BY PRATEEK KUMAR

```
pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
```

```
from nltk.classify import NaiveBayesClassifier
from nltk.corpus import subjectivity
from nltk.sentiment import SentimentAnalyzer
from nltk.sentiment.util import *
```

```
import nltk
nltk.download('subjectivity')
n_instances = 1000 #no_of_senetnces to be used
subj_docs = [(sent, 'subj') for sent in subjectivity.sents(categories='subj')[:n_instances]] # 1000 subjective sentences from the subj
obj_docs = [(sent, 'obj') for sent in subjectivity.sents(categories='obj')[:n_instances]] # 1000 subjective sentences from the objecti
len(subj_docs), len(obj_docs)
#(1000, 1000)
```

```
[nltk_data] Downloading package subjectivity to /root/nltk_data...
[nltk_data]   Package subjectivity is already up-to-date!
(1000, 1000)
```

```
train_subj_docs = subj_docs[:900]
test_subj_docs = subj_docs[900:100]
train_obj_docs = obj_docs[:900]
test_obj_docs = obj_docs[900:1000]
training_docs = train_subj_docs+train_obj_docs
testing_docs = test_subj_docs+test_obj_docs
sentim_analyzer = SentimentAnalyzer()
all_words_neg = sentim_analyzer.all_words([mark_negation(doc) for doc in training_docs])
```

```
unigram_feats = sentim_analyzer.unigram_word_feats(all_words_neg, min_freq=4)
#len(unigram_feats)
sentim_analyzer.add_feat_extractor(extract_unigram_feats, unigrams=unigram_feats)
```

```
training_set = sentim_analyzer.apply_features(training_docs)
test_set = sentim_analyzer.apply_features(testing_docs)
```

```
trainer = NaiveBayesClassifier.train
classifier = sentim_analyzer.train(trainer, training_set)
##Training classifier
for key,value in sorted(sentim_analyzer.evaluate(test_set).items()):
  print('{0}: {1}'.format(key, value))
```

```
Training classifier
Evaluating NaiveBayesClassifier results...
Accuracy: 0.79
F-measure [obj]: 0.8826815642458102
Precision [obj]: 1.0
Recall [obj]: 0.79
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

sentences = ["PRATEEK is smart, handsome, and funny.","He is terrific person.", "Radha is kind and lovable person.", "You look very fu

```

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
from nltk import tokenize
lines_list = tokenize.sent_tokenize(str(sentences))
```

```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
True
```

```
for sentence in sentences:
  sid = SentimentIntensityAnalyzer()
  print('text: '+sentence)
  ss = sid.polarity_scores(sentence)
  for k in sorted(ss):
    print('{0}: {1}, '.format(k, ss[k]), end='')
  print()
```

```
text: PRATEEK is smart, handsome, and funny.
compound: 0.8316, neg: 0.0, neu: 0.254, pos: 0.746,
text: He is terrific person.
compound: 0.4767, neg: 0.0, neu: 0.492, pos: 0.508,
text: Radha is kind and lovable person.
compound: 0.8126, neg: 0.0, neu: 0.351, pos: 0.649,
text: You look very funny!.
compound: 0.0, neg: 0.0, neu: 1.0, pos: 0.0,
text: R u ok!!!
compound: 0.4724, neg: 0.0, neu: 0.0, pos: 1.0,
text: VADER is VERY SMART, really handsome, and INCREDIBLY FUNNY!!!
compound: 0.9469, neg: 0.0, neu: 0.294, pos: 0.706,
text: The book was good.
compound: 0.4404, neg: 0.0, neu: 0.508, pos: 0.492,
text: The book was kind of good.
compound: 0.3832, neg: 0.0, neu: 0.657, pos: 0.343,
text: The plot was good, but the characters are uncompelling and the dialog is not great.
compound: -0.7042, neg: 0.327, neu: 0.579, pos: 0.094,
text: A really bad, horrible book.
compound: -0.8211, neg: 0.791, neu: 0.209, pos: 0.0,
text: At least it isn't a horrible book.
compound: 0.431, neg: 0.0, neu: 0.637, pos: 0.363,
text: :) and :D
compound: 0.7925, neg: 0.0, neu: 0.124, pos: 0.876,
text:
compound: 0.0, neg: 0.0, neu: 0.0, pos: 0.0,
text: Today sux
compound: -0.3612, neg: 0.714, neu: 0.286, pos: 0.0,
text: Today sux!
compound: -0.4199, neg: 0.736, neu: 0.264, pos: 0.0,
text: Today SUX!
compound: -0.5461, neg: 0.779, neu: 0.221, pos: 0.0,
text: Today kinda sux! But I'll get by, lol
compound: 0.5249, neg: 0.138, neu: 0.517, pos: 0.344,
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Sample sentences for sentiment analysis
sentences = ["PRATEEK is smart, handsome, and funny.","He is terrific person.", "Radha is kind and lovable person.", "You look very fu


# Initialize the list to store sentiment results
sentiment_results = []

# Loop through sentences and perform sentiment analysis
for sentence in sentences:
    sid = SentimentIntensityAnalyzer()
    ss = sid.polarity_scores(sentence)

    # Store the sentiment analysis result in a dictionary
    result = {
        "text": sentence,
        "compound": ss["compound"]
    }

    # Add the result to the sentiment_results list
    sentiment_results.append(result)

# Display the sentiment_results list
print(sentiment_results)

```

```
[{'text': 'PRATEEK is smart, handsome, and funny.', 'compound': 0.8316}, {'text': 'He is terrific person.', 'compound': 0.4767}, {'
```

```
pip install twilio

```

```
Requirement already satisfied: twilio in /usr/local/lib/python3.10/dist-packages (8.7.0)
Requirement already satisfied: pytz in /usr/local/lib/python3.10/dist-packages (from twilio) (2023.3)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from twilio) (2.31.0)
```

```
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/lib/python3/dist-packages (from twilio) (2.3.0)
Requirement already satisfied: aiohttp>=3.8.4 in /usr/local/lib/python3.10/dist-packages (from twilio) (3.8.5)
Requirement already satisfied: aiohttp-retry>=2.8.3 in /usr/local/lib/python3.10/dist-packages (from twilio) (2.8.3)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio) (23.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp>=3.8.4->twilio) (1.3.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.0.0->twilio) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.0.0->twilio) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.0.0->twilio) (2023.7
```

```
from twilio.rest import Client

# Twilio credentials
twilio_account_sid = 'TWILIO SID'
twilio_auth_token = 'TWILIO AUTH TOKEN'
twilio_phone_number = '+13187502471'
recipient_phone_number = 'YOUR PHONE NUMBER'

# Create a function to send SMS
def send_sms(message):
    client = Client(twilio_account_sid, twilio_auth_token)
    client.messages.create(
        to=recipient_phone_number,
        from_=twilio_phone_number,
        body=message
    )

# Sentiment scores and corresponding text
sentiment_scores = [0.8316, 0.8439, 0.8545, 0.9227, 0.9342, 0.9469, 0.4404, 0.3832, -0.7042, -0.8211, 0.431, 0.7925, 0.0, -0.3612, -0.
sentiment_texts =  ["PRATEEK is smart, handsome, and funny.","He is terrific person.", "Radha is kind and lovable person.", "You look


# Loop through sentiment scores and texts
for score, text in zip(sentiment_scores, sentiment_texts):
    severity = None
    if score > 0.5:
        severity = "Positive"
    elif score < -0.5:
        severity = "Negative"

    if severity:
        message = f"Sentiment Alert: {severity} sentiment detected.\nText: {text}\nSentiment Score: {score:.4f}"
        send_sms(message)

```