Applied Machine Learning Homework 5

UNI-rk3165

Name- Rakshith Kamath

Due 2 May,2022 (Monday) 11:59PM EST

Natural Language Processing

We will train a supervised training model to predict if a tweet has a positive or negative sentiment.

In [1]:

```
import re
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegressionCV
from sklearn.metrics import classification_report
```

Dataset loading & dev/test splits

1.1) Load the twitter dataset from NLTK library

In [2]:

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

```
[nltk_data] Downloading package twitter_samples to
[nltk_data]     /Users/rakshithkamath/nltk_data...
[nltk_data]   Package twitter_samples is already up-to-date!
```

In [3]:

```
import nltk
nltk.download("stopwords")
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/rakshithkamath/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/rakshithkamath/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

1.2) Load the positive & negative tweets

In [4]:

```
all_positive_tweets = twitter_samples.strings('positive_tweets.json')
all_negative_tweets = twitter_samples.strings('negative_tweets.json')
```

1.3) Create a development & test split (80/20 ratio):

In [5]:

```
#code here
pos_label = ['pos']*len(all_positive_tweets)
neg_label = ['neg']*len(all_negative_tweets)

tweets=all_positive_tweets+all_negative_tweets
labels=pos_label+neg_label

df=pd.DataFrame({'tweets':tweets,'sentiment':labels})
df.head(10)
```

Out[5]:

	tweets	sentiment
0	#FollowFriday @France_Inte @PKuchly57 @Milipol...	pos
1	@Lamb2ja Hey James! How odd :/ Please call our...	pos
2	@DespiteOfficial we had a listen last night :)...	pos
3	@97sides CONGRATS :)	pos
4	yeaaaah yippppy!!! my accnt verified rqst has...	pos
5	@BhaktisBanter @PallaviRuhail This one is irre...	pos
6	We don't like to keep our lovely customers wai...	pos
7	@Impatientraider On second thought, there's ju...	pos
8	Jgh , but we have to go to Bayan :D bye	pos
9	As an act of mischievousness, am calling the E...	pos

In [6]:

```
y = df.sentiment
df.drop(['sentiment'], axis=1, inplace=True)
df.head()
```

Out[6]:

	tweets
0	#FollowFriday @France_Inte @PKuchly57 @Milipol...
1	@Lamb2ja Hey James! How odd :/ Please call our...
2	@DespiteOfficial we had a listen last night :)...
3	@97sides CONGRATS :)
4	yeaaaah yippppy!!! my accnt verified rqst has...

In [7]:

```
X_dev,X_test, y_dev, y_test = train_test_split(df, y, test_size=.2, random_state
print(f"The amount of positive and negative sentiment tweets in dev")
print(y_dev.value_counts())
print(f"The amount of positive and negative sentiment tweets in test")
print(y_test.value_counts())
```

```
The amount of positive and negative sentiment tweets in dev
neg    4012
pos    3988
Name: sentiment, dtype: int64
The amount of positive and negative sentiment tweets in test
pos    1012
neg     988
Name: sentiment, dtype: int64
```

Data preprocessing

We will do some data preprocessing before we tokenize the data. We will remove `#` symbol, hyperlinks, stop words & punctuations from the data. You can use the `re` package in python to find and replace these strings.

1.4) Replace the `#` symbol with '' in every tweet

In [8]:

```
#code here
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:re.sub(r'#','',x)})
X_dev.head(10)
```

Out[8]:

	tweets
9254	:(((((matt
1561	@Lachdog_AU @Posica all good, thanks anyway :)
1670	my bf is mean :)
6087	zzzz missed my stop :(
6669	@bexmader that means 3am for me in Australia :(((
5933	@ButDinero your so fake I texted you :(
8829	This actually made me cry this is so disgustin...
7945	@lynfogeek "We're sorry, but Google Play Music...
3508	@Yorkshireccc @YCCCDizzy Have a good match 2ni...
2002	After Earth! :)) http://t.co/nrqNiBm7Ks

In [9]:

```
#code here
X_test[['tweets']]=X_test.apply({'tweets':lambda x:re.sub(r'#','',x)})
X_test.head(10)
```

Out[9]:

	tweets
6252	I love you, how but you? @Taecyeon2pm8 did you...
4684	@mayusushita @dildeewana_ @sonalp2591 @deepti_...
1731	Your love, O Lord, is better than life. :) &lt...
4742	@yasminyasir96 yeah but it will be better if w...
4521	Ok good night I wish troye wasn't ugly and I m...
6340	@scottybev I'm not surprised, that sounds hell...

	tweets
576	Dry, hot, scorching summer FF :) @infocffm @Me...
5202	@hanbined sad pray for me :(((
6363	Popol day too :(
439	My Song of the Week is Ducktails - Surreal Exp...

1.5) Replace hyperlinks with '' in every tweet

In [10]:

```
#code here
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:re.sub(r'@\w*','',x)})
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:re.sub(r'http\S+','',x)})
X_dev.head(10)
```

Out[10]:

	tweets
9254	:(((((matt
1561	all good, thanks anyway :)
1670	my bf is mean :)
6087	zzzz missed my stop :(
6669	that means 3am for me in Australia :(((
5933	your so fake I texted you :(
8829	This actually made me cry this is so disgustin...
7945	"We're sorry, but Google Play Music is curren...
3508	Have a good match 2nite boys - lets go out o...
2002	After Earth! :))

In [11]:

```
#code here
X_test[['tweets']]=X_test.apply({'tweets':lambda x:re.sub(r'@\w*','',x)})
X_test[['tweets']]=X_test.apply({'tweets':lambda x:re.sub(r'http\S+','',x)})
X_test.head(10)
```

Out[11]:

	tweets
6252	I love you, how but you? did you feel the sam...
4684	Thanks Guys :)
1731	Your love, O Lord, is better than life. :) &lt;3
4742	yeah but it will be better if we use her offi...
4521	Ok good night I wish troye wasn't ugly and I m...
6340	I'm not surprised, that sounds hellish! Why w...
576	Dry, hot, scorching summer FF :)
5202	sad pray for me :(((
6363	Popol day too :(

	tweets
439	My Song of the Week is Ducktails - Surreal Exp...

1.6) Remove all stop words

In [12]:

```
#code here
stop_words = stopwords.words('english')

def remove_stop_words(sent):
    token_words = word_tokenize(sent)
    stopwords_removed = [word for word in token_words if word not in stop_words]
    return ' '.join(stopwords_removed)
```

In [13]:

```
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:remove_stop_words(x)})
X_dev.head(10)
```

Out[13]:

	tweets
9254	: (((((matt
1561	good , thanks anyway :)
1670	bf mean :)
6087	zzzz missed stop : (
6669	means 3am Australia : (((
5933	fake I texted : (
8829	This actually made cry disgusting whAT THE ACT...
7945	We 're sorry , Google Play Music currently ...
3508	Have good match 2nite boys - lets go comp high...
2002	After Earth ! :))

In [14]:

```
X_test[['tweets']]=X_test.apply({'tweets':lambda x:remove_stop_words(x)})
X_test.head(10)
```

Out[14]:

	tweets
6252	I love , ? feel ? Emm I think : (
4684	Thanks Guys :)
1731	Your love , O Lord , better life . :) & lt ; 3
4742	yeah better use official Account :) Like The ...
4521	Ok good night I wish troye n't ugly I met toda...
6340	I 'm surprised , sounds hellish ! Why would th...
576	Dry , hot , scorching summer FF :)
5202	sad pray : (((
6363	Popol day : (

	tweets
439	My Song Week Ducktails - Surreal Exposure SOTW...

1.7) Remove all punctuations

In [15]:

```
#code here
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:re.sub(r'[^\w\s]', '',x)})
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:re.sub(r'_', '',x)})
X_dev.head(10)
```

Out[15]:

	tweets
9254	matt
1561	good thanks anyway
1670	bf mean
6087	zzzz missed stop
6669	means 3am Australia
5933	fake I texted
8829	This actually made cry disgusting whAT THE ACT...
7945	We re sorry Google Play Music currently expe...
3508	Have good match 2nite boys lets go comp high ...
2002	After Earth

In [16]:

```
#code here
X_test[['tweets']]=X_test.apply({'tweets':lambda x:re.sub(r'[^\w\s]', '',x)})
X_test[['tweets']]=X_test.apply({'tweets':lambda x:re.sub(r'_', '',x)})
X_test.head(10)
```

Out[16]:

	tweets
6252	I love feel Emm I think
4684	Thanks Guys
1731	Your love O Lord better life It 3
4742	yeah better use official Account Like The Ot...
4521	Ok good night I wish troye nt ugly I met today...
6340	I m surprised sounds hellish Why would thing
576	Dry hot scorching summer FF
5202	sad pray
6363	Popol day
439	My Song Week Ducktails Surreal Exposure SOTW ...

1.8) Apply stemming on the development & test datasets using Porter algorithm

In [17]:
```
#code here
porter = PorterStemmer()
def stem(sent):
    token_words = word_tokenize(sent)
    stem_sent = [porter.stem(word) for word in token_words]
    return ' '.join(stem_sent)
```

In [18]:
```
X_dev[['tweets']]=X_dev.apply({'tweets':lambda x:stem(x)})
X_dev.head(10)
```

Out[18]:

	tweets
9254	matt
1561	good thank anyway
1670	bf mean
6087	zzzz miss stop
6669	mean 3am australia
5933	fake i text
8829	thi actual made cri disgust what the actual fu...
7945	we re sorri googl play music current experienc...
3508	have good match 2nite boy let go comp high enj...
2002	after earth

In [19]:
```
X_test[['tweets']]=X_test.apply({'tweets':lambda x:stem(x)})
X_test.head(10)
```

Out[19]:

	tweets
6252	i love feel emm i think
4684	thank guy
1731	your love o lord better life lt 3
4742	yeah better use offici account like the other
4521	ok good night i wish troy nt ugli i met today ...
6340	i m surpris sound hellish whi would thing
576	dri hot scorch summer ff
5202	sad pray
6363	popol day
439	my song week ducktail surreal exposur sotw jin...

Model training

1.9) Create bag of words features for each tweet in the development dataset

In [20]:

```
#code here
bag_of_words = CountVectorizer()
X_dev_bag=bag_of_words.fit_transform(X_dev.tweets)
```

1.10) Train a supervised learning model of choice on the development dataset

In [21]:

```
#code here
lr_bag = LogisticRegressionCV(cv=10, max_iter=10000)
lr_bag.fit(X_dev_bag, y_dev)
```

Out[21]:

```
LogisticRegressionCV(cv=10, max_iter=10000)
```

1.11) Create TF-IDF features for each tweet in the development dataset

In [22]:

```
#code here
tf_idf = TfidfVectorizer()
X_dev_tf_idf=tf_idf.fit_transform(X_dev.tweets)
```

1.12) Train the same supervised learning algorithm on the development dataset with TF-IDF features

In [23]:

```
#code here
lr_tf_idf = LogisticRegressionCV(cv=10, max_iter=10000)
lr_tf_idf.fit(X_dev_tf_idf, y_dev)
```

Out[23]:

```
LogisticRegressionCV(cv=10, max_iter=10000)
```

1.13) Compare the performance of the two models on the test dataset

In [24]:

```
#code here
X_test_bag = bag_of_words.transform(X_test.tweets)
print(f"Performance of bag of words on test dataset-{lr_bag.score(X_test_bag, y_
print(classification_report(y_test,lr_bag.predict(X_test_bag)))
```

```
Performance of bag of words on test dataset-0.745
              precision    recall  f1-score   support

         neg       0.72      0.78      0.75       988
         pos       0.77      0.71      0.74      1012

    accuracy                           0.74      2000
   macro avg       0.75      0.75      0.74      2000
weighted avg       0.75      0.74      0.74      2000
```

In [25]:

```
X_test_tf_idf = tf_idf.transform(X_test.tweets)
print(f"Performance of tf idf on test dataset-{lr_tf_idf.score(X_test_tf_idf, y_
print(classification_report(y_test,lr_tf_idf.predict(X_test_tf_idf)))
```

```
Performance of tf idf on test dataset-0.76
              precision    recall  f1-score   support

         neg       0.74      0.78      0.76       988
         pos       0.78      0.74      0.76      1012
```

```
    accuracy                           0.76      2000
   macro avg       0.76      0.76      0.76      2000
weighted avg       0.76      0.76      0.76      2000
```

Answer- Bag of Words model constructs a vocabulary extracting the unique words from the documents and keeps the vector with the term frequency of the particular word in the corresponding document.In TF-IDF, apart from the term frequencies we also take inverse of number of documents that a particular term appears or the inverse of document frequency.

Hence,in this study, Term ordering is not considered and Rareness of a term is not considered in BOW hence TF-IDF is better approach.