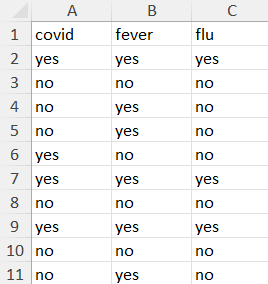
**PRACTICAL NO 5**

**AIM :** Text Categorization

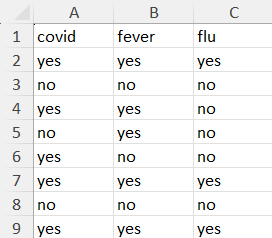
* Implement a text classification algorithm (e.g., Naive Bayes or Support Vector Machines).
* Train the classifier on a labelled dataset and evaluate its performance.

**SOLUTION :**

Dataset.csv file



Test.csv file



**INPUT:**

import pandas as pd

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

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, classification\_report

df=pd.read\_csv(r"C:\Users\nupur\sem 6 journals\IR\Dataset.csv")

data= df["covid"]+" "+df["fever"]

X=data.astype(str)

y=df['flu']

X\_train, X\_test, y\_train, y\_test=train\_test\_split(X,y,test\_size=0.2, random\_state=42)

vectorizer=CountVectorizer()

X\_train\_counts=vectorizer.fit\_transform(X\_train)

X\_test\_counts = vectorizer.transform(X\_test)

classifier=MultinomialNB()

classifier.fit(X\_train\_counts, y\_train)

data1=pd.read\_csv(r"C:\Users\nupur\sem 6 journals\IR\Test.csv")

new\_data=data1["covid"]+" "+data1["fever"]

new\_data\_counts=vectorizer.transform(new\_data.astype(str))

predictions=classifier.predict(new\_data\_counts)

new\_data=predictions

print(new\_data)

accuracy=accuracy\_score(y\_test, classifier.predict(X\_test\_counts))

print(f"\nAccuracy : {accuracy:.2f}")

print("Classification Report :")

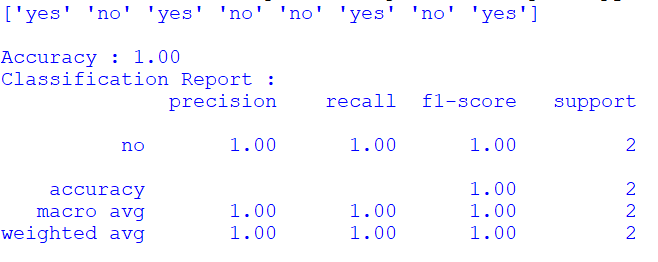
print(classification\_report(y\_test,classifier.predict(X\_test\_counts)))

predictions\_df=pd.DataFrame(predictions,columns=['flu\_prediction'])

data1=pd.concat([data1,predictions\_df],axis=1)

data1.to\_csv(r"C:\Users\nupur\sem 6 journals\IR\test1.csv",index=False)

**OUTPUT:**

****

test1.csv

