In [1]:

```
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 [2]:

```
#Read the data
df=pd.read_csv('news.csv')
df.shape
df.head()
```

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

Out[3]:

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

In [4]:

```
#Split the dataset into train and test
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2, random_state=7)
```

In [5]:

```
print("Shape of xtrain is",x_train.shape)
```

```
Shape of xtrain is (5068,)
```

In [6]:

```
print("Shape of ytrain is",y_train.shape)
print("Shape of xtest is",x_test.shape)
print("Shape of ytest is",y_test.shape)
```

```
Shape of ytrain is (5068,)
Shape of xtest is (1267,)
Shape of ytest is (1267,)
```

TF-IDF vectorizer is a sklearn implementation helps in finding the TF-IDF values. This TF-IDF values signifies the importance of words in the given sentence. Stop_words are the english words which does not add much meaning to the sentence.

In [7]:

```
#Initialize a TfidfVectorizer from sklearn
vectorizer=TfidfVectorizer(stop_words='english')

#fit and transform train set, transform test set
tfidf_train = vectorizer.fit_transform(x_train)
tfidf_test = vectorizer.transform(x_test)
```

PassiveAggressiveClassifier is one of the incremental learning models. If the prediction is correct, model is kept as it is. But if prediction is incorrect, make changes to the model, i.e some changes may correct the predictions.

In [8]:

```
#initialize a PassiveAggressiveClassifier
pac=PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)
```

Out[8]:

```
PassiveAggressiveClassifier(max_iter=50)
```

In [9]:

```
y_pred=pac.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 92.42%
```

In [10]:

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

Out[10]:

```
array([[587,  51],
       [ 45, 584]], dtype=int64)
```