DADS7203_assignment1: Train, Test Model ในการทำ sentiment analysis โดยใช้ dataset Tcas61_2.csv

รายชื่อกลุ่ม:

1. 6410422007 ประภัสสร เตียวพานิชย์กิจ
2. 6410422013 วนิดา ถาวรสุข
3. 6410422016 โชติกา บุญทวีโชค
4. 6410422021 ศุภฤกษ์ ฤทธิกุลสิทธิชัย
5. 6410422022 จุฑามาศ พิสกุล

File github: https://github.com/chotika-boon/DADS7203_assignment1

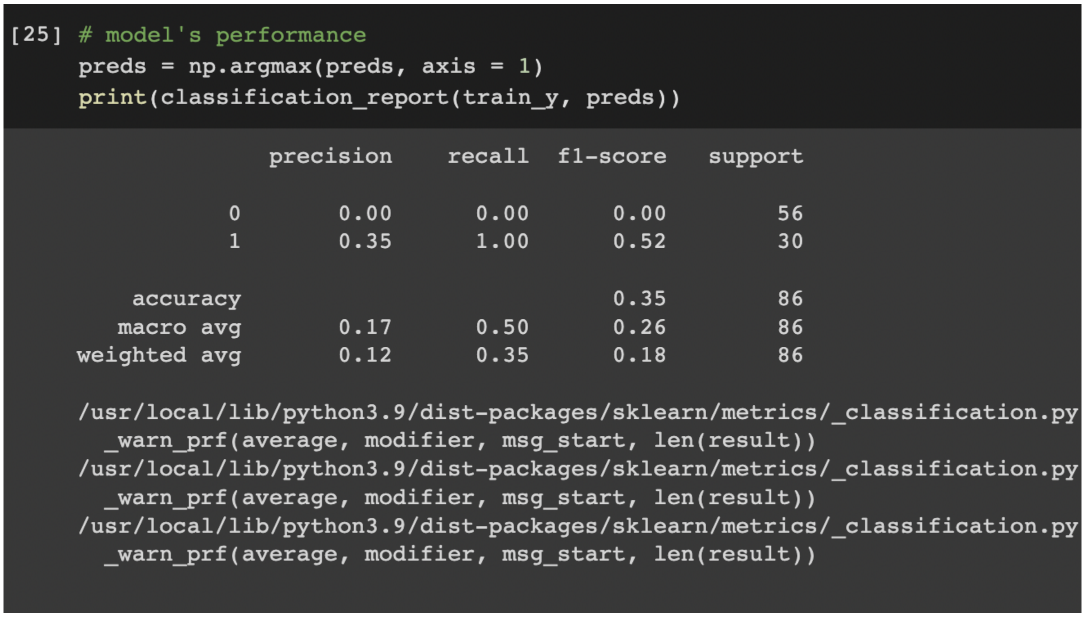
Summary Model:

No	Details	Train F1 Score	Test F1 Score	Code File
1	**No Fine Tune Model** **Model**: BERT **Pretrained**: bert-base uncased **Batch_size**: 16 **max_seq_len**: 8 **Epoch**: 15	0.35	0.48	DADS7203_assignment1_Model1.ipynb
2	**Fine Tune Model** **Model**: BERT **Pretrained**: "Geotrend/bert-base-th-cased **Batch_size**: 32 **Epoch**: 50 **max_seq_len**: 8 **Optimizer**: AdamW lr = 5e-2	0.93	0.79	DADS7203_assignment1_Model2.ipynb
3	**Model**: BERT Pretrained: poom-sci/WangchanBERTa-finetuned-sentiment **Batch_size**: 32 **Epoch**: 30 **max_seq_len**: 25 **Optimizer**: AdamW lr = 2e-5, eps = 1e-8	1.00	0.95	DADS7203_assignment1_Model3.ipynb

