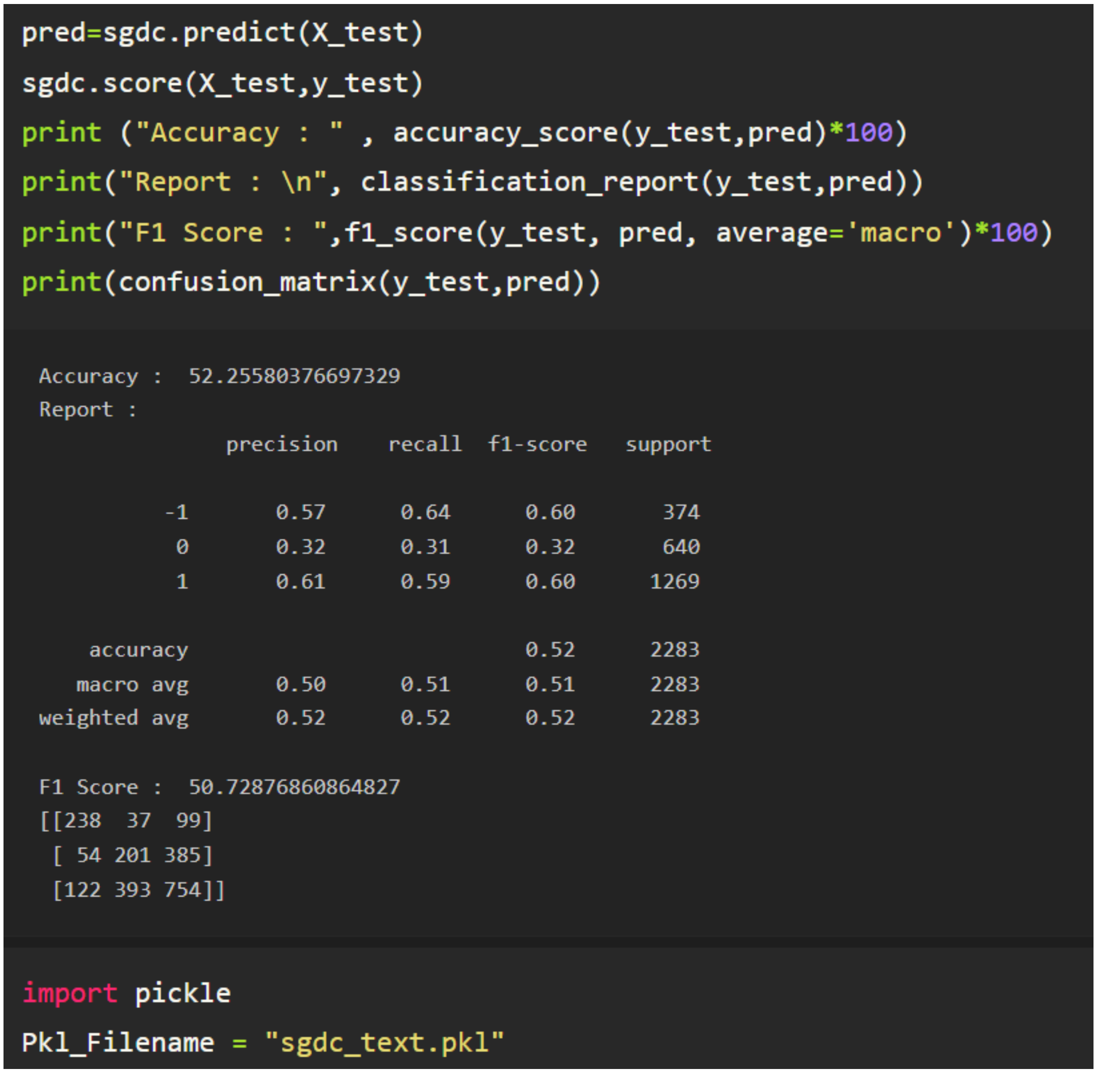
F1 SCORES PER MODEL

- SGDCClassifier for text

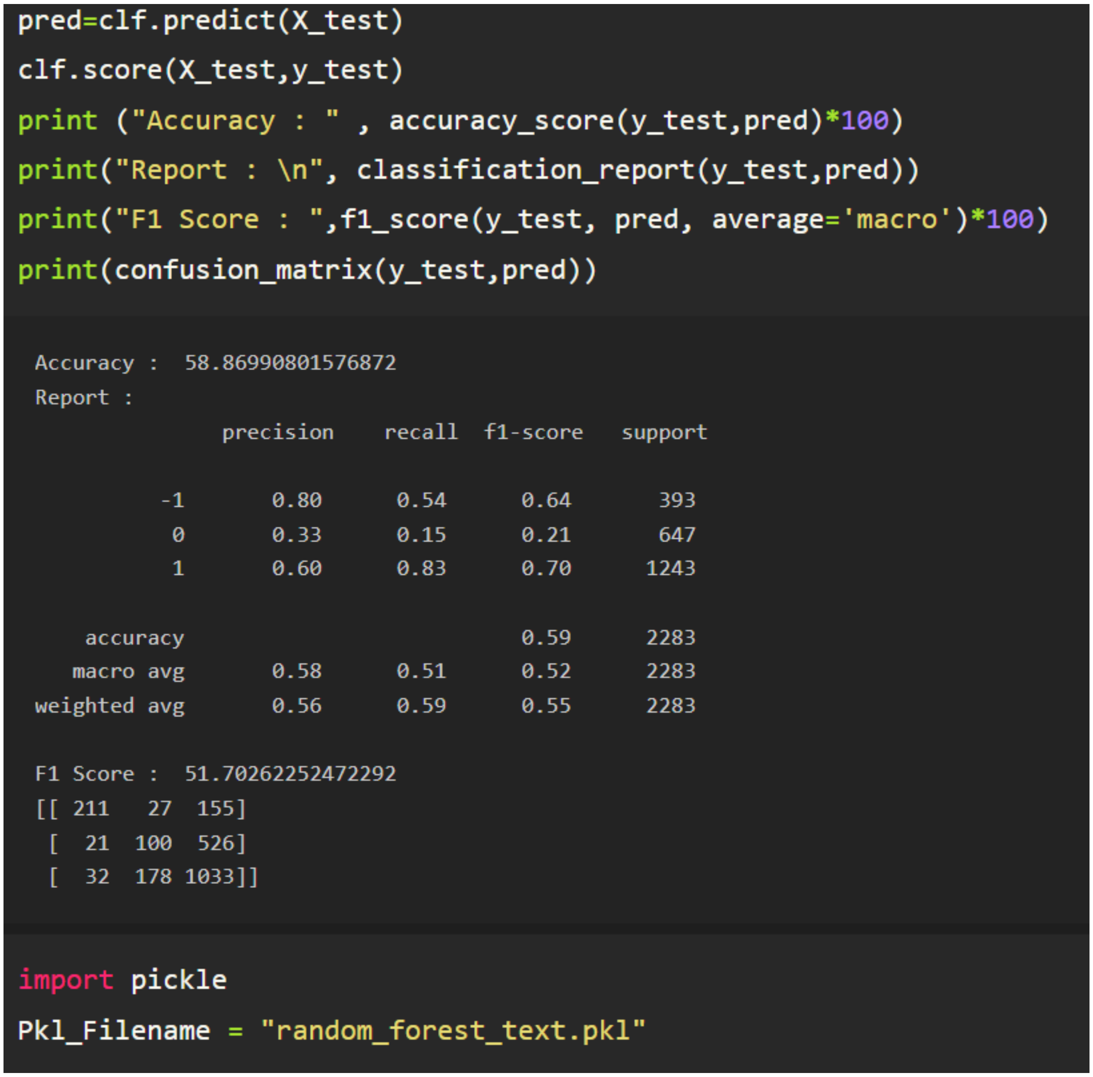


```
pred=sgdc.predict(X_test)
sgdc.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
print("Report : \n", classification_report(y_test,pred))
print("F1 Score : ",f1_score(y_test, pred, average='macro')*100)
print(confusion_matrix(y_test,pred))
```

```
Accuracy :  52.25580376697329
Report :
               precision    recall  f1-score   support

          -1       0.57      0.64      0.60       374
           0       0.32      0.31      0.32       640
           1       0.61      0.59      0.60      1269

    accuracy                           0.52      2283
   macro avg       0.50      0.51      0.51      2283
