COVID FAKE NEWS DETECTION

In [3]:

```
# import libraries

from nltk.stem.porter import PorterStemmer

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

import pickle

from sklearn.linear_model import LogisticRegressionCV

import re

import pandas as pd

import warnings

warnings.filterwarnings("ignore")
```

In [4]:

```
df = pd.read_csv(r"C:\Users\Vaish\Desktop\NLP(AD)\covid_fake.csv")
```

In [5]:

```
df.head()
```

Out[5]:

	title	text	source	label
0	Due to the recent outbreak for the Coronavirus...	You just need to add water, and the drugs and ...	coronavirusmedicalkit.com	Fake
1	NaN	Hydroxychloroquine has been shown to have a 10...	RudyGiuliani	Fake
2	NaN	Fact: Hydroxychloroquine has been shown to hav...	CharlieKirk	Fake
3	NaN	The Corona virus is a man made virus created i...	JoanneWrightForCongress	Fake
4	NaN	Doesn’t @BillGates finance research at the Wuh...	JoanneWrightForCongress	Fake

In [6]:

```
df.shape
```

Out[6]:

```
(1164, 4)
```

In [7]:

```
df['label'].value_counts()
```

```
Out[7]: label
        TRUE     584
        Fake     345
        fake     230
        Name: count, dtype: int64
```

```
In [8]: df.loc[5:15]
```

Out[8]:

	title	text	source	label
5	CORONA UNMASKED: Chinese Intelligence Officer ...	NaN	NaN	NaN
6	NaN	Urgent: Health Bulletin to the Public. Ministr...	Ministry of Health	Fake
7	NaN	Pls tell ur families, relatives and friendsMOH...	NWLLAB	Fake
8	NaN	SERIOUS EXCELLENT ADVICE by Japanese doctors t...	Japanese doctors treating COVID-19 cases	Fake
9	Basic protective measures against the new coro...	Stay aware of the latest information on the CO...	https://www.who.int/emergencies/diseases/novel...	TRUE
10	NaN	The new Coronavirus may not show signs of infe...	Taiwan Experts	Fake
11	NaN	A vaccine meant for cattle can be used to figh...	facebook	Fake
12	NaN	Using a hair dryer to breathe in hot air can c...	Youtube	Fake
13	NaN	Corona virus before it reaches the lungs it re...	twitter	Fake
14	Exposing yourself to the sun or to temperature...	You can catch COVID-19, no matter how sunny or...	https://www.who.int/emergencies/diseases/novel...	TRUE
15	You can recover from	Most of the people who	https://www.who.int/emergencies/diseases/novel...	NaN

title	text	source	label
the coronavirus disease (...	catch COVID-19 can reco...		

In [9]:

```
df.isna().sum()
```

Out[9]:

```
title     82
text      10
source    20
label      5
dtype: int64
```

In [10]:

```
df.loc[df['label'] == 'Fake', ['label']] = 'FAKE'

df.loc[df['label'] == 'fake', ['label']] = 'FAKE'

df.loc[df['source'] == 'facebook', ['source']] = 'Facebook'

df.text.fillna(df.title, inplace=True)

df.loc[5]['label'] = 'FAKE'

df.loc[15]['label'] = 'TRUE'

df.loc[43]['label'] = 'FAKE'

df.loc[131]['label'] = 'TRUE'

df.loc[242]['label'] = 'FAKE'

#df = df.sample(frac=1).reset_index(drop=True)

df.title.fillna('missing', inplace=True)

df.source.fillna('missing', inplace=True)

df['title_text'] = df['title'] + ' ' + df['text']
```

In [11]:

```
# checking for missing values again

df.isna().sum()
```

Out[11]:

```
title         0
text          0
source        0
label         0
title_text    0
dtype: int64
```

In [12]:

```
df['label'].value_counts()
```

Out[12]:

```
label
TRUE     586
FAKE     578
Name: count, dtype: int64
```

In [13]:

```
df.head()
```

Out[13]:

	title	text	source	label	title_text
0	Due to the recent outbreak for the Coronavirus...	You just need to add water, and the drugs and ...	coronavirusmedicalkit.com	FAKE	Due to the recent outbreak for the Coronavirus...
1	missing	Hydroxychloroquine has been shown to have a 10...	RudyGiuliani	FAKE	missing Hydroxychloroquine has been shown to h...
2	missing	Fact: Hydroxychloroquine has been shown to hav...	CharlieKirk	FAKE	missing Fact: Hydroxychloroquine has been show...
3	missing	The Corona virus is a man made virus created i...	JoanneWrightForCongress	FAKE	missing The Corona virus is a man made virus c...
4	missing	Doesn’t @BillGates finance research at the Wuh...	JoanneWrightForCongress	FAKE	missing Doesn’t @BillGates finance research at...

In [14]:

```
df.shape
```

Out[14]:

```
(1164, 5)
```

In [15]:

```
df['title_text'][3]
```

Out[15]:

```
'missing The Corona virus is a man made virus created in a Wuhan laboratory. Ask @
BillGates who financed it.'
```

In [16]:

```
def preprocessor(text):
    text = re.sub('<[^>]*>', '', text)
    text = re.sub(r'[^\w\s]','', text)
    text = re.sub(r'[\n]', '', text)
    text = text.lower()
    return text
df['title_text'] = df['title_text'].apply(preprocessor)
df['title_text'][3]
```

Out[16]:

```
'missing the corona virus is a man made virus created in a wuhan laboratory ask bi
llgates who financed it'
```

In [17]:

```
porter = PorterStemmer()
```

```
def tokenizer_porter(text):

    return [porter.stem(word) for word in text.split()]
```

In [18]:
```
tfidf = TfidfVectorizer(strip_accents=None,

                        lowercase=False,

                        preprocessor=None,

                        tokenizer=tokenizer_porter,

                        use_idf=True,

                        norm='l2',

                        smooth_idf=True)

X = tfidf.fit_transform(df['title_text'])

y = df.label.values
```

In [19]:
```
X.shape
```

Out[19]:
```
(1164, 27020)
```

In [21]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size
```

In [35]:
```
clf = LogisticRegressionCV(cv=5, scoring='accuracy', random_state=0, n_jobs=-1,verb
clf.fit(X_train, y_train)
fake_news_model = open('fake_news_model.sav', 'wb')
pickle.dump(clf, fake_news_model)
fake_news_model.close()
```

Model Evaluation

In [38]:
```
filename = 'fake_news_model.sav'
saved_clf = pickle.load(open(filename, 'rb'))
saved_clf.score(X_test, y_test)
```

Out[38]:
```
0.9314285714285714
```

In [39]:
```
from sklearn.metrics import classification_report, accuracy_score
y_pred = clf.predict(X_test)
print("---Test Set Results---")
print(classification_report(y_test, y_pred))
```

```
---Test Set Results---
              precision    recall  f1-score   support

        FAKE       0.92      0.89      0.91       132
        TRUE       0.94      0.95      0.95       218

    accuracy                           0.93       350
   macro avg       0.93      0.92      0.93       350
weighted avg       0.93      0.93      0.93       350
```

In [40]:

```
# sample prediction

clf.predict(X_test[59])
```

Out[40]:

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

In [48]:

```
test = "Corona virus before it reaches the lungs"

inp = [test]

vect = tfidf.transform(inp)
prediction = clf.predict(vect)
print(prediction)
```

```
['FAKE']
```

In [ ]: