

1) 專題實作資料集及探勘目的說明

使用 **Disaster Tweets** 資料集，這是一個由推文組成的文本分類問題，目的是預測每條推文是否與災難相關。資料集的主要目的是對推文進行二分類：是否是與災難有關的推文（**target = 1**），或是與災難無關的推文（**target = 0**）。這項任務有助於了解如何處理自然語言處理（**NLP**）中的分類問題，並應用現有的深度學習模型來解決實際問題。

2) 採用 data mining 方法

使用 **BERT (Bidirectional Encoder Representations from Transformers)** 模型來提取文本特徵，並用 **SVM (Support Vector Machine)** 進行分類。具體方法包括：

- **BERT 模型**：首先，使用 BERT 模型對推文進行語言模型預處理，生成每條推文的特徵向量（logits）。
- **SVM**：用 BERT 提取的特徵向量訓練一個 SVM 模型，並用來進行分類。

3) 程式/環境設定, 執行方式說明

程式設置：

- 使用 **Python** 編程語言，並導入以下關鍵庫：
 - **PyTorch**：用於建立和訓練深度學習模型。
 - **Transformers**：用於載入並使用 BERT 模型。
 - **Scikit-learn**：用於 SVM 模型訓練及計算評估指標（如 **F1** 分數）。
 - **Pandas & Numpy**：用於數據處理和數學計算。

詳細在 **conda** 中裝的內容:

```

(mining) nieves@minun:~/final_project$ conda list
# packages in environment at /datas/store162/nieves/miniconda3/envs/mining:
#
# Name                                Version                                Build                                Channel
_libgcc_mutex                         0.1                                    main
_openmp_mutex                         5.1                                   1_gnu
accelerate                           1.2.1                                pypi_0                                pypi
bzip2                                 1.0.8                                h5eee18b_6
ca-certificates                      2024.11.26                           h06a4308_0
certifi                              2024.12.14                           pypi_0                                pypi
charset-normalizer                   3.4.0                                pypi_0                                pypi
contourpy                            1.3.1                                pypi_0                                pypi
cyclor                               0.12.1                               pypi_0                                pypi
filelock                             3.16.1                               pypi_0                                pypi
fonttools                            4.55.3                               pypi_0                                pypi
fsspec                               2024.12.0                             pypi_0                                pypi
huggingface-hub                     0.27.0                               pypi_0                                pypi
idna                                  3.10                                  pypi_0                                pypi
jinja2                               3.1.4                                pypi_0                                pypi
joblib                               1.4.2                                pypi_0                                pypi
kiwisolver                           1.4.7                                pypi_0                                pypi
ld_impl_linux-64                    2.40                                  h12ee557_0
libffi                               3.4.4                                h6a678d5_1
libgcc-ng                           11.2.0                               h1234567_1
libgomp                             11.2.0                               h1234567_1
libstdcxx-ng                        11.2.0                               h1234567_1
libuuid                              1.41.5                               h5eee18b_0
markupsafe                           3.0.2                                pypi_0                                pypi
matplotlib                           3.10.0                               pypi_0                                pypi
mpmath                               1.3.0                                pypi_0                                pypi
ncurses                              6.4                                  h6a678d5_0
networkx                             3.4.2                                pypi_0                                pypi
numpy                                2.2.0                                pypi_0                                pypi
nvidia-cublas-cu12                  12.4.5.8                             pypi_0                                pypi
nvidia-cuda-cupti-cu12              12.4.127                             pypi_0                                pypi
nvidia-cuda-nvrtc-cu12              12.4.127                             pypi_0                                pypi
nvidia-cuda-runtime-cu12            12.4.127                             pypi_0                                pypi
nvidia-cudnn-cu12                   9.1.0.70                             pypi_0                                pypi
nvidia-cufft-cu12                   11.2.1.3                             pypi_0                                pypi
nvidia-curand-cu12                  10.3.5.147                           pypi_0                                pypi
nvidia-cusolver-cu12                11.6.1.9                             pypi_0                                pypi
nvidia-cuspars-cu12                 12.3.1.170                           pypi_0                                pypi
nvidia-nccl-cu12                    2.21.5                               pypi_0                                pypi
nvidia-nvjitlink-cu12              12.4.127                             pypi_0                                pypi

