In [ ]:

```
#Detecting Fake News:
    To build a model to accurately classify a piece of news as REAL or FAKE.
```

In [ ]:

```
#Fake News: It includes training and a dataset with a unique id for a news article, TITLE o
```

In [1]:

```
import pandas as pd
import numpy as np
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
```

In [2]:

```
#Get STopwords
```

In [3]:

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

```
[nltk_data] Error loading stopwords: <urlopen error [Errno 11001]
[nltk_data]     getaddrinfo failed>
```

Out[3]:

```
False
```

In [4]:

```
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", won', "won't", 'wouldn', "wouldn't"]
```

In [5]:

```
#Data Pre-processing
```

In [6]:

```
dataset = pd.read_csv('news.csv')
```

In [7]:

```
dataset
```

Out[7]:

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows × 4 columns

In [8]:

```
dataset.head()
```

Out[8]:

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

In [9]:

```
dataset.tail()
```

Out[9]:

	Unnamed: 0	title	text	label
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

In [10]:

```
dataset.isnull().sum()
```

Out[10]:

```
Unnamed: 0    0
title         0
text          0
label         0
dtype: int64
```

In [11]:

```
#Preprocessing test data
dataset.shape
```

Out[11]:

```
(6335, 4)
```

In [12]:

```
# Merge title and text cols
dataset['content'] = dataset['title']+ ' '+ dataset['text']
```

In [13]:

```
dataset
```

Out[13]:

	Unnamed: 0	title	text	label	content
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	You Can Smell Hillary's Fear Daniel Greenfield...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Watch The Exact Moment Paul Ryan Committed Pol...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	Kerry to go to Paris in gesture of sympathy U....
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	Bernie supporters on Twitter erupt in anger ag...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	The Battle of New York: Why This Primary Matte...
...	...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL	State Department says it can't find emails fro...
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE	The 'P' in PBS Should Stand for 'Plutocratic' ...
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE	Anti-Trump Protesters Are Tools of the Oligarc...
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL	In Ethiopia, Obama seeks progress on peace, se...
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL	Jeb Bush Is Suddenly Attacking Trump. Here's W...

6335 rows × 5 columns

In [14]:

```
#Stemming procedure
```

In [15]:

```
Port_stem = PorterStemmer()
```

In [49]:

```
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
```

In [50]:

```
import re
```

In [51]:

```
from nltk.stem import PorterStemmer
stemmer=PorterStemmer()
```

In [52]:

```
dataset['title'] = dataset['title'].apply(stemming)
```

In [53]:

```
print(dataset['title'])
```

```
0                                     smell hillari fear
1       watch exact moment paul ryan commit polit suic...
2                           kerri go pari gestur sympathi
3          berni support twitter erupt anger dnc tri warn
4                           battl new york primari matter
                              ...
6330       state depart say find email clinton specialist
6331                        p pb stand plutocrat pentagon
6332             anti trump protest tool oligarchi inform
6333    ethiopia obama seek progress peac secur east a...
6334                jeb bush suddenli attack trump matter
Name: title, Length: 6335, dtype: object
```

In [54]:

```
#Separating the target and features
X = dataset['text'].values
X
```

Out[54]:

```
array(['Daniel Greenfield, a Shillman Journalism Fellow at the Freedom Cen
ter, is a New York writer focusing on radical Islam. \nIn the final stretc
h of the election, Hillary Rodham Clinton has gone to war with the FBI. \n
The word “unprecedented” has been thrown around so often this election tha
t it ought to be retired. But it’s still unprecedented for the nominee of
a major political party to go war with the FBI. \nBut that’s exactly what
Hillary and her people have done. Coma patients just waking up now and wat
ching an hour of CNN from their hospital beds would assume that FBI Direct
or James Comey is Hillary’s opponent in this election. \nThe FBI is under
attack by everyone from Obama to CNN. Hillary’s people have circulated a l
etter attacking Comey. There are currently more media hit pieces lambastin
g him than targeting Trump. It wouldn’t be too surprising if the Clintons
or their allies were to start running attack ads against the FBI. \nThe FB
I’s leadership is being warned that the entire left-wing establishment wil
l form a lynch mob if they continue going after Hillary. And the FBI’s cre
dibility is being attacked by the media and the Democrats to preemptively
head off the results of the investigation of the Clinton Foundation and Hi
llary Clinton. \nThe covert struggle between FBI agents and Obama’s DOJ pe
```

