FAKE NEWS DETECTION

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import nltk
import re
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [2]:

```
data=pd.read_csv('/home/silpa/Downloads/data (2).csv')
data.head()
```

Out[2]:

	URLs	Headline	Body	Label
0	http://www.bbc.com/news/world-us-canada-414191...	Four ways Bob Corker skewered Donald Trump	Image copyright Getty Images\nOn Sunday mornin...	1
1	https://www.reuters.com/article/us-filmfestiva...	Linklater's war veteran comedy speaks to moder...	LONDON (Reuters) - “Last Flag Flying”, a comed...	1
2	https://www.nytimes.com/2017/10/09/us/politics...	Trump’s Fight With Corker Jeopardizes His Legi...	The feud broke into public view last week when...	1
3	https://www.reuters.com/article/us-mexico-oil-...	Egypt's Cheiron wins tie-up with Pemex for Mex...	MEXICO CITY (Reuters) - Egypt’s Cheiron Holdin...	1
4	http://www.cnn.com/videos/cnnmoney/2017/10/08/...	Jason Aldean opens 'SNL' with Vegas tribute	Country singer Jason Aldean, who was performin...	1

In [3]:

```
data.shape  # no. of rows and columns in the dataset
```

Out[3]:

```
(4009, 4)
```

In [4]:

```
data[data['Label']==1].shape  #  no. of rows and columns with real news
```

Out[4]:

