Predictive Algorithm

Homework 09

Jenifer M. Nemoda

04/29/2021

```
# Set seed for reproducibility
import random; random.seed(53)

# Import all we need from sklearn
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn import metrics
```

```
import pandas as pd

# Load data
tweet_df = pd.read_csv('tweets_hw9_tidy.csv')

# Create target
y = tweet_df.author

# Split training and testing data
X_train, X_test, y_train, y_test = train_test_split(tweet_df['status'], y, test_size=0.33,
                random_state=53)
```

```
# Initialize count vectorizer
count_vectorizer = CountVectorizer(stop_words='english',
                                   min_df=0.05, max_df=0.9)

# Create count train and test variables
count_train = count_vectorizer.fit_transform(X_train)
count_test = count_vectorizer.transform(X_test)

# Initialize tfidf vectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english',
                                   min_df=0.05, max_df=0.9)

# Create tfidf train and test variables
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)
```

```
tfidf_nb = MultinomialNB()
tfidf_nb.fit(tfidf_train, y_train)
```

```
## MultinomialNB()
```

```
tfidf_nb_pred = tfidf_nb.predict(tfidf_test)
tfidf_nb_score = metrics.accuracy_score(y_test, tfidf_nb_pred)

count_nb = MultinomialNB()
count_nb.fit(count_train, y_train)
```

```
## MultinomialNB()
```

```
count_nb_pred = count_nb.predict(count_test)
count_nb_score = metrics.accuracy_score(y_test, count_nb_pred)

print('NaiveBayes Tfidf Score: ', tfidf_nb_score)
```

```
## NaiveBayes Tfidf Score:  0.6893939393939394
```

```
print('NaiveBayes Count Score: ', count_nb_score)
```

```
## NaiveBayes Count Score:  0.6893939393939394
```

```
from helper_functions import plot_confusion_matrix


tfidf_nb_cm = metrics.confusion_matrix(y_test, tfidf_nb_pred, labels=['BorisJohnson', 'AOC'])
count_nb_cm = metrics.confusion_matrix(y_test, count_nb_pred, labels=['BorisJohnson', 'AOC'])

plot_confusion_matrix(tfidf_nb_cm, classes=['BorisJohnson', 'AOC'], title="TF-IDF NB Confusion Matrix")
```

```
## Confusion matrix, without normalization
```

```
plot_confusion_matrix(count_nb_cm, classes=['BorisJohnson', 'AOC'], title="Count NB Confusion Matrix", 
```

```
## Confusion matrix, without normalization
```

```
import matplotlib.pyplot as plt
tfidf_svc = LinearSVC()
tfidf_svc.fit(tfidf_train, y_train)
```

```
## LinearSVC()
```

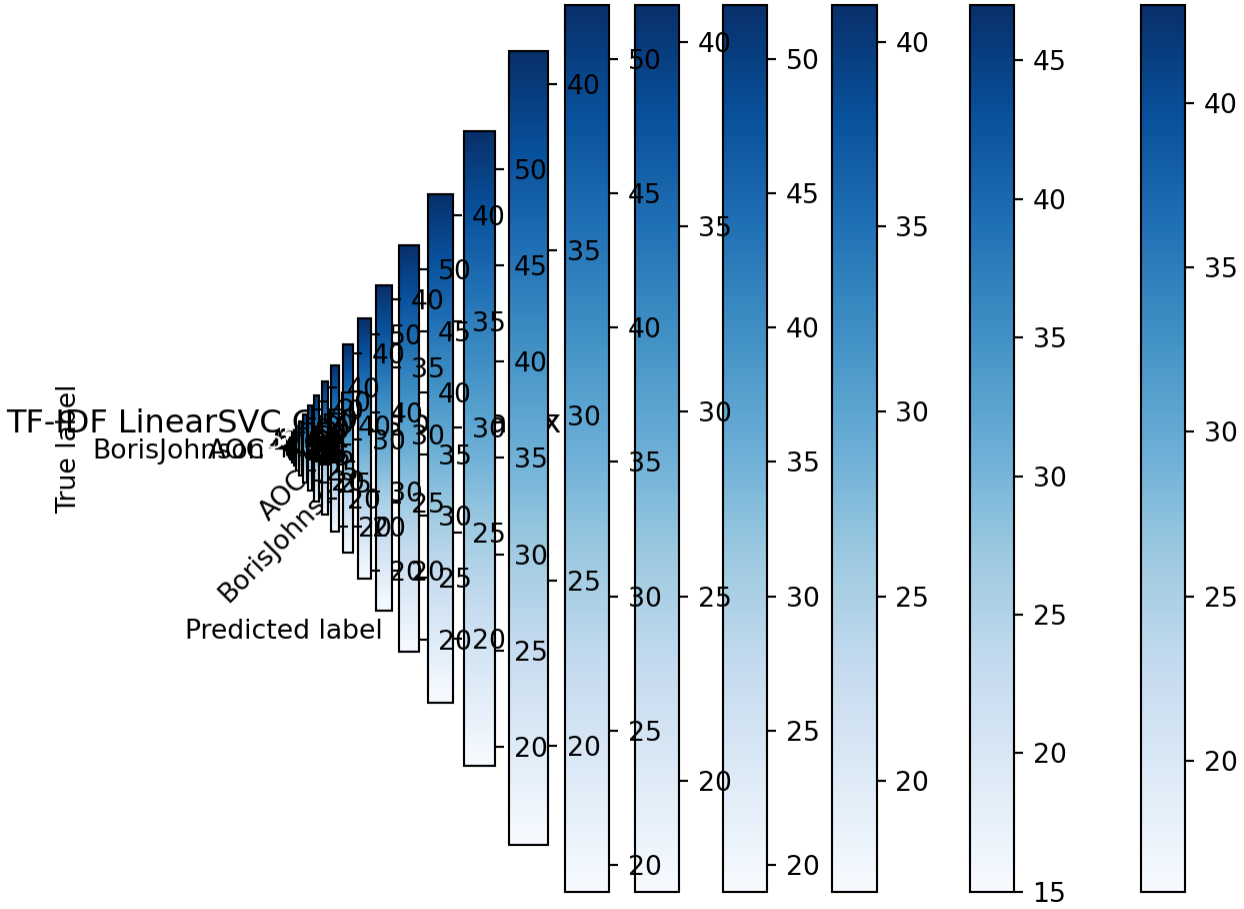
```
tfidf_svc_pred = tfidf_svc.predict(tfidf_test)
tfidf_svc_score = metrics.accuracy_score(y_test, tfidf_svc_pred)

print("LinearSVC Score:   %0.3f" % tfidf_svc_score)
```

```
## LinearSVC Score:   0.750
```