In [55]:

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

Out[55]:

```
array(['FAKE', 'FAKE', 'REAL', ..., 'FAKE', 'REAL', 'REAL'], dtype=object)
```

In [56]:

```
#Converting text data to numerical data
```

In [57]:

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(X)
```

In [58]:

```
print(X)
```

```
  (0, 20169)    0.014178228995882907
  (0, 18853)    0.009272970141159279
  (0, 9478)     0.014151936756099957
  (0, 2306)     0.021002759070217543
  (0, 29229)    0.020971160535555908
  (0, 9970)     0.02373789102685452
  (0, 35452)    0.01739914666626697
  (0, 11488)    0.015903774476540594
  (0, 7360)     0.00851757554951921
  (0, 54591)    0.01681725296153299
  (0, 41241)    0.010445411064565417
  (0, 15571)    0.017943875463470493
  (0, 42123)    0.018847229406544267
  (0, 60169)    0.00943587772821063
  (0, 66227)    0.023232767820230864
  (0, 56882)    0.014191435022357902
  (0, 34685)    0.008473013319232815
  (0, 37735)    0.008879850243481452
  (0, 52895)    0.01869543128192668
  (0, 62044)    0.021839533627909375
  (0, 48810)    0.021956440574128107
  (0, 12613)    0.01996403366596103
  (0, 6194)     0.01148458488643172
  (0, 57065)    0.015510228181162313
  (0, 10714)    0.02504773510038326
  :     :
  (6334, 46170) 0.013537432252378223
  (6334, 23833) 0.03604699183792198
  (6334, 57284) 0.01396094413508426
  (6334, 10081) 0.041850073029148845
  (6334, 7183)  0.007903701587123184
  (6334, 31871) 0.06050294014879994
  (6334, 59817) 0.08573269806256655
  (6334, 60056) 0.054765985661499136
  (6334, 42600) 0.01893221285253174
  (6334, 55750) 0.03073662370850829
  (6334, 5267)  0.015235307818133429
  (6334, 7339)  0.028295606425907522
  (6334, 65918) 0.015196188393628629
  (6334, 64851) 0.016342132115853072
  (6334, 60504) 0.22767608902556247
  (6334, 27517) 0.05713266993191476
  (6334, 12544) 0.012604398227018886
  (6334, 28373) 0.013073956875861993
  (6334, 42528) 0.07484524651108244
  (6334, 30160) 0.09571062532218513
  (6334, 42775) 0.0426287986677159
  (6334, 41252) 0.05244809600061579
  (6334, 31743) 0.07940432731188607
  (6334, 59827) 0.2688015008211872
  (6334, 5737)  0.03216756143121747
```

In [59]:

```
X.shape
```

Out[59]:

```
(6335, 67659)
```

In [60]:

```
#Splitting the training and test data
```

In [61]:

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

In [35]:

```
print(X.shape,X_train.shape, Y.shape, Y_train.shape)
```

```
(6335, 67659) (5068, 67659) (6335,) (5068,)
```

In [36]:

```
#Model Training
model = LogisticRegression()
model.fit(X_train, Y_train)
```

Out[36]:

```
LogisticRegression()
```

In [62]:

```
X_new = X_test[0]
prediction = model.predict(X_new)
print(prediction)

if (prediction[0] == 0):
  print("The news is real")
else:
  print("The news is fake")
```

```
['FAKE']
The news is fake
```

In [63]:

```
#Acc score on training data
train_prediction = model.predict(X_train)
train_acc = accuracy_score(train_prediction, Y_train)
print("Training data accuracy:", train_acc)
```

```
Training data accuracy: 0.9516574585635359
```

In [64]:

```
ac=accuracy_score(train_prediction,Y_train)
```

In [ ]:

```
print(ac*100)
```

In [2]:

```
95.16424553
```

In [ ]:

```

```