DETECTING

FAKE NEWS

IN SOCIAL MEDIA NETWORKS

Truong Minh Hong

Instructor: Nguyen Thanh Tuan

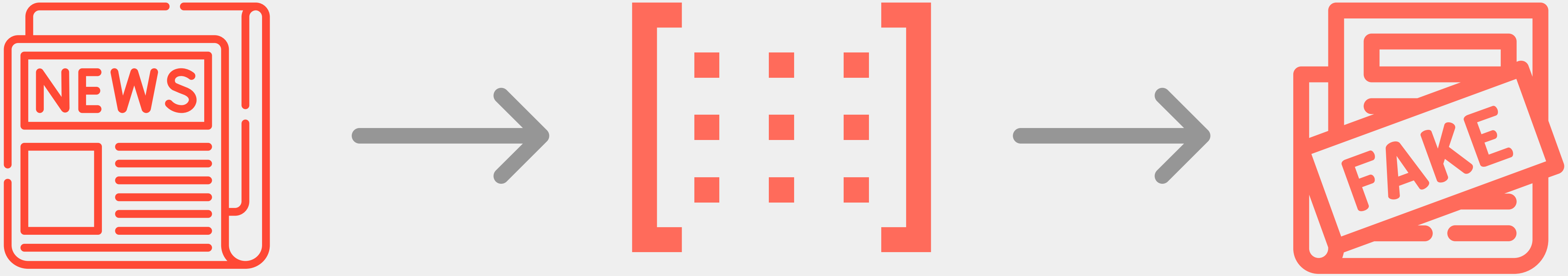
WHAT IS FAKE NEWS ?

Fake news is untrue information presented as news. It often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue.

Rumors about coronavirus are spreading.
Cold weather and snow CAN kill the coronavirus.
The coronavirus CAN be transmitted through mosquito bites.
dryers are
tive in killing
coronavirus
The coron
CANNOT b
in areas wit

1. Define Problem

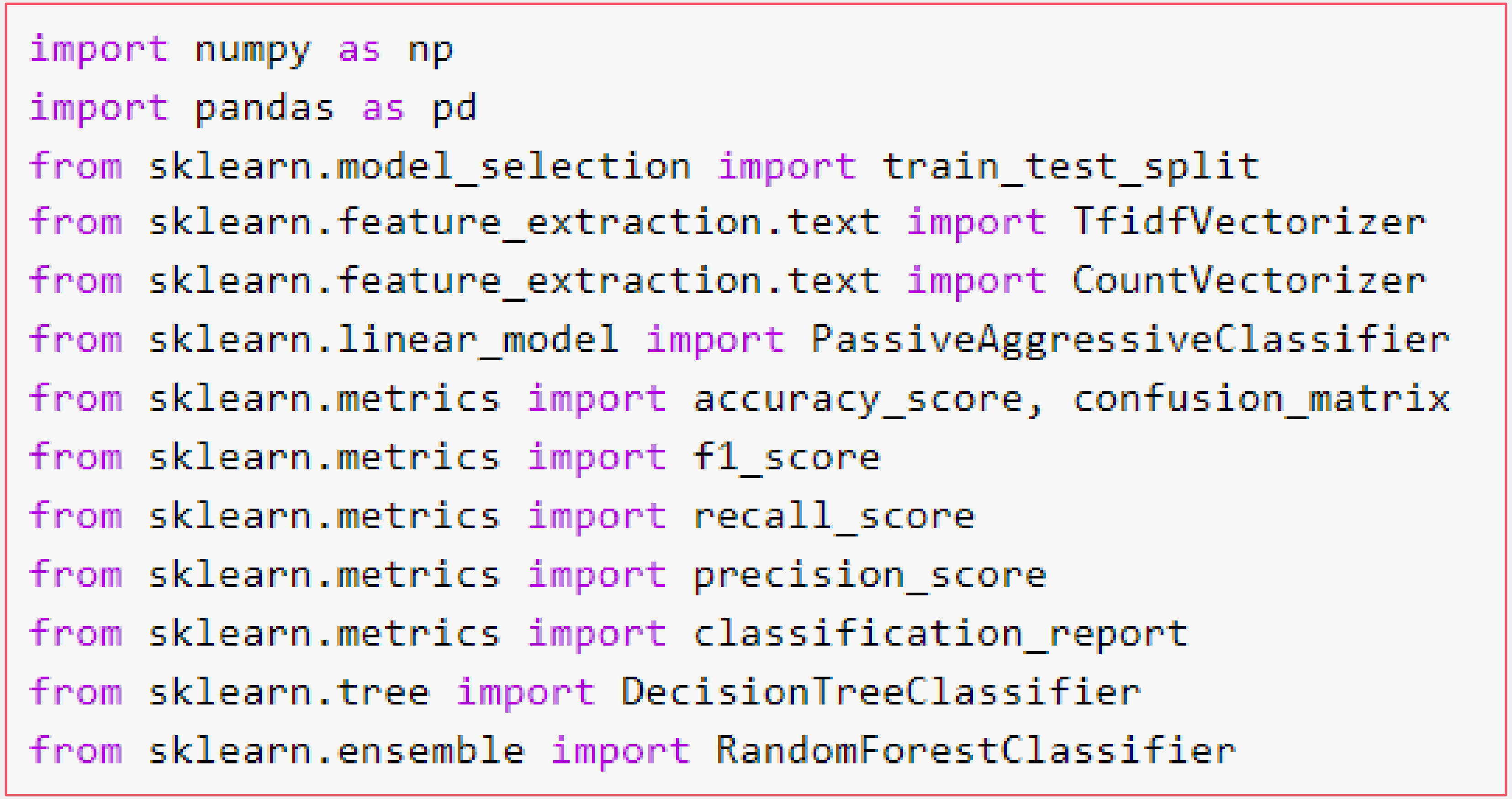
Detect news with the given title and brief content whether it is fake or not by using machine learning.



We will transform dataset to dealable data (Vector) by using TF - IDF Vectorizer and Count Vectorizer.

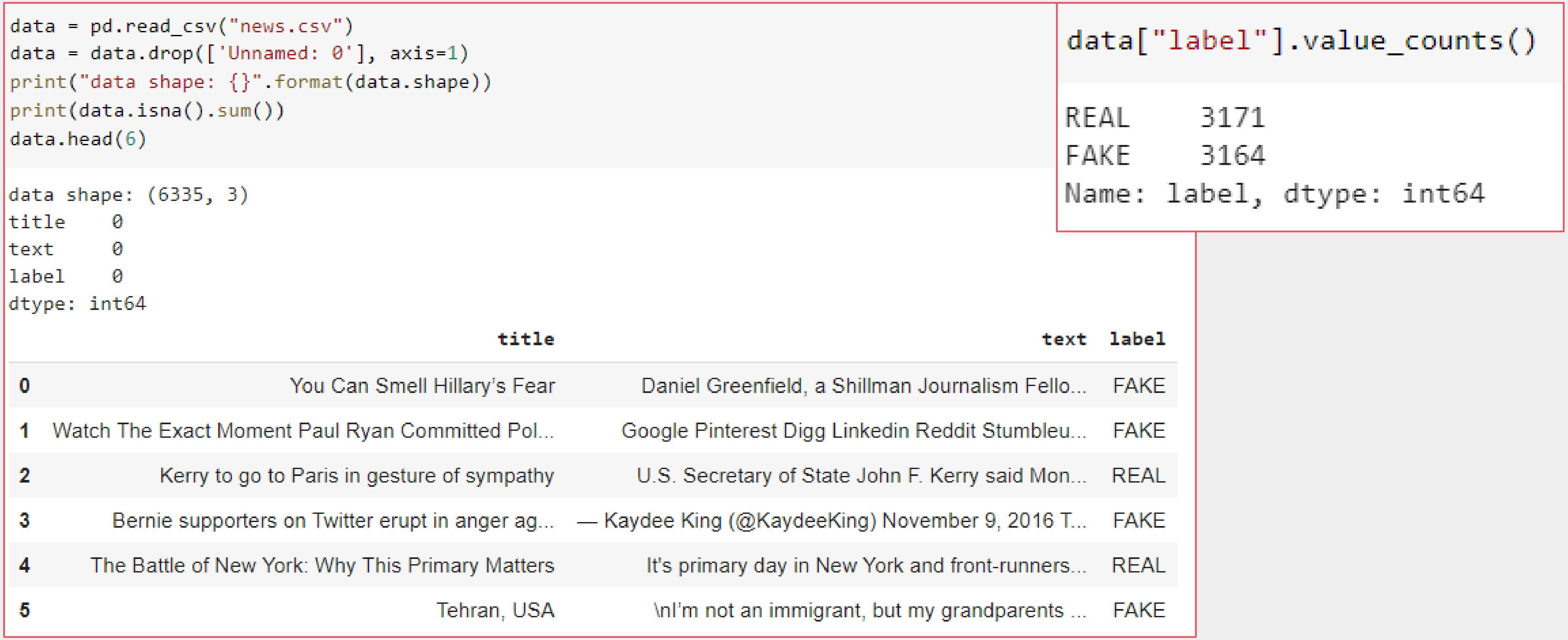
After that we will use Passive Aggressive Classifier algorithm to predict whether it is fake news or not.

2. Import Library & Load Dataset



```
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.feature_extraction.text import CountVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

2. Import Library & Load Dataset



```
data = pd.read_csv("news.csv")
data = data.drop(['Unnamed: 0'], axis=1)
print("data shape: {}".format(data.shape))
print(data.isna().sum())
data.head(6)
```

```
data shape: (6335, 3)
title    0
text     0
label    0
dtype: int64
```

	title	text	label
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
5	Tehran, USA	\nI'm not an immigrant, but my grandparents ...	FAKE