weighted avg       0.52      0.52      0.52      2283

F1 Score :  50.72876860864827
[[238  37  99]
 [ 54 201 385]
 [122 393 754]]
```

```
import pickle
Pkl_Filename = "sgdc_text.pkl"
```

- Random Forest Classifier for text

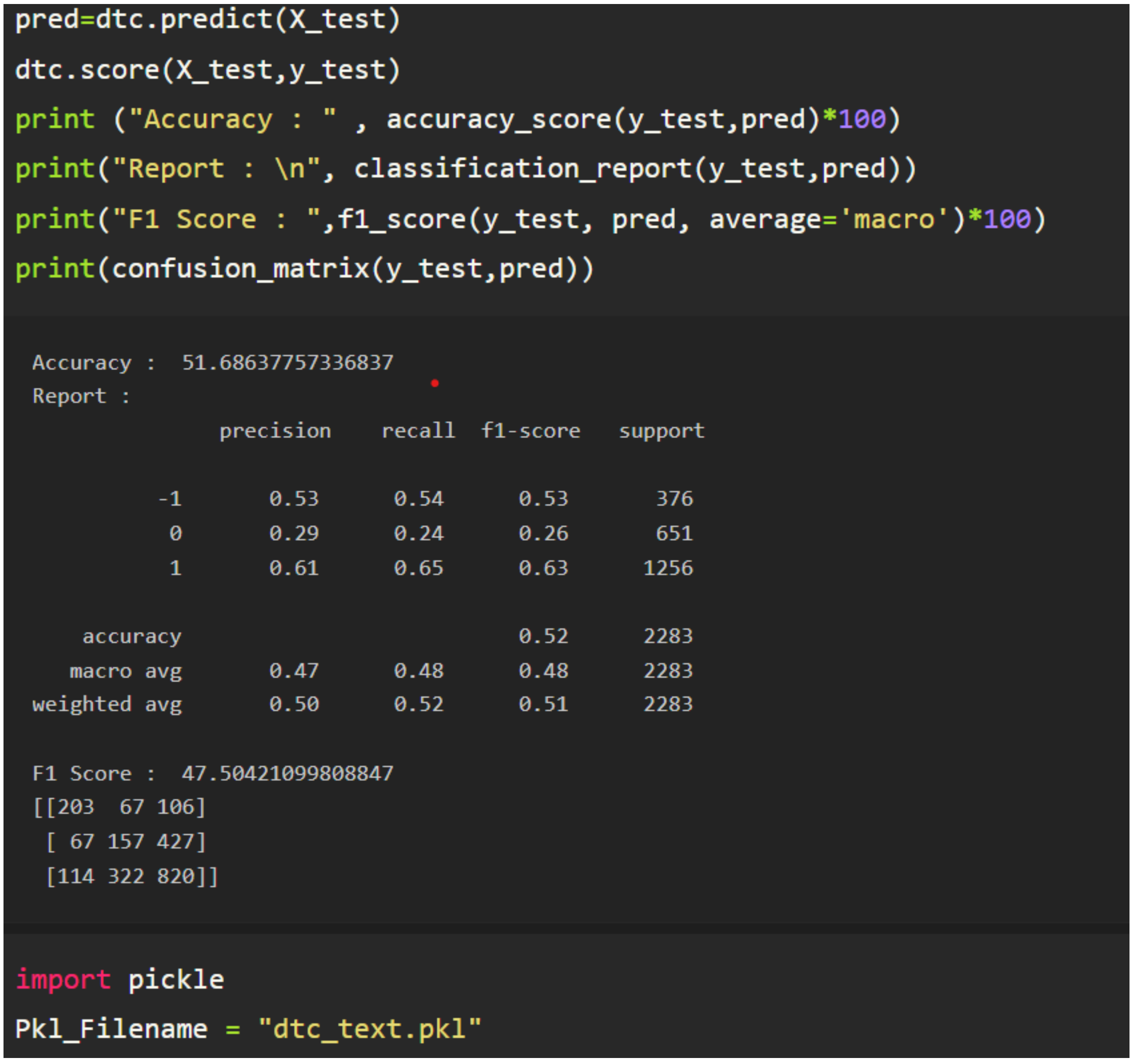


```
pred=clf.predict(X_test)
clf.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
print("Report : \n", classification_report(y_test,pred))
print("F1 Score : ",f1_score(y_test, pred, average='macro')*100)
print(confusion_matrix(y_test,pred))
```

```
Accuracy :  58.86990801576872
Report : 
              precision    recall  f1-score   support

          -1       0.80      0.54      0.64       393
           0       0.33      0.15      0.21       647
           1       0.60      0.83      0.70      1243

