

```
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 [6]: #Read the data
df=pd.read_csv(r'C:\Users\sathi\AppData\Local\Temp\Temp1_news.zip\news.csv')
#Get shape and head
df.shape
df.head()
```

Out[6]:

|   | 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 [7]: #DataFlair - Get the labels
labels=df.label
labels.head()
```

Out[7]:

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

```
In [8]: #DataFlair - Split the dataset
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2, random_state=7)
```

```
In [9]: #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 [10]: #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 [12]: #DataFlair - Build confusion matrix
confusion_matrix(y_test,y_pred, labels=['FAKE', 'REAL'])
```

Out[12]:

```
array([[588,  50],
       [ 42, 587]], dtype=int64)
```

```
In [ ]:
```