mlfornlp

November 1, 2020

[85]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
import re,collections
from math import log
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪confusion_matrix, roc_auc_score, f1_score
from sklearn.cluster import KMeans
```

[40]:
```
df_real = pd.read_csv('True.csv')
df_real['RealNews?'] = True
df_fake = pd.read_csv('Fake.csv')
df_fake['RealNews?'] = False
df = df_real.append(df_fake)
```

[41]:
```
df.columns
```

[41]:
```
Index(['title', 'text', 'subject', 'date', 'RealNews?'], dtype='object')
```

[42]:
```
df['document']=df[['title','text']].agg(' '.join,axis=1).apply(lambda x:x.
↪lower())
df['text']=df['text'].apply(lambda x:x.lower())
```

1 Question 1

[43]:
```
# CountVectorizer produces a feature matrix of token counts for text.
tf_vectorizer = CountVectorizer(stop_words='english')
x = tf_vectorizer.fit_transform(df['text'])
#fit the lda model with 10 topics
lda = LatentDirichletAllocation(n_components=10,random_state=0)
lda.fit(x)
```

[43]: LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
                                  evaluate_every=-1, learning_decay=0.7,
                                  learning_method='batch', learning_offset=10.0,
                                  max_doc_update_iter=100, max_iter=10,
                                  mean_change_tol=0.001, n_components=10, n_jobs=None,
                                  perp_tol=0.1, random_state=0, topic_word_prior=None,
                                  total_samples=1000000.0, verbose=0)

[44]:
```
#print the top_n words for each topic
def print_top_words(model, feature_names, n_top_words):
    for topic_idx, topic in enumerate(model.components_):
        message = "Topic #%d: " % topic_idx
        message += " ".join([feature_names[i]
                             for i in topic.argsort()[:-n_top_words - 1:-1]])
        print(message)
```

[45]:
```
#print the topics of lda model
print("\nTopics in LDA model:")
n_top_words = 20
tf_feature_names = tf_vectorizer.get_feature_names()
print_top_words(lda, tf_feature_names, n_top_words)
```

```
Topics in LDA model:
Topic #0: trump said republican house president senate republicans tax
washington reuters white obama congress senator new democrats donald democratic
year plan
Topic #1: police said year people city old told killed officers man family years
arrested according death attack shooting group black officer
Topic #2: said state million 000 election government year money billion company
new according percent companies reuters federal city public states vote
Topic #3: court said law trump president states obama federal order supreme
immigration justice judge administration department united case legal ban
climate
Topic #4: trump said russia russian president fbi clinton intelligence
investigation campaign house news election information committee comey security
washington director department
Topic #5: clinton party hillary election percent said democratic voters sanders
new campaign vote poll political presidential candidate support democrats polls
year
Topic #6: trump people donald president just like twitter said obama white don
know image time going news featured right com think
Topic #7: people women gun america like state american children new students law
muslim school rights right group world just religious public
Topic #8: said government minister eu european reuters president israel britain
united prime mexico trade states turkey union talks border parliament deal
Topic #9: said united north military china state reuters korea states iran
```

```
president nuclear syria security trump government foreign war al told
```

[46]:
```
print(lda.components_.shape)
len(tf_feature_names)
```

```
(10, 121690)
```

[46]:
```
121690
```

1.1 Most of the topics are related to politics. Topics such as fbi investigation, law on climate change, tax plan are found by the lda model. Its a good representation of real-world topics

2 Question 2

[47]:
```
#randomly sample 5 real news and 5 fake news
df_sampled = df[df['RealNews?']==True].sample(n=5).append(df[df['RealNews?
↪']==False].sample(n=5))
```

[48]:
```
df_sampled
```

[48]:
```
                                                    title  \
20627  Trump visit to Britain still unfixed nine mont…
8423   Clinton leads Trump by eight points: Reuters/I…
18904  Iraq sends delegation to Iran 'to coordinate m…
19052  China busts underground bank in Guangzhou: Chi…
15771  Myanmar sees 'bad consequences' if U.S. impose…
15507  https://100percentfedup.com/video-hillary-aske…
22173  BIGGER THAN SNOWDEN: Wikileaks 'Vault 7' Class…
19886  WHOA! DEMOCRATIC Strategist Gives Crooked Hill…
5524    Sean Hannity Just Made A Bizarre Implication …
20061  WHY TRUMP SUPPORTERS ARE LAUGHING After WikiLe…

