Applied Machine Learning Homework 5: NLP

Due May 2,2023 (Tuesday) 11:59PM EST

Instructions

1) Please push the .ipynb and .pdf to Github Classroom prior to the deadline, .py file is optional (not needed).
2) Please include your Name and UNI below.

Name: Shruti Agarwal

UNI: sa4136

Natural Language Processing

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

Dataset loading & dev/test splits

1.1) Load the twitter dataset from NLTK library

```
In [1]: import nltk
nltk.download('twitter_samples')
from nltk.corpus import twitter_samples
nltk.download('punkt')
nltk.download('stopwords')

import warnings
warnings.filterwarnings("ignore")

from nltk.corpus import stopwords
stop = stopwords.words('english')
import pandas as pd
import string
import re
from sklearn.model_selection import train_test_split
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report, accuracy_score
import numpy as np
# Feel free to import any other packages you need
```

```
[nltk_data] Downloading package twitter_samples to
[nltk_data]     /Users/shrutiagarwal/nltk_data...
[nltk_data]   Unzipping corpora/twitter_samples.zip.
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/shrutiagarwal/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/shrutiagarwal/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

1.2) Load the positive & negative tweets

```
In [2]: all_positive_tweets = twitter_samples.strings('positive_tweets.json')
all_negative_tweets = twitter_samples.strings('negative_tweets.json')
```

1.3) Make a data frame that has all tweets and their corresponding labels

```
In [3]: # Your Code Here
all_tweets = all_negative_tweets + all_positive_tweets

# Create a list of labels
labels = ["negative"] * len(all_negative_tweets) + ["positive"] * len(all_pc

# Create a dataframe with two columns: "tweet" and "label"
df = pd.DataFrame({"Tweet": all_tweets, "Label": labels})

# Print the dataframe
df
```

Out[3]:

	Tweet	Label
0	hopeless for tmr :(	negative
1	Everything in the kids section of IKEA is so c...	negative
2	@Hegelbon That heart sliding into the waste ba...	negative
3	“@ketchBurning: I hate Japanese call him "bani...	negative
4	Dang starting next week I have "work" :(	negative
...	...	...
9995	@chriswiggin3 Chris, that's great to hear :) D...	positive
9996	@RachelLiskeard Thanks for the shout-out :) It...	positive
9997	@side556 Hey! :) Long time no talk...	positive
9998	@staybubbly69 as Matt would say. WELCOME TO AD...	positive
9999	@DanielOConnel18 you could say he will have eg...	positive

10000 rows × 2 columns

1.4) Look at the class distribution of the tweets

In [4]:
```
# Your Code Here

class_dist = df["Label"].value_counts()
class_dist
```

Out[4]:
```
negative    5000
positive    5000
Name: Label, dtype: int64
```

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

In [5]:
```
# Your Code Here

# split the dataframe into development and test sets
dev_set, test_set = train_test_split(df, test_size=0.2, random_state=42)

# print the sizes of the resulting sets
print("Development set size:", len(dev_set))
print("Test set size:", len(test_set))
```

```
Development set size: 8000
Test set size: 2000
```

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.

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

In [6]:
```python
# Your Code Here

# define a function to remove the '#' symbol from a string
def remove_hashtags(text):
    return re.sub(r'#', '', text)

# apply the function to every tweet in the dataframe
dev_set["Tweet"] = dev_set["Tweet"].apply(remove_hashtags)
test_set["Tweet"] = test_set["Tweet"].apply(remove_hashtags)

# print the resulting dataframe
dev_set
```

Out[6]:

	Tweet	Label
9254	Friday!:) http://t.co/HUoq4txhmb	positive
1561	sorry for always changing my layout :(	negative
1670	<3 <3 awsme song <3 :-* :-(:-(:'(h...	negative
6087	@bwoyblunder @rajudasonline Sorted :). Thanks....	positive
6669	@narrhallamarsch Good Flight! :)	positive
...	...	...
5734	@ChaSilveo follow @jnlazts & http://t.co/...	positive
5191	Hi BAM ! @BarsAndMelody \nCan you follow my be...	positive
5390	@hostclubhowell no prob!:)	positive
860	@dullandwicked @_GrahamPatrick @JohnBoyStyle H...	negative
7270	Unreal training boys!\nAwesome work Zaine, Zac...	positive

8000 rows × 2 columns

1.7) Replace hyperlinks with '' in every tweet

In [7]:
```python
# Your Code Here

def remove_hyperlinks(text):
    return re.sub(r'http\S+', '', text)

