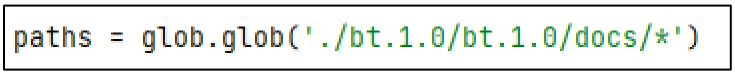
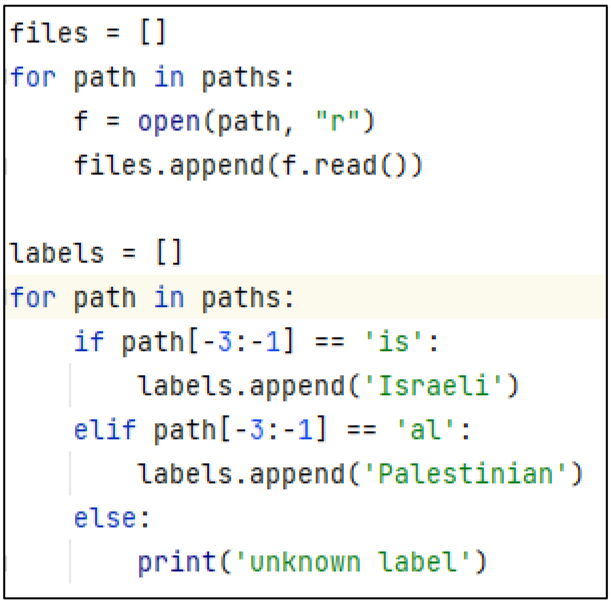
No 11

To load the dataset, I used glob. glob is used to list all the data in a given path.



```
paths = glob.glob('./bt.1.0/bt.1.0/docs/*')
```

I loaded all the data and for the label, I took it from the file name. The code is as follows:

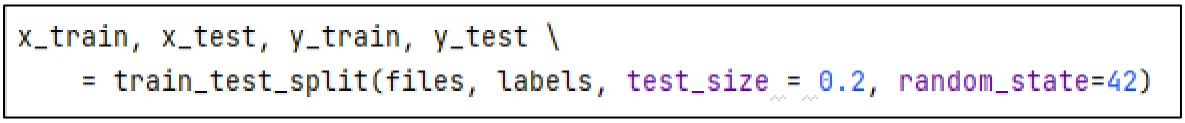


```
files = []
for path in paths:
    f = open(path, "r")
    files.append(f.read())

labels = []
for path in paths:
    if path[-3:-1] == 'is':
        labels.append('Israeli')
    elif path[-3:-1] == 'al':
        labels.append('Palestinian')
    else:
        print('unknown label')
```

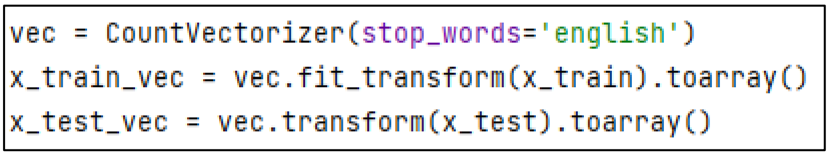
I saved 'is' as Israeli, and 'al' as Palestinian. -3 to -1 is the position of the label in the file name.

Next, I split the data into train and test by using sklearn.



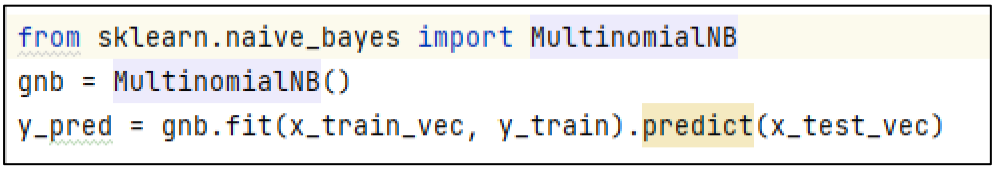
```
x_train, x_test, y_train, y_test \
    = train_test_split(files, labels, test_size = 0.2, random_state=42)
```

Once I have the train and test data, I converted it to vector using CountVetorizer by sklearn.



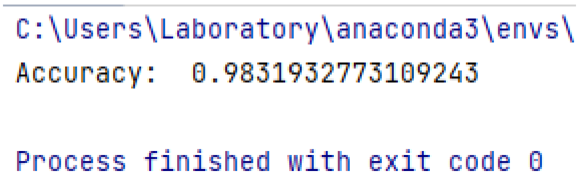
```
vec = CountVectorizer(stop_words='english')
x_train_vec = vec.fit_transform(x_train).toarray()
x_test_vec = vec.transform(x_test).toarray()
```

The model that I used to solve this problem is Multinomial Naïve Bayes. This is because it is commonly used in text classification where the data are represented as word vector counts. I implement this model by using sklearn library as follows:



```
from sklearn.naive_bayes import MultinomialNB
gnb = MultinomialNB()
y_pred = gnb.fit(x_train_vec, y_train).predict(x_test_vec)
```

The result was as follows:



```
C:\Users\Laboratory\anaconda3\envs\
Accuracy:  0.9831932773109243

Process finished with exit code 0
```