

## fake-news-detection

June 27, 2024

```
[8]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split as ttp
from sklearn.metrics import classification_report
import re
import string
import matplotlib.pyplot as plt
```

```
[9]: data_true=pd.read_csv("/content/drive/MyDrive/True.csv")
data_fake=pd.read_csv("/content/drive/MyDrive/Fake.csv")
```

```
[10]: #from google colab import drive
#drive.mount('/content/drive')
```

```
[11]: data_true.shape,data_fake.shape
```

```
[11]: ((21417, 4), (23481, 4))
```

```
[12]: data_true["class"]=1
data_fake["class"]=0
```

Taking last 10 values for manual testing

```
[13]: data_true_manual_testing=data_true.tail(10)
for i in range(21416,21406,-1):
    data_true.drop([i],axis=0,inplace=True)
data_fake_manual_testing=data_fake.tail(10)
for i in range(21416,21406,-1):
    data_fake.drop([i],axis=0,inplace=True)
```

```
[14]: data_manual_testing=pd.
concat([data_fake_manual_testing,data_true_manual_testing],axis=0)
data_manual_testing.to_csv("manual_testing.csv")
```

```
[15]: data_merge=pd.concat([data_true,data_fake],axis=0)
data_merge.head(10)#displaying first 10 values after above operation
```

```
[15]:
```

	title \
0	As U.S. budget fight looms, Republicans flip t...
1	U.S. military to accept transgender recruits o...
2	Senior U.S. Republican senator: 'Let Mr. Muell...
3	FBI Russia probe helped by Australian diplomat...
4	Trump wants Postal Service to charge 'much mor...
5	White House, Congress prepare for talks on spe...
6	Trump says Russia probe will be fair, but time...
7	Factbox: Trump on Twitter (Dec 29) – Approval ...
8	Trump on Twitter (Dec 28) – Global Warming
9	Alabama official to certify Senator-elect Jone...

	text	subject \
0	WASHINGTON (Reuters) – The head of a conservat...	politicsNews
1	WASHINGTON (Reuters) – Transgender people will...	politicsNews
2	WASHINGTON (Reuters) – The special counsel inv...	politicsNews
3	WASHINGTON (Reuters) – Trump campaign adviser ...	politicsNews
4	SEATTLE/WASHINGTON (Reuters) – President Donal...	politicsNews
5	WEST PALM BEACH, Fla./WASHINGTON (Reuters) – T...	politicsNews
6	WEST PALM BEACH, Fla (Reuters) – President Don...	politicsNews
7	The following statements were posted to the ve...	politicsNews
8	The following statements were posted to the ve...	politicsNews
9	WASHINGTON (Reuters) – Alabama Secretary of St...	politicsNews