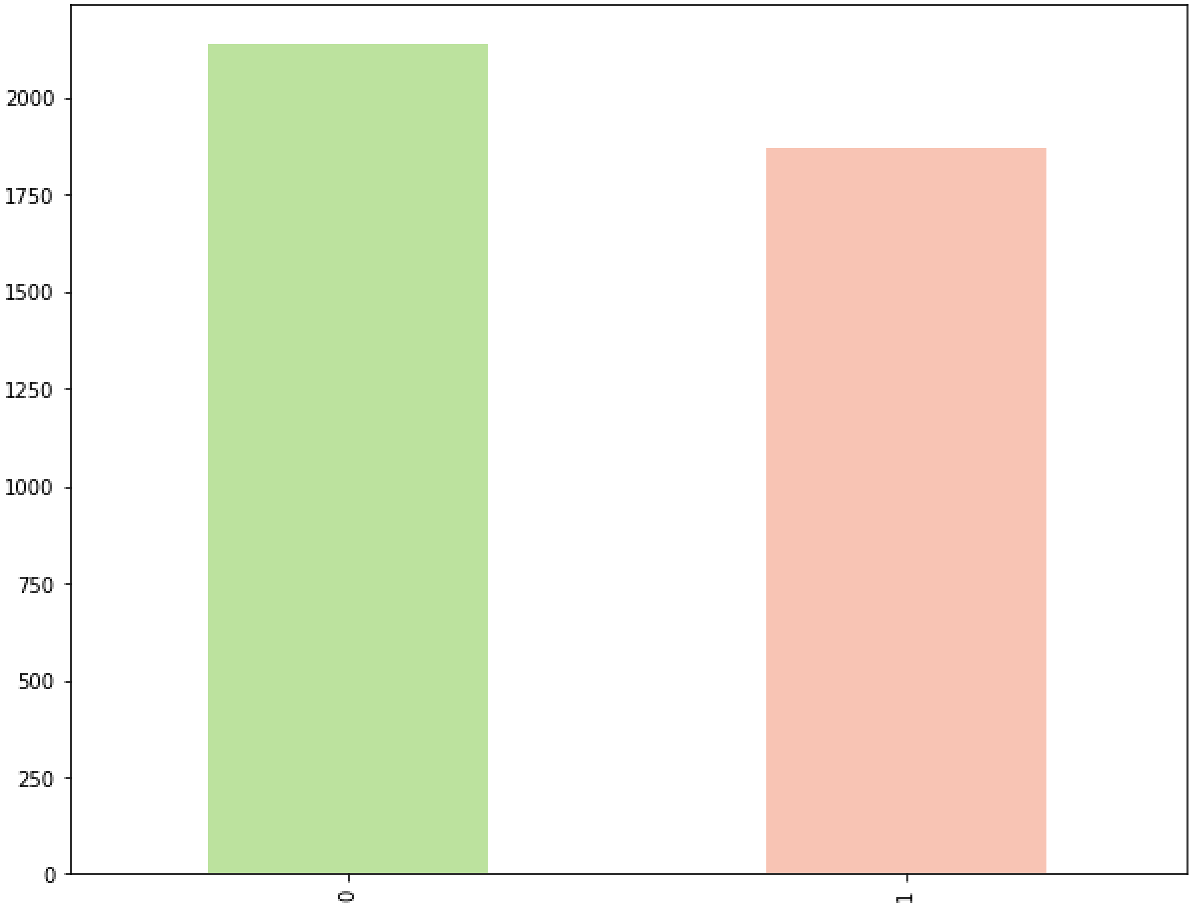
```
(1872, 4)
```

In [5]:

```
# count of fake and real news
clr=['#BCE29E','#F8C4B4']
data['Label'].value_counts().plot(kind='bar',figsize=(10,8),color=clr)
```

Out[5]:

```
<AxesSubplot:>
```



```
In [6]: data.columns   # column names

Out[6]: Index(['URLs', 'Headline', 'Body', 'Label'], dtype='object')
```

```
In [7]: data.isna().sum()   # checking null values

Out[7]: URLs         0
        Headline     0
        Body        21
        Label        0
        dtype: int64
```

```
In [8]: df = data.copy() #Creating a copy of data
```

```
In [9]: df.dropna(inplace=True)   # dropping the null values
```

```
In [10]: df.isna().sum()

Out[10]: URLs        0
         Headline    0
         Body        0
         Label       0
         dtype: int64
```

```
In [11]: df.drop(['URLs','Headline'],axis=1,inplace=True)   # dropping the unwanted columns
         df
```

Out[11]:

	Body	Label
0	Image copyright Getty Images\nOn Sunday mornin...	1
1	LONDON (Reuters) - “Last Flag Flying”, a comed...	1
2	The feud broke into public view last week when...	1
3	MEXICO CITY (Reuters) - Egypt’s Cheiron Holdin...	1

4	Country singer Jason Aldean, who was performin...	1
...	...	...
4003	Vietnam Is in Great Danger, You Must Publish a...	0
4004	Trends to Watch\n% of readers think this story...	0
4005	Trump Jr. Is Soon To Give A 30-Minute Speech F...	0
4007	SHANGHAI (Reuters) - China said it plans to ac...	1
4008	Vice President Mike Pence Leaves NFL Game Beca...	0

3988 rows × 2 columns

In [12]:
```
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to /home/silpa/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /home/silpa/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[12]:
```
True
```

In [13]:
```
body=df['Body']
```

In [14]:
```
sn=SnowballStemmer('english')    # snowball stemmer
st=stopwords.words('english')
def fns(news):
    news=news.apply(lambda x:word_tokenize(x)).apply(lambda x:" ".join(x)) #tokenization
    news=news.str.replace('[^a-zA-Z0-9]+',' ') #removing special characters
    #stemming
    news=news.apply(lambda x:[sn.stem(i.lower()) for i in word_tokenize(x)]).apply(lambd
    #removing stopwords
    news=news.apply(lambda x:[i for i in word_tokenize(x) if i not in st]).apply(lambda
    return news
```

In [15]:
```
body=fns(body)  # applying def funcion
print(body)
```

```
0       imag copyright getti imag sunday morn donald t...
1       london reuter last flag fli comedi drama vietn...
2       feud broke public view last week mr corker sai...
3       mexico citi reuter egypt cheiron hold limit ri...
4       countri singer jason aldean perform las vega s...
                              ...
4003    vietnam great danger must publish tell armi go...
4004    trend watch reader think stori fact add two ce...
4005    trump jr soon give 30 minut speech 100 000 rea...
4007    shanghai reuter china said plan accept data ov...
4008    vice presid mike penc leav nfl game becaus ant...
Name: Body, Length: 3988, dtype: object
```

In [16]:
```
vec=TfidfVectorizer() #vectorization
train_data_vec=vec.fit_transform(body)
print(train_data_vec)
```

```
  (0, 14924)    0.03993181579517344
  (0, 29498)    0.04681102741277237
  (0, 18518)    0.037615085490131116
  (0, 6097)     0.020672226937044794
  (0, 26973)    0.020791267908858387
  (0, 26793)    0.02014731617513696
  (0, 15481)    0.023577886967839596
  (0, 22235)	0.02554630246651472
  (0, 16634)	0.020310837159359413
  (0, 2911)	0.028495839483776632
  (0, 21876)	0.023138150051537264
  (0, 1883)	0.02201635444950773
  (0, 29622)	0.0438922924106594
  (0, 9377)	0.035692563892593836
  (0, 4050)	0.021206997394998157
  (0, 17895)	0.022091372069417997
  (0, 31145)	0.0276371630473814
  (0, 30906)	0.02184505117046639
  (0, 8886)	0.03146628307384299
  (0, 5252)	0.022401697740226097
  (0, 10353)	0.01275127370607288
  (0, 17082)	0.016378074874552378
  (0, 20557)	0.016249876164040666
  (0, 7309)	0.03311002938428662
  (0, 8847)	0.019072830861995106
  :	:
  (3987, 17706)	0.013847798016851752
  (3987, 8024)	0.04368251895056217
  (3987, 17290)	0.015327829775248979
  (3987, 2757)	0.04736510756027361
  (3987, 29265)	0.01237663441784551
  (3987, 26950)	0.11148843369420122
  (3987, 12338)	0.014338909337773402
  (3987, 23297)	0.030862684376152236
  (3987, 30306)	0.2364824389818362
  (3987, 29996)	0.014323333890470257
  (3987, 19356)	0.03427643987284211
  (3987, 10370)	0.04847494196528498
  (3987, 29239)	0.08020992798440885
  (3987, 23872)	0.014862881265121582
  (3987, 599)	0.02738027673908934
  (3987, 30999)	0.021841259475281084
  (3987, 11754)	0.022933779709462664
  (3987, 22231)	0.04092559131049021
  (3987, 24740)	0.01182430004271673
  (3987, 22571)	0.1870643485976439
  (3987, 28548)	0.012364978562535892
  (3987, 19923)	0.05130727789194103
  (3987, 29264)	0.01856901366284994
  (3987, 29081)	0.05638877241973
  (3987, 27514)	0.02065844103816319
```

In [17]:

```
y=df['Label'].values
y
```

Out[17]:

```
array([1, 1, 1, ..., 0, 1, 0])
```

In [18]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(train_data_vec,y,test_size=0.3,random_sta
```

In [19]:

```
from sklearn.svm import SVC  # using support vector machine
classifier=SVC()
classifier.fit(x_train,y_train)
y_pred=classifier.predict(x_test)
print(y_pred)
```

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

In [20]:

```
# checking accuracy of the model
from sklearn.metrics import accuracy_score,confusion_matrix
score=accuracy_score(y_test,y_pred)
print("Accuracy of SVM model:", score)
```

```
Accuracy of SVM model: 0.9791144527986633
```

In [21]:

```
#creating a function for detecing news

def fake_news_detection(news):
    input_news = {"text":[news]}
    new_test_in = pd.DataFrame(input_news)   # to data frame
    new_test_in["text"] = fns(new_test_in["text"]) # applying to the already created def
    new_x_test = new_test_in["text"]
    vectorized_data = vec.transform(new_x_test)  # vectorization
    prediction = classifier.predict(vectorized_data)   # prediction

    if prediction == 1:
        print("Real News")
    else:
        print("Fake News")
```

In [22]:

```
fake_news_detection("""The second Covid-19 wave in India is now on the "downswing," the
```

```
Real News
```

In [23]:

```
fake_news_detection("JetNation FanDuel League; Week 4 of readers think this story is Fac
```

```
Fake News
```