    accuracy                           0.59      2283
   macro avg       0.58      0.51      0.52      2283
weighted avg       0.56      0.59      0.55      2283

F1 Score :  51.70262252472292
[[ 211   27  155]
 [  21  100  526]
 [  32  178 1033]]
```

```
import pickle
Pkl_Filename = "random_forest_text.pkl"
```

- Decision Tree Classifier for text

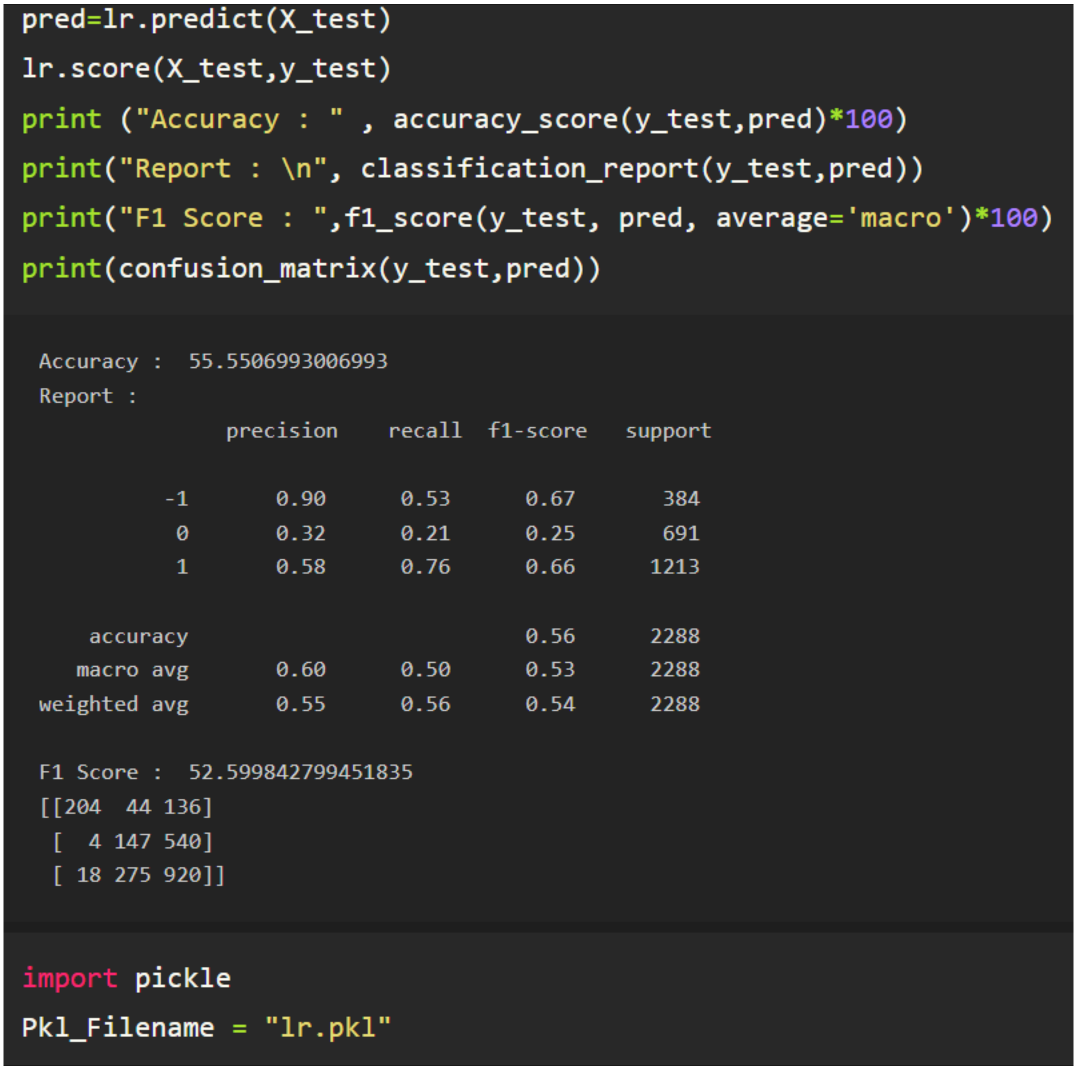


```
pred=dtc.predict(X_test)
dtc.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
print("Report : \n", classification_report(y_test,pred))
print("F1 Score : ",f1_score(y_test, pred, average='macro')*100)
print(confusion_matrix(y_test,pred))
```

```
Accuracy :  51.68637757336837
Report :
               precision    recall  f1-score   support

