**Fake News Detection Using Machine Learning**

In this digital age, fake news is a huge issue considering it hurts real-world communities by disseminating misinformation, destroying reputations, and igniting social unrest.

Fake news can be a result of misinformation, or it can be an intentional attempt to intentionally mislead people. Now it has become harder and harder to recognize whether the news is legitimate news from fake news as social media has grown a lot.

At the same time identifying and rectifying fake news is a significant concern for any news organization, so here comes machine learning, which can help in doing so.

Machine Learning Techniques have shown promising results in detecting fake news with the help of analyzing vast amounts of data, in which it identifies patterns and it provides outcomes that are based on those patterns. Machine Learning can be applied in various ways and fields for the detection of false information.

**Code:**

Now, we will try to implement machine learning methods for the detection of fake news. Here we will have two datasets: "**Fake.csv"** and **"True.csv".**

One contains fake news, and the other contains true news.

**Importing Libraries**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

import re

import string

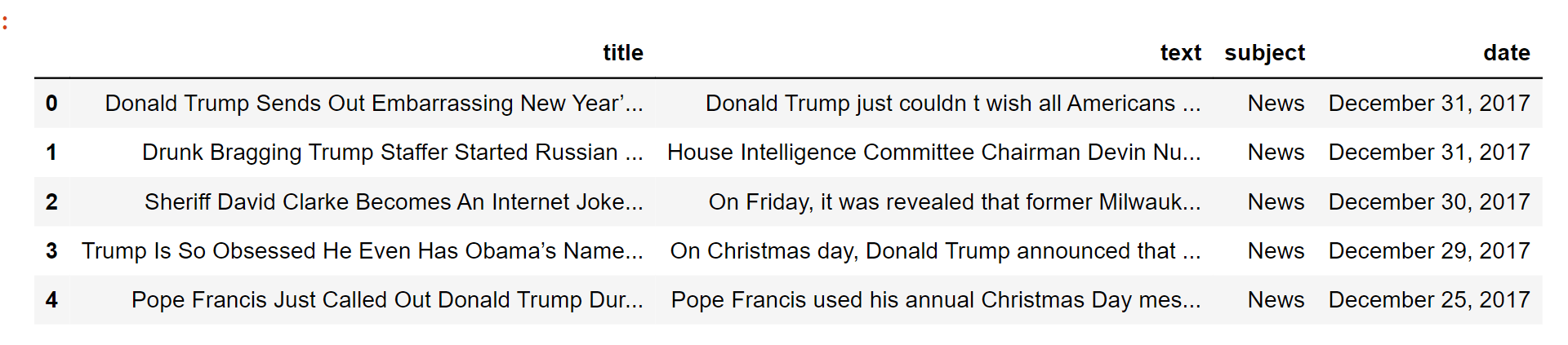
**Importing Dataset**

dataframe\_fake = pd.read\_csv("Fake.csv")

dataframe\_true = pd.read\_csv("True.csv")

dataframe\_fake.head()

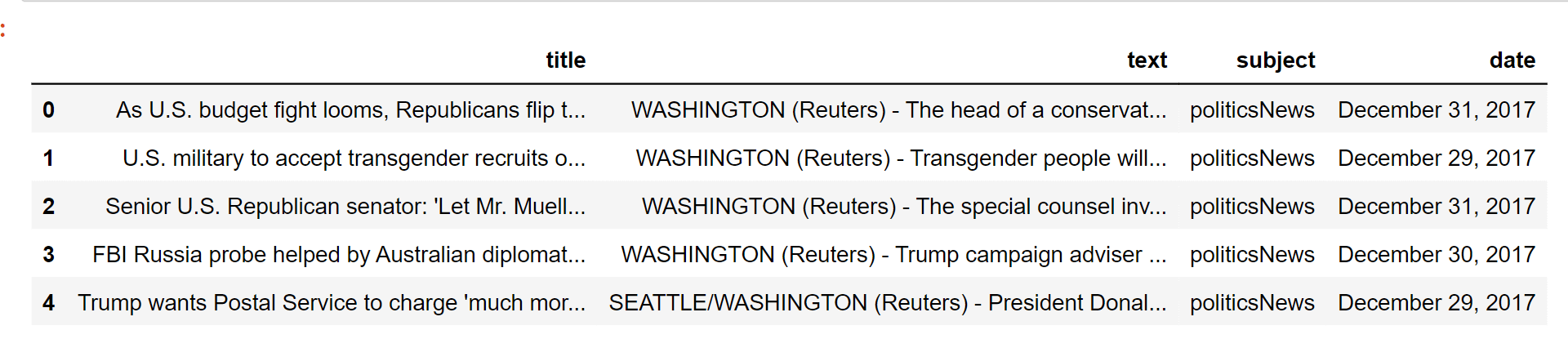
**Output:**



**Code:**

dataframe\_true.head()

**Output:**



Now we will insert a column in both of the datasets named "class", which will be the target feature. In a fake dataframe, we will give a value of 1 to the class and on the other hand, with true, we will allocate 0.