Capture Result

Model 1

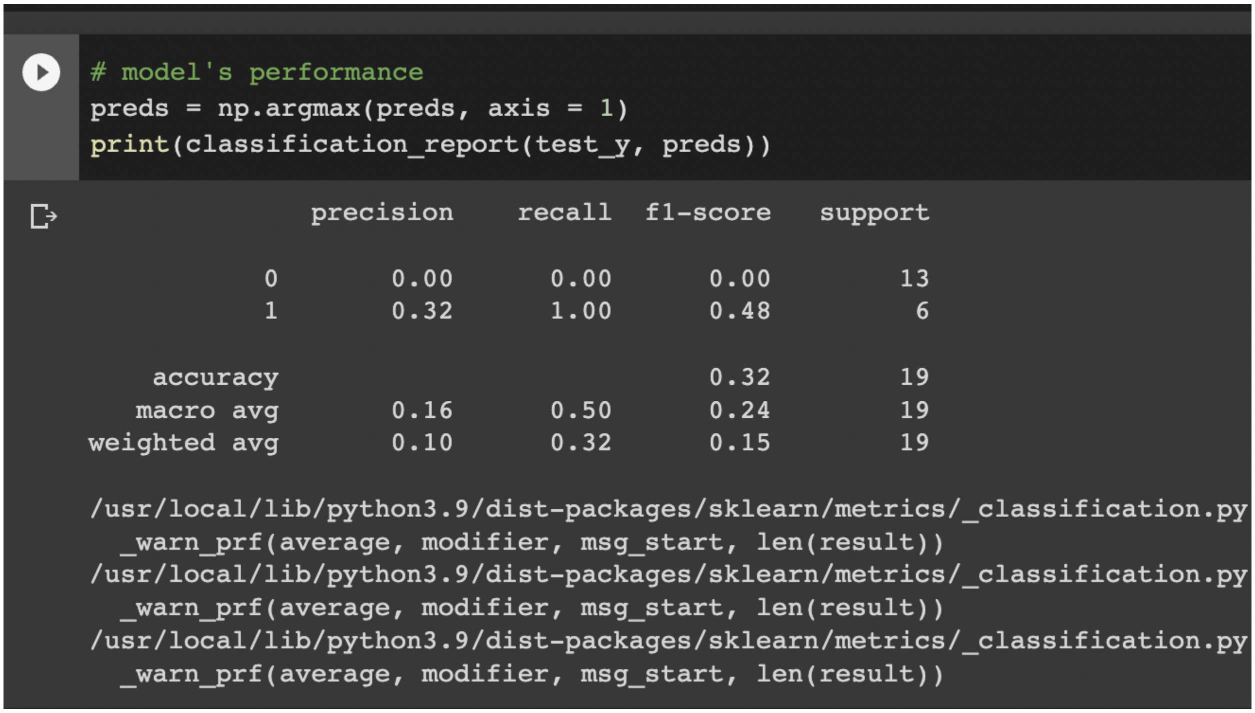
Train F1 Score



```
[25] # model's performance
     preds = np.argmax(preds, axis = 1)
     print(classification_report(train_y, preds))
```

```
                  precision    recall  f1-score   support

               0       0.00      0.00      0.00        56
               1       0.35      1.00      0.52        30

        accuracy                           0.35        86
       macro avg       0.17      0.50      0.26        86
    weighted avg       0.12      0.35      0.18        86

    /usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py
      _warn_prf(average, modifier, msg_start, len(result))
    /usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py
      _warn_prf(average, modifier, msg_start, len(result))
    /usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py
      _warn_prf(average, modifier, msg_start, len(result))
```

Test F1 Score

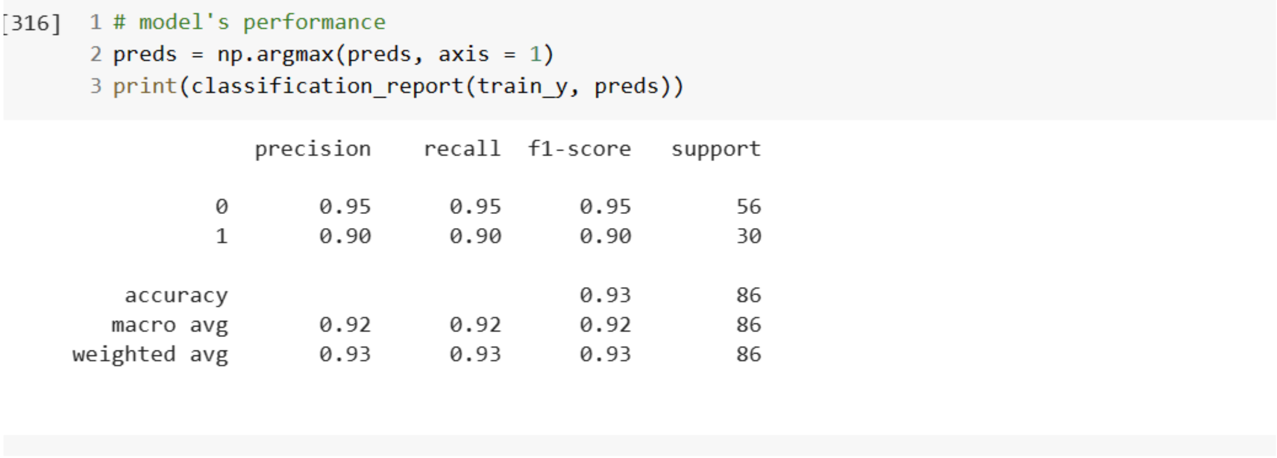


```
# model's performance
preds = np.argmax(preds, axis = 1)
print(classification_report(test_y, preds))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        13
           1       0.32      1.00      0.48         6

    accuracy                           0.32        19
   macro avg       0.16      0.50      0.24        19
weighted avg       0.10      0.32      0.15        19

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py
  _warn_prf(average, modifier, msg_start, len(result))
```