```
data["label"].value_counts()
```

```
REAL    3171
FAKE    3164
Name: label, dtype: int64
```

3. TF - IDF Vectorizer

Term - Frequency (TF):
Measures the frequency of a word in a documents.

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Inverse Document Frequency (IDF):
Measures the rank of the specific word for its relevancy within the text.

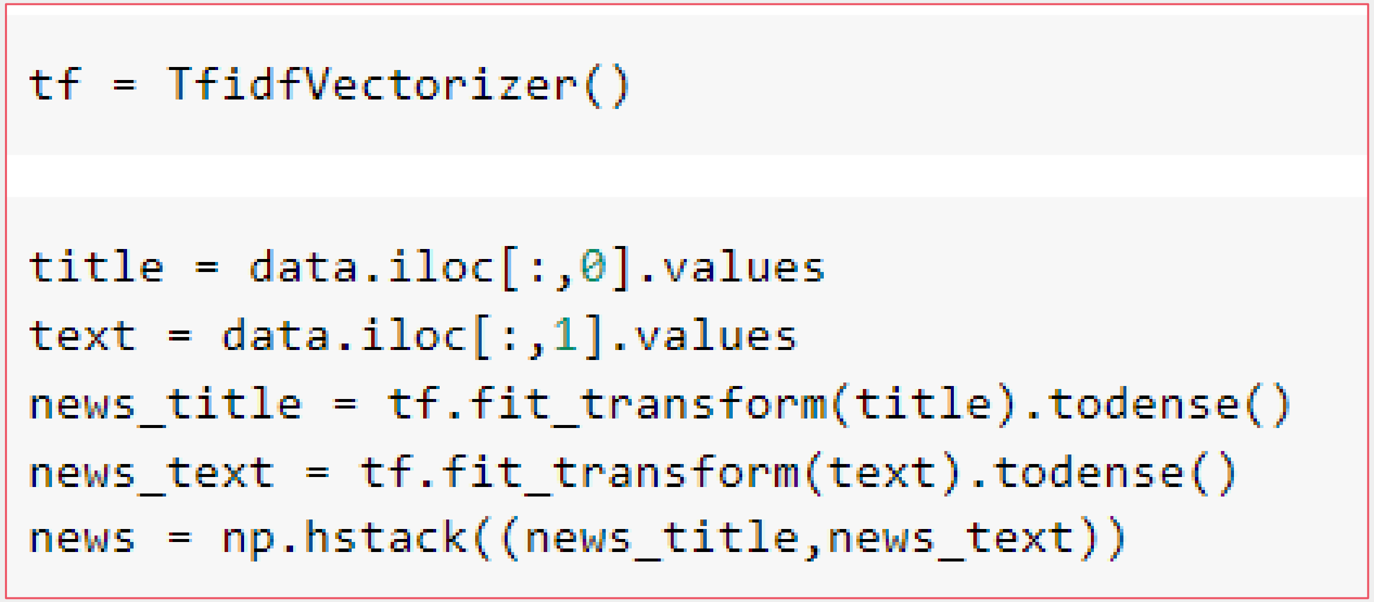
$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

TF - IDF measures how important a word in a given document.

$$\boxed{\text{TF - IDF} = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)}$$

3. TF - IDF Vectorizer

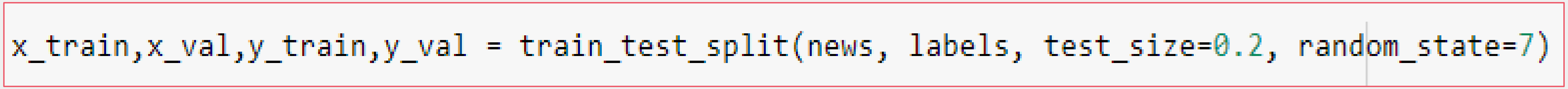
Initialize TfidfVectorizer;
Transform title and text to vector



```
tf = TfidfVectorizer()
```

```
title = data.iloc[:,0].values
text = data.iloc[:,1].values
news_title = tf.fit_transform(title).todense()
news_text = tf.fit_transform(text).todense()
news = np.hstack((news_title,news_text))
```

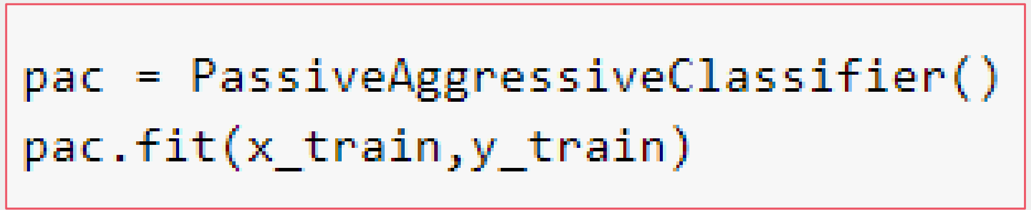
Divde Dataset into training and validation set