          -1       0.53      0.54      0.53       376
           0       0.29      0.24      0.26       651
           1       0.61      0.65      0.63      1256

    accuracy                           0.52      2283
   macro avg       0.47      0.48      0.48      2283
weighted avg       0.50      0.52      0.51      2283

F1 Score :  47.50421099808847
[[203  67 106]
 [ 67 157 427]
 [114 322 820]]
```

```
import pickle
Pkl_Filename = "dtc_text.pkl"
```

- Logistic Regression for images

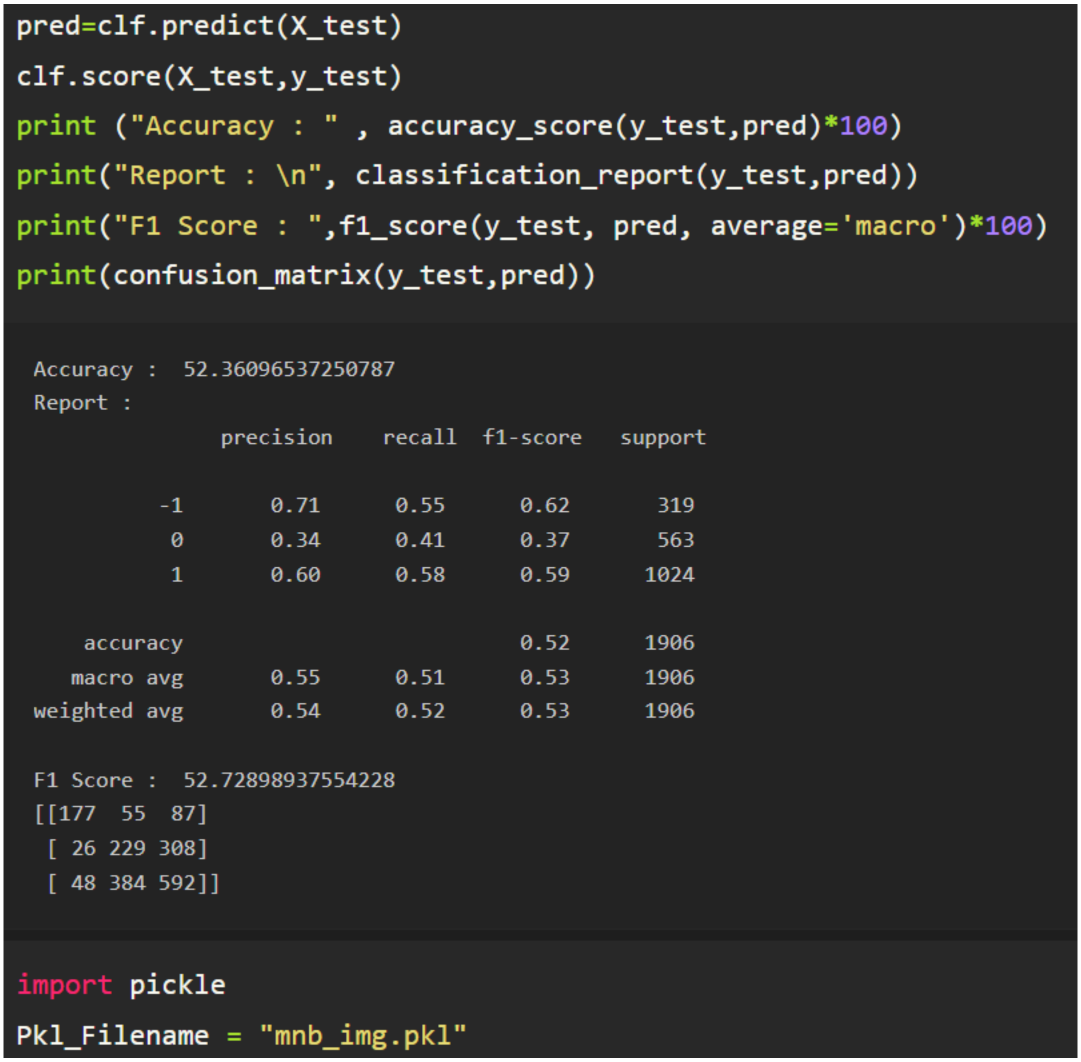


```
pred=lr.predict(X_test)
lr.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
print("Report : \n", classification_report(y_test,pred))
print("F1 Score : ",f1_score(y_test, pred, average='macro')*100)
print(confusion_matrix(y_test,pred))
```