Model 2

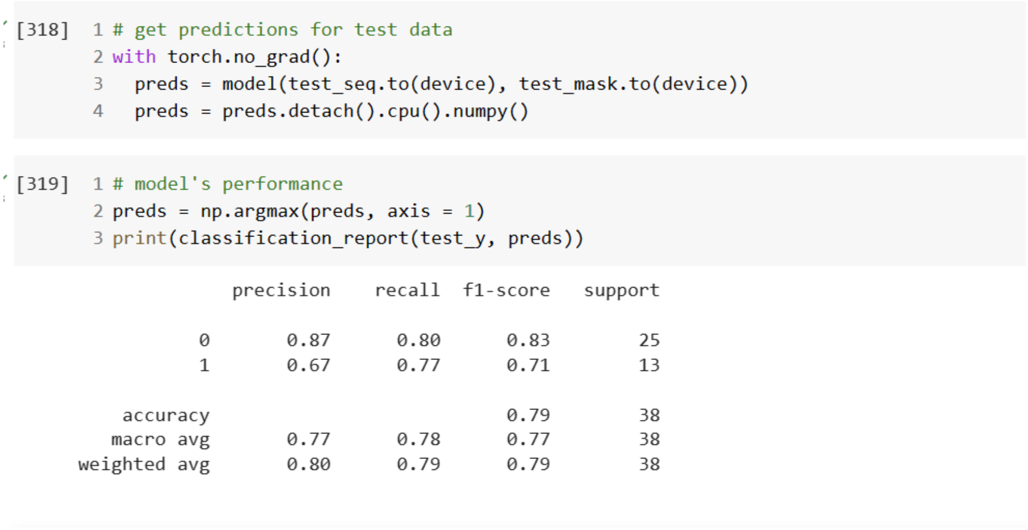
Train F1 Score



```
[316]  # model's performance
       preds = np.argmax(preds, axis = 1)
       print(classification_report(train_y, preds))
```

```
              precision    recall  f1-score   support

           0       0.95      0.95      0.95        56
           1       0.90      0.90      0.90        30

    accuracy                           0.93        86
   macro avg       0.92      0.92      0.92        86
weighted avg       0.93      0.93      0.93        86
```

Test F1 Score

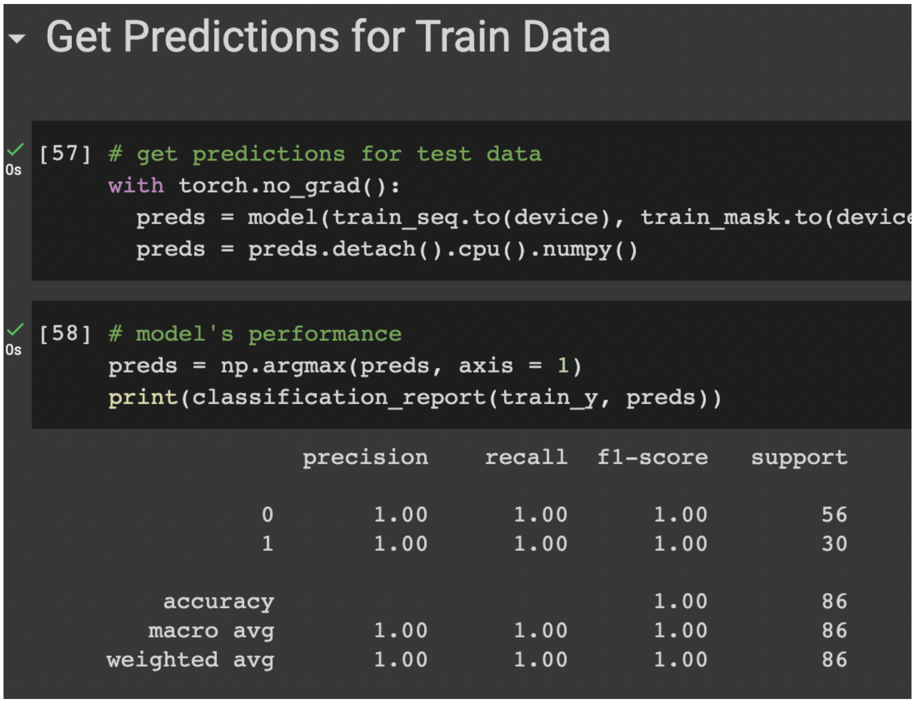


```
[318]  # get predictions for test data
       with torch.no_grad():
         preds = model(test_seq.to(device), test_mask.to(device))
         preds = preds.detach().cpu().numpy()
```

```
[319]  # model's performance
       preds = np.argmax(preds, axis = 1)
       print(classification_report(test_y, preds))
```

```
              precision    recall  f1-score   support

           0       0.87      0.80      0.83        25
           1       0.67      0.77      0.71        13

    accuracy                           0.79        38
   macro avg       0.77      0.78      0.77        38
weighted avg       0.80      0.79      0.79        38
```

Model 3
Train F1 Score



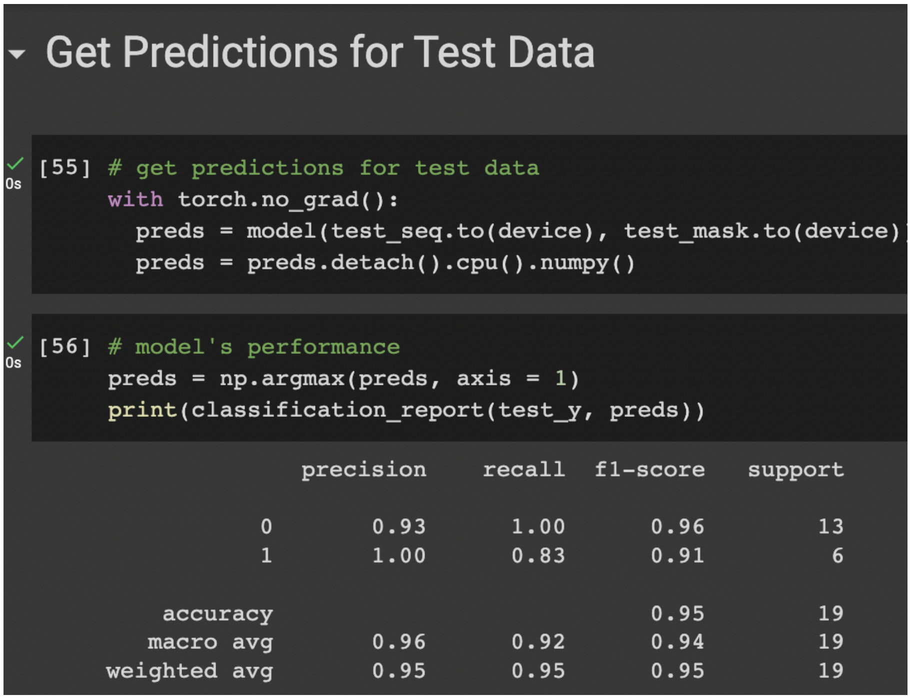
Get Predictions for Train Data

```
[57] # get predictions for test data
     with torch.no_grad():
       preds = model(train_seq.to(device), train_mask.to(devic
       preds = preds.detach().cpu().numpy()
```

```
[58] # model's performance
     preds = np.argmax(preds, axis = 1)
     print(classification_report(train_y, preds))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        56
           1       1.00      1.00      1.00        30

    accuracy                           1.00        86
   macro avg       1.00      1.00      1.00        86
weighted avg       1.00      1.00      1.00        86
```

Test F1 Score



Get Predictions for Test Data

```
[55] # get predictions for test data
     with torch.no_grad():
       preds = model(test_seq.to(device), test_mask.to(device)
       preds = preds.detach().cpu().numpy()
```

```
[56] # model's performance
     preds = np.argmax(preds, axis = 1)
     print(classification_report(test_y, preds))
```

```
              precision    recall  f1-score   support

           0       0.93      1.00      0.96        13
           1       1.00      0.83      0.91         6

    accuracy                           0.95        19
   macro avg       0.96      0.92      0.94        19
weighted avg       0.95      0.95      0.95        19
```