```
x_train,x_val,y_train,y_val = train_test_split(news, labels, test_size=0.2, random_state=7)
```

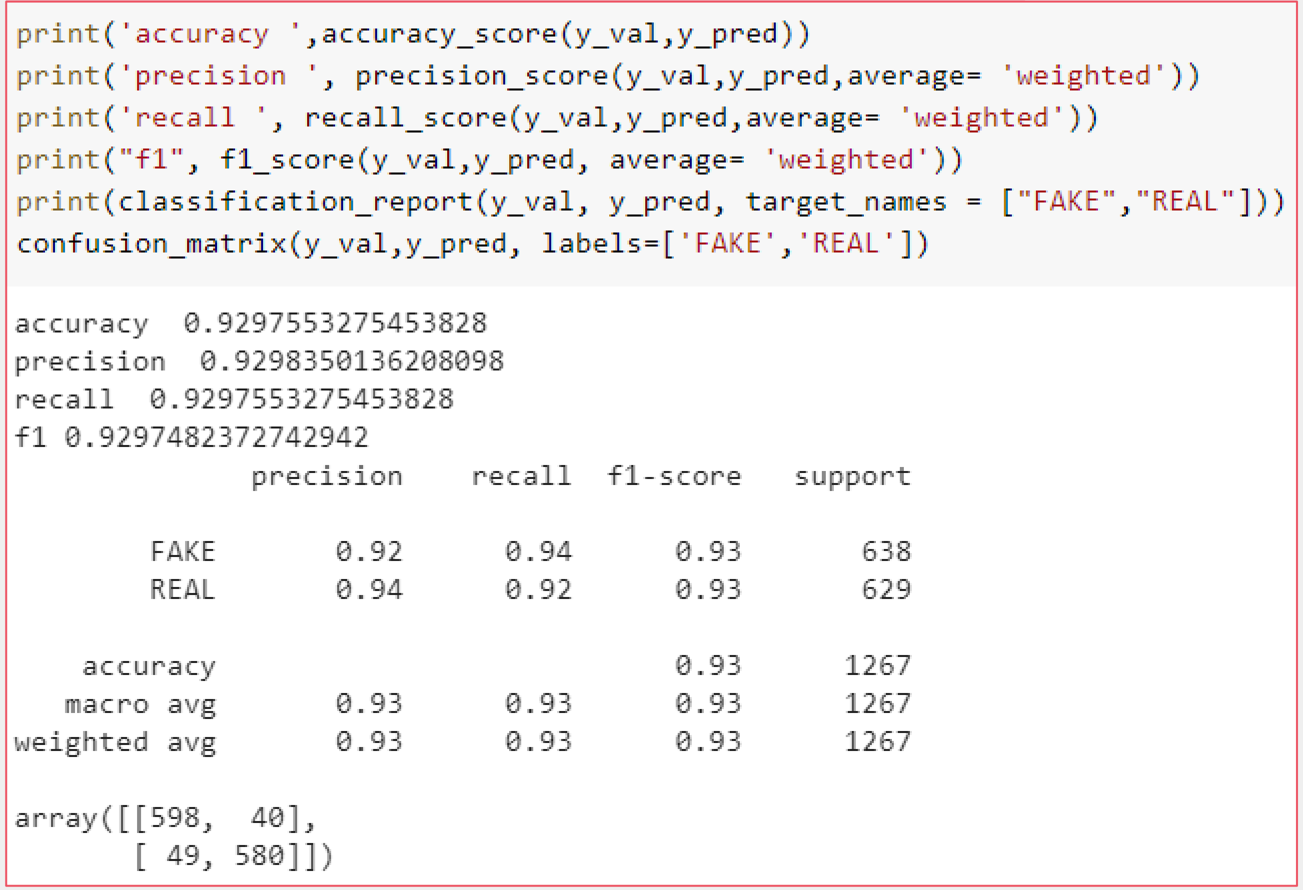
3. TF - IDF Vectorizer

Initialize Passive Agressive Classifier and fit training data



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

Result

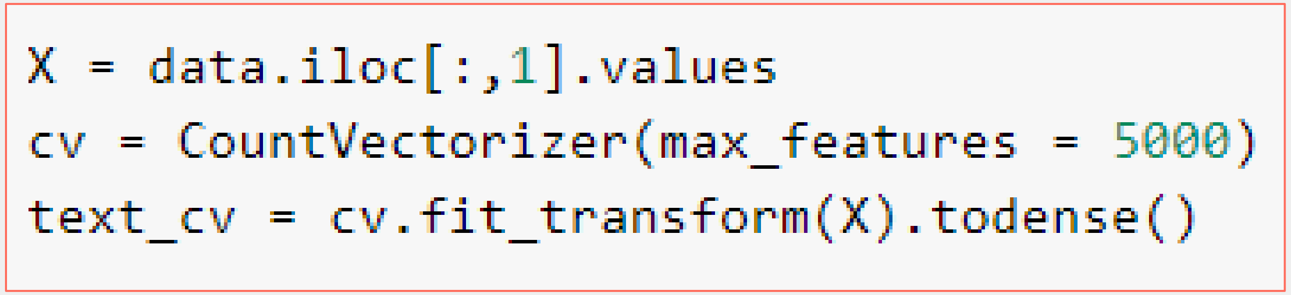


```
print('accuracy ',accuracy_score(y_val,y_pred))
print('precision ', precision_score(y_val,y_pred,average= 'weighted'))
print('recall ', recall_score(y_val,y_pred,average= 'weighted'))
print("f1", f1_score(y_val,y_pred, average= 'weighted'))
print(classification_report(y_val, y_pred, target_names = ["FAKE","REAL"]))
confusion_matrix(y_val,y_pred, labels=['FAKE','REAL'])
```

```
accuracy  0.9297553275453828
precision  0.9298350136208098
recall  0.9297553275453828
f1 0.9297482372742942
              precision    recall  f1-score   support

        FAKE       0.92      0.94      0.93       638
        REAL       0.94      0.92      0.93       629

    accuracy                           0.93      1267
   macro avg       0.93      0.93      0.93      1267
weighted avg       0.93      0.93      0.93      1267