                                                     text       subject  \
20627  london (reuters) - nine months after prime min…     worldnews
8423   new york (reuters) - democratic presidential c…  politicsNews
18904  baghdad (reuters) - a top ranking delegation f…     worldnews
19052  shanghai (reuters) - chinese police have broke…     worldnews
15771  yangon (reuters) - proposed u.s. sanctions tar…     worldnews
15507  https://100percentfedup.com/video-hillary-aske…      politics
22173   he who controls the spice controls the univer…       US_News
19886  the most unpopular, deplorable woman in americ…     left-news
5524   sean hannity made a bizarre implication during…          News
20061  fans of #crookedhillary are not going to like …     left-news
```

```
                                                       date  RealNews?  \
20627                              September 8, 2017        True
8423                                 August 19, 2016        True
18904                             September 27, 2017        True
19052                             September 26, 2017        True
15771                               November 3, 2017        True
15507  https://100percentfedup.com/video-hillary-aske…       False
22173                                   March 13, 2017       False
19886                                      Oct 2, 2016       False
5524                                      July 10, 2016      False
20061                                      Aug 28, 2016      False

                                                   document
20627  trump visit to britain still unfixed nine mont…
8423   clinton leads trump by eight points: reuters/i…
18904  iraq sends delegation to iran 'to coordinate m…
19052  china busts underground bank in guangzhou: chi…
15771  myanmar sees 'bad consequences' if u.s. impose…
15507  https://100percentfedup.com/video-hillary-aske…
22173  bigger than snowden: wikileaks ‘vault 7’ class…
19886  whoa! democratic strategist gives crooked hill…
5524    sean hannity just made a bizarre implication …
20061  why trump supporters are laughing after wikile…
```

2.1 Predict on test documents

[56]:
```python
#get topics for each document
x_sampled = tf_vectorizer.transform(df_sampled['text'])
topics_prob = lda.transform(x_sampled)
#print(topics_prob.shape)
docs_topics = np.argmax(topics_prob,axis=1)
for i in range(len(docs_topics)):
    print("document {} belongs to topic {}".format(i+1, docs_topics[i]))
```

```
document 1 belongs to topic 8
document 2 belongs to topic 5
document 3 belongs to topic 9
document 4 belongs to topic 9
document 5 belongs to topic 9
document 6 belongs to topic 6
document 7 belongs to topic 4
document 8 belongs to topic 5
document 9 belongs to topic 6
document 10 belongs to topic 4
```

2.2 topic 9(US and North Korea nuclear war)is prevalent in real news articles. topics 4 and 6 are prevalent in fake news articles.

3 Question 3

[58]:
```
df_train, df_test = train_test_split(df, test_size=0.2, shuffle=True)
print(df_train.shape,df_test.shape)
```

```
(35918, 6) (8980, 6)
```

[68]:
```
# CountVectorizer produces a feature matrix of token counts for text.
tf_vectorizer = CountVectorizer(stop_words='english')
x_train = tf_vectorizer.fit_transform(df_train['text'])
#fit the lda model with 10 topics
lda = LatentDirichletAllocation(n_components=10,random_state=0)
lda.fit(x_train)
```

[68]:
```
LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
                          evaluate_every=-1, learning_decay=0.7,
                          learning_method='batch', learning_offset=10.0,
                          max_doc_update_iter=100, max_iter=10,
                          mean_change_tol=0.001, n_components=10, n_jobs=None,
                          perp_tol=0.1, random_state=0, topic_word_prior=None,
                          total_samples=1000000.0, verbose=0)
```

[69]:
```
#get lda vectors for train and test docs
x_train = lda.transform(x_train)
x_test = tf_vectorizer.transform(df_test['text'])
x_test = lda.transform(x_test)
y_train,y_test= df_train['RealNews?'],df_test['RealNews?']
print("x_train shape:",x_train.shape)
print("y_train shape:",y_train.shape)
print("x_test shape:",x_test.shape)
print("y_test shape:",y_test.shape)
```

