**Fake News Detection**

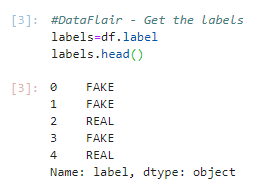
The dataset we’ll use for this python project- we’ll call it news.csv. This dataset has a shape of 7796×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE.

Steps for the project :-

1. You’ll need to install the following libraries with pip.
2. You’ll need to install Jupyter Lab to run your code. Get to your command prompt and run the following command.
3. Make necessary imports.
4. Now, let’s read the data into a DataFrame, and get the shape of the data and the first 5 records.
5. And get the labels from the DataFrame.
6. Split the dataset into training and testing sets.
7. Let’s initialize a **[TfidfVectorizer](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html" \t "_blank)** 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 the most common words in a language that are to be filtered out before processing the natural language data. And a TfidfVectorizer turns a collection of raw documents into a matrix of TF-IDF features.
8. Now, fit and transform the vectorizer on the train set, and transform the vectorizer on the test set.
9. Next, we’ll initialize a PassiveAggressiveClassifier. This is. We’ll fit this on tfidf\_train and y\_train.
10. Then, we’ll predict on the [test set](https://github.com/GeorgeMcIntire/fake_real_news_dataset) from the TfidfVectorizer and calculate the accuracy with accuracy\_score() from sklearn.metrics.
11. Finally print out a confusion matrix to gain insight into the number of false and true negatives and positives.

We took a political dataset, implemented a TfidfVectorizer, initialized a PassiveAggressiveClassifier, and fit our model.

**Output** :-



**Input-:**