array([[598,  40],
       [ 49, 580]])
```

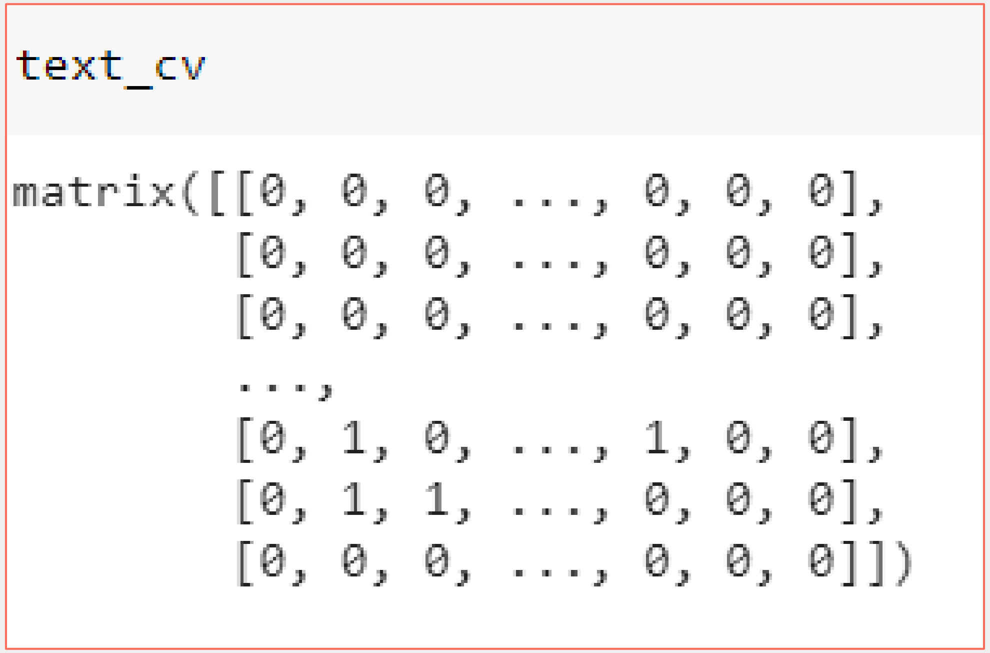
4. CountVectorizer

Transform text to matrix



```
X = data.iloc[:,1].values
cv = CountVectorizer(max_features = 5000)
text_cv = cv.fit_transform(X).todense()
```

Matrix

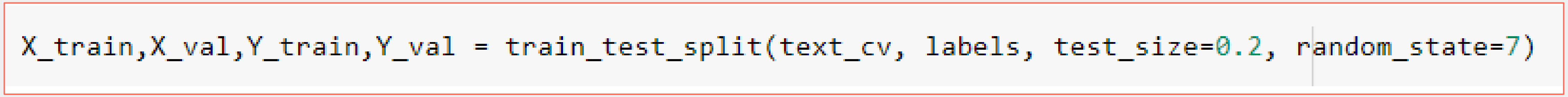


```
text_cv
```

```
matrix([[0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        ...,
        [0, 1, 0, ..., 1, 0, 0],
        [0, 1, 1, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]])
```

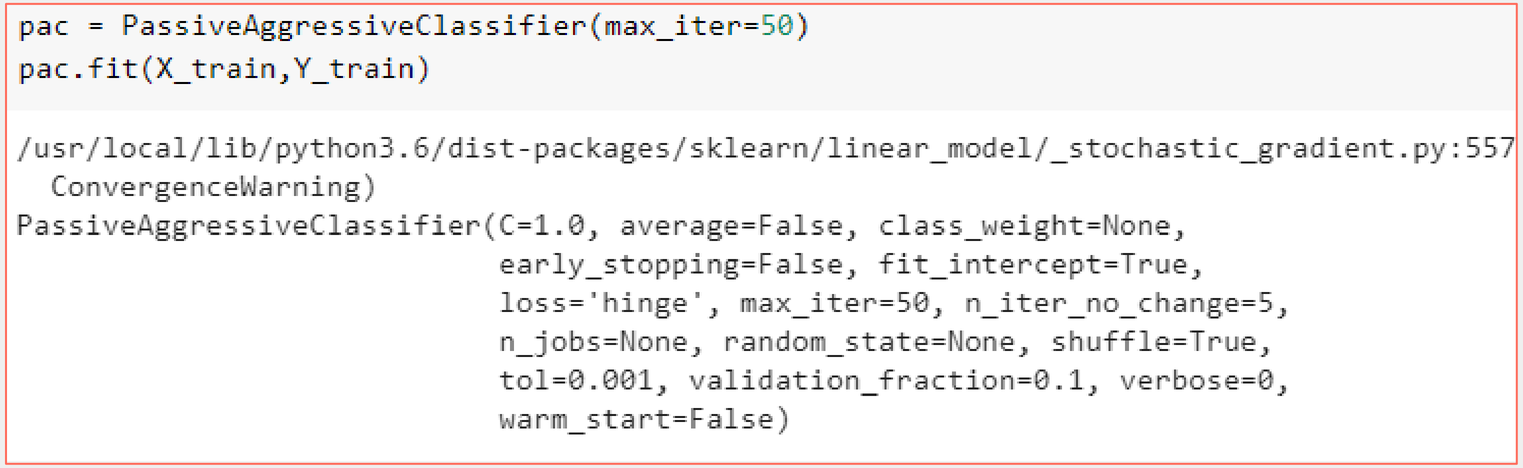
4. CountVectorizer

Spilt dataset into training and validation set