# apply the function to every tweet in the dataframe
dev_set["Tweet"] = dev_set["Tweet"].apply(remove_hyperlinks)
test_set["Tweet"] = test_set["Tweet"].apply(remove_hyperlinks)

# print the resulting dataframe
dev_set
```

Out[7]:

	Tweet	Label
9254	Friday!:)	positive
1561	sorry for always changing my layout :(	negative
1670	<3 <3 awsme song <3 :-* :-(:-(:'(	negative
6087	@bwoyblunder @rajudasonline Sorted :). Thanks....	positive
6669	@narrhallamarsch Good Flight! :)	positive
...	...	...
5734	@ChaSilveo follow @jnlazts & follow u ba...	positive
5191	Hi BAM ! @BarsAndMelody \nCan you follow my be...	positive
5390	@hostclubhowell no prob!:)	positive
860	@dullandwicked @_GrahamPatrick @JohnBoyStyle H...	negative
7270	Unreal training boys!\nAwesome work Zaine, Zac...	positive

8000 rows × 2 columns

1.8) Remove all stop words

In [8]:

```
# Your Code Here

# define a function to remove stop words from a string
def remove_stopwords(text):
    stop_words = set(stopwords.words('english'))
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_wor
    return ' '.join(filtered_words)

# apply the function to every tweet in the dataframe
dev_set["Tweet"] = dev_set["Tweet"].apply(remove_stopwords)
test_set["Tweet"] = test_set["Tweet"].apply(remove_stopwords)

# print the resulting dataframe
dev_set
```

Out[8]:

	Tweet	Label
9254	Friday!:)	positive
1561	sorry always changing layout :(	negative
1670	<3 <3 awsme song <3 :-* :-(:-(:'(	negative
6087	@bwoyblunder @rajudasonline Sorted :). Thanks....	positive
6669	@narrhallamarsch Good Flight! :)	positive
...	...	...
5734	@ChaSilveo follow @jnlazts & follow u back :)	positive
5191	Hi BAM ! @BarsAndMelody follow bestfriend @969...	positive
5390	@hostclubhowell prob!:)	positive
860	@dullandwicked @_GrahamPatrick @JohnBoyStyle n...	negative
7270	Unreal training boys! Awesome work Zaine, Zac ...	positive

8000 rows × 2 columns

1.9) Remove all punctuations

In [9]:

```
# Your Code Here

# define a function to remove punctuation from a string
def remove_punctuation(text):
    return re.sub(r'[^\w\s]', '', text)

# apply the function to every tweet in the dataframe
dev_set["Tweet"] = dev_set["Tweet"].apply(remove_punctuation)
test_set["Tweet"] = test_set["Tweet"].apply(remove_punctuation)

# print the resulting datafram
dev_set
```

Out[9]:

	Tweet	Label
9254	Friday	positive
1561	sorry always changing layout	negative
1670	lt3 lt3 awsme song lt3	negative
6087	bwoyblunder rajudasonline Sorted Thanks Daaru...	positive
6669	narrhallamarsch Good Flight	positive
...	...	...
5734	ChaSilveo follow jnlazts amp follow u back	positive
5191	Hi BAM BarsAndMelody follow bestfriend 969Hor...	positive
5390	hostclubhowell prob	positive
860	dullandwicked _GrahamPatrick JohnBoyStyle nobo...	negative
7270	Unreal training boys Awesome work Zaine Zac Is...	positive

8000 rows × 2 columns

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

In [10]:

```
# Your Code Here

porter = PorterStemmer()
dev_set['stemmed_tweet'] = dev_set['Tweet'].apply(lambda x: ' '.join([porter
test_set['stemmed_tweet'] = test_set['Tweet'].apply(lambda x: ' '.join([port

dev_set
```

Out[10]:

	Tweet	Label	stemmed_tweet
9254	Friday	positive	friday
1561	sorry always changing layout	negative	sorri alway chang layout
1670	lt3 lt3 awsme song lt3	negative	lt3 lt3 awsm song lt3
6087	bwoyblunder rajudasonline Sorted Thanks Daaru...	positive	bwoyblund rajudasonlin sort thank daaru parti ...
6669	narrhallamarsch Good Flight	positive	narrhallamarsch good flight
...	...	...	...
5734	ChaSilveo follow jnlazts amp follow u back	positive	chasilveo follow jnlazt amp follow u back
5191	Hi BAM BarsAndMelody follow bestfriend 969Hor...	positive	hi bam barsandmelodi follow bestfriend 969hora...
5390	hostclubhowell prob	positive	hostclubhowel prob
860	dullandwicked _GrahamPatrick JohnBoyStyle nobo...	negative	dullandwick _grahampatrick johnboystyl nobodi ...
7270	Unreal training boys Awesome work Zaine Zac Is...	positive	unreal train boy awesom work zain zac isaac oss

8000 rows × 3 columns

In [11]:

```
test_set
```

Out[11]:

	Tweet	Label	stemmed_tweet
6252	Malan_Sanjaya yes switched back lap optimized...	positive	malan_sanjaya ye switch back lap optim window ...
4684	MTAP tomorrow means sleep early tonight	negative	mtap tomorrow mean sleep earli tonight
1731	Gotham3 sad view	negative	gotham3 sad view
4742	Jessica calls quits power abs 515	negative	jessica call quit power ab 515
4521	like cant actually put pressure ankle hop arou...	negative	like cant actual put pressur ankl hop around h...
...	...	...	...
6412	Agree Phone WiFi LifeStyle QatarDay	positive	agre phone wifi lifestyl qatarday
8285	Rl191459Alex Hey thank following	positive	rl191459alex hey thank follow
7853	See yah Sunday carmenkvarnen	positive	see yah sunday carmenkvarnen
1095	didnt took photos	negative	didnt took photo
6929	LondonLycra see legs lycra p	positive	londonlycra see leg lycra p

2000 rows × 3 columns

Model training

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

In [28]:
```
# Your Code Here

vectorizer = CountVectorizer(stop_words='english', ngram_range=(1,2), max_fe
bow_features = vectorizer.fit_transform(dev_set['stemmed_tweet'])
bow_features = bow_features.toarray()

print(bow_features)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

1.12) Train a Logistic Regression model on the development dataset

In [29]:
```
# Your Code Here

lr_model_bow = LogisticRegression()
lr_model_bow.fit(bow_features, dev_set['Label'])

print("Accuracy on development set: ", lr_model_bow.score(bow_features, dev_
```

```
Accuracy on development set:  0.784625
```

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

In [30]:
```
# Your Code Here

tfidf_vectorizer = TfidfVectorizer(max_df=0.90, min_df=2, max_features=1000,
tfidf_features = tfidf_vectorizer.fit_transform(dev_set['stemmed_tweet'])

print(tfidf_features.toarray())
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

1.14) Train the Logistic Regression model on the development dataset with TF-IDF features

In [31]:
```python
# Your Code Here

lr_model_tfidf = LogisticRegression()
lr_model_tfidf.fit(tfidf_features, dev_set['Label'])

print("Accuracy on development set: ", lr_model_tfidf.score(tfidf_features,
```

```
Accuracy on development set:  0.786875
```

1.15) Compare the performance of the two models on the test dataset using a classification report and the scores obtained. Explain the difference in results obtained.

In [32]:
```python
# Your Code Here

bow_test_features = vectorizer.transform(test_set['stemmed_tweet'])
bow_test_features = bow_test_features.toarray()

y_pred_bow = lr_model_bow.predict(bow_test_features)

print("Accuracy on Test set: ", lr_model_bow.score(bow_test_features, test_s

print(classification_report(test_set["Label"], y_pred_bow))
```

```
Accuracy on Test set:  0.729
              precision    recall  f1-score   support

    negative       0.72      0.77      0.74      1012
    positive       0.75      0.69      0.71       988

    accuracy                           0.73      2000
   macro avg       0.73      0.73      0.73      2000
weighted avg       0.73      0.73      0.73      2000
```

In [33]:
```python
tfidf_test_features = tfidf_vectorizer.transform(test_set['stemmed_tweet'])
y_pred_tfidf = lr_model_tfidf.predict(tfidf_test_features)

print("Accuracy on Test set: ", lr_model_tfidf.score(tfidf_test_features, te

print(classification_report(test_set['Label'], y_pred_tfidf))
```

```
Accuracy on Test set:  0.738
              precision    recall  f1-score   support

    negative       0.72      0.78      0.75      1012
    positive       0.75      0.70      0.72       988

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

*Explanation here

The performance of the TF-IDF model is expected to be better than the BOW model because it can capture more meaningful features and reduce the impact of noise caused by common words.

The bag-of-words approach represents a text document as a bag of words, without considering their order or context. It counts the frequency of each word in the document and constructs a feature vector for each document based on the frequency of each word. It lacks the ability to capture the semantic relationship between words and treats all words equally.

On the other hand, the TF-IDF approach considers the importance of words in a document relative to their frequency in the entire corpus. It reduces the weight of common words and increases the weight of rare words that are more informative. Therefore, it can capture the semantic meaning of words and their importance in a document.

In [ ]:

```

```