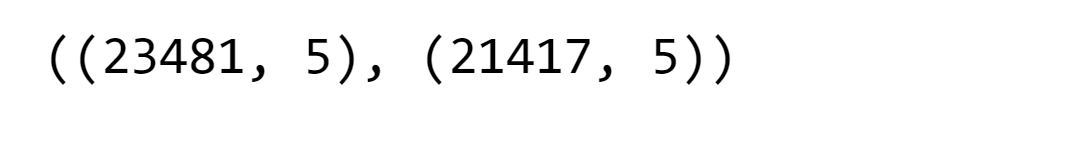
**Code:**

dataframe\_true["class"] = 0

dataframe\_true["class"] = 1

# Now, we will look at the shape of both the dataset

dataframe\_fake.shape, dataframe\_true.shape

**Output:** 

dataframe\_fake dataset contains 23481 rows and 5 columns.

dataframe\_true dataset contains 21417 rows and 5 columns.

Let's have some manual testing

# We will remove the last 10 rows for manual testing

dataframe\_fakedataframe\_fake\_manual\_testing = dataframe\_fake.tail(10)

for i in range(23480,23470,-1):

dataframe\_fake.drop([i], axis = 0, inplace = True)

dataframe\_truedataframe\_true\_manual\_testing = dataframe\_true.tail(10)

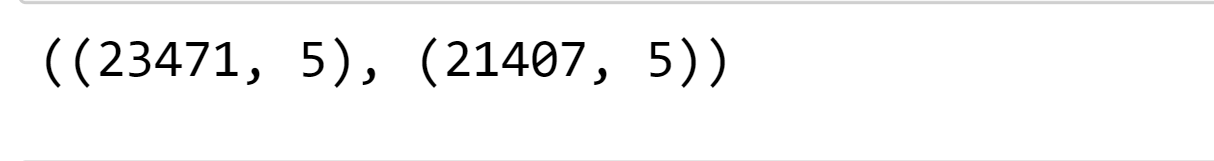
for i in range(21416,21406,-1):

dataframe\_true.drop([i], axis = 0, inplace = True)

# Let's have a look at the change in the shape of both the dataset

dataframe\_fake.shape, dataframe\_true.shape

**Output:**



If you look here, there is a decrease in the number of rows. It is because we took 10 rows from each dataset for manual testing.

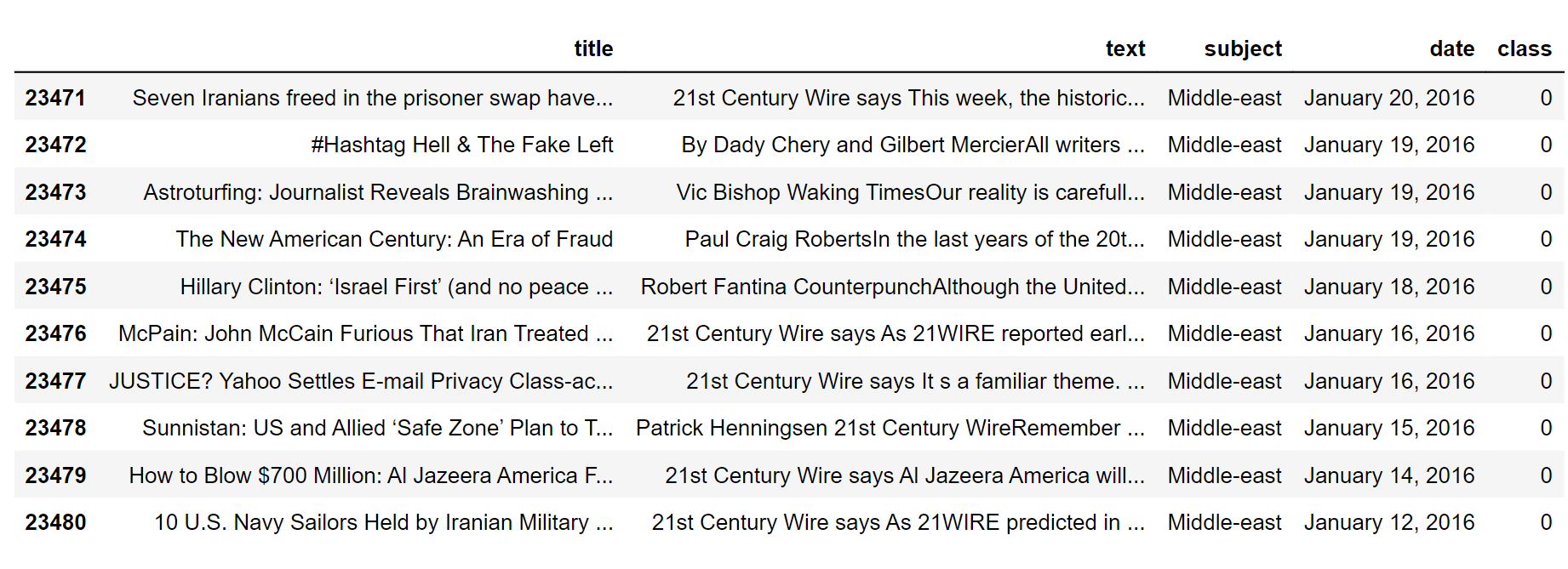
#Inserting the class column in both of the manual testing datasets