```
X_train,X_val,Y_train,Y_val = train_test_split(text_cv, labels, test_size=0.2, random_state=7)
```

Initialize PAC and fit training set

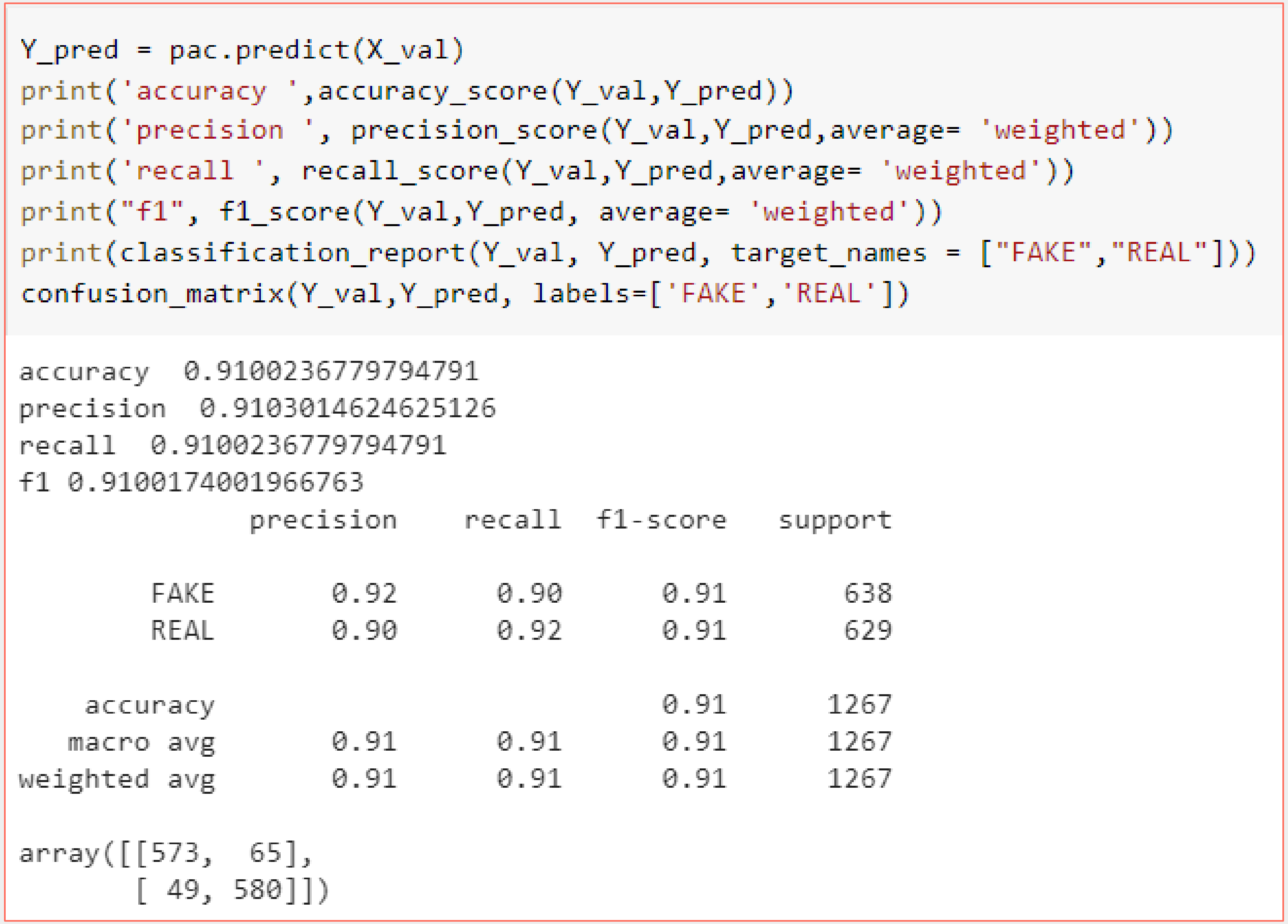


```
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(X_train,Y_train)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_stochastic_gradient.py:557
  ConvergenceWarning)
PassiveAggressiveClassifier(C=1.0, average=False, class_weight=None,
                            early_stopping=False, fit_intercept=True,
                            loss='hinge', max_iter=50, n_iter_no_change=5,
                            n_jobs=None, random_state=None, shuffle=True,
                            tol=0.001, validation_fraction=0.1, verbose=0,
                            warm_start=False)
```

4. CountVectorizer

Result

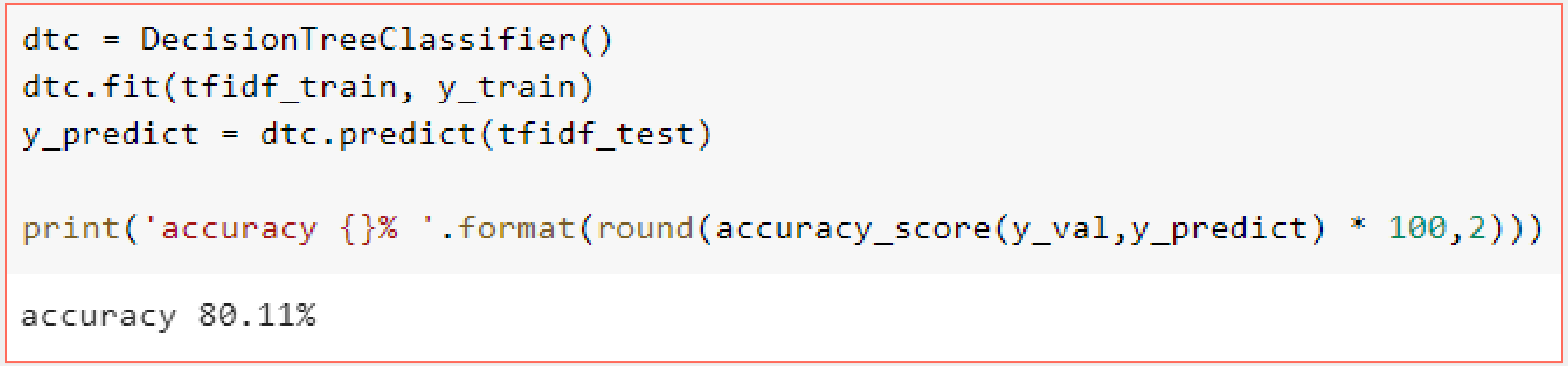


```
Y_pred = pac.predict(X_val)
print('accuracy ',accuracy_score(Y_val,Y_pred))
print('precision ', precision_score(Y_val,Y_pred,average= 'weighted'))
print('recall ', recall_score(Y_val,Y_pred,average= 'weighted'))
print("f1", f1_score(Y_val,Y_pred, average= 'weighted'))
print(classification_report(Y_val, Y_pred, target_names = ["FAKE","REAL"]))
confusion_matrix(Y_val,Y_pred, labels=['FAKE','REAL'])
```

```
accuracy  0.9100236779794791
precision  0.9103014624625126
recall  0.9100236779794791
f1 0.9100174001966763
              precision    recall  f1-score   support

        FAKE       0.92      0.90      0.91       638
        REAL       0.90      0.92      0.91       629

    accuracy                           0.91      1267
   macro avg       0.91      0.91      0.91      1267
weighted avg       0.91      0.91      0.91      1267

array([[573,  65],
       [ 49, 580]])
```

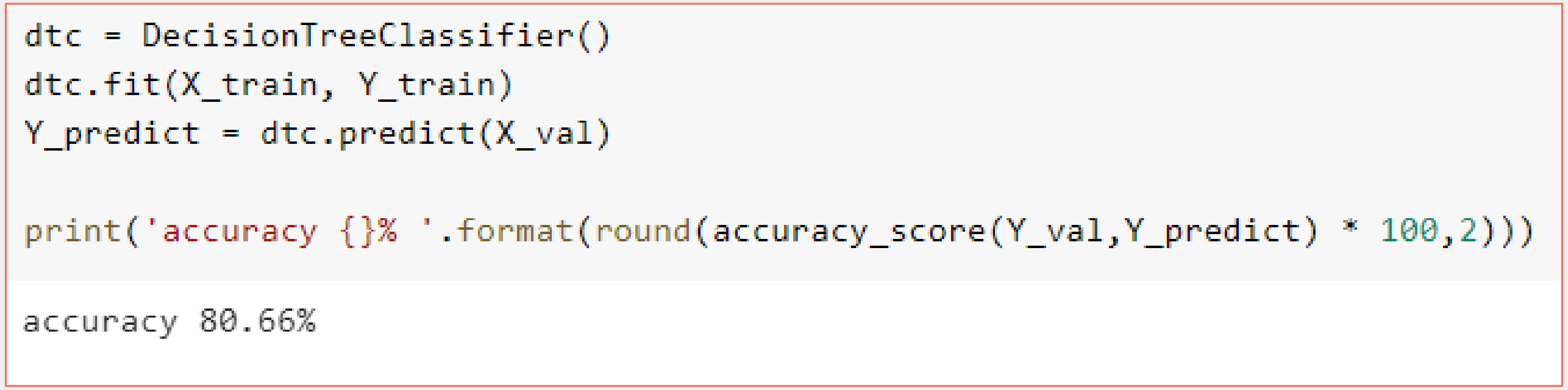
5. Other Algorithms

Decision Tree Classifier



```
dtc = DecisionTreeClassifier()
dtc.fit(tfidf_train, y_train)
y_predict = dtc.predict(tfidf_test)

print('accuracy {}% '.format(round(accuracy_score(y_val,y_predict) * 100,2)))
```

```
accuracy 80.11%
```

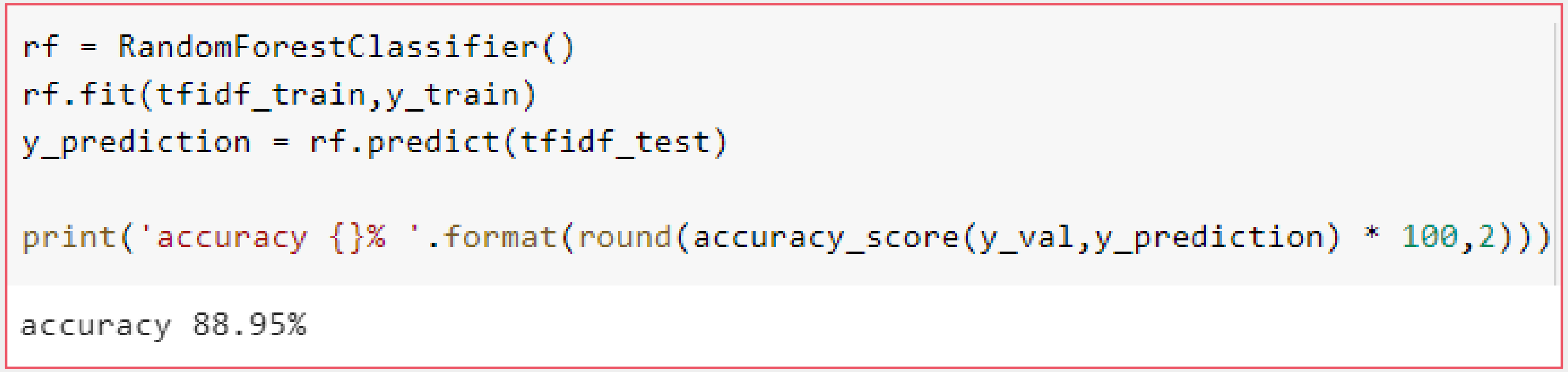


```
dtc = DecisionTreeClassifier()
dtc.fit(X_train, Y_train)
Y_predict = dtc.predict(X_val)

print('accuracy {}% '.format(round(accuracy_score(Y_val,Y_predict) * 100,2)))
```

