1. 問題描述

本次競賽的目標是通過推文的文本數據，建立一個情感分類模型，將推文分類到不同的情感標籤中（如快樂、憤怒、悲傷等）。以下報告將詳細描述數據處理、特徵工程和模型訓練的過程，並總結關鍵觀察與結果。

2. 數據預處理

2.1 數據導入

通過 Kaggle 的接口，我們下載了競賽的數據集，主要包括：

- **data_identification.csv**: 用於標註推文是訓練集、驗證集還是測試集。
- **emotion.csv**: 每條推文的情感標籤。
- **tweets_DM.json**: 包含推文文本的 JSON 文件。

我們將這些數據合併，形成完整的數據框架。

2.2 文本清理

在推文文本中，包含許多無關的內容（如標籤、用戶提及、URL 等），為了提高模型的性能，我們進行了以下清理：

- 移除 **<LH>** 標籤。
- 移除 hashtags (#) 和用戶提及 (@)。
- 去除 URL。
- 僅保留英文字母和空格，並轉換為小寫。
- 程式如下：

```
# 清理文本數據
def clean_text(text):
    text = re.sub(r"<LH>", "", text)
    text = re.sub(r"#[A-Za-z0-9_]+", "", text)  # 去除hashtags
    text = re.sub(r"@[A-Za-z0-9_]+", "", text)  # 去除用戶提及
    text = re.sub(r"https?://\\S+", "", text)   # 去除URL
    text = re.sub(r"[^a-zA-Z ]", "", text)      # 保留字母和空格
    return text.lower().strip()

data["text"] = data["text"].apply(clean_text)
```

3. 特徵工程

3.1 TF-IDF 向量化

為了將文本數據轉換為數值形式，我們使用了 **TF-IDF** 向量化。TF-IDF 是一種常用的文本特徵提取技術，可以捕捉單詞的重要性，同時降低高頻無用詞的影響。

- 設置 `max_features=1000`，提取最具代表性的 1000 個詞。
- 僅對訓練集中標記為 `"train"` 的數據進行向量化。

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=1000)
X = vectorizer.fit_transform(data[data["identification"] == "train"]["text"])
y = data[data["identification"] == "train"]["emotion"]
```

3.2 特徵與標籤

最終，`X` 為 TF-IDF 特徵矩陣，`y` 為對應的情感標籤

4. 模型選擇與訓練

4.1 模型選擇

選擇了 **Multinomial Naive Bayes** 模型，因為該模型在文本分類任務中特別高效，並且能很好地處理稀疏特徵（如 TF-IDF 特徵矩陣）。

```
# 訓練模型
model = MultinomialNB()
model.fit(X_train, y_train)
```

4.2 訓練與驗證

- 使用 **80%-20%** 的劃分，將數據分為訓練集和驗證集。
- 利用 `MultinomialNB` 訓練模型，並在驗證集上進行性能評估。

5. 結果

Accuracy: 0.42337855059719076

```
Accuracy: 0.42337855059719076

Classification Report:
              precision    recall  f1-score   support

       anger       0.93      0.04      0.08      7964
anticipation       0.56      0.31      0.40     49725