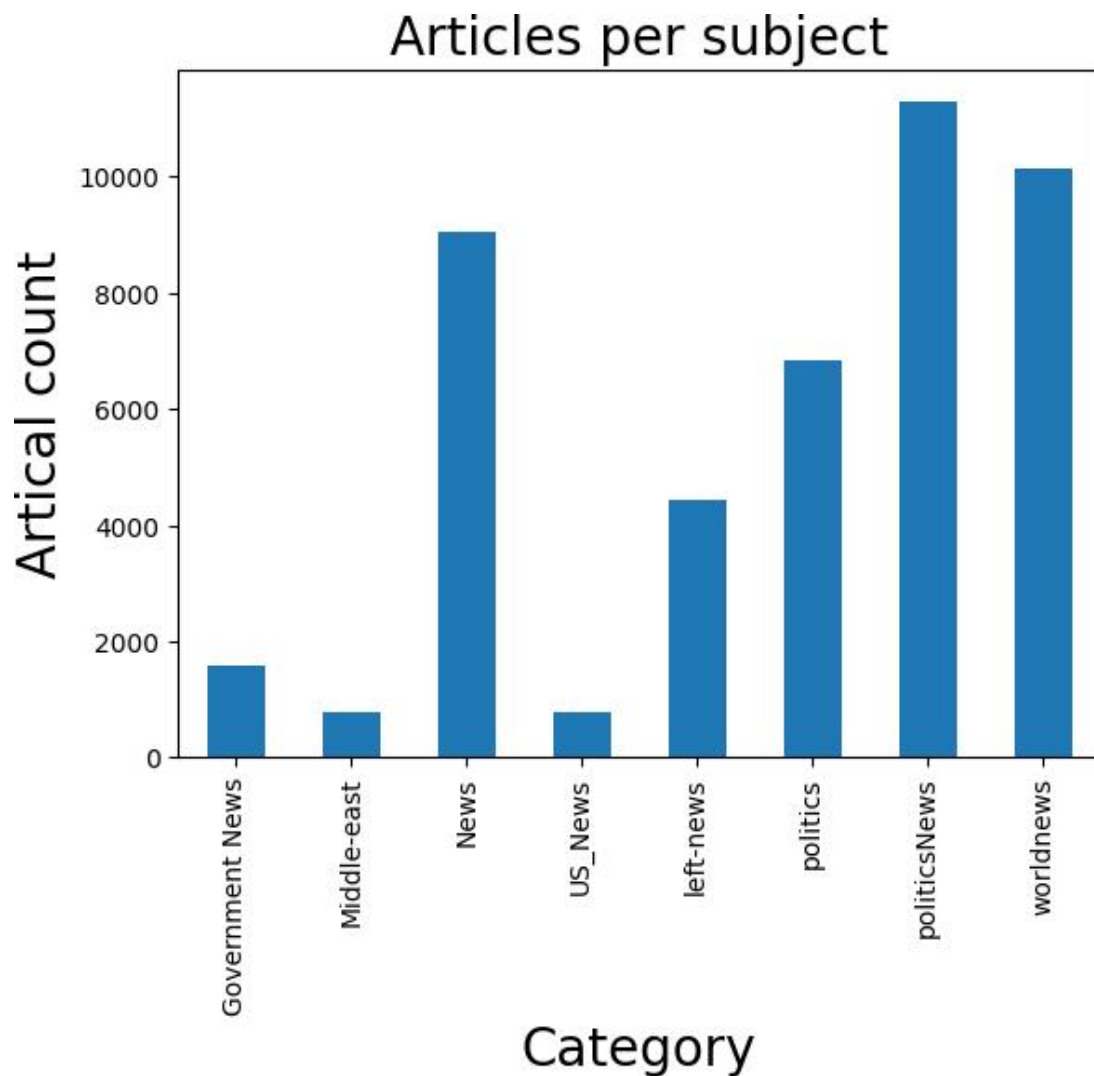
	date	class
0	December 31, 2017	1
1	December 29, 2017	1
2	December 31, 2017	1
3	December 30, 2017	1
4	December 29, 2017	1
5	December 29, 2017	1
6	December 29, 2017	1
7	December 29, 2017	1
8	December 29, 2017	1
9	December 28, 2017	1

```
[16]: print(data_merge.groupby(['subject'])['text'].count())
data_merge.groupby(['subject'])['text'].count().plot(kind="bar")
plt.title("Articles per subject",size=20)
plt.xlabel("Category",size=20)
plt.ylabel("Artical count",size=20)
plt.show()
```

subject	
Government News	1570
Middle-east	778
News	9050

US_News	783
left-news	4449
politics	6841
politicsNews	11272
worldnews	10135

Name: text, dtype: int64



```
[17]: print(data_merge.groupby(['class'])['text'].count())
print("0 = Fake News\n1 = True News")
data_merge.groupby(['class'])['text'].count().plot(kind="pie")
plt.title("Fake News vs True News",size=20)
plt.show()
```

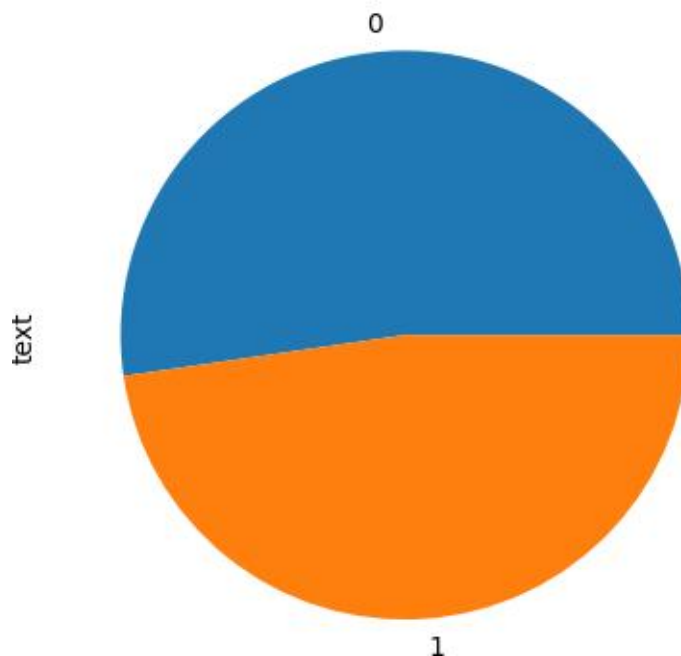
class

```

0    23471
1    21407
Name: text, dtype: int64
0 = Fake News
1 = True News

