Practice

October 23, 2021

```
[15]: import nltk
      import pandas as pd
      import numpy as np
```

```
[ ]: nltk.download()
```

0.1 1) The Data (Source:- Kaggle)

```
[39]: messages = pd.read_csv('Sentiment Analysis Dataset 2.csv',error_bad_lines=False)
      messages
```

```
b'Skipping line 8836: expected 4 fields, saw 5\n'
b'Skipping line 535882: expected 4 fields, saw 7\n'
```

```
[39]:           ItemID  Sentiment SentimentSource  \
      0              1          0    Sentiment140
      1              2          0    Sentiment140
      2              3          1    Sentiment140
      3              4          0    Sentiment140
      4              5          0    Sentiment140
      …            …        …              …
      1578607  1578623          1    Sentiment140
      1578608  1578624          1    Sentiment140
      1578609  1578625          0    Sentiment140
      1578610  1578626          0    Sentiment140
      1578611  1578627          0    Sentiment140

                                                   SentimentText
      0                              is so sad for my APL frie…
      1                            I missed the New Moon trail…
      2                                   omg its already 7:30 :O
      3                   .. Omgaga. Im sooo  im gunna CRy. I'…
      4                  i think mi bf is cheating on me!!!   …
      …                                                      …
      1578607                 Zzzzzz… Finally! Night tweeters!
      1578608                            Zzzzzzz, sleep well people
      1578609              ZzzZzZzzzZ… wait no I have homework.
      1578610        ZzZzzzZZZZzzz meh, what am I doing up again?
```

```
1578611                      Zzzzzzzzzzzzzzzzzzz, I wish

[1578612 rows x 4 columns]
```

0.2 2) Cleaning and Slicing of the Data

[40]:
```
messages.drop(['ItemID','SentimentSource'],axis=1,inplace=True)
```

[41]:
```
messages.head()
```

[41]:
```
   Sentiment                                      SentimentText
0          0                       is so sad for my APL frie…
1          0                     I missed the New Moon trail…
2          1                            omg its already 7:30 :O
3          0            .. Omgaga. Im sooo  im gunna CRy. I'…
4          0           i think mi bf is cheating on me!!!   …
```

[19]:
```
from nltk.corpus import stopwords
```

[20]:
```
stopwords.words('english')
```

[20]:
```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
 'did',
 'doing',
 'a',
 'an',
 'the',
 'and',
 'but',
 'if',
 'or',
 'because',
 'as',
 'until',
 'while',
 'of',
 'at',
 'by',
 'for',
 'with',
 'about',
 'against',
 'between',
 'into',
 'through',
 'during',
 'before',
 'after',
 'above',
 'below',
 'to',
 'from',
 'up',
 'down',
 'in',
 'out',
 'on',
 'off',
 'over',
 'under',
 'again',
 'further',
 'then',
 'once',
 'here',
 'there',
 'when',
 'where',
 'why',
 'how',
 'all',
 'any',
 'both',
 'each',
 'few',
 'more',
 'most',
 'other',
 'some',
 'such',
 'no',
 'nor',
 'not',
 'only',
 'own',
 'same',
 'so',
 'than',
 'too',
 'very',
 's',
 't',
 'can',
 'will',
 'just',
 'don',
 "don't",
 'should',
 "should've",
 'now',
 'd',
 'll',
 'm',
 'o',
 're',
 've',
 'y',
 'ain',
 'aren',
 "aren't",
 'couldn',
 "couldn't",
 'didn',
 "didn't",
 'doesn',
 "doesn't",
 'hadn',
 "hadn't",
 'hasn',
 "hasn't",
 'haven',
 "haven't",
 'isn',
 "isn't",
 'ma',
 'mightn',
 "mightn't",
 'mustn',
 "mustn't",
 'needn',
 "needn't",
  'shan',
  "shan't",
  'shouldn',
  "shouldn't",
  'wasn',
  "wasn't",
  'weren',
  "weren't",
  'won',
  "won't",
  'wouldn',
  "wouldn't"]
```

As the dataset is too big for us to train on our Local Machine we are Slicing the Data

[43]:
```
messages = messages.iloc[0:15000,:]
messages
```

[43]:
```
       Sentiment                                        SentimentText
0              0                              is so sad for my APL frie…
1              0                            I missed the New Moon trail…
2              1                                   omg its already 7:30 :O
3              0                 .. Omgaga. Im sooo  im gunna CRy. I'…
4              0                i think mi bf is cheating on me!!!   …
…            …                                                      …
14995          0   …well i can, but the other person needs to b…
14996          0   …well I was going to RPM. Vespa needs oil, I…
14997          1   …well it's bed time again …I will bid you …
14998          1   …went to Chinatown, ate lots of Chinese nood…
14999          1   …what can I say …what a surprise…http:/…

[15000 rows x 2 columns]
```

[ ]:
```
from nltk.stem import WordNetLemmatizer
lem = WordNetLemmatizer()
import re
```

[44]:
```
clean_data = []
for i in range(len(messages)):
    sent = re.sub('[^a-zA-Z]',' ',messages['SentimentText'][i])
    sent = sent.lower()
    sent = sent.split()
    sent = [lem.lemmatize(word) for word in sent if word not in stopwords.
 ↪words('english')]
    sent = ' '.join(sent)
    clean_data.append(sent)
    if i%100==0:
        print(i)
```

0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100
4200
4300
4400
4500
4600
4700

4800
4900
5000
5100
5200
5300
5400
5500
5600
5700
5800
5900
6000
6100
6200
6300
6400
6500
6600
6700
6800
6900
7000
7100
7200
7300
7400
7500
7600
7700
7800
7900
8000
8100
8200
8300
8400
8500
8600
8700
8800
8900
9000
9100
9200
9300
9400
9500

9600
9700
9800
9900
10000
10100
10200
10300
10400
10500
10600
10700
10800
10900
11000
11100
11200
11300
11400
11500
11600
11700
11800
11900
12000
12100
12200
12300
12400
12500
12600
12700
12800
12900
13000
13100
13200
13300
13400
13500
13600
13700
13800
13900
14000
14100
14200
14300

```
14400
14500
14600
14700
14800
14900
```

0.3 3) Converting words into Vectors

```
[58]: from sklearn.feature_extraction.text import TfidfVectorizer
      tf = TfidfVectorizer(max_features=5000)
```

```
[59]: X = tf.fit_transform(clean_data).toarray()
```

```
[60]: X.shape
```

```
[60]: (15000, 5000)
```

```
[61]: y = pd.get_dummies(messages['Sentiment'],drop_first=True)
```

```
[62]: from sklearn.model_selection import train_test_split
```

```
[63]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
       ↪random_state=42)
```

0.4 4) Training and testing on our Model

```
[64]: from sklearn.naive_bayes import MultinomialNB
```

```
[65]: log = MultinomialNB()
```

```
[66]: log.fit(X_train,y_train)
```

```
C:\Users\Sushant\anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(**kwargs)
```

```
[66]: MultinomialNB()
```

```
[67]: y_pred = log.predict(X_test)
```

```
[68]: from sklearn.metrics import confusion_matrix,classification_report
```

```
[69]: print(confusion_matrix(y_test,y_pred))
      print(classification_report(y_test,y_pred))
```

```
[[1406  253]
 [ 499  842]]
              precision    recall  f1-score   support

           0       0.74      0.85      0.79      1659
           1       0.77      0.63      0.69      1341

    accuracy                           0.75      3000
   macro avg       0.75      0.74      0.74      3000
weighted avg       0.75      0.75      0.75      3000
```