

# Greetings Welcome To

## #Day12,13 of #30DaysofMachineLearning

### Project 02- Fake/Real News Detection

#### 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
```

## ▼ Importing the Dependencies

```
import numpy as np
import pandas as pd
import re #re Stands For Regular Expression it is very uSefull for searching the
#text in the News/document

from nltk.corpus import stopwords #nltk Stands for natural language toolkit and
#corpus generally means the body of the documents important context of text

from nltk.stem.porter import PorterStemmer #we will perform a function called Streaming so
#that function takes a word and remove the prefix and sufix of that word and return the ro

from sklearn.feature_extraction.text import TfidfVectorizer #TfidfVectorizer This function
#are using to convert the text into feature vectors. feature vector are Nothing but Numbe

from sklearn.model_selection import train_test_split #train_test_split With The Help of Th
#will Split our Dataset into training data and Testing data

from sklearn.linear_model import LogisticRegression #importing Our Model
from sklearn.metrics import accuracy_score #importing The Accuracy Score
```

## ▼ We need To download stopwords from nltk

```
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", "you\'
```



## ▼ Data Pre-processing

```
# loading the dataset to a pandas DataFrame
data = pd.read_csv('../input/fakereal-news-prediction/train.csv')
```

```
data.shape
```

```
(20800, 5)
```

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

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucas	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

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

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

```
# Replacing the null values with empty string
```

```
data = data.fillna('')

# Merging the author name and news title
data['content'] = data['author']+' '+ data['title']
```

```
#printing The Data From Content Column
print(data['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 = data.drop(columns='label', axis=1)
Y = data['label']
```

```
print("printing X data")
print(X)
```

```
printing X data
      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 Aistr...
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("Printing The y Data")
print(Y)

```

```

Printing The y Data
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

```

## ▼ 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 stopwords]
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

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

```
print(data['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 = data['content'].values
Y = data['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 y
 'alex ansari nato russia hold parallel exercis balkan'
 'david swanson keep f aliv']
```

```
print(Y)
```

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

```
Y.shape
```

```
(20800,)
```

```
# 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)     0.24684501285337135
(0, 2483)     0.3676519686797209
(0, 267)      0.2701012497770876
(1, 16799)    0.30071745655510157
(1, 6816)     0.19046601982968486
(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)  0.29542040034203126
(20797, 8988)  0.36160868928090795
(20797, 8364)  0.22322585870464118
(20797, 7042)  0.21799048897828685
(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
```

## ▼ Splitting the dataset to training & test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, ran
```

## ▼ Training the Model: Logistic Regression

```
model = LogisticRegression()
```

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

```
LogisticRegression()
```

## Evaluation

## ▼ Accuracy score

```
# 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 is : ', training_data_accuracy)
```

```
Accuracy score of the training data is : 0.9865985576923076
```

```
# 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 is : ', test_data_accuracy)
```

```
Accuracy score of the test data is : 0.9790865384615385
```

## ▼ Making a Predictive System

```
X_new = X_test[3]

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

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

```
else:  
    print('The news is Fake')
```

```
    [0]  
    The news is Real
```

```
print(Y_test[3])
```

```
0
```

```
print(Y_test[4])
```

```
0
```

```
print(Y_test[5])
```

```
1
```

IF You like Please UpVote:)

Thanks !

Ahmed Ali..