```
accuracy 80.66%
```

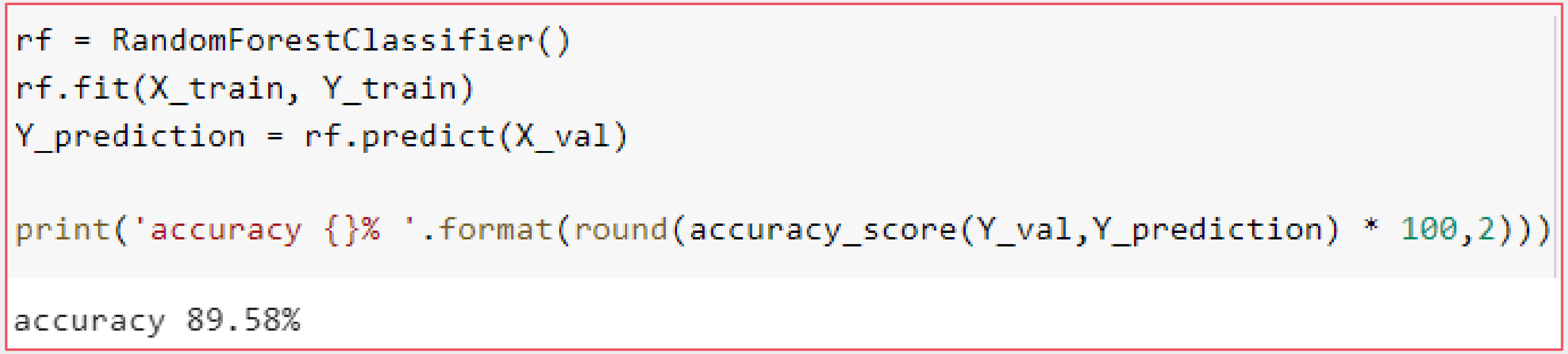
5. Other Algorithms

Random Forest Classifier



```
rf = RandomForestClassifier()
rf.fit(tfidf_train,y_train)
y_prediction = rf.predict(tfidf_test)

print('accuracy {}% '.format(round(accuracy_score(y_val,y_prediction) * 100,2)))
```

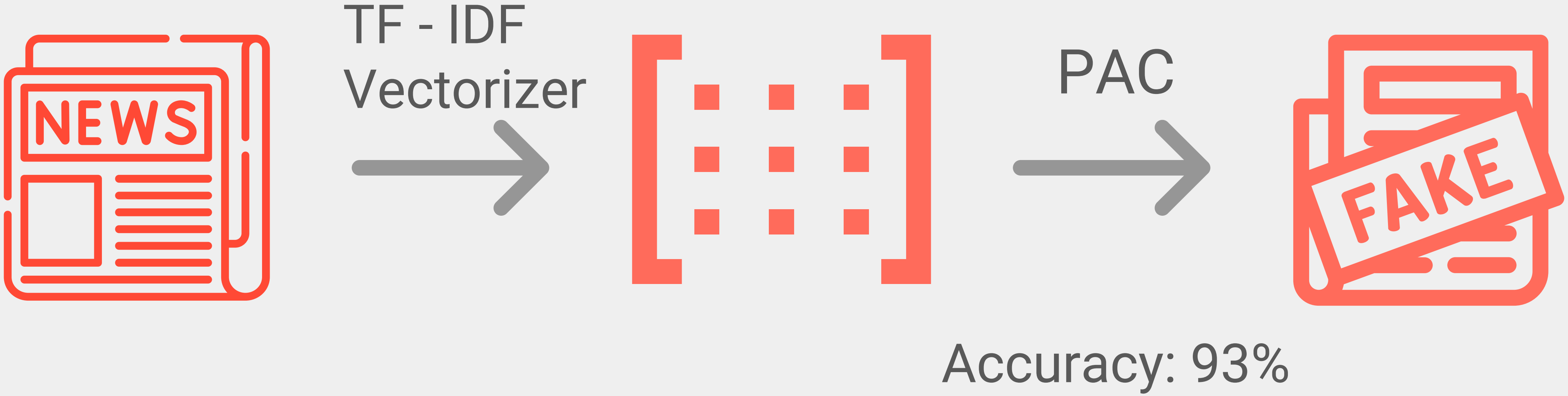
```
accuracy 88.95%
```



```
rf = RandomForestClassifier()
rf.fit(X_train, Y_train)
Y_prediction = rf.predict(X_val)

print('accuracy {}% '.format(round(accuracy_score(Y_val,Y_prediction) * 100,2)))
```

```
accuracy 89.58%
```

6. Conclusion



TF - IDF Vectorizer

PAC

Accuracy: 93%

The Tribune
SIMLA, WEDNESDAY, DECEMBER 10, 1947
Partition Of Assets And Liabilities
Complete Agreement Reached Between India & Pakistan
Kashmir Issue Not Discussed
Question To Be Decided Later
PATEL'S STATEMENT IN PARLIAMENT
PROTEST AGAINST W. BENGAL GOVT.
The Hindustan Times
GANDHIJI'S TALKS WITH THE VICEROY
DISCUSSION ON AMNESTY FOR POLITICAL PRISONERS
MAHATMA PRESSES FOR ABOLITION OF SALT DUTY
INTERCESSION WITH C.-IN-C. ON BEHALF OF I.N.A. MEN
LEAGUE DREAM OF EMPIRE
The Hindustan Times
SPEEDY EVACUATION BOTH WAYS ESSENTIAL
PATEL ON GRAVITY OF PROBLEM
UNDISTURBED PASSAGE TO BE ENSURED

THANK YOU FOR LISTENING!