```

## Fake News vs True News



```

[18]: data = data_merge.drop(["title", "subject", "date"], axis=1)
      data.head(10)

```

```

[18]:
      text  class
0  WASHINGTON (Reuters) – The head of a conservat...    1
1  WASHINGTON (Reuters) – Transgender people will...    1
2  WASHINGTON (Reuters) – The special counsel inv...    1
3  WASHINGTON (Reuters) – Trump campaign adviser ...    1
4  SEATTLE/WASHINGTON (Reuters) – President Donal...    1
5  WEST PALM BEACH, Fla./WASHINGTON (Reuters) – T...    1
6  WEST PALM BEACH, Fla (Reuters) – President Don...    1
7  The following statements were posted to the ve...    1
8  The following statements were posted to the ve...    1
9  WASHINGTON (Reuters) – Alabama Secretary of St...    1

```

```
[19] : data=data.sample(frac=1)
data.head(10)
```

```
[19] :
text class
14708 AMMAN (Reuters) – The Syrian army backed by Ru... 1
18355 New York City s radical leftist Democrat Mayor... 0
13944 Open borders Ryan might be in for a shock when... 0
16062 MANILA (Reuters) – Philippine troops on Tuesda... 1
885 FRONT ROYAL, Va. (Reuters) – Ralph and Mike Wa... 1
20541 BEIRUT (Reuters) – The Syrian army and its all... 1
6330 WASHINGTON (Reuters) – President-elect Donald ... 1
18889 FBI director James Comey wanted to go public w... 0
14328 RIYADH (Reuters) – U.S. Secretary of State Rex... 1
20201 By now, pretty much everyone has seen pictures... 0
```

```
[20] : data.isnull().sum()
```

```
[20] : text      0
class      0
dtype: int64
```

```
[21] : def filtering(data):
text=data.lower()
text=re.sub('\[.*?\]', '', text)
text=re.sub("\W", "",text)
text=re.sub('https?://\S+|www\.\S+', '', text)
text=re.sub('<.*?>+', '', text)
text=re.sub('[%s]' % re.escape(string.punctuation), '', text)
text=re.sub('\n', '', text)
text=re.sub('\w*\d\w*', '', text)
return text
```

```
[22] : data["text"]=data["text"].apply(filtering)
data.head(10)
```

```
[22] :
text class
14708 ammanreuters the syrian army backed by ru... 1
18355 new york city s radical leftist democrat mayor... 0
13944 open borders ryan might be in for a shock when... 0
16062 manila reuters philippine troops on tuesda... 1
885 front royal va reuters ralph and mike wa... 1
20541 beirut reuters the syrian army and its all... 1
6330 washington reuters president elect donald ... 1
18889 fbi director james comey wanted to go public w... 0
14328 riyadh reuters u s secretary of state rex... 1
20201 by now pretty much everyone has seen pictures... 0
```

```
[23] : x=data['text']#independent
      y=data['class']#dependent
```

```
[26]: !pip install scikit-learn
      from sklearn.model_selection import train_test_split

      x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.25)
      # Now you can use train_test_split
```

Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)  
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)  
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)  
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)  
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)

```
[27]: from sklearn.feature_extraction.text import TfidfVectorizer

      vectorization = TfidfVectorizer()
      xv_train = vectorization.fit_transform(x_train)
      xv_test = vectorization.transform(x_test)
```

```
[28]: from sklearn.linear_model import LogisticRegression
```

```
[29]: LR = LogisticRegression()
      LR.fit(xv_train, y_train)
```

```
[29] : LogisticRegression()
```

```
[30] : LogisticRegression()
```

```
[30] : LogisticRegression()
```

```
[31] : pred_lr = LR.predict(xv_test)
```

```
[32] : LR.score(xv_test, y_test)
```

```
[32]: 0.9861853832442068
```

```
[33] : print(classification_report(y_test, pred_lr))
```

	precision	recall	f1-score	support
0	0.99	0.98	0.99	5850

1	0.98	0.99	0.99	5370
accuracy			0.99	11220
macro avg	0.99	0.99	0.99	11220
weighted avg	0.99	0.99	0.99	11220

```
[35]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
from sklearn import tree

# ... (rest of your code)

def predict_news(text):
    text_vectorized = vectorization.transform([filtering(text)]) # Use_
    'vectorization' instead of 'vectorizer'
    prediction = model.predict(text_vectorized) # Use 'model' instead of 'LR'
    if prediction == 1:
        return "This news is likely true."
    else:
        return "This news is likely fake."

user_input = input("Enter news text: ")
result = predict_news(user_input)
print(result)

# ... (rest of your code)
```

