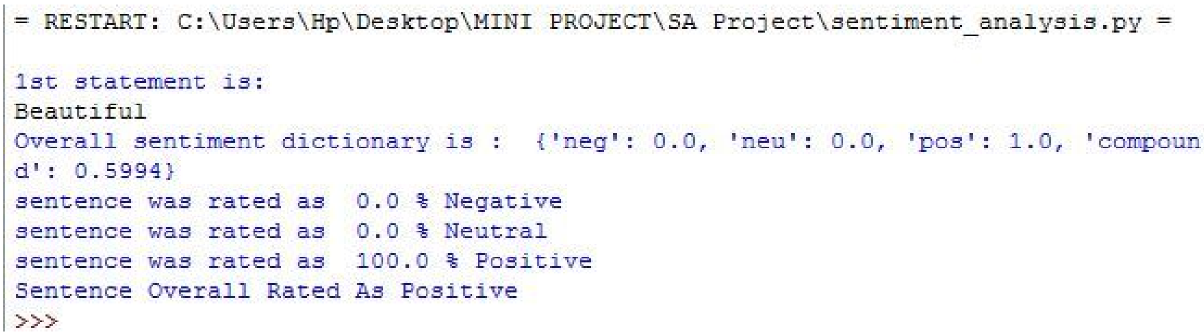
```
= RESTART: C:\Users\Hp\Desktop\MINI PROJECT\SA Project\sentiment_analysis.py =

1st statement is:
The
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compoun
d': 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
>>> 
```



```
...
= RESTART: C:\Users\Hp\Desktop\MINI PROJECT\SA Project\sentiment_analysis.py =

1st statement is:
Jay is a bad person
Overall sentiment dictionary is :  {'neg': 0.538, 'neu': 0.462, 'pos': 0.0, 'com
pound': -0.5423}
sentence was rated as  53.800000000000004 % Negative
sentence was rated as  46.2 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Negative
>>> 
```



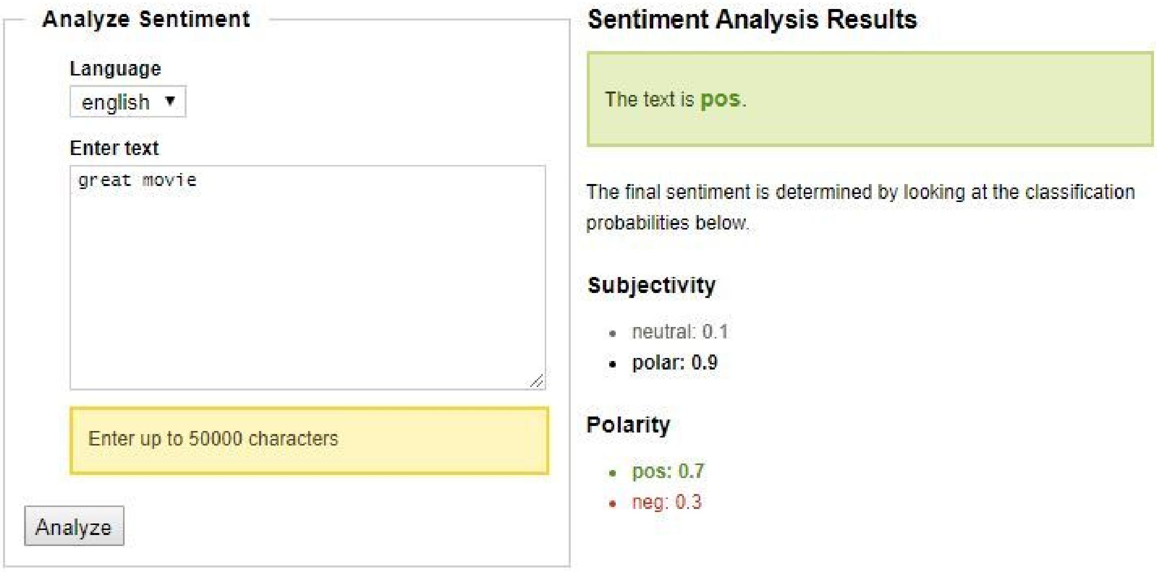
```
= RESTART: C:\Users\Hp\Desktop\MINI PROJECT\SA Project\sentiment_analysis.py =

1st statement is:
Beautiful
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.0, 'pos': 1.0, 'compoun
d': 0.5994}
sentence was rated as  0.0 % Negative
sentence was rated as  0.0 % Neutral
sentence was rated as  100.0 % Positive
Sentence Overall Rated As Positive
>>>
```

4.CONCLUSION AND FUTURE PROJECTS

Sentiment analysis is a useful tool for any organization or group for which public sentiment or attitude towards them is important for their success - whichever way that success is defined.On social media, blogs, and online forums millions of people are busily discussing and reviewing businesses, companies, and organizations. And those opinions are being ‘listened to’ and analysed.The results from sentiment analysis help businesses understand the conversations and discussions taking place about them, and helps them react and take action accordingly.They can quickly identify any negative sentiments being expressed, and turn poor customer experiences into very good ones.

I would like to create an online Sentiment Analyzer as an enhancement to the current project. The look of the online analyzer will look like the follows:



Analyze Sentiment

Language: english ▾

Enter text: great movie

Enter up to 50000 characters

Sentiment Analysis Results

The text is **pos**.

The final sentiment is determined by looking at the classification probabilities below.

Subjectivity

- neutral: 0.1
- **polar: 0.9**

Polarity

- pos: 0.7
- neg: 0.3

5. REFERENCES

Following are the links to the websites which gave me a clear insight to the project:

https://web.stanford.edu/class/cs124/lec/sentiment.pdf

https://cloud.google.com/natural-language/docs/sentiment-tutorial

https://datascience.stackexchange.com/questions/8390/sentiment-analysis-tutorial

http://fjavieralba.com/basic-sentiment-analysis-with-python.html

http://andybromberg.com/sentiment-analysis-python/

https://github.com/abromberg/sentiment_analysis_python

http://andybromberg.com/sentiment-analysis/

http://sentiment.christopherpotts.net/

https://www.slideshare.net/snehapenmetsa/project-sentiment-analysis-58103144

https://medium.com/@martinpella/naive-bayes-for-sentiment-analysis-49b37db18bf8

https://triton.ml/blog/sentiment-analysis

https://github.com/martinpella/metacritic

6.SOURCE CODE

Sentiment_analysis.py :

```
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 is:")
        sentence = input()

        # function calling
        sentiment_scores(sentence)
```