About the Dataset: 1.id: unique id for a news article 2.title: the title of a news article 3.author: author of the news article 4.text: the text of the article; could be incomplete 5.label: a label that marks whether the news article is real or fake: 1: Fake news 0: real News

```
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.sparse as sp
import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
# printing the stopwords in English
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you'
```

```
# loading the dataset to a pandas DataFrame
my_dataset = pd.read_csv('/content/sample_data/train.csv')
```

```
my_dataset.shape
```

```
(20800, 5)
```

```
# print the first 5 rows of the dataframe
my_dataset.head()
```

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0

```
titles = my_dataset["title"]
labels = my_dataset["label"]
```

15 Civilians Killed In Single US Videos 15 Civilians Killed

```
x=[]
y=[]
```

unpublished... has been sentenced to...

```
x=list(titles)
y=list(labels)
```

Data Cleaning

```
# counting the number of missing values in the dataset
my_dataset.isnull().sum()
```

```
id           0
title      558
author    1957
text        39
label        0
dtype: int64
```

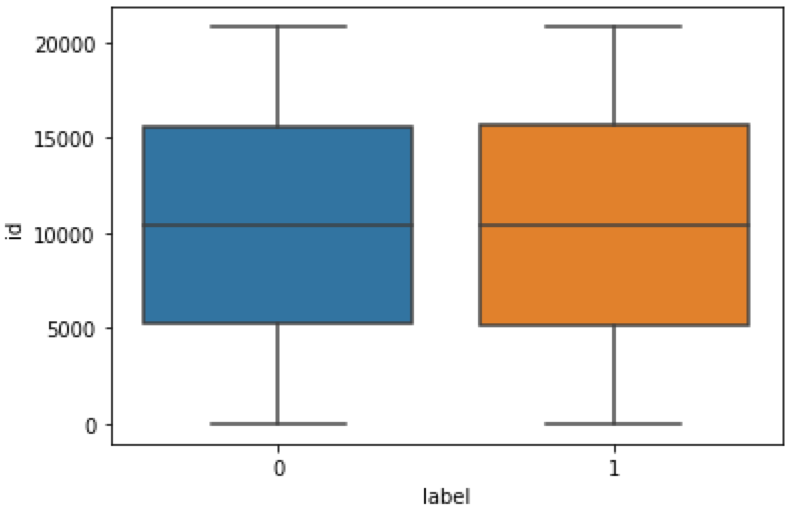
```
# replacing the null values with empty string
my_dataset = my_dataset.fillna('')
```

