a4_q1

April 30, 2020

1 Hate Speech

1.0.1 Importing libraries

[26]:
```
import nltk
import pandas
import numpy
import string
import re
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn import metrics
```

1.0.2 Reading the train data set

[27]:
```
data_set = pandas.read_csv("hate_speech_train.csv")
text = data_set.iloc[:,0]
labels = data_set.iloc[:,1]
```

1.0.3 Reading the test data set

[28]:
```
test_ds = pandas.read_csv("hate_speech_test.csv")
test_text = test_ds.iloc[:,0]
```

1.0.4 Exploring the data set

[29]:
```
print("data set shape :",data_set.shape)
print("data set columns :",list(data_set.columns))

# Checking if data set has null values
# print(data_set.isnull().sum())

# filt_0 = (data_set['labels'] == 0)
# print(filt_0)
# print("no. of label 0 rows ",data_set.loc[filt_0].shape[0])

# filt_1 = (data_set['labels'] == 1)
# print(filt_1)
# print("no. of label 1 rows ",data_set.loc[filt_1].shape[0])

# Printing the data set
data_set.head()
```

```
data set shape : (5266, 2)
data set columns : ['text', 'labels']
```

[29]:

	text	labels
0	@realDonaldTrump This is one of the worst time…	0
1	How about the crowd in Oval in today's #AUSvIN…	1
2	@skroskz @shossy2 @JoeBiden Biden & his so…	0
3	#etsy shop: Benedict Donald so called presiden…	1
4	@realDonaldTrump Good build a wall around Arka…	0

[30]:
```
string.punctuation
```

[30]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

1.0.5 Removal of punctuations

[31]:
```
def remove_punct(txt):
    no_punct_txt = []
    for lv in txt:
        if(lv not in string.punctuation):
            no_punct_txt.append(lv)
    return "".join(no_punct_txt)

data_set['no_punct_txt'] = data_set['text'].apply(lambda x : remove_punct(x))
# print(data_set.shape)
# print(data_set.head())
test_ds['no_punct_txt'] = test_ds['text'].apply(lambda x : remove_punct(x))
```

1.0.6 Tokenization

[32]:
```
def tokenize(txt):
    tokens = re.split('\W+',txt)
    return tokens

data_set['tokenized_txt'] = data_set['no_punct_txt'].apply(lambda x: tokenize(x.
 ↪lower()))
# print(data_set.head())
test_ds['tokenized_txt'] = test_ds['no_punct_txt'].apply(lambda x: tokenize(x.
 ↪lower()))
```

[33]:
```
stop_words = nltk.corpus.stopwords.words('english')
print(stop_words[:179])
```

```
['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"]
```

1.0.7 Removal of stop words

[34]:
```
def remove_stpwrds(txt):
    no_stpwrds = []
    for lv in txt:
        if lv not in stop_words:
            no_stpwrds.append(lv)
    return no_stpwrds
```

```
data_set['no_stop_words'] = data_set['tokenized_txt'].apply(lambda x:␣
 ↪remove_stpwrds(x))
# print(data_set.head())
test_ds['no_stop_words'] = test_ds['tokenized_txt'].apply(lambda x:␣
 ↪remove_stpwrds(x))
```

1.0.8 Stemming

[35]:
```
ps = PorterStemmer()
def stemming(txt):
    stem_txt = []
    for lv in txt:
        stem_txt.append(ps.stem(lv))
    return stem_txt

data_set['stem_txt'] = data_set['no_stop_words'].apply(lambda x: stemming(x))
# print(data_set.head())
test_ds['stem_txt'] = test_ds['no_stop_words'].apply(lambda x: stemming(x))
```

[36]:
```
wn = nltk.WordNetLemmatizer()
```

1.0.9 Lemmatization

[37]:
```
def lemmatization(txt):
    lemmatized_txt = []
    for lv in txt:
         lemmatized_txt.append(wn.lemmatize(lv))
    return lemmatized_txt

data_set['lemmatized_txt'] = data_set['no_stop_words'].apply(lambda x:␣
 ↪lemmatization(x))
# print(data_set.head())
test_ds['lemmatized_txt'] = test_ds['no_stop_words'].apply(lambda x:␣
 ↪lemmatization(x))
```

