

## Homework 3

Yu-Chieh Kuo B07611039<sup>†</sup>

<sup>†</sup>Department of Information Management, National Taiwan University

### Usage

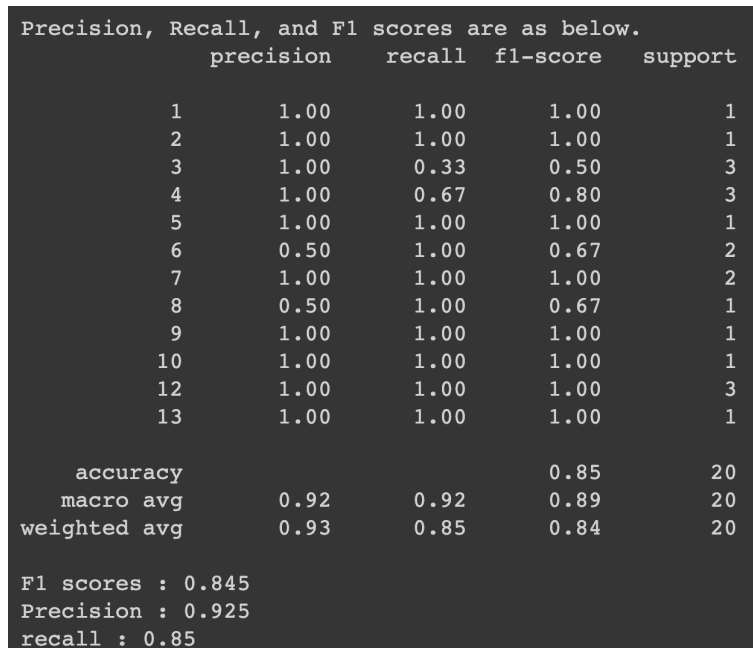
---

```
cd b07611039
cp -R /PA3-data .
cp -R /Pretrained model . # move pre-trained model to this directory
python -V # Python 3.7.12 in my environment
pip install keras-bert
pip install tensorflow # I did this assignment Google colab, where the
tensorflow package was installed and imported by default.
pip install sklearn
pip install matplotlib
pip install pickle
pip install pandas
# Requirements: keras-bert, sklearn, matplotlib, tensorflow, pickle, pandas
python3 pa3.py
```

---

### Precision, Recall, F1 score, and Precision Recall Curve

The precision, recall, F1 score, and precision recall curve are represented as below.



Precision, Recall, and F1 scores are as below.					
	precision	recall	f1-score	support	
1	1.00	1.00	1.00	1	
2	1.00	1.00	1.00	1	
3	1.00	0.33	0.50	3	
4	1.00	0.67	0.80	3	
5	1.00	1.00	1.00	1	
6	0.50	1.00	0.67	2	
7	1.00	1.00	1.00	2	
8	0.50	1.00	0.67	1	
9	1.00	1.00	1.00	1	
10	1.00	1.00	1.00	1	
12	1.00	1.00	1.00	3	
13	1.00	1.00	1.00	1	
accuracy			0.85	20	
macro avg	0.92	0.92	0.89	20	
weighted avg	0.93	0.85	0.84	20	
F1 scores : 0.845					
Precision : 0.925					
recall : 0.85					

Figure 1: Precision, Recall, and F1 scores for SVM with linear kernel by using the BERT-BASE pre-trained model

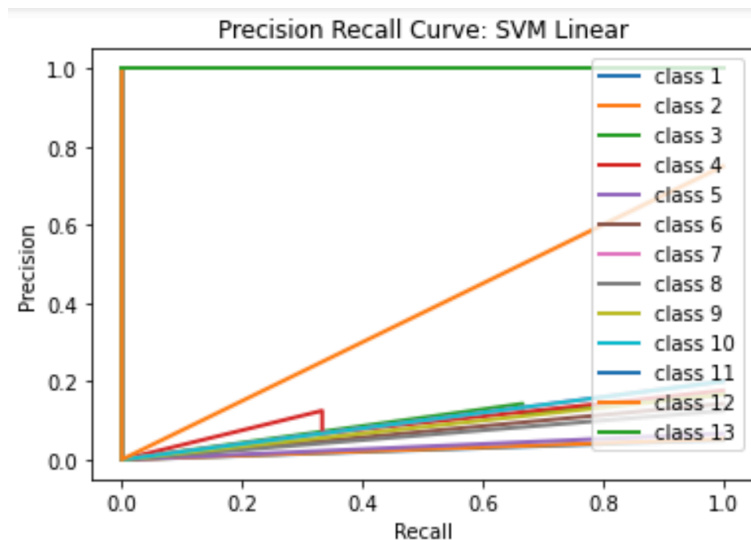


Figure 2: Precision curve for SVM with linear kernel by using the BERT-BASE pre-trained model

## Score on Kaggle

The submission score on Kaggle is 0.87444. I compare with the results between using TFIDF vector and CLS embedding and it shows that the results by using TFIDF vector is better than CLS embedding. As Kaggle records the best score in the competition, my score, consequently, arises to 0.98222, leading me to the second prize. However, that score is not the true result in BERT, therefore, **I only select the correct score from BERT for the final score.**

In discussion, I have studied that it shall combine BERT with modern deep learning skill to reinforce its precision and improve its result. In other words, if we use the traditional method such as SVM after training by BERT, the result will not be excellent, or more improved.

## Implement

1. Preprocess the data as previous programming assignments.
2. Download and load pre-trained model.

---

```
from keras_bert import load_vocabulary
from keras_bert import Tokenizer
model_path = './'
dict_path = './vocab.txt'
bert_token_dict = load_vocabulary(dict_path)
bert_tokenizer = Tokenizer(bert_token_dict)
```

---

3. Retrieve CLS embedding data.

---

```
CLS_embedding_set = []
for i in range(0, len(dataset), batch_size):
    embeddings = extract_embeddings(model_path, dataset[i:i + batch_size])
    CLS_embedding_set.extend([item[0] for item in embeddings])

CLS_embedding_set = np.array(CLS_embedding_set)
```

---

4. Spilt training set and testing set.

---

```
training_embedding = CLS_embedding_set[docs['id'].isin(labels['training_id'])]
testing_embedding = CLS_embedding_set[~docs['id'].isin(labels['training_id'])]
```

---

## 5. Train SVC with linear kernel.

---

```
x_train, x_test, y_train, y_test = train_test_split(training_embedding,
                                                    labels['classes'], test_size = 0.1)
SVC_Linear_model = SVC(kernel='linear', C = 1.0)
SVC_Linear_model.fit(x_train, y_train)

prediction = []
expectation = []

prediction.extend(SVC_Linear_model.predict(x_test))
expectation.extend(y_test)
```

---

## 6. Illustrate precision recall curve, check precision, recall and F1 score.