Data Exploration

```
my_dataset
```

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...	...	...

```
plot = sns.boxplot(x='label',y="id",data=my_dataset)
```



```
#print(my_dataset.groupby("label")('1').count())
# data.groupby([‘target’])[‘text’].count().plot(kind=”bar”)
# plt.show()
```

```
# merging the author name and news title
my_dataset['content'] = my_dataset['author']+' '+my_dataset['title']
```

```
print(my_dataset['content'])
```

```
0        Darrell Lucus House Dem Aide: We Didn’t Even S...
1        Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...
2        Consortiumnews.com Why the Truth Might Get You...
3        Jessica Purkiss 15 Civilians Killed In Single ...
4        Howard Portnoy Iranian woman jailed for fictio...
                                ...
20795    Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...
```

```
20796    Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...
20797    Michael J. de la Merced and Rachel Abrams Macy...
20798    Alex Ansary NATO, Russia To Hold Parallel Exer...
20799              David Swanson What Keeps the F-35 Alive
Name: content, Length: 20800, dtype: object
```

```
# separating the data & label
X = my_dataset.drop(columns='label', axis=1)
Y = my_dataset['label']
```

```
print(X)
```

```
          id                                              title  \
0          0  House Dem Aide: We Didn’t Even See Comey’s Let...
1          1  FLYNN: Hillary Clinton, Big Woman on Campus - ...
2          2                  Why the Truth Might Get You Fired
3          3  15 Civilians Killed In Single US Airstrike Hav...
4          4  Iranian woman jailed for fictional unpublished...
...      ...                                                ...
20795  20795  Rapper T.I.: Trump a ’Poster Child For White S...
20796  20796  N.F.L. Playoffs: Schedule, Matchups and Odds -...
20797  20797  Macy’s Is Said to Receive Takeover Approach by...
20798  20798  NATO, Russia To Hold Parallel Exercises In Bal...
20799  20799                          What Keeps the F-35 Alive

                                          author  \
0                                  Darrell Lucus
1                                Daniel J. Flynn
2                             Consortiumnews.com
3                                Jessica Purkiss
4                                 Howard Portnoy
...                                          ...
20795                              Jerome Hudson
20796                           Benjamin Hoffman
20797  Michael J. de la Merced and Rachel Abrams
20798                                Alex Ansary
20799                              David Swanson

                                                    text  \
0      House Dem Aide: We Didn’t Even See Comey’s Let...
1      Ever get the feeling your life circles the rou...
2      Why the Truth Might Get You Fired October 29, ...
3      Videos 15 Civilians Killed In Single US Airstr...
4      Print \nAn Iranian woman has been sentenced to...
...                                                  ...
20795  Rapper T. I. unloaded on black celebrities who...
20796  When the Green Bay Packers lost to the Washing...
20797  The Macy’s of today grew from the union of sev...
20798  NATO, Russia To Hold Parallel Exercises In Bal...
20799    David Swanson is an author, activist, journa...

                                                 content
0      Darrell Lucus House Dem Aide: We Didn’t Even S...
1      Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...
2      Consortiumnews.com Why the Truth Might Get You...
3      Jessica Purkiss 15 Civilians Killed In Single ...
4      Howard Portnoy Iranian woman jailed for fictio...
```

```
...                                                  ...
20795  Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...
20796  Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...
20797  Michael J. de la Merced and Rachel Abrams Macy...
20798  Alex Ansary NATO, Russia To Hold Parallel Exer...
20799            David Swanson What Keeps the F-35 Alive

[20800 rows x 5 columns]
```

```
print(Y)
```

```
0        1
1        0
2        1
3        1
4        1
        ..
20795    0
20796    0
20797    0
20798    1
20799    1
Name: label, Length: 20800, dtype: int64
```

```
port_stem = PorterStemmer()
```

```
def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ',content)
    stemmed_content = stemmed_content.lower()
    stemmed_content = stemmed_content.split()
    stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in stc
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

```
my_dataset['content'] = my_dataset['content'].apply(stemming)
```

```
print(my_dataset['content'])
```

```
0        darrel lucu hous dem aid even see comey letter...
1        daniel j flynn flynn hillari clinton big woman...
2                   consortiumnew com truth might get fire
3        jessica purkiss civilian kill singl us airstri...
4        howard portnoy iranian woman jail fiction unpu...
                               ...
20795    jerom hudson rapper trump poster child white s...
20796    benjamin hoffman n f l playoff schedul matchup...
20797    michael j de la merc rachel abram maci said re...
20798    alex ansari nato russia hold parallel exercis ...
20799                            david swanson keep f aliv
Name: content, Length: 20800, dtype: object
```

```
#separating the data and label
X = my_dataset['content'].values
```

```
Y = my_dataset['label'].values

print(X)
```

```
['darrel lucu hous dem aid even see comey letter jason chaffetz tweet'
 'daniel j flynn flynn hillari clinton big woman campu breitbart'
 'consortiumnew com truth might get fire' ...
 'michael j de la merc rachel abram maci said receiv takeov approach hudson bay new
 'alex ansari nato russia hold parallel exercis balkan'
 'david swanson keep f aliv']
```

```
print(Y)
```

```
[1 0 1 ... 0 1 1]
```

Data Analysis and Visualization

```
#Analyzing the frequency of news with word cloud(Data Visualization Technique)
from wordcloud import WordCloud
fake_data = my_dataset[my_dataset["label"] == "1"]
all_words = ' '.join([text for text in fake_data.text])
counts = my_dataset['label'].value_counts()
counts.index = counts.index.map(str)
wordcloud = WordCloud().generate_from_frequencies(counts)
plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

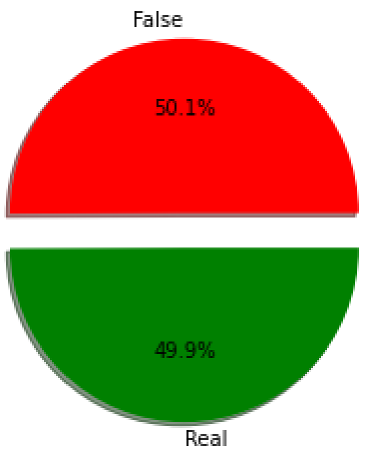
```
#Analysing number of false and true news
c_real = 0
c_false = 0
```

```
for i in Y:
  if Y[i]==0:
    c_real=c_real+1
  else :
    c_false = c_false+1
print("Number of real news: "+ str(c_real))
print("Number of false news: "+ str(c_false))
```

```
     Number of real news: 10413
     Number of false news: 10387
```

```
#Visualizing percentage of fake and real news by cateogery in train file
label_size =[c_real,c_false]
plt.pie(label_size,explode=[0.1,0.1],colors=['red','green'],shadow=True,labels=['False','F
```

```
     ([<matplotlib.patches.Wedge at 0x7f15c2c2f110>,
       <matplotlib.patches.Wedge at 0x7f15c2c2fd90>],
      [Text(-0.0023562469051222937, 1.1999976867063213, 'False'),
       Text(0.00235624690512188, -1.1999976867063213, 'Real')],
      [Text(-0.0013744773613213377, 0.6999986505786873, '50.1%'),
       Text(0.0013744773613210966, -0.6999986505786873, '49.9%')])
```



```
# width =0.3
# plt.bar(np.arange(c_real, c_real, width=width)
# plt.bar(np.arange(len(Y))+ width, Y, width=width)
# plt.show()
```

```
Y.shape
print(Y)
```

```
     [1 0 1 ... 0 1 1]
```

```
# converting the textual data to numerical data
vectorizer = TfidfVectorizer()
vectorizer.fit(X)

X = vectorizer.transform(X)
```

```
print(X)
```

```
  (0, 15686)    0.28485063562728646
  (0, 13473)    0.2565896679337957
  (0, 8909)     0.3635963806326075
  (0, 8630)     0.29212514087043684
  (0, 7692)     0.24785219520671603
  (0, 7005)     0.21874169089359144
  (0, 4973)     0.233316966909351
  (0, 3792)     0.2705332480845492
  (0, 3600)     0.3598939188262559
  (0, 2959)     0.2468450128533713
  (0, 2483)     0.3676519686797209
  (0, 267)      0.27010124977708766
  (1, 16799)    0.30071745655510157
  (1, 6816)     0.1904660198296849
  (1, 5503)     0.7143299355715573
  (1, 3568)     0.26373768806048464
  (1, 2813)     0.19094574062359204
  (1, 2223)     0.3827320386859759
  (1, 1894)     0.15521974226349364
  (1, 1497)     0.2939891562094648
  (2, 15611)    0.41544962664721613
  (2, 9620)     0.49351492943649944
  (2, 5968)     0.3474613386728292
  (2, 5389)     0.3866530551182615
  (2, 3103)     0.46097489583229645
  :     :
  (20797, 13122)        0.2482526352197606
  (20797, 12344)        0.27263457663336677
  (20797, 12138)        0.24778257724396507
  (20797, 10306)        0.08038079000566466
  (20797, 9588) 0.174553480255222
  (20797, 9518) 0.2954204003420313
  (20797, 8988) 0.36160868928090795
  (20797, 8364) 0.22322585870464118
  (20797, 7042) 0.21799048897828688
  (20797, 3643) 0.21155500613623743
  (20797, 1287) 0.33538056804139865
  (20797, 699)  0.30685846079762347
  (20797, 43)   0.29710241860700626
  (20798, 13046)        0.22363267488270608
  (20798, 11052)        0.4460515589182236
  (20798, 10177)        0.3192496370187028
  (20798, 6889) 0.32496285694299426
  (20798, 5032) 0.4083701450239529
  (20798, 1125) 0.4460515589182236
  (20798, 588)  0.3112141524638974
  (20798, 350)  0.28446937819072576
  (20799, 14852)        0.5677577267055112
  (20799, 8036) 0.45983893273780013
  (20799, 3623) 0.37927626273066584
  (20799, 377)  0.5677577267055112
```

```
type(X)
```

```
scipy.sparse.csr.csr_matrix
```

