

```
In [1]: import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

from sklearn.linear_model import PassiveAggressiveClassifier
```

```
In [2]: data=pd.read_csv("news.csv")

data.head()
```

Out[2]:

	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 [3]: X = data['text']
y = data['label']
```

```
In [4]: data.shape
```

Out[4]: (6335, 4)

```
In [5]: data.isnull().sum()
```

Out[5]:

Unnamed: 0	0
title	0
text	0
label	0

dtype: int64

```
In [6]: labels=data.label
```

```
In [7]: labels
```

Out[7]:

0	FAKE
1	FAKE
2	REAL
3	FAKE
4	REAL
...	
6330	REAL
6331	FAKE
6332	FAKE
6333	REAL
6334	REAL

Name: label, Length: 6335, dtype: object

```
In [8]: labels.head()
```

Out[8]:

0	FAKE
1	FAKE
2	REAL
3	FAKE
4	REAL

Name: label, dtype: object

```
In [9]: X_train, X_test, y_train, y_test = train_test_split(data['text'],labels, test_size=0.2, random_state=42)
```

```
In [10]: X_train.head()
```

Out[10]:

1142	Donald Trump received a key endorsement for hi...
2654	Nina November 6, 2016 @ 2:39 pm \nPolish gover...
5395	Time: Investigating Hillary is an Attack on Al...
1170	Taki's Magazine October 28, 2016 \nThis electi...
4371	Pakistan Pakistan's cricketer turned politica...

Name: text, dtype: object

```
In [11]: vectorizer = TfidfVectorizer(stop_words='english')
```

```
In [12]: X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

```
In [13]: model = PassiveAggressiveClassifier(max_iter=50)
model.fit(X_train, y_train)
```

Out[13]:

▼ PassiveAggressiveClassifier

PassiveAggressiveClassifier(max_iter=50)

```
In [14]: y_pred = model.predict(X_test)
```

```
In [15]: score=accuracy_score(y_test,y_pred)
```

```
In [19]: accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy = {round(accuracy * 100, 2)}%")

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

Accuracy = 93.76%
Confusion Matrix:
[[592  36]
 [ 43 596]]
```

```
In [24]: import joblib
loaded_model = joblib.load(model_filename)
```

```
In [ ]:
```