6. NLP – Twitter Sentiment Analysis – Textblob

Contents

6. NLP – Twitter Sentiment Analysis – Textblob

1. TextBlob

- ✓ TextBlob provides an API that can perform different Natural Language Processing (NLP) tasks like,
  - Part-of-Speech Tagging
  - Noun Phrase Extraction
  - Sentiment Analysis
  - Classification (Naive Bayes, Decision Tree)
  - Language Translation and Detection
  - Spelling Correction etc.

- ✓ TextBlob is built upon Natural Language Toolkit (NLTK).
- ✓ Sentiment Analysis means analysing the sentiment of a given text or document and categorizing the text/document into a specific class or category (like positive and negative).
- ✓ Basically, the classification is done for two classes: positive and negative.
- ✓ However, we can add more classes like neutral, highly positive, highly negative, etc.

2. Installing textblob library

- ✓ pip install -U textblob

3. Simple TextBlob Sentiment Analysis Example

- ✓ We can apply textblob on any text to do Sentiment Analysis
- ✓ The sentiment property gives the sentiment scores to the given text.
- ✓ There are two scores given: Polarity and Subjectivity.
  - The polarity score is a float within the range [-1.0, 1.0] where negative value indicates negative text and positive value indicates that the given text is positive.
  - The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

Program Name: Sentiment analysis
demo1.py

```
from textblob import TextBlob

text = TextBlob("It was a wonderful movie. I liked it very much.")

print (text.sentiment)
print ('polarity: {}'.format(text.sentiment.polarity))
print ('subjectivity: {}'.format(text.sentiment.subjectivity))
```

Output

```
Sentiment(polarity=0.62, subjectivity=0.6866666666666666)
polarity: 0.62
subjectivity: 0.6866666666666666
```

Program Name: Sentiment analysis
demo2.py

```
from textblob import TextBlob

text = TextBlob("I liked the acting of the lead actor but I didn't like the movie overall.")

print (text.sentiment)
print ('polarity: {}'.format(text.sentiment.polarity))
print ('subjectivity: {}'.format(text.sentiment.subjectivity))
```

Output

```
Sentiment(polarity=0.19999999999999998, subjectivity=0.26666666666666666)
polarity: 0.19999999999999998
subjectivity: 0.26666666666666666
```

Program Name: Sentiment analysis
demo3.py

```
from textblob import TextBlob

text = TextBlob("I liked the acting of the lead actor and I liked the movie overall.")

print (text.sentiment)
print ('polarity: {}'.format(text.sentiment.polarity))
print ('subjectivity: {}'.format(text.sentiment.subjectivity))
```

Output

```
Sentiment(polarity=0.3, subjectivity=0.4)
polarity: 0.3
subjectivity: 0.4
```

4. Using NLTK's Twitter Corpus

- ✓ We use the twitter_samples corpus to train the TextBlob's NaiveBayesClassifier.
- ✓ Using the twitter_samples corpus, we create a train set and test set containing a certain amount of positive and negative tweets.
- ✓ And, then we test the accuracy of the trained classifier.

Program Name: Getting twitter sample datasets
demo4.py

```
from nltk.corpus import twitter_samples
import nltk
nltk.download('twitter_samples')

print (twitter_samples.fileids())
```

Output

```
['negative_tweets.json', 'positive_tweets.json', 'tweets.20150430-223406.json']
```

Program Name: Getting twitter sample datasets and checking length
demo5.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')
print(len(pos_tweets))

neg_tweets = twitter_samples.strings('negative_tweets.json')
print(len(neg_tweets))
```

Output

```
5000
5000
```

Program Name: Getting twitter sample datasets and checking length
demo6.py

```
from nltk.corpus import twitter_samples

all_tweets = twitter_samples.strings('tweets.20150430-223406.json')
print (len(all_tweets))
```

Output

```
20000
```

Program Name: Getting positive tweets
demo7.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

for i in pos_tweets_set:
        print(i)
```

Output

```
('#FollowFriday @France_Inte @PKuchly57 @Milipol_Paris for being top engaged members in my community th
is week :)', 'pos')
('@Lamb2ja Hey James! How odd :/ Please call our Contact Centre on 02392441234 and we will be able to a
ssist you :) Many thanks!', 'pos')
('@DespiteOfficial we had a listen last night :) As You Bleed is an amazing track. When are you in Scot
land?!', 'pos')
('@97sides CONGRATS :)', 'pos')
('yeaaaah yippppy!!!  my accnt verified rqst has succeed got a blue tick mark on my fb profile :) in 15
 days', 'pos')
('@BhaktisBanter @PallaviRuhail This one is irresistible :)\n#FlipkartFashionFriday http://t.co/EbZ0L2V
ENM', 'pos')
("We don't like to keep our lovely customers waiting for long! We hope you enjoy! Happy Friday! - LWWF
:) https://t.co/smyYriipxI", 'pos')
('@Impatientraider On second thought, there's just not enough time for a DD :) But new shorts entering
system. Sheep must be buying.', 'pos')
```

Program Name: Getting negative tweets
demo8.py

```
from nltk.corpus import twitter_samples

neg_tweets = twitter_samples.strings('negative_tweets.json')

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

for i in neg_tweets_set:
        print(i)
```

Output

```
('hopeless for tmr :(', 'neg')
("Everything in the kids section of IKEA is so cute. Shame I'm nearly 19 in 2 months :(", 'neg')
('@Hegelbon That heart sliding into the waste basket. :(', 'neg')
('“@ketchBurning: I hate Japanese call him "bani" :( :(”\n\nMe too', 'neg')
('Dang starting next week I have "work" :(', 'neg')
("oh god, my babies' faces :( https://t.co/9fcwGvaki0", 'neg')
('@RileyMcDonough make me smile :((', 'neg')
('@f0ggstar @stuartthull work neighbour on motors. Asked why and he said hates the updates on search :(
 http://t.co/XvmTUikWln', 'neg')
('why?:("@tahuodyy: sialan:( https://t.co/Hv1i0xcrL2"', 'neg')
('Athabasca glacier was there in #1948 :-( #athabasca #glacier #jasper #jaspernationalpark #alberta #ex
plorealberta #… http://t.co/dZZdqmf7Cz', 'neg')
("I have a really good m&amp;amp;g idea but I'm never going to meet them :(((", 'neg')
('@Rampageinthebox mare ivan :(', 'neg')
```

Program Name: Splitting Training and testing datasets
demo9.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

from random import shuffle

shuffle(pos_tweets_set)
shuffle(neg_tweets_set)

test_set = pos_tweets_set[:300] + neg_tweets_set[:300]
train_set = pos_tweets_set[300:600] + neg_tweets_set[300:600]

print("Training and testing datasets")
```

Output

```
Training and testing datasets
```

Program Name: Creating model and training the model
demo10.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

from random import shuffle

shuffle(pos_tweets_set)
shuffle(neg_tweets_set)

test_set = pos_tweets_set[:300] + neg_tweets_set[:300]
train_set = pos_tweets_set[300:600] + neg_tweets_set[300:600]

from textblob.classifiers import NaiveBayesClassifier
classifier = NaiveBayesClassifier(train_set)

print("Model got trained")
```

Output

```
Model got trained
```

Program Name: Testing model accuracy
demo11.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

from random import shuffle

shuffle(pos_tweets_set)
shuffle(neg_tweets_set)

test_set = pos_tweets_set[:300] + neg_tweets_set[:300]
train_set = pos_tweets_set[300:600] + neg_tweets_set[300:600]

from textblob.classifiers import NaiveBayesClassifier
classifier = NaiveBayesClassifier(train_set)

accuracy = classifier.accuracy(test_set)
print (accuracy)
```

Output

```
0.73666666666667
```

Program Name: Testing the model
demo12.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

from random import shuffle

shuffle(pos_tweets_set)
shuffle(neg_tweets_set)

test_set = pos_tweets_set[:300] + neg_tweets_set[:300]
train_set = pos_tweets_set[300:600] + neg_tweets_set[300:600]

from textblob.classifiers import NaiveBayesClassifier
classifier = NaiveBayesClassifier(train_set)

text = "It was a wonderful movie. I liked it very much."
print (classifier.classify(text))
```

Output

```
0.735
pos
```

Program Name: Testing the model
demo13.py

```
from nltk.corpus import twitter_samples

pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

from random import shuffle

shuffle(pos_tweets_set)
shuffle(neg_tweets_set)

test_set = pos_tweets_set[:300] + neg_tweets_set[:300]
train_set = pos_tweets_set[300:600] + neg_tweets_set[300:600]

from textblob.classifiers import NaiveBayesClassifier
classifier = NaiveBayesClassifier(train_set)

text = "I don't like movies having happy ending."
print (classifier.classify(text))
```

Output

```
0.726666666666666
neg
```

Program Name: Testing the model
demo14.py

```
from nltk.corpus import twitter_samples
from textblob import TextBlob

pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')

pos_tweets_set = []

for tweet in pos_tweets:
        pos_tweets_set.append((tweet, 'pos'))

neg_tweets_set = []

for tweet in neg_tweets:
        neg_tweets_set.append((tweet, 'neg'))

from random import shuffle

shuffle(pos_tweets_set)
shuffle(neg_tweets_set)

test_set = pos_tweets_set[:300] + neg_tweets_set[:300]
train_set = pos_tweets_set[300:600] + neg_tweets_set[300:600]

from textblob.classifiers import NaiveBayesClassifier
classifier = NaiveBayesClassifier(train_set)

text = "It was a wonderful movie. I liked it very much."
print(classifier.classify(text))

blob = TextBlob(text, classifier=classifier)

print(blob)
print(blob.classify())
```

Output

```
0.695
pos
It was a wonderful movie. I liked it very much.
pos
```