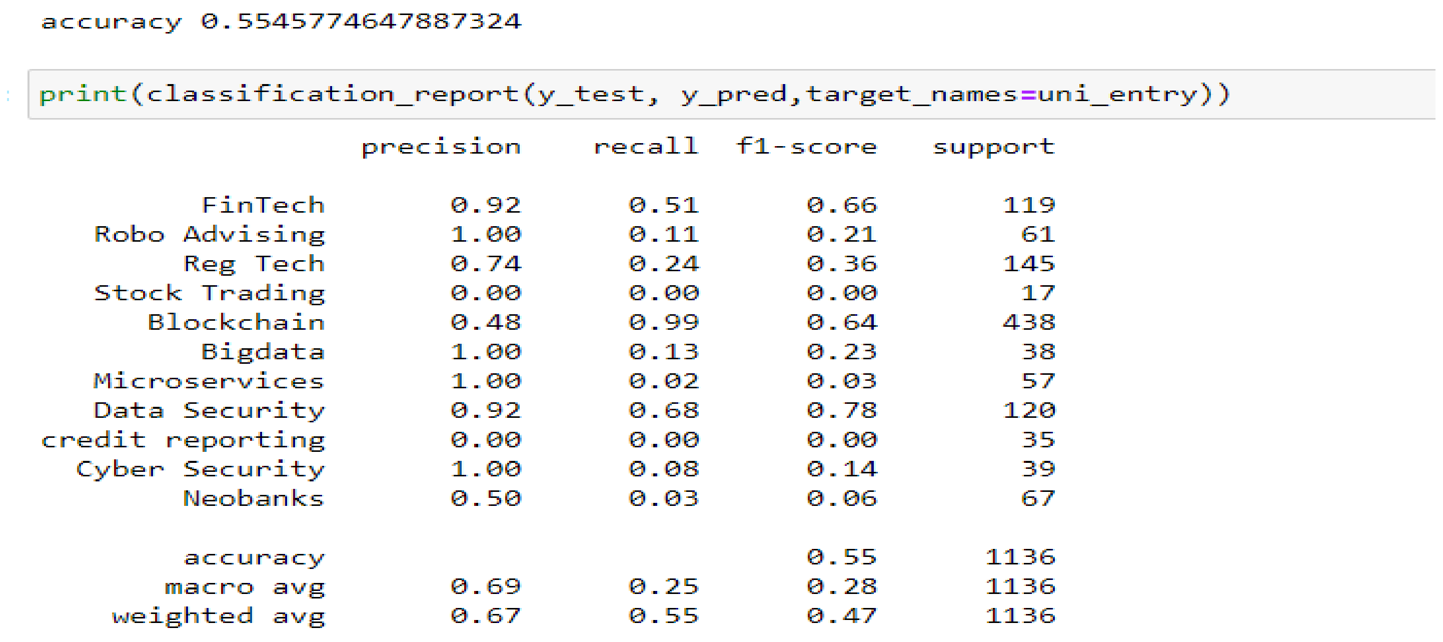
TEXT CLASSIFICATION

In this problem, we have given 22705 rows of text in 11 different classes. We have to split the data set in the train and test data set. The data set is clean so no need to cleaning data. Next, we need to differentiate data into test and train.
Because data is very ordered so I made a random subset of the data and use that subset as original data. I split the data in train and test with **95%** of data in the training set and **5%** of data in the testing set.

After splitting the data we need to convert this text into token words, using **countVectorizer** in the SkLearn library after that we will normalize the count matrix using **the tf-id** method, after that, we use this matrix to classify using naive Bayes classifier.
I use the pipeline to work as a compound classifier.



```
accuracy 0.5545774647887324
```

```
print(classification_report(y_test, y_pred,target_names=uni_entry))
```

```
                  precision    recall  f1-score   support

         FinTech       0.92      0.51      0.66       119
    Robo Advising      1.00      0.11      0.21        61
        Reg Tech       0.74      0.24      0.36       145
    Stock Trading      0.00      0.00      0.00        17
      Blockchain       0.48      0.99      0.64       438
         Bigdata       1.00      0.13      0.23        38
    Microservices      1.00      0.02      0.03        57
    Data Security      0.92      0.68      0.78       120
 credit reporting      0.00      0.00      0.00        35
   Cyber Security      1.00      0.08      0.14        39
        Neobanks       0.50      0.03      0.06        67

        accuracy                           0.55      1136
       macro avg       0.69      0.25      0.28      1136
    weighted avg       0.67      0.55      0.47      1136
```

The accuracy is very less due to several factors I think! First due to the high uncorrelation between the text in the data set. second, could be the very sort text in the data set for every class.
The classes which have more data set have better **F1** score in this data set like blockchain and fintech.
We can test this data set on a different classifier to see the result and compare that to this result.