```
Accuracy :  55.5506993006993
Report : 
               precision    recall  f1-score   support

          -1       0.90      0.53      0.67       384
           0       0.32      0.21      0.25       691
           1       0.58      0.76      0.66      1213

    accuracy                           0.56      2288
   macro avg       0.60      0.50      0.53      2288
weighted avg       0.55      0.56      0.54      2288

F1 Score :  52.599842799451835
[[204  44 136]
 [  4 147 540]
 [ 18 275 920]]
```

```
import pickle
Pkl_Filename = "lr.pkl"
```

- MultinomialNB for image

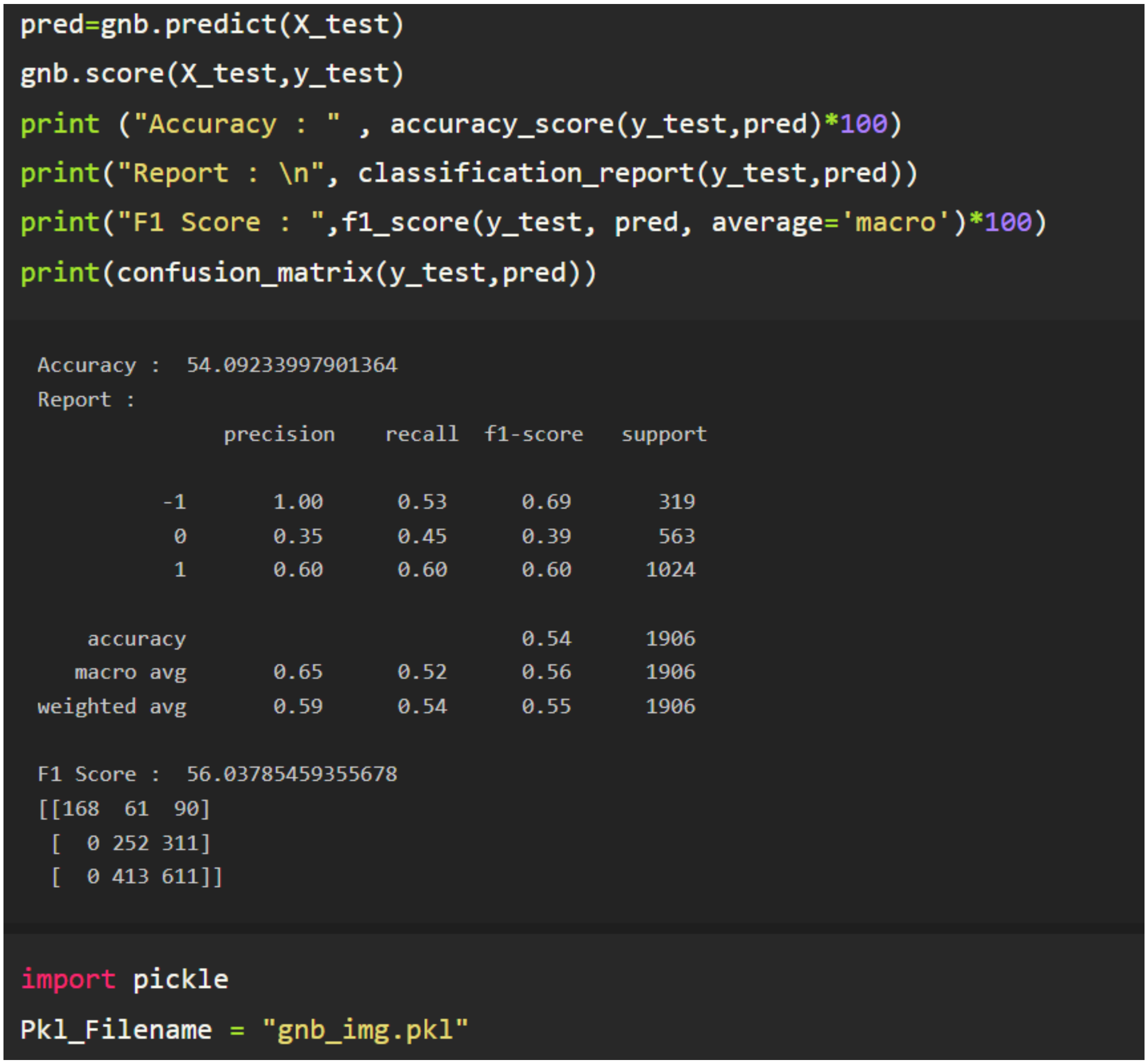


```
pred=clf.predict(X_test)
clf.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
print("Report : \n", classification_report(y_test,pred))