1.0.10 Vectorization

[38]:
```
print(test_ds.head())
# tmp1 = data_set.iloc[:,0]
tmp2 = data_set.iloc[:,5]
tmp3 = test_ds.iloc[:,4]
print(tmp3.head())
value1=[' '.join([word for word in row]) for row in tmp2]
```

```
value2=[' '.join([word for word in row]) for row in tmp3]
vectorizer = TfidfVectorizer().fit(value1)
vectorized_ds = vectorizer.transform(value1)
# print(type(vectorized_ds))
# print(vectorized_ds)
vectorized_ts = vectorizer.transform(value2)
print(vectorized_ds.shape)
print(vectorized_ts.shape)
```

```
                                                text  \
0  #Assange is not a #rapist  https://t.co/M4sfW7…
1  #GandiNaaliAbuse | Where an MP says that he wi…
2  Candle light silent protest in MYSORE, by Myso…
3  #ShameOnICC  1. ICC on Dhoni's gloves         …
4  #ICC …look at pak team…wht is going on…

                                      no_punct_txt  \
0       Assange is not a rapist  httpstcoM4sfW7csXC
1  GandiNaaliAbuse  Where an MP says that he will…
2  Candle light silent protest in MYSORE by Mysor…
3  ShameOnICC  1 ICC on Dhonis gloves            …
4  ICC look at pak teamwht is going onnw this is …

                                     tokenized_txt  \
0  [assange, is, not, a, rapist, httpstcom4sfw7csxc]
1  [gandinaaliabuse, where, an, mp, says, that, h…
2  [candle, light, silent, protest, in, mysore, b…
3  [shameonicc, 1, icc, on, dhonis, gloves, vs, 2…
4  [icc, look, at, pak, teamwht, is, going, onnw,…

                                     no_stop_words  \
0             [assange, rapist, httpstcom4sfw7csxc]
1  [gandinaaliabuse, mp, says, cut, throat, musli…
2  [candle, light, silent, protest, mysore, mysor…
3  [shameonicc, 1, icc, dhonis, gloves, vs, 2icc,…
4  [icc, look, pak, teamwht, going, onnw, appropr…

                                          stem_txt  \
0              [assang, rapist, httpstcom4sfw7csxc]
1  [gandinaaliabus, mp, say, cut, throat, muslim,…
2  [candl, light, silent, protest, mysor, mysor, …
3  [shameonicc, 1, icc, dhoni, glove, vs, 2icc, p…
4  [icc, look, pak, teamwht, go, onnw, appropriat…

                                    lemmatized_txt
0             [assange, rapist, httpstcom4sfw7csxc]
1  [gandinaaliabuse, mp, say, cut, throat, muslim…
```

```
2  [candle, light, silent, protest, mysore, mysor…
3  [shameonicc, 1, icc, dhonis, glove, v, 2icc, p…
4  [icc, look, pak, teamwht, going, onnw, appropr…
0                  [assang, rapist, httpstcom4sfw7csxc]
1    [gandinaaliabus, mp, say, cut, throat, muslim,…
2    [candl, light, silent, protest, mysor, mysor, …
3    [shameonicc, 1, icc, dhoni, glove, vs, 2icc, p…
4    [icc, look, pak, teamwht, go, onnw, appropriat…
Name: stem_txt, dtype: object
(5266, 16253)
(586, 16253)
```

1.0.11 Splitting

[39]:
```
train_text, validate_text, train_labels, validate_labels =␣
 ↪train_test_split(vectorized_ds, labels, test_size=0.3, random_state=42)
# print(type(train_text))
# print(type(validate_text))
# print(type(train_labels))
# print(type(validate_labels))
```

1.0.12 Svm classifier

[40]:
```
svclassifier = SVC(kernel = 'linear' , C = 1.0)

svclassifier.fit(train_text, train_labels)

pred_labels = svclassifier.predict(validate_text)
# f1_score(validate_labels, pred_labels, average='macro')
f1_score(validate_labels, pred_labels, average='micro')
# f1_score(validate_labels, pred_labels, average='weighted')
# f1_score(validate_labels, pred_labels, average=None)
```

[40]:
```
0.660759493670886
```

[41]:
```
accuracy = metrics.accuracy_score(validate_labels, pred_labels)
print("accuracy",accuracy)
```

```
accuracy 0.660759493670886
```

1.0.13 Logistic regression on train data

[42]:
```
classifier = LogisticRegression(random_state = 0)

print(train_text.shape)
print(type(train_text))
print(train_labels.shape)
print(type(train_labels))
print(validate_labels.shape)
print(type(validate_text))
print(validate_text.shape)
print(type(validate_labels))

classifier.fit(train_text, train_labels)
pred_labels = classifier.predict(validate_text)
print(type(pred_labels))
print("f1 score:", f1_score(validate_labels, pred_labels))
print ("accuracy : ", accuracy_score(validate_labels, pred_labels))
```

```
(3686, 16253)
<class 'scipy.sparse.csr.csr_matrix'>
(3686,)
<class 'pandas.core.series.Series'>
(1580,)
<class 'scipy.sparse.csr.csr_matrix'>
(1580, 16253)
<class 'pandas.core.series.Series'>
<class 'numpy.ndarray'>
f1 score: 0.7658089838639337
accuracy :  0.660126582278481
```

1.0.14 Logistic regression on test data

[43]:
```
# print(vectorized_ds.shape)
# print(type(vectorized_ds))
# print(labels.shape)
# print(type(labels))
# print(vectorized_ts.shape)
# print(type(vectorized_ts))
classifier = LogisticRegression(random_state = 0)
classifier.fit(vectorized_ds, labels)
pred_labels = classifier.predict(vectorized_ts)
# print(type(pred_labels))
# print(pred_labels.shape)
# print(pred_labels)
# pred_labels = list(pred_labels)
```

```
# print(type(pred_labels))
```

[44]:
```
numpy.savetxt("submission.csv",pred_labels,header='labels',fmt='%d',comments='')
```