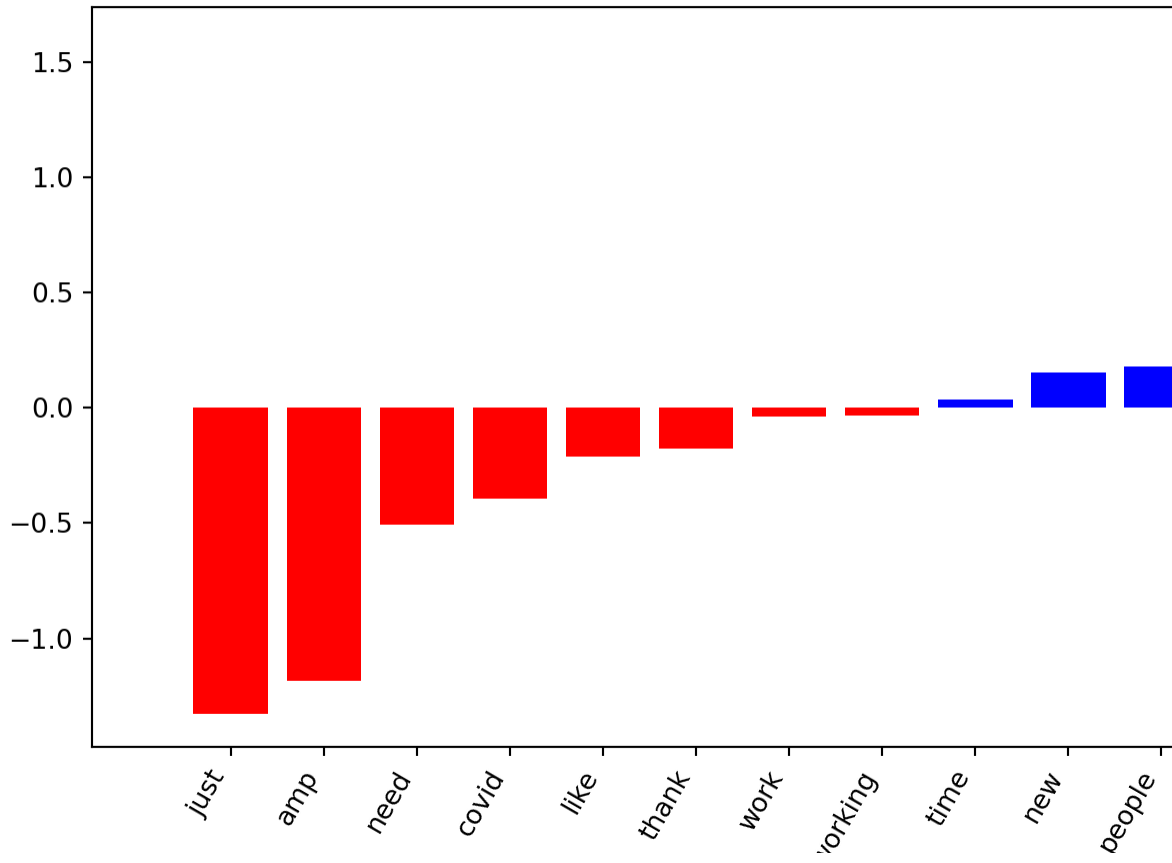
```
svc_cm = metrics.confusion_matrix(y_test, tfidf_svc_pred, labels=['BorisJohnson', 'AOC'])
cm = plot_confusion_matrix(svc_cm, classes=['BorisJohnson', 'AOC'], title="TF-IDF LinearSVC Confusion Ma
```

```
## Confusion matrix, without normalization
```

```
plt.show(cm)
```




```
from helper_functions import plot_and_return_top_features
from pprint import pprint

top_features = plot_and_return_top_features(tfidf_svc, tfidf_vectorizer)
```




```
pprint(top_features)
```

```
## [(-1.3261665904572686, 'just'),
##  (-1.1842285022870298, 'amp'),
##  (-0.5049679259148636, 'need'),
##  (-0.39570978148489616, 'covid'),
##  (-0.21036200521472087, 'like'),
##  (-0.17794809163610864, 'thank'),
##  (-0.03704269060841623, 'work'),
##  (-0.03465552083339783, 'working'),
##  (0.03345841750689321, 'time'),
##  (0.15221673839480623, 'new'),
##  (0.17718143413091803, 'people'),
##  (0.32526058661044527, 'make'),
##  (0.3283897601264688, 've'),
##  (0.3437571213159321, 'million'),
##  (0.43121245638869055, 'lives'),
##  (0.49826454205415016, 'https'),
##  (0.7844324489875879, 'today'),
##  (0.813220376945947, 'country'),
##  (1.0722274024195568, 'vaccine'),
##  (1.3255181932134072, 'jab'),
##  (1.4009955673876548, 'pandemic'),
```

```
##  (1.5898650185776968, 'uk')]
```

```
bj_tweet = "I would like to send my warm wishes to Her Majesty The Queen on her 95th birthday."
aoc_tweet = "That a family had to lose a son, brother and father; that a teenage girl had to film and p

bj_tweet_vectorized = tfidf_vectorizer.transform([bj_tweet])
aoc_tweet_vectorized = tfidf_vectorizer.transform([aoc_tweet])

bj_tweet_pred = tfidf_svc.predict(bj_tweet_vectorized)
aoc_tweet_pred = tfidf_svc.predict(aoc_tweet_vectorized)

print("Predicted Boris Johnson tweet", bj_tweet_pred)
```

```
## Predicted Boris Johnson tweet ['AOC']
```

```
print("Predicted AOC tweet", aoc_tweet_pred)


#Tweets for Manual Test

#This is the story of the UK vaccine rollout which has already saved thousands of lives.
#Tweeted by: BJ
#Predicted: BJ