```

| | | | |
|-------------------|-------------|-----------------|------|
| nvidia-nvtx-cu12 | 12.4.127 | pypi_0 | pypi |
| openssl | 3.0.15 | h5eee18b_0 | |
| packaging | 24.2 | pypi_0 | pypi |
| pandas | 2.2.3 | pypi_0 | pypi |
| pillow | 11.0.0 | pypi_0 | pypi |
| pip | 24.2 | py311h06a4308_0 | |
| protobuf | 5.29.2 | pypi_0 | pypi |
| psutil | 6.1.1 | pypi_0 | pypi |
| pyparsing | 3.2.0 | pypi_0 | pypi |
| python | 3.11.11 | he870216_0 | |
| python-dateutil | 2.9.0.post0 | pypi_0 | pypi |
| pytz | 2024.2 | pypi_0 | pypi |
| pyyaml | 6.0.2 | pypi_0 | pypi |
| readline | 8.2 | h5eee18b_0 | |
| regex | 2024.11.6 | pypi_0 | pypi |
| requests | 2.32.3 | pypi_0 | pypi |
| safetensors | 0.4.5 | pypi_0 | pypi |
| scikit-learn | 1.6.0 | pypi_0 | pypi |
| scipy | 1.14.1 | pypi_0 | pypi |
| sentencepiece | 0.2.0 | pypi_0 | pypi |
| setuptools | 75.1.0 | py311h06a4308_0 | |
| six | 1.17.0 | pypi_0 | pypi |
| sqlite | 3.45.3 | h5eee18b_0 | |
| sympy | 1.13.1 | pypi_0 | pypi |
| threadpoolctl | 3.5.0 | pypi_0 | pypi |
| tiktoken | 0.8.0 | pypi_0 | pypi |
| tk | 8.6.14 | h39e8969_0 | |
| tokenizers | 0.21.0 | pypi_0 | pypi |
| torch | 2.5.1 | pypi_0 | pypi |
| torchvision | 0.20.1 | pypi_0 | pypi |
| tqdm | 4.67.1 | pypi_0 | pypi |
| transformers | 4.47.1 | pypi_0 | pypi |
| triton | 3.1.0 | pypi_0 | pypi |
| typing-extensions | 4.12.2 | pypi_0 | pypi |
| tzdata | 2024.2 | pypi_0 | pypi |
| urllib3 | 2.2.3 | pypi_0 | pypi |
| wheel | 0.44.0 | py311h06a4308_0 | |
| xz | 5.4.6 | h5eee18b_1 | |
| zlib | 1.2.13 | h5eee18b_1 | |

執行方式：

可以把資料準備齊全(final_project.py、train.csv、test.csv、sample_submission.csv)並且下載完需要的套件後，直接在終端機下 python3 final_project.py，跑完後會出現 submission.csv，那就是預測出來的結果

程式邏輯：

1. **資料預處理**：對文本數據進行清理，包括移除 URL、標籤（@username 和 #hashtag）、特殊字符和表情符號。final_project.py 是有 text_clean 的版本，final_project_noclean.py 是沒有 text_clean 的版本，結果顯示沒有 text_clean 的版本預測結果更為準確

以下比較有做 clean_text 跟沒做 clean_text 在每個 epoch 的 loss 值以及總體的 F1 score

| | 有clean_text | 無clean_text |
|-------------|-------------|-------------|
| epoch1 loss | 0.4865 | 0.4705 |
| epoch2 loss | 0.3309 | 0.2985 |
| F1 score | 0.8 | 0.8103 |

2. **文本標記化**：將每條推文轉換為 BERT 可處理的格式（input_ids 和 attention_mask），並進行填充與截斷。
3. **模型訓練**：使用 Trainer 類別訓練 BERT 模型，並根據 F1 分數來選擇最優模型。
4. **SVM 訓練**：將 BERT 提取的特徵餵入 SVM 模型進行訓練，並用它來對驗證集進行預測。
5. **測試與提交**：使用 SVM 對測試集進行預測並生成提交文件。

必要環境：

- Python 3.x
- PyTorch 1.x
- Transformers 4.x
- Scikit-learn 0.24.x

- 需要 GPU 支持來加速訓練，特別是 BERT 模型的處理。

4) 改變控制參數/技術說明(須說明為何想改變控制的想法)

Transformer (BERT) 參數設定：

- **學習率 (Learning Rate)**：設為 $5e-5$ 。這是為了平衡訓練速度與模型收斂的穩定性所做的設置。學習率過高可能會導致模型在訓練過程中震盪，甚至無法收斂；而過低則可能導致訓練過程緩慢，甚至停滯不前。因此，選擇中等的學習率有助於加快收斂並避免過擬合。
- **訓練輪數 (Epochs)**：設為 2。這是為了避免過擬合。BERT 模型本身已經經過了大量預訓練，過多的訓練輪數可能會導致在小範圍數據集上的過擬合。選擇 2 輪可以在不過度訓練的情況下，達到較好的效果。
- **批次大小 (Batch Size)**：設為 48。這個大小的選擇能夠在有限的 GPU 記憶體下有效地加速訓練過程。較大的批次大小可以加速每次迭代的訓練，但需要較多的記憶體；較小的批次大小會導致更頻繁的梯度更新，但可能使訓練時間增長。選擇 48 是在速度與記憶體利用之間的折衷。

SVM (Support Vector Machine) 參數設定：

- **核函數 (Kernel)**：我嘗試了 linear 和 rbf 兩種核函數，來觀察它們對模型性能的影響。在實驗中，linear 表現優於 rbf，因為 BERT 特徵屬於高維數據，且在一個線性空間中能被更有效地區分。同時，linear 核函數還具有運算成本低的優勢，有助於提升計算效率並降低過擬合的風險。

以下是使用兩種模型且在沒做 clean_text 時的 F1 score

| 特徵提取方法 | SVM 核函數 | F1 Score |
|--------|---------|----------|
| BERT | RBf | 0.807 |
| | Linear | 0.81 |

- **懲罰參數 (C)**：默認設置。在 SVM 中，C 參數用來平衡分類器的複雜度與錯誤容忍度。較大的 C 值會鼓勵模型嘗試減少訓練錯誤，但可能會導

致過擬合。較小的 C 值則會讓模型對錯誤有更高的容忍度，但可能會欠擬合。在此選擇了默認的設置(1.0)，因為經過測試，這能夠提供良好的平衡。

為何這樣設置：

- 在設置 **Transformer** 的參數時，學習率和訓練輪數是最常調整的兩個超參數。選擇較小的學習率是為了防止模型在收斂過程中不穩定，並且在有限的輪數內快速達到較好的效果。
- **SVM** 的線性核設置使得它能夠在高維空間（由 **BERT** 特徵提供）上進行高效的分類，並且能夠處理較大的數據集。通過選擇適當的懲罰參數，**SVM** 可以在避免過擬合的情況下達到較好的預測準確度。

5) 評估方法(例如 Accuracy, Error Rate, Precision, Recall, F-measure,執行時間等)

評估指標：

- **F1 分數**：主要評估指標，因為 **F1** 分數綜合了精確率（Precision）和召回率（Recall），對於不平衡類別問題特別有效。這個指標有助於衡量模型在不同類別間的預測能力。

計算方法：





- **F1 分數**是透過對比模型預測的結果（ y_{pred} ）與真實標籤（ y_{true} ）來計算的，使用 `f1_score` 函數來得到。

6) 結果及討論

- **SVM F1 分數**：在驗證集上的 **F1** 分數會被計算並輸出。這反映了 **SVM** 模型在使用 **BERT** 提取的特徵後對災難推文的分類效果。
- **SVM 測試集預測結果**：將 **SVM** 模型對測試集的預測結果儲存在 `submission.csv` 中，這是最終的提交結果。

BERT 負責捕捉語言上下文和語意，而 **SVM** 則作為一個強大的分類器來進行最終預測。

Leaderboard 結果:

| Submission and Description | | Public Score  |
|---|--|--|
|  | submission.csv Complete · 21h ago | 0.83634 |
|  | submission.csv Complete · 1d ago · no clean_text | 0.83634 |
|  | submission.csv Complete · 2d ago | 0.82745 |

第一個為沒做 `clean_text` 且 SVM 選擇 `rbf` 的結果，第二個為沒做 `clean_text` 且 SVM 選擇 `linear` 的結果，第三個是做了 `clean_text` 且 SVM 選 `linear` 的結果