dataframe\_fake\_manual\_testing["class"] = 0

dataframe\_true\_manual\_testing["class"] = 1

dataframe\_fake\_manual\_testing.head(10)

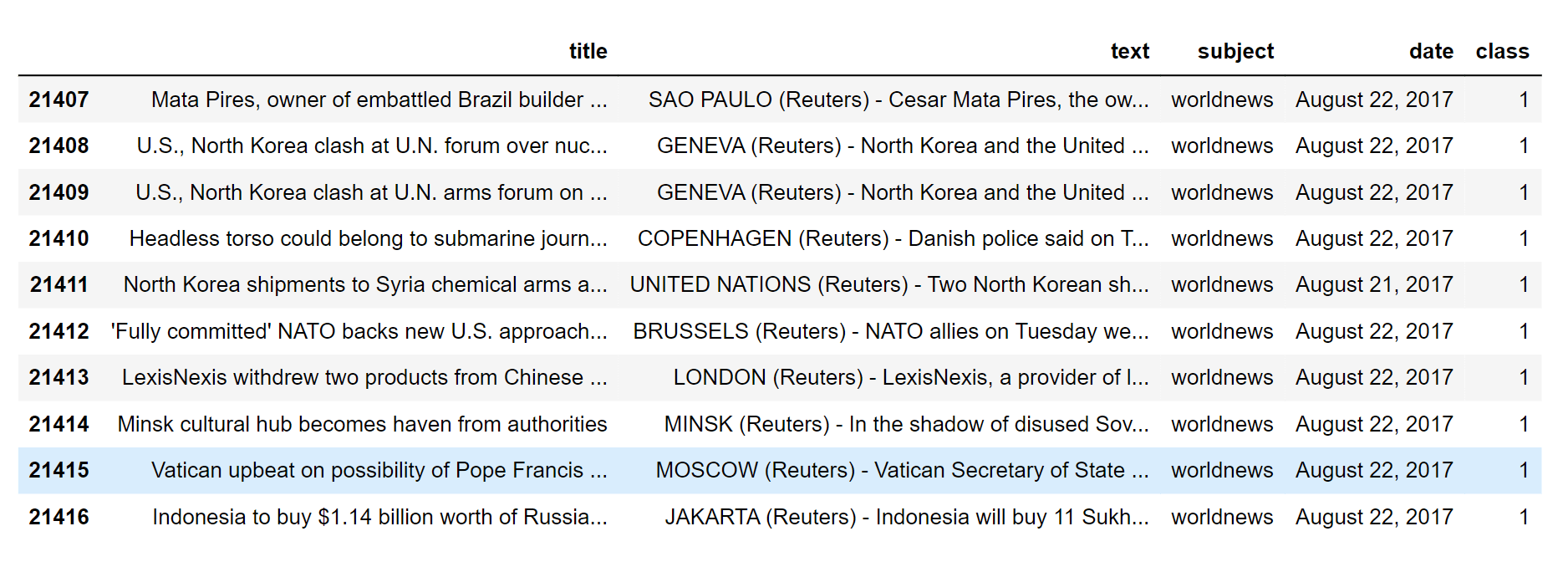
**Output:**



**Code:**

dataframe\_true\_manual\_testing.head(10)

**Output:**



**Merging True and Fake Dataframes**

Here, we will merge '**dataframe\_fake**' and '**dataframe\_true**' to form a new dataset so that we perform the machine learning operations on it.

dataframe\_merge = pd.concat([dataframe\_fake, dataframe\_true], axis =0 )

dataframe\_merge.head(10)

**Output:**

