Model for classification of News as REAL or FAKE

To build a model to accurately classify a piece of news as REAL or FAKE. Using sklearn, build a TfidfVectorizer on the provided dataset. Then, initialize a PassiveAggressive Classifier and fit the model. In the end, the accuracy score and the confusion matrix tell us how well our model fares.

In [1]:
```
# import the neccessary modules
import numpy as np
import pandas as pd
import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
```

In [2]:
```
#Read the data
data=pd.read_csv('news.csv')
#Get shape and head
print(data.shape)
data.head()
```

```
(6335, 4)
```

Out[2]:

	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 [3]:
```
#DataFlair - Get the labels
labels=data.label
labels.head()
```

Out[3]:
```
0    FAKE
1    FAKE
2    REAL
3    FAKE
4    REAL
Name: label, dtype: object
```

In [4]:
```
#DataFlair - Split the dataset
x_train,x_test,y_train,y_test=train_test_split(data['text'], labels, test_size=0.2, random_s
tate=7)
```

In [5]:
```
#DataFlair - Initialize a TfidfVectorizer
vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)
#DataFlair - Fit and transform train set, transform test set
tfidf_train=vectorizer.fit_transform(x_train)
tfidf_test=vectorizer.transform(x_test)
```

In [6]:
```
#DataFlair - Initialize a PassiveAggressiveClassifier
pac=PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)
#DataFlair - Predict on the test set and calculate accuracy
y_pred=pac.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')


#print classification report

print(classification_report(y_test,y_pred))
pd.crosstab(y_test,y_pred)
```

```
Accuracy: 92.9%
              precision    recall  f1-score   support

        FAKE       0.93      0.92      0.93       638
        REAL       0.92      0.93      0.93       629

    accuracy                           0.93      1267
   macro avg       0.93      0.93      0.93      1267
weighted avg       0.93      0.93      0.93      1267
```

Out[6]:

col_0	FAKE	REAL
label		
FAKE	590	48
REAL	42	587

In [7]:
```
#DataFlair - Build confusion matrix
confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
```

Out[7]:
```
array([[590,  48],
       [ 42, 587]], dtype=int64)
```

In [ ]:
```

```