splitting the data set in to training and test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, stratify=Y, rar
```

```
print(X_train)
```

```
  (0, 11684)	0.5070580156214289
  (0, 10315)	0.39867999740572146
  (0, 5147)	0.5278034356036302
  (0, 114)	0.5526028970496242
  (1, 15611)	0.34068220157778806
  (1, 12693)	0.33794458026137175
  (1, 10306)	0.11725436798714685
  (1, 10289)	0.35264942087438866
  (1, 6816)	0.21005241836193117
  (1, 5233)	0.2782015960508413
  (1, 4680)	0.2805114469722502
  (1, 4068)	0.3514994338166583
  (1, 2813)	0.2105814708038776
  (1, 2526)	0.3320399992928536
  (1, 1954)	0.3914934354594198
  (2, 16704)	0.32002037998152355
  (2, 12430)	0.32002037998152355
  (2, 10611)	0.31787162360806587
  (2, 9082)	0.33295341853189286
  (2, 6383)	0.2810399578483559
  (2, 5390)	0.3797958930662605
  (2, 3105)	0.29715407865561705
  (2, 2526)	0.23906998304316238
  (2, 2070)	0.29971898869481917
  (2, 1983)	0.2892920262544172
  :	:
  (14557, 8026)	0.4341030219850026
  (14557, 6398)	0.36578019031535824
  (14557, 3648)	0.34411587758353523
  (14557, 908)	0.2293218471469665
  (14557, 536)	0.2592097704417449
  (14558, 15231)	0.2218396553848159
  (14558, 8680)	0.7222309741587801
  (14558, 7278)	0.38996055895126003
  (14558, 6816)	0.14478467103493528
  (14558, 5400)	0.20447693286274574
  (14558, 4035)	0.3550990105395703
  (14558, 436)	0.29703808745848487
  (14559, 16996)	0.08245668067938565
  (14559, 15295)	0.08090589645691927
  (14559, 12219)	0.3115606907305463
  (14559, 12137)	0.23002341966320766
  (14559, 10306)	0.0797042735905404
  (14559, 9341)	0.2694570955700903
  (14559, 8388)	0.38457262303621853
  (14559, 8067)	0.2963473515946786
  (14559, 7386)	0.38457262303621853
  (14559, 4805)	0.2712441338310002
  (14559, 3944)	0.3042758190266888
  (14559, 3469)	0.3585652480042937
  (14559, 813)	0.28173435664213364
```

```
print(X_test)
```

```
  (0, 15705)	0.45003463066461935
  (0, 13463)	0.27408268150674925
  (0, 12305)	0.2670868113223411
  (0, 9872)	0.2952009846812773
  (0, 5599)	0.2667477556464323
  (0, 4094)	0.3615652927181354
  (0, 3326)	0.432231674736212
  (0, 1236)	0.31141866041273353
  (0, 368)	0.2796048998537153
  (1, 15173)	0.42575771209033225
  (1, 14572)	0.3876959831513743
  (1, 11092)	0.2640956935704667
  (1, 7395)	0.3737192892699826
  (1, 6774)	0.3969651256507491
  (1, 4222)	0.2837890310876728
  (1, 2462)	0.354944314618474
  (1, 2144)	0.30806908491504775
  (2, 15582)	0.13233753904564444
  (2, 14524)	0.3003056663043611
  (2, 11886)	0.350980425781301
  (2, 10749)	0.3665403200383284
  (2, 10174)	0.3799069003178553
  (2, 7824)	0.226689047951983
  (2, 4533)	0.42885343252619573
  (2, 4530)	0.32675030353054124
  :	:
  (6237, 1894)	0.11007362902863657
  (6237, 1425)	0.25025997923501425
  (6237, 350)	0.21126676090780142
  (6238, 11506)	0.3806498882273162
  (6238, 9155)	0.43610084301214197
  (6238, 7824)	0.2364734559380314
  (6238, 7520)	0.40370414475223176
  (6238, 2986)	0.3569270969725366
  (6238, 2631)	0.34482993704937287
  (6238, 2323)	0.37006453086249474
  (6238, 469)	0.25054458769254745
  (6239, 15142)	0.25096907872339486
  (6239, 15058)	0.2555755954429748
  (6239, 14444)	0.19399060814385596
  (6239, 14273)	0.3478801778542246
  (6239, 9657)	0.22397801136622394
  (6239, 8792)	0.352122320037735
  (6239, 6840)	0.29412888112928376
  (6239, 6816)	0.15322038094646717
  (6239, 3848)	0.28389262368317575
  (6239, 2129)	0.3242140578611897
  (6239, 1880)	0.21942167300737828
  (6239, 1403)	0.33731657623558986
  (6239, 908)	0.20326241985833352
  (6239, 469)	0.19986087420718443
```

```
X_test.data
```

```
array([0.45003463, 0.27408268, 0.26708681, ..., 0.33731658, 0.20326242,
       0.19986087])
```

```
type(X_test)

     scipy.sparse.csr.csr_matrix
```

```
X_test.shape

     (6240, 17128)
```

```
type(X_test.shape)

     tuple
```

```
i = X_test.shape[0]
```

```
j= i-1
print(j)

     6239
```

```
X_test.nnz

     63309
```

```
print(Y_train)
print(len(Y_train))

     [1 1 1 ... 0 1 0]
     14560
```

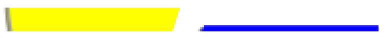
```
print(Y_test)

     [1 1 0 ... 0 0 1]
```

```
#visualizing test data and train data
label_size =[len(Y_train),len(Y_test)]
plt.pie(label_size,explode=[0.1,0.1],colors=['Yellow','blue'],shadow=True,labels=['Train',
     ([<matplotlib.patches.Wedge at 0x7f15c6265050>,
       <matplotlib.patches.Wedge at 0x7f15c6265ad0>],
      [Text(-0.705342266393061, 0.9708204196655015, 'Train'),
       Text(0.7053422663930609, -0.9708204196655015, 'Test')],
      [Text(-0.41144965539595224, 0.5663119114715425, '70.0%'),
```

```
model = LogisticRegression()
```

```
model.fit(X_train, Y_train)
```

```
LogisticRegression()
```



```
# accuracy score on the training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy score of the training data : ', training_data_accuracy)
```

```
Accuracy score of the training data :  0.9850961538461539
```

```
# accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy score of the test data : ', test_data_accuracy)
```

```
Accuracy score of the test data :  0.9743589743589743
```

```
print("Enter the val needs to be tested to know whether the news is fake or not")
print("The value must be between 0 and "+ str(j) + " as this is the range of test data")
val = int(input())
```

```
Enter the val needs to be tested to know whether the news is fake or not
The value must be between 0 and 6239 as this is the range of test data
600
```

```
try:
    X_new = X_test[val]
    #X_new = X_test[3]
    prediction = model.predict(X_new)
    view =model.predict_proba(X_new)
    print(prediction)
    print(view)
    if (prediction[0]==0):
      print('The news is Real')
    else:
      print('The news is Fake')
except:
    print("Out of range")
```

```
[1]
```

```
    [[0.12596282 0.87403718]]
    The news is Fake
```

```
try:
 print(Y_test[val])
except:
  print("Out of range")
```

```
    1
```