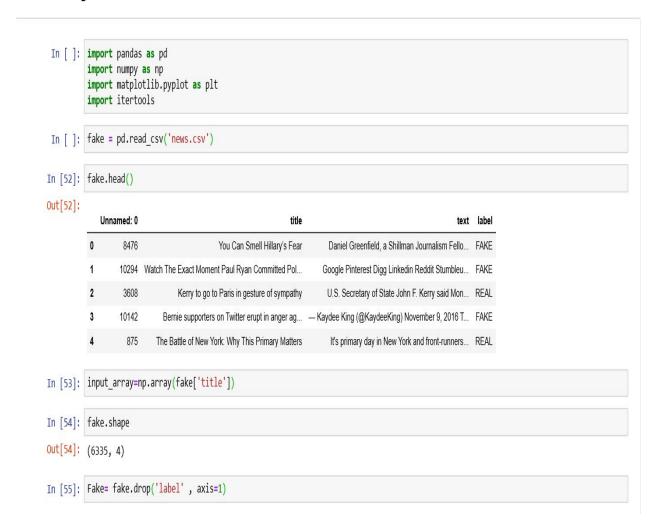
# **REPORT**

Topic: Fake news detection

**Problem Statement:** The main objective is to detect the fake news, build a Machine Learning model to differentiate between "Real" news and "Fake" news.

### **Code implemented:**



```
Fake= fake.drop('label' , axis=1)
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.model selection import train test split
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy score, confusion matrix
from sklearn import metrics
labels=fake.label
labels.head()
0
     FAKE
1
     FAKE
2
     REAL
     FAKE
3
     REAL
Name: label, dtype: object
x train,x test,y train,y test=train test split(fake['text'], labels, test size=0.2, random state=7)
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)
tfidf_train=tfidf_vectorizer.fit_transform(x_train)
tfidf_test=tfidf_vectorizer.transform(x_test)
  In [60]: #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('accuracy is ')
           print(score*100)
           accuracy is
           92.73875295974744
  In [61]: confusion matrix(y test,y pred, labels=['FAKE','REAL'])
  Out[61]: array([[589, 49],
                  [ 43, 586]], dtype=int64)
```

# **Strategy adopted:**

Tfidvectorizer and PassiveAgressiveClassifier is used in making the fake news detection model.

#### Tfidvectorizer:

**TF (Term Frequency):** The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

**IDF** (**Inverse Document Frequency**): Words that occur many times a document, but also occur many times in many others, may be irrelevant. IDF is a measure of how significant a term is in the entire corpus.

The TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features.

The news dataset contains only words and thus we cannot apply algorithms on the dataset. Therefore the dataset had to be converted into a matrix that can be computed on.

## PassiveAgressiveClassifier:

Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

The advantage is that the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. Therefore it is ideal for news dataset where is such a large amount of data.

After converting the dataset to a matrix of TF-IDF features and fitting the data to PassiveAgressiveClassifier object we use accuracy score to find the accuracy. We get an accuracy of 92.73 %. By using confusion matrix we find that our prediction model has 589 true positives, 49 false negatives , 43 false positive and 586 true negatives.