Importing the Dependencies

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 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]   Unzipping corpora/stopwords.zip.
True
```

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

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

Data Pre-processing

```
# loading the dataset to a pandas DataFrame
data = pd.read_csv('train.csv')
data.head()
```

```
<ipython-input-37-f28df1ebfd34>:2: DtypeWarning: Columns (0,4,5,6,7,8,9,10,11,12
  data = pd.read_csv('train.csv')
```

	id	title	author	text	label	Unnamed: 5	Unnamed: 6	Unnamed: 7
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	NaN	NaN	NaN
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0	NaN	NaN	NaN
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1	NaN	NaN	NaN
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1	NaN	NaN	NaN
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1	NaN	NaN	NaN

5 rows × 686 columns

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


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


print(news_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...
```

```
                                  ...
25111    Jerome Hudson Rapper T.I.: Trump a ‘Poster Chi...
25112    Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...
25113    Michael J. de la Merced and Rachel Abrams Macy...
25114    Alex Ansary NATO, Russia To Hold Parallel Exer...
25115                David Swanson What Keeps the F-35 Alive
Name: content, Length: 25116, dtype: object
```

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

```
print(X)
print(Y)
```

```
           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...
...       ...                                                ...
25111   20795  Rapper T.I.: Trump a ‘Poster Child For White S...
25112   20796  N.F.L. Playoffs: Schedule, Matchups and Odds -...
25113   20797  Macy’s Is Said to Receive Takeover Approach by...
25114   20798  NATO, Russia To Hold Parallel Exercises In Bal...
25115   20799                          What Keeps the F-35 Alive

                                      author  \
0                               Darrell Lucus
1                              Daniel J. Flynn
2                            Consortiumnews.com
3                               Jessica Purkiss
4                                Howard Portnoy
...                                         ...
25111                           Jerome Hudson
25112                         Benjamin Hoffman
25113  Michael J. de la Merced and Rachel Abrams
25114                             Alex Ansary
25115                           David Swanson

                                                    text Unnamed: 5  \
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...
...                                                  ...        ...
25111  Rapper T. I. unloaded on black celebrities who...
25112  When the Green Bay Packers lost to the Washing...
25113  The Macy’s of today grew from the union of sev...
25114  NATO, Russia To Hold Parallel Exercises In Bal...
25115    David Swanson is an author, activist, journa...

      Unnamed: 6 Unnamed: 7 Unnamed: 8 Unnamed: 9 Unnamed: 10  ...  \
```

```
0                                                                    ...
1                                                                    ...
2                                                                    ...
3                                                                    ...
4                                                                    ...
...          ...        ...        ...        ...          ...     ...
25111                                                                ...
25112                                                                ...
25113                                                                ...
25114                                                                ...
25115                                                                ...

      Unnamed: 677 Unnamed: 678 Unnamed: 679 Unnamed: 680 Unnamed: 681  \
0
1
2
3
4
```

Stemming:

Stemming is the process of reducing a word to its Root word

example: actor, actress, acting --> act

```
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 stop
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content


news_dataset['content'] = news_dataset['content'].apply(stemming)


print(news_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...
                              ...
25111    jerom hudson rapper trump poster child white s...
25112    benjamin hoffman n f l playoff schedul matchup...
25113    michael j de la merc rachel abram maci said re...
25114    alex ansari nato russia hold parallel exercis ...
25115                            david swanson keep f aliv
Name: content, Length: 25116, dtype: object
```

```
#separating the data and label
X = news_dataset['content'].values
Y = news_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
 'alex ansari nato russia hold parallel exercis balkan'
 'david swanson keep f aliv']
```

```
print(Y)
```

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

```
Y.shape
```

```
(25116,)
```

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

X = vectorizer.transform(X)
```

```
print(X)
```

```
  (0, 294)      0.2711143595304882
  (0, 2676)     0.3695330951455916
  (0, 3190)     0.25021444167545076
  (0, 3884)     0.36190187718479516
  (0, 4092)     0.2719193039217898
  (0, 5380)     0.2242110980219714
  (0, 7617)     0.21896464665093704
  (0, 8387)     0.2514419730052186
  (0, 9387)     0.2931650914234994
  (0, 9694)     0.365543809573664
  (0, 14648)    0.24170537647240825
  (0, 17070)    0.2880854334390338
  (1, 1617)     0.28684095327432707
  (1, 2043)     0.16033000789330373
  (1, 2405)     0.3819555242659426
  (1, 3036)     0.19465794276348
  (1, 3850)     0.2655601620053947
  (1, 5954)     0.7141085899028713
  (1, 7407)     0.1937754459590677
  (1, 18279)    0.3003541536609626
  (2, 3174)     0.3289786239632894
  (2, 3349)     0.4735061475862881
  (2, 5829)     0.3938520403026429
  (2, 6467)     0.3373308801281171
```