```
x_train shape: (35918, 10)
y_train shape: (35918,)
x_test shape: (8980, 10)
y_test shape: (8980,)
```

[70]:
```
#train the logistic regression clf
lr = LogisticRegression(random_state=0)
lr.fit(x_train,y_train)

#prediction
y_pred = lr.predict(x_test)
y_prob = lr.predict_proba(x_test)[:,1]
```

```
#model evaluation
print("Logistic regression model performance on lda vectors:")
print("Accuracy score is {}".format(accuracy_score(y_test,y_pred)))
print("Precision score is {}".format(precision_score(y_test, y_pred)))
print("Recall score is {}".format(recall_score(y_test, y_pred)))
print("F1 score is {}".format(f1_score(y_test, y_pred)))

print("Area Under the Curve(ROC curve) is {}".
 ↪format(roc_auc_score(y_test,y_prob)))
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()

print("true negative", tn)
print("false positive", fp)
print("false negative", fn)
print("true positive", tp)
specificity = tn / (tn+fp)
print("specificity is {}".format(specificity))
```

```
/home/vijay/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver
will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

```
Logistic regression model performance on lda vectors:
Accuracy score is 0.884075723830735
Precision score is 0.8751450452541193
Recall score is 0.8823116518483856
F1 score is 0.8787137364557847
Area Under the Curve(ROC curve) is 0.9532763508825242
true negative 4168
false positive 538
false negative 503
true positive 3771
specificity is 0.8856778580535487
```

[81]:
```
#get the most useful topics for classification
topics_importance_score = lr.coef_
useful_topics = np.flip(np.argsort(topics_importance_score))
print("useful topics ranked from most useful to least useful", useful_topics)
print("scores for corresponding topics",topics_importance_score)
```

```
useful topics ranked from most useful to least useful [[2 3 9 1 4 0 8 6 7 5]]
scores for corresponding topics [[ 1.38789583  2.53022141  5.22121393
4.06925796  1.45932434 -7.8030476
  -2.16746468 -6.43829936 -1.63321257  3.21623314]]
```

3.1 The top 3 most useful topics in classification of real or fake news is topic 2, topic 3 and topic 9.

4 Question 4

[82]:
```
#take only real news
df_real = df[df['RealNews?']==True]
```

[84]:
```
#get the lda vectors for real news docs
# CountVectorizer produces a feature matrix of token counts for text.
tf_vectorizer = CountVectorizer(stop_words='english')
x = tf_vectorizer.fit_transform(df_real['text'])
#fit the lda model with 10 topics
lda = LatentDirichletAllocation(n_components=10,random_state=0)
lda.fit(x)
x = lda.transform(x)
print("shape of real news lda vectors is", x.shape)
```

```
shape of real news lda vectors is (21417, 10)
```

[101]:
```
#cluster the docs using k-means
kmeans = KMeans(n_clusters=10, random_state=0)
kmeans.fit(x)
```

[101]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=10, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=0, tol=0.0001, verbose=0)
```

[103]:
```
#get 5 documents from each cluster
y_pred = kmeans.labels_
n_clusters=10
#stores docs index for each cluster
docs_index_by_clusters=[]
for i in range(n_clusters):
    #get all docs index for each cluster
    docs_index = np.where(y_pred==i)[0]
    #consider only 5 docs for each cluster
    docs_index = docs_index[:5]
    docs_index_by_clusters.append(docs_index)
print(docs_index_by_clusters)
```

```
[array([0, 4, 5, 7, 8]), array([ 919, 1227, 1345, 1349, 1351]), array([ 15,  34,
66, 175, 178]), array([ 2,  3,  6,  9, 10]), array([43, 47, 61, 64, 65]),
array([348, 363, 504, 571, 765]), array([ 33, 101, 105, 121, 128]), array([ 1,
18, 20, 21, 22]), array([ 24, 452, 456, 461, 485]), array([447, 448, 549, 552,
572])]
```

[106]:
```python
#print the titles of documents belonging to each cluster to find similarity
for i in range(len(docs_index_by_clusters)):
    print("document titles belonging to cluster",i)
    print("----------------------------------------")
    docs_indexes = docs_index_by_clusters[i]
    for j in docs_indexes:
        print(df_real.iloc[j]['title'])
    print("-------------------------------------")
```

