Sentiment Analysis using textblob

In [1]:

```
#!pip install textblob
import textblob
```

In [2]:

```
from textblob import TextBlob
```

In [3]:

```
txt1 = TextBlob('This was a good movie')
```

In [4]:

```
txt1.sentiment
```

Out[4]:

```
Sentiment(polarity=0.7, subjectivity=0.6000000000000001)
```

In [5]:

```
txt2 = TextBlob('This was a bad movie')
```

In [6]:

```
txt2.sentiment
```

Out[6]:

```
Sentiment(polarity=-0.6999999999999998, subjectivity=0.6666666666666666)
```

In [7]:

```
txt3 = TextBlob('This was an okay movie')
```

In [8]:

```
txt3.sentiment
```

Out[8]:

```
Sentiment(polarity=0.5, subjectivity=0.5)
```

Sentiment Analysis using Vader

In [9]:

```
import vaderSentiment
```

In [10]:
```
!pip install vaderSentiment
```

```
Requirement already satisfied: vaderSentiment in c:\users\yashm\anaconda3\lib\site-packages (3.3.2)
Requirement already satisfied: requests in c:\users\yashm\anaconda3\lib\site-packages (from vaderSentiment) (2.25.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\yashm\anaconda3\lib\site-packages (from requests->vaderSen
timent) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\yashm\anaconda3\lib\site-packages (from requests->vaderSentimen
t) (2.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\yashm\anaconda3\lib\site-packages (from requests->vade
rSentiment) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\yashm\anaconda3\lib\site-packages (from requests->vaderSe
ntiment) (2019.11.28)


WARNING: You are using pip version 20.3.3; however, version 22.2.2 is available.
You should consider upgrading via the 'C:\Users\yashm\anaconda3\python.exe -m pip install --upgrade pip' command.
```

In [11]:
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

In [12]:
```
sentiment_analyser = SentimentIntensityAnalyzer()
```

In [13]:
```
sent = "This was a great movie"
```

In [14]:
```
sent_analysis = sentiment_analyser.polarity_scores(sent)
```

In [15]:
```
sent_analysis
```

Out[15]:
```
{'neg': 0.0, 'neu': 0.494, 'pos': 0.506, 'compound': 0.6249}
```

In [16]:
```
sent_analysis
```

Out[16]:
```
{'neg': 0.0, 'neu': 0.494, 'pos': 0.506, 'compound': 0.6249}
```

Excercise

In [17]:

```
import pandas as pd

data = pd.read_csv('Tweets.csv')
```

In [18]:

```
data[['airline_sentiment','airline_sentiment_confidence','text']]


```

Out[18]:

	airline_sentiment	airline_sentiment_confidence	text
0	neutral	1.0000	@VirginAmerica What @dhepburn said.
1	positive	0.3486	@VirginAmerica plus you've added commercials t...
2	neutral	0.6837	@VirginAmerica I didn't today... Must mean I n...
3	negative	1.0000	@VirginAmerica it's really aggressive to blast...
4	negative	1.0000	@VirginAmerica and it's a really big bad thing...
...	...	...	...
14635	positive	0.3487	@AmericanAir thank you we got on a different f...
14636	negative	1.0000	@AmericanAir leaving over 20 minutes Late Flig...
14637	neutral	1.0000	@AmericanAir Please bring American Airlines to...
14638	negative	1.0000	@AmericanAir you have my money, you change my ...
14639	neutral	0.6771	@AmericanAir we have 8 ppl so we need 2 know h...

14640 rows × 3 columns

Excercise:

1. Preprocess the text with the techniques dicussed yesterday
2. Creat a user defined function that returns the compound sentiment from the vader function
3. If the compound sentiment is greater than 0.5 it is positive and if it is less than -0.5 it is negative and if it is between -0.5 to 0.5 it is neutral
4. Compare the above output with the column of airline sentiment and evaluate your accuracy

In [19]:

```
df = data[['airline_sentiment','airline_sentiment_confidence','text']].iloc[:500]

from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize



def remove_punc(string):
    punc = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
    for char in string:
        if char in punc:
            string = string.replace(char,"")
    return string

def stem_text(string):
    ps = SnowballStemmer(language = 'english')
    words = word_tokenize(string)
    sentence = []
    for word in words:
        sentence.append(ps.stem(word))
    return " ".join(sentence)

def lower(string):
    return string.lower()



def clean_text(string):
    string = remove_punc(string)
    string = stem_text(string)
    return string.lower()


df['clean_text'] = df['text'].apply(clean_text)
```

In [24]:

```python
def vader_sentiment_analyser(sent):
    sentiment_analyser = SentimentIntensityAnalyzer()
    compound = sentiment_analyser.polarity_scores(sent)['compound']
    if compound>0.5:
        return 'positive'
    elif compound<-0.5:
        return 'negative'
    else:
        return 'neutral'

```

In [25]:

```python
df['vader_score'] = df['clean_text'].apply(vader_sentiment_analyser)
```

In [26]:

```python
from sklearn.metrics import classification_report
```

In [27]:

```python
print(classification_report(df['vader_score'],df['airline_sentiment']))
```

```
              precision    recall  f1-score   support

    negative       0.21      0.95      0.34        40
     neutral       0.86      0.40      0.55       363
    positive       0.44      0.69      0.54        97

    accuracy                           0.50       500
   macro avg       0.51      0.68      0.48       500
weighted avg       0.73      0.50      0.53       500

```

In [ ]:

```python

```