In [1]:

```
pip install numpy pandas sklearn
```

```
Requirement already satisfied: numpy in ./opt/anaconda3/lib/python3.9/site-packages (1.2
1.5)
Requirement already satisfied: pandas in ./opt/anaconda3/lib/python3.9/site-packages (1.
4.4)
Requirement already satisfied: sklearn in ./opt/anaconda3/lib/python3.9/site-packages
(0.0.post1)
Requirement already satisfied: python-dateutil>=2.8.1 in ./opt/anaconda3/lib/python3.9/s
ite-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in ./opt/anaconda3/lib/python3.9/site-packag
es (from pandas) (2022.7)
Requirement already satisfied: six>=1.5 in ./opt/anaconda3/lib/python3.9/site-packages
(from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
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
```

In [9]:

```
#Read the data  into a DataFrame
df=pd.read_csv('/Users/jalilkhan/Downloads/news.csv')

#Get shape and head
df.shape
df.head()
```

Out[9]:

	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 [10]:

```
#DataFlair - Get the labels

labels=df.label
labels.head()
```

Out[10]:

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

In [11]:

```
#DataFlair - Split the dataset

x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2, random
```

initialize a TfidfVectorizer with stop words from the English language and a maximum document frequency of 0.7 -- terms with a higher document frequency will be discarded.

Stop words are to be filtered out before processing the natural language data.

A TfidfVectorizer turns a collection of raw documents into a matrix of TF-IDF features.

In [12]:

```
#DataFlair - Initialize a TfidfVectorizer
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)

#DataFlair - Fit and transform train set, transform test set
tfidf_train=tfidf_vectorizer.fit_transform(x_train)
tfidf_test=tfidf_vectorizer.transform(x_test)
```

In [13]:

```
#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)}%')
```

```
Accuracy: 92.74%
```

In [14]:

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

Out[14]:

```
array([[591,  47],
       [ 45, 584]])
```

So with this model, we have 589 true positives, 587 true negatives, 42 false positives, and 49 false negatives.

what we did here is we took a political dataset, implemented a TfidfVectorizer, initialized a PassiveAggressiveClassifier, and fit our model. We ended up obtaining an accuracy of 92.82% in magnitude.

In [ ]:

```

```