```
  (2, 10451)    0.4757873319489284
  :     :
  (25113, 3932) 0.21371008348018833
  (25113, 7656) 0.2216005995344619
  (25113, 9104) 0.22557147358170038
  (25113, 9775) 0.3629705176720701
  (25113, 10332)        0.2978183783941726
  (25113, 10414)        0.1782507786420312
  (25113, 11195)        0.08589792606878596
  (25113, 13201)        0.2509263052625155
  (25113, 13422)        0.26602859281597036
  (25113, 14266)        0.23506540036371604
  (25113, 16287)        0.3113354554763349
  (25113, 16655)        0.08673957288852255
  (25113, 18500)        0.08885964892553022
  (25114, 383)  0.28778802112656215
  (25114, 640)  0.3150505522876953
  (25114, 1219) 0.4477091867359452
  (25114, 5445) 0.4106366484638114
  (25114, 7490) 0.3173446208314138
  (25114, 11057)        0.32229857515469124
  (25114, 12015)        0.43913659520232545
  (25114, 14187)        0.22678249995679425
  (25115, 414)  0.5641330209353659
  (25115, 3907) 0.3878802220711985
  (25115, 8758) 0.44424508431121
  (25115, 16164)        0.5778833559479472
```

Splitting the dataset to training & test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3,  random_state=2)
```

```
print(X_train)
```

```
  (0, 335)      0.25571605545537907
  (0, 737)      0.3122829696476194
  (0, 800)      0.41795325427348295
  (0, 2043)     0.1316954445738917
  (0, 2236)     0.38126412995980286
  (0, 4092)     0.2601286906239681
  (0, 9308)     0.26553098430056377
  (0, 12584)    0.2578616179927715
  (0, 14862)    0.37396380700597887
  (0, 16734)    0.2960477735043463
  (0, 16961)    0.11674942493635179
  (0, 17794)    0.23218053635828198
  (1, 1986)     0.30971645586489044
  (1, 2181)     0.2593728567882714
  (1, 2704)     0.19972149673036055
  (1, 2874)     0.21614854044824247
  (1, 4352)     0.3340324617476545
  (1, 4899)     0.2342313873827553
  (1, 5562)     0.22373377274088468
```

```
(1, 7568)     0.2219646290902192
(1, 8506)     0.3175444613139986
(1, 8770)     0.27757244453301827
(1, 11088)    0.27619655912439417
(1, 11195)    0.08090217719456319
(1, 12937)    0.238109260572727
:     :
(17576, 3036) 0.1986787505242242
(17576, 4545) 0.3591874229116876
(17576, 7445) 0.31995856935281153
(17576, 8630) 0.32725266857669155
(17576, 8812) 0.39172283562737004
(17576, 9605) 0.31861981778132464
(17576, 9630) 0.3124183660909029
(17576, 11522)        0.4073285323492249
(17577, 512)  0.2620529252434432
(17577, 3047) 0.3370129051238706
(17577, 3920) 0.2696727196445279
(17577, 4753) 0.3927415096235872
(17577, 5004) 0.23545227475286076
(17577, 5671) 0.33984058155804636
(17577, 11210)        0.4348792937299835
(17577, 14511)        0.3277754632424745
(17577, 17709)        0.3511340884484408
(17579, 2517) 0.3213169485220343
(17579, 3058) 0.3723339463827434
(17579, 5987) 0.3842761748170999
(17579, 9227) 0.3842761748170999
(17579, 12236)        0.23928112918931796
(17579, 13712)        0.46307363432837145
(17579, 14648)        0.274409046675963
(17579, 16788)        0.34028446835787796
```

Training the Model: Logistic Regression

```
model = LogisticRegression()

model.fit(X_train, Y_train)
```

```
LogisticRegression()
```

Evaluation

accuracy score

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

```
Accuracy score of the training data :  0.9379443717649736
```

```
# 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 :  0.8951559389515594
```

Making a Predictive System

```
X_new = X_test[1]

prediction = model.predict(X_new)
print(prediction)

if (prediction[0]=='0'):
  print('The news is Real')
else:
  print('The news is Fake')
```

```
['1']
The news is Fake
```

```
print(Y_test[1])
```

```
1
```