```
document titles belonging to cluster 0
----------------------------------------
As U.S. budget fight looms, Republicans flip their fiscal script
Trump wants Postal Service to charge 'much more' for Amazon shipments
White House, Congress prepare for talks on spending, immigration
Factbox: Trump on Twitter (Dec 29) - Approval rating, Amazon
Trump on Twitter (Dec 28) - Global Warming
-------------------------------------
document titles belonging to cluster 1
----------------------------------------
North Korean defector pushes diplomatic solution in U.S. Congress
North Korea not ready to meet with South Korea in Russia: agencies
Turkey urges U.S. to review visa suspension as lira, stocks tumble
Turkey's Erdogan says U.S. decision to suspend visa services 'upsetting'
Turkey summons U.S. consulate worker for questioning: Anadolu
-------------------------------------
document titles belonging to cluster 2
----------------------------------------
Virginia officials postpone lottery drawing to decide tied statehouse election
As Republicans aim to ride economy to election victory, a warning from voters in
key district
In Georgia, battle of the 'Staceys' tests Democrats' future
After Alabama upset, Democrats see new prospects in U.S. South
Control of Virginia state House at stake as recounts begin
-------------------------------------
document titles belonging to cluster 3
----------------------------------------
Senior U.S. Republican senator: 'Let Mr. Mueller do his job'
FBI Russia probe helped by Australian diplomat tip-off: NYT
Trump says Russia probe will be fair, but timeline unclear: NYT
Alabama official to certify Senator-elect Jones today despite challenge: CNN
Jones certified U.S. Senate winner despite Moore challenge
-------------------------------------
document titles belonging to cluster 4
----------------------------------------
U.S. House approves $81 billion for disaster aid
U.S. launches effort to reduce reliance on imports or critical minerals
```

White House says tax bill will not hurt Puerto Rico
Fight over Alaska Arctic drilling has just begun, opponents vow
Senator Cornyn trying to get Big Corn behind U.S. biofuels reform
----
document titles belonging to cluster 5

U.S. defense chief urges Pakistan to redouble efforts against militants
U.S. embassy to Russia to resume some visa services after diplomatic row
Russian envoy to U.S. to inspect San Francisco consulate: RIA
Putin, Trump to discuss North Korea on Tuesday: IFX cites Kremlin aide
White House condemns missile attacks on Saudi by Yemen's Houthis
----
document titles belonging to cluster 6

Callista Gingrich becomes Trump's envoy to pope as differences mount
Trump strategy document says Russia meddles in domestic affairs worldwide
Trump: U.S. has 'no choice' but to deal with North Korea arms challenge
Trump to say in security speech that China is competitor: officials
Trump officials brief Hill staff on Saudi reactors, enrichment a worry
----
document titles belonging to cluster 7

U.S. military to accept transgender recruits on Monday: Pentagon
U.S. appeals court rejects challenge to Trump voter fraud panel
Federal judge partially lifts Trump's latest refugee restrictions
Exclusive: U.S. memo weakens guidelines for protecting immigrant children in court
Trump travel ban should not apply to people with strong U.S. ties: court
----
document titles belonging to cluster 8

Failed vote to oust president shakes up Peru's politics
Britain's U.S. ambassador discussed Trump retweets with senior White House staff: source
British minister hopes condemnation of Trump tweet has impact
Trump fires back at Britain's May: 'Don't focus on me'
Factbox: Who are Britain First, whose leader's posts Trump re-tweeted?
----
document titles belonging to cluster 9

Trump angers UK with truculent tweet to May after sharing far-right videos
U.N. rights boss condemns "spreading hatred through tweets"
U.S. calls Myanmar moves against Rohingya 'ethnic cleansing'
U.S. hopes to pressure Myanmar to permit Rohingya repatriation
U.S. Congress members decry 'ethnic cleansing' in Myanmar; Suu Kyi doubts allegations
----

4.1 There is some similarity between documents in each cluster. For example, docs in cluster 9 are about US opposition on Myanmar's ethnic cleansing, cluster 2 is about elections

[ ]:

```

```