Enter news text: modhi is died

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-35-b0ee8519c6b8> in <cell line: 19>()
    17
    18 user_input = input("Enter news text: ")
---> 19 result = predict_news(user_input)
    20 print(result)
    21

<ipython-input-35-b0ee8519c6b8> in predict_news(text)
    10 def predict_news(text):
    11     text_vectorized = vectorization.transform([filtering(text)]) # Use_
    'vectorization' instead of 'vectorizer'
---> 12     prediction = model.predict(text_vectorized) # Use 'model' instead_
of 'LR'
```

```

13     if prediction == 1:
14         return "This news is likely true."

```

**NameError:** name 'model' is not defined

Decision Tree Classifier

```
[36]: from sklearn.tree import DecisionTreeClassifier
```

```

DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)

```

```
[36]: DecisionTreeClassifier()
```

```
[37]: DecisionTreeClassifier()
```

```
[37]: DecisionTreeClassifier()
```

```
[38]: pred_dt = DT.predict(xv_test)
```

```
[39]: DT.score(xv_test, y_test)
```

```
[39]: 0.9944741532976827
```

```
[40]: print(classification_report(y_test, pred_lr))
```

	precision	recall	f1-score	support
0	0.99	0.98	0.99	5850
1	0.98	0.99	0.99	5370
accuracy			0.99	11220
macro avg	0.99	0.99	0.99	11220
weighted avg	0.99	0.99	0.99	11220

Gradient Boost Classifier

```
[41]: from sklearn.ensemble import GradientBoostingClassifier
      from sklearn.metrics import accuracy_score
```

```

# Assuming xv_train and y_train are already defined and preprocessed
# Example:
# xv_train = vectorizer.fit_transform(x_train)
# y_train = y_train

```

```

# Initialize and train GradientBoostingClassifier
GB = GradientBoostingClassifier(random_state=0)

```



```
GB.fit(xv_train, y_train)
```

```
# Example of making predictions and evaluating
```

```
y_pred = GB.predict(xv_test) # Assuming xv_test is the vectorized test data
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print("Accuracy:", accuracy)
```

Accuracy: 0.9939393939393939

```
[42] : GradientBoostingClassifier(random_state=0)
```

```
[42] : GradientBoostingClassifier(random_state=0)
```

```
[43] : pred_gb = GB.predict(xv_test)
```

```
[44] : pred_gb = GB.predict(xv_test)
```

```
[45] : GB.score(xv_test, y_test)
```

```
[45]: 0.9939393939393939
```

```
[46] : print(classification_report(y_test, pred_gb))
```

	precision	recall	f1-score	support
0	1.00	0.99	0.99	5850
1	0.99	1.00	0.99	5370
accuracy			0.99	11220
macro avg	0.99	0.99	0.99	11220
weighted avg	0.99	0.99	0.99	11220

```
[47] : from sklearn.ensemble import RandomForestClassifier
```

```
RF = RandomForestClassifier(random_state = 0)
```

```
RF.fit(xv_train, y_train)
```

```
[47] : RandomForestClassifier(random_state=0)
```

```
[48] : RF.score(xv_test, y_test)
```

```
[48]: 0.9905525846702318
```

Testing the Model

```
[49]: def output_lable(n):
    if n==0:
        return "Fake News"
    elif n==1:
        return "Not A Fake News"
```

```
[52]: def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    # Define the wordopt function here or make sure it is defined elsewhere and
    has been executed
    def wordopt(text):
        # Implement the logic for wordopt here
        return processed_text
    new_def_test['text'] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    pred_DT = DT.predict(new_xv_test)
    pred_GB = GB.predict(new_xv_test)
    pred_RF = RF.predict(new_xv_test)

    return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction:
    {} \nRFC Prediction:{}".format(output_lable(pred_LR[0]),
                                   output_lable(pred_DT[0]),
                                   output_lable(pred_GB[0]),
                                   output_lable(pred_RF[0])))
```

```
[60]: def predict_news(text):
    text_vectorized = vectorization.transform([filtering(text)])
    prediction = RF.predict(text_vectorized)
    if prediction == 1:
        return "This news is likely true." # Indent this line
    else:
        return "This news is likely fake." # Indent this line

user_input = input("Enter news text: ")
result = predict_news(user_input)
print(result)
```

Enter news text: WASHINGTON (Reuters) – The head of a conservative Republican faction in the U.S.

This news is likely true.