

```
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
```

```
In [2]: # The fake news Dataset
# The dataset we'll use for this python project-we'll call it news.csv. .
# 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
```

```
In [3]: import os
os.chdir(r"C:\Users\HP\Downloads\FakeNews")
#Read the data
df=pd.read_csv('news.csv')

#Get shape and head

print("df shape" , df.shape )

df shape (6335, 4)
```

```
In [4]: df.head()
```

```
Out[4]:
```

	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 [5]: # get the labels from the DataFrame

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

```
Out[5]:
```

0	FAKE
1	FAKE
2	REAL
3	FAKE
4	REAL

Name: label, dtype: object

```
In [6]: #as the news is the data so its the x input and label is the y output
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2,
random_state=7)
```

```
In [7]: # Initialize a TfidfVectorizer with stop words from the
# English language and a maximum document frequency of 0.7
# TfidfVectorizer turns a collection of
# raw documents into a matrix of TF-IDF features
```

```
# Now, fit and transform the vectorizer on the train set,  
# and transform the vectorizer on the test set.  
  
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 [8]: # initialize a PassiveAggressiveClassifier. This is.  
# We'll fit this on tfidf_train and y_train  
  
# Then, we'll predict on the test set from the TfidfVectorizer  
# and calculate the accuracy with accuracy_score() from sklearn.metrics  
  
# Initialize a PassiveAggressiveClassifier  
pac=PassiveAggressiveClassifier(max_iter=50)  
pac.fit(tfidf_train,y_train)  
  
# 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: 92.9%

In []: