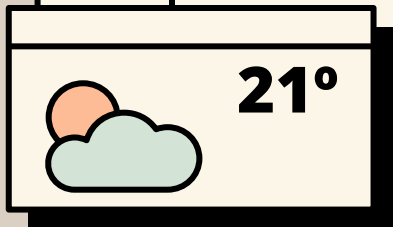
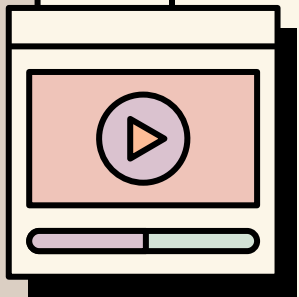


Lab4

大型語言模型 情感分析



>>>>>

第四組 | 蕭名妍 劉貞莉 黃詩涵

~~~~~  
.....



# 資料簡介

- 資料集：IMDB Dataset.csv
- 共2個欄位、50,000筆資料

“review”：電影評論

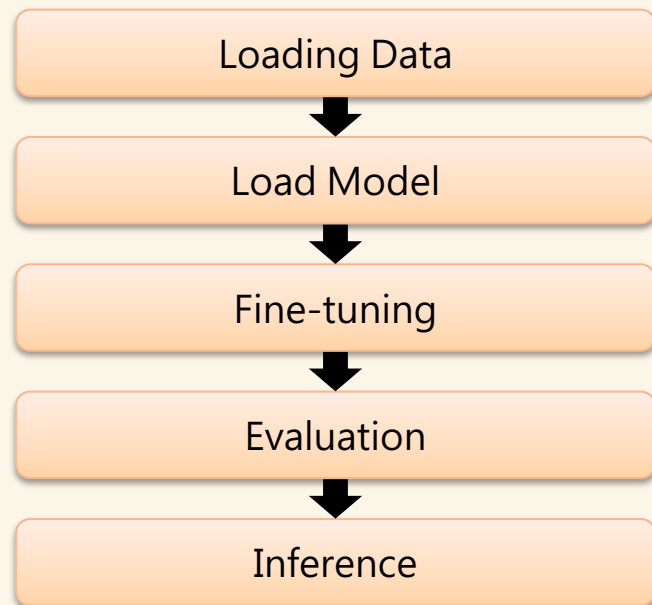
“sentiment”：電影評論的正負向情感

positive 與 negative 各25,000筆

|       | review                                            | sentiment |
|-------|---------------------------------------------------|-----------|
| 0     | One of the other reviewers has mentioned that ... | positive  |
| 1     | A wonderful little production. <br /><br />The... | positive  |
| 2     | I thought this was a wonderful way to spend ti... | positive  |
| 3     | Basically there's a family where a little boy ... | negative  |
| 4     | Petter Mattei's "Love in the Time of Money" is... | positive  |
| ...   | ...                                               | ...       |
| 49995 | I thought this movie did a down right good job... | positive  |
| 49996 | Bad plot, bad dialogue, bad acting, idiotic di... | negative  |
| 49997 | I am a Catholic taught in parochial elementary... | negative  |
| 49998 | I'm going to have to disagree with the previou... | negative  |
| 49999 | No one expects the Star Trek movies to be high... | negative  |

50000 rows x 2 columns

# 實作流程



| 流程一     | 流程二      | 流程三      |
|---------|----------|----------|
| 10,000筆 | 10,000筆  | 50,000筆  |
| CPU     | MPS(GPU) | MPS(GPU) |
| 8,000筆  | 8,000筆   | 40,000筆  |
| CPU     | CPU      | CPU      |
| 2,000筆  | 2,000筆   | 10,000筆  |

# Metal Performance Shaders, MPS (GPU on Mac)

```
# source: https://developer.apple.com/metal/pytorch/
if torch.backends.mps.is_available():
    mps_device = torch.device("mps")
    x = torch.ones(1, device=mps_device)
    print (x)
else:
    print ("MPS device not found.")

tensor([1.], device='mps:0')
```

# Load Model

```
# load pre-trained BERT model and tokenizer
```

```
# 分詞器(tokenizer)用於將文本分割成單詞或子詞，並將其轉化為模型可以理解的輸入格式
```

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

```
# bert-base-uncased 是Hugging Face Model Hub上提供的一個已經訓練好的BERT模型的名稱
```

```
# 這個模型是基於小寫字母的英文文本訓練的 ("uncased"表示不區分大小寫)
```

```
# 創建一個BERT模型，用於序列分類任務，這是一個適用於文本分類任務的BERT模型變體，這個模型也已在大規模文本數據上進行了訓練
```

```
model = BertForSequenceClassification.from_pretrained('bert-base-uncased')
```

# Fine-tuning

```
# BERT fine-tuning
# convert data to PyTorch tensors
input_ids, attention_masks, labels = tokenize_reviews(X_train, y_train)
```

|            | 流程一    | 流程二    | 流程三     |
|------------|--------|--------|---------|
| CPU / MPS  | CPU    | MPS    | MPS     |
| Traing Set | 8,000筆 | 8,000筆 | 40,000筆 |
| 執行時間       | 10s    | 9s     | 53s     |

# Fine-tuning

```
for epoch in range(epochs): # 模型訓練將在多個時期中進行
    model.train() # 將模型切換到訓練模式。在訓練模式下，模型會計算梯度，並在反向傳播時更新模型參數
    for batch in train_data_loader: # 這是一個內部循環，用於遍歷訓練數據集中的所有批次 (batches)。train_data_loader 是:
        input_ids, attention_mask, labels = batch # 從當前批次中解壓出輸入張量 (input_ids和attention_mask) 和標籤 (labels)
        input_ids, attention_mask, labels = input_ids.to(mps_device), attention_mask.to(mps_device), labels.to(mps_device)
        optimizer.zero_grad() # 將模型參數的梯度設置為零。在每個批次之前，需要清零梯度，以便計算新的梯度。這是因為PyTorch會累加梯度
        outputs = model(input_ids, attention_mask=attention_mask, labels=labels) # 將輸入數據傳遞給模型以進行前向傳播
        loss = outputs.loss # 從模型輸出中計算的損失值
        loss.backward() # compute the gradient # 執行反向傳播，計算模型參數的梯度。梯度用於優化器的參數更新
    optimizer.step() # update parameters(optimizer) # 使用優化器來更新模型參數，以最小化損失函數。它執行參數的梯度更新
```

|            | 流程一     | 流程二    | 流程三     |
|------------|---------|--------|---------|
| CPU / MPS  | CPU     | MPS    | MPS     |
| Traing Set | 8,000筆  | 8,000筆 | 40,000筆 |
| 執行時間       | 25m 59s | 6m 30s | 37m 43s |

# Evaluation

```
device = torch.device('cpu')
model.to(device) # 移動到cpu
model.eval() # set to evaluation mode
test_input_ids, test_attention_masks, test_labels = tokenize_reviews(X_test, y_test) # 將文本數據轉換為適用於B
test_input_ids, test_attention_masks, test_labels = test_input_ids.to(device), test_attention_masks.to(dev

with torch.no_grad(): # 這是一個上下文管理器，它告訴PyTorch在接下來的代碼塊中不計算梯度。這對於推理階段非常重要，因為在推
    logits = model(test_input_ids, attention_mask=test_attention_masks) # 分類器，使用訓練好的BERT模型進行推理
    # 模型返回的 `logits` 包含了每個類別的得分或原始預測值。這些得分可以用於計算類別概率，或者通過 `argmax` 操作找到模型的最終預
predicted_labels = np.argmax(logits.logits.cpu().numpy(), axis=1) # 從 `logits` 中找到每個樣本的最高得分的類別，即
# `logits` 是一個PyTorch張量，因此我們使用 `.cpu().numpy()` 將其轉換為NumPy數組，以便更容易進行操作
```

|             | 流程一    | 流程二    | 流程三      |
|-------------|--------|--------|----------|
| CPU / MPS   | CPU    | CPU    | CPU      |
| Testing Set | 2,000筆 | 2,000筆 | 10,000筆  |
| 執行時間        | 59s    | 1m 2s  | 234m 34s |



# Evaluation

|             | 流程一    | 流程二    | 流程三     |
|-------------|--------|--------|---------|
| CPU / MPS   | CPU    | MPS    | MPS     |
| Traing Set  | 8,000筆 | 8,000筆 | 40,000筆 |
| CPU / MPS   | CPU    | CPU    | CPU     |
| Testing Set | 2,000筆 | 2,000筆 | 10,000筆 |
| Accuracy    | 0.8830 | 0.8750 | 0.8924  |
| Precision   | 0.8830 | 0.8753 | 0.8940  |
| Recall      | 0.8830 | 0.8750 | 0.8927  |
| F1 Score    | 0.8830 | 0.8750 | 0.8923  |

# Parameters

```
# 輸出模型的參數數量
def get_learnable_params(module):
    return [p for p in module.parameters() if p.requires_grad]

model_params = get_learnable_params(model)
clf_params = get_learnable_params(model.classifier)

print(f"""
整個分類模型的參數量：{sum(p.numel() for p in model_params)}
線性分類器的參數量：{sum(p.numel() for p in clf_params)}
""")
```

整個分類模型的參數量：109483778  
線性分類器的參數量：1538

# Inference

|      | review                                            | sentiment | predicted_sentiment |
|------|---------------------------------------------------|-----------|---------------------|
| 0    | I really liked this Summerslam due to the look... | positive  | positive            |
| 1    | Not many television shows appeal to quite as m... | positive  | positive            |
| 2    | The film quickly gets to a major chase scene w... | negative  | negative            |
| 3    | Jane Austen would definitely approve of this o... | positive  | positive            |
| 4    | Expectations were somewhat high for me when I ... | negative  | negative            |
| ...  | ...                                               | ...       | ...                 |
| 9995 | Although Casper van Dien and Michael Rooker ar... | negative  | negative            |
| 9996 | I liked this movie. I wasn't really sure what ... | positive  | positive            |
| 9997 | Yes non-Singaporean's can't see what's the big... | positive  | negative            |
| 9998 | As far as films go, this is likable enough. En... | negative  | negative            |
| 9999 | I saw Anatomy years ago -- dubbed at a friends... | positive  | negative            |

10000 rows x 3 columns

# Inference

- 預測為negative，但實際是positive（大小訓練集皆預測錯誤）

What was **always missing** with the Matrix story was how things came to be in the real world. Say no more, because this part of the story covered most of the bases. What was truly interesting was how political it was, maybe even a cheap shot at the current presidential administration. Fascism and violence were the only things man could think of in regards to fighting the robotic horde, who were meant as nothing more than servants to humanity. What I also found interesting was the use of fear and how it was perpetuated by the idea of the unknown. We as humans tend to **fall into that trap** quite often, letting the **lack of logic** and thought overtake us because people **can't believe** the contrary. **Well represented** and put together, this a true testament to how **illogical** humans can be.

➤ 推測判斷錯誤的原因：

影評對於該電影生動地呈現人類不合邏輯的思想與應對給予正面評價，但用了許多負面詞彙去說明電影如何描述「人類」的缺點，可能被模型誤判為「電影」本身的缺點

# Inference

- 預測為positive，但實際是negative（大小訓練集皆預測錯誤）

Deep Sea 3D is a **stunning** insight in to an underwater world only a few have had the opportunity to view first hand. ... It is a **beautifully** captured under sea feast for the eyes.<br /><br />Our ears on the other hand, are **not given** the same treatment. The film is narrated by Hollywood stars Jonny Depp and Kate Winslet. Both sound so **ridiculous** it positively **spoils the enjoyment** of the visuals. ... The script is fairly **silly** and contains very little depth. The soundtrack is filled with strange, unrealistic sound effects which I assume are meant to be funny but in fact detract attention from the material which should have been allowed to speak for itself. ... The film is a **technical marvel** but with it's childish script, **annoying** narration and misplaced sound effects it **cannot be taken seriously**.

➤ 推測判斷錯誤的原因：

影評認為電影的視覺呈現令人驚嘆，但旁白與音效差強人意，

一褒一貶間，模型忽略了「無法讓人認真看待此作品」的重點，錯誤預測成正面評價

# Inference

- 預測為positive，但實際是negative（訓練集增加時預測正確）

I'm **glad** I rented this movie for one reason: its **shortcomings** made me want to read Allende's book and get the full story. <br /><br />Pros: the movie is **beautiful**, the period is depicted well and consistently (to the best of my knowledge), and Meryl and Glenn do **good jobs**.<br /><br />Cons: This is the **worst** acting job I've ever seen from Jeremy Irons--I kept wondering if something was wrong with his mouth. (And I **hate** the terribly English way he says "Transito.") Winona Ryder does **nothing believable** except look young and idealistic. Most of the other performances are OK, but so few things hang together in the character arcs and the relationship development that I was **frustrated** and **angry** well before the end. <br /><br />I'm very curious now whether this movie is typical of Bille August's work. I may have to drop another couple of bucks to rent Smilla's Sense of Snow.

➤ 推測判斷錯誤的原因：

影評中包含glad, shortcomings, good jobs, worst, hate等辭彙，

而影評人高興的原因是因為這部電影的缺點讓他想要讀原著，具有反諷意味

# Inference

- 預測為negative，但實際是positive（訓練集增加時預測正確）

While caricatures and/or references to entertainment industry people or things or even brands of products is **usually** a staple in shorts like this one, they **aren't** used in quantity here. Most of the individual gags are rather **generic**. As I'm going to give examples, there will be spoilers below: ... . The gags are very **good** and it's an **excellent** example of a Bob Clampett cartoon. Clampett had hit his stride as a director by this point and while it **isn't** anywhere near his **best** work, it's nothing to sneeze at either. This short can be found on Looney Tunes Golden Collection, Vol. 3, which is an excellent set that I highly recommend. This short itself is also **recommended**.

➤ 推測判斷錯誤的原因：

影評中提及該短片「未使用」常見元素 → 反而影評人所樂見的，稱讚笑點非常出色

提及該短片遠遠「不是」導演最好的作品 → 當時已是優秀導演，雖不是最佳但依舊推薦



**THANKS!**