print("F1 Score : ",f1_score(y_test, pred, average='macro')*100)
print(confusion_matrix(y_test,pred))
```

```
Accuracy :  52.36096537250787
Report :
               precision    recall  f1-score   support

          -1       0.71      0.55      0.62       319
           0       0.34      0.41      0.37       563
           1       0.60      0.58      0.59      1024

    accuracy                           0.52      1906
   macro avg       0.55      0.51      0.53      1906
weighted avg       0.54      0.52      0.53      1906

F1 Score :  52.72898937554228
[[177  55  87]
 [ 26 229 308]
 [ 48 384 592]]
```

```
import pickle
Pkl_Filename = "mnb_img.pkl"
```

- GaussianNB for image

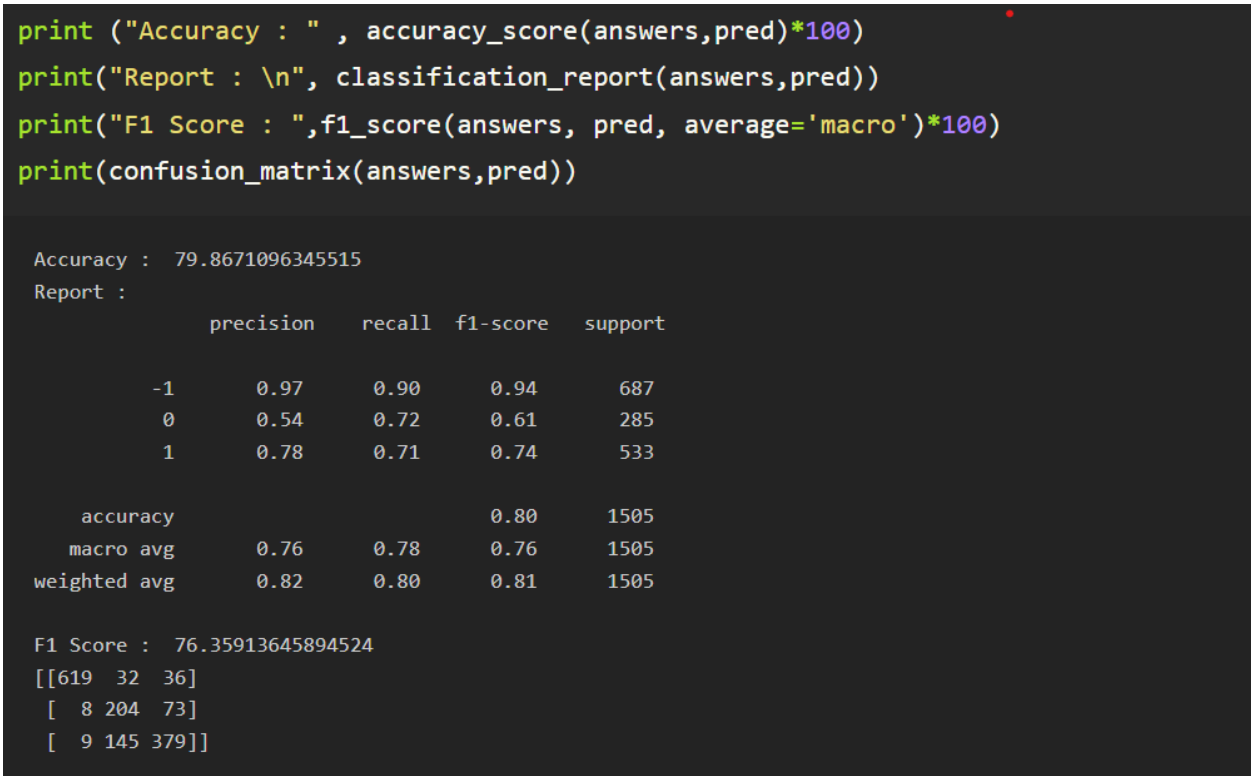


```
pred=gnb.predict(X_test)