#The world's tropical forests are the lungs of our planet and yet we're losing these great, teeming eco
#Tweeted by: BJ
#Predicted: BJ

#We can build back better from this pandemic by building back greener.
#Tweeted by: BJ
#Predicted: BJ

#In supporting #StephenLawrenceDay we are reminded of Stephen's many talents, and our duty to stop raci
#Tweeted by: BJ
#Predicted: AOC
#!! Error

#I would like to send my warm wishes to Her Majesty The Queen on her 95th birthday.
#Tweeted by: BJ
#Predicted: AOC
#!! Error

#Last January I was sworn in for my first term in Congress.
#Tweeted by: AOC
#Predicted: AOC

#Yes! And as we continue the fight for #MedicareForAll nationally, did you know that progressives estab
#Tweeted by: AOC
#Predicted: AOC

#I'm glad you like it @chelliepingree! Happy Earth Day Earth globe americas
#Tweeted by: AOC
#Predicted: AOC

#That a family had to lose a son, brother and father; that a teenage girl had to film and post a murder
```

```
#Tweeted by: AOC
#Predicted: AOC

#I welcome the Court of Appeal's decision to overturn the convictions of 39 former sub-postmasters in t
#Tweeted by: AOC
#Predicted: BJ
#!! Error
```

```
## Predicted AOC tweet ['AOC']
```

```
#Unrelated users test

test_tweet = "Buckingham Palace have released a new picture of Queen Elizabeth in celebration of her 95

test_tweet_vectorized = tfidf_vectorizer.transform([test_tweet])

test_tweet_pred = tfidf_svc.predict(test_tweet_vectorized)

print("Predicted test tweet", test_tweet_pred)

#Tweet for unrelated users manual test:

#Get vaccinated, folks.
#Tweeted by: President Biden
#Predicted: AOC

#Your answer to the dreaded low battery. #GalaxyS21 5G intuitively manages your usage so you never miss
#Tweeted by: Samsung Mobile US
#Predicted: AOC

#Am hosting SNL on May 8
#Tweeted by: Elon Musk
#Predicted: AOC

#It's the Championship - all or nothing. Choose your team before the games start.
#Tweeted by: Wendy's
#Predicted: AOC

#Buckingham Palace have released a new picture of Queen Elizabeth in celebration of her 95th birthday.
#Tweeted by: The Telegraph
#Predicted: AOC
```

```
## Predicted test tweet ['AOC']
```