IMPORTING LIBRARIES

```
import itertools
import numpy as np
import pandas as pd
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
```

+ Code + Text

LOADING THE NEWS DATASET

```
news_df = pd.read_csv('news.csv')
```

```
news_df
```

	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
...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows × 4 columns

```
news_df.head()
```

	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

CHECKING FOR NULL VALUES IN THE DATASET

```
news_df.isnull().sum()
```

```
Unnamed: 0    0
title         0
text          0
label         0
dtype: int64
```

DIMENSIONS OF THE DATASET

```
news_df.shape
```

```
(6335, 4)
```

SPLIT THE NEWS AND THE LABELS

```
labels = news_df.label.copy()
news = news_df.text.copy()
```

INITIALIZE THE tf-idf VECTORIZER

```
tfidf_vec = TfidfVectorizer(stop_words='english', max_df=0.7)
```

FIT THE VECTORIZER TO THE DATA

```
news = tfidf_vec.fit_transform(news)
```

SPLIT THE TRAINING AND THE TESTING DATA

```
x_train, x_test, y_train, y_test = train_test_split(news, labels, test_size=0.2, random_state=42, stratify=labels)
```

INITIALIZE A PASSIVEAGGRESSIVECLASSIFIER

```
pac = PassiveAggressiveClassifier()
```

FIT THE CLASSIFIER TO HE TRAINING DATA

```
pac.fit(x_train, y_train)
```

▾ PassiveAggressiveClassifier
PassiveAggressiveClassifier()

```
PassiveAggressiveClassifier()
```

▾ PassiveAggressiveClassifier
PassiveAggressiveClassifier()

PREDICT THE LABELS OF THE TEST DATASET

```
y_pred = pac.predict(x_test)
```

EVALUATION OR PERFORMANCE METRICE

```
score = accuracy_score(y_test, y_pred)
```

```
score
```

```
0.9431728492501973
```

```
confusion_matrix(y_test, y_pred, labels=['FAKE', 'REAL'])
```

```
array([[600,  33],
       [ 39, 595]])
```