gnb.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
print("Report : \n", classification_report(y_test,pred))
print("F1 Score : ",f1_score(y_test, pred, average='macro')*100)
print(confusion_matrix(y_test,pred))
```

```
Accuracy :  54.09233997901364
Report :
               precision    recall  f1-score   support

          -1       1.00      0.53      0.69       319
           0       0.35      0.45      0.39       563
           1       0.60      0.60      0.60      1024

    accuracy                           0.54      1906
   macro avg       0.65      0.52      0.56      1906
weighted avg       0.59      0.54      0.55      1906

F1 Score :  56.03785459355678
[[168  61  90]
 [  0 252 311]
 [  0 413 611]]
```

```
import pickle
Pkl_Filename = "gnb_img.pkl"
```

- Combined F1 Score



```python
print ("Accuracy : " , accuracy_score(answers,pred)*100)
print("Report : \n", classification_report(answers,pred))
print("F1 Score : ",f1_score(answers, pred, average='macro')*100)
print(confusion_matrix(answers,pred))
```

```
Accuracy :  79.8671096345515
Report : 
               precision    recall  f1-score   support

          -1       0.97      0.90      0.94       687
           0       0.54      0.72      0.61       285
           1       0.78      0.71      0.74       533

    accuracy                           0.80      1505
   macro avg       0.76      0.78      0.76      1505
weighted avg       0.82      0.80      0.81      1505

F1 Score :  76.35913645894524
[[619  32  36]
 [  8 204  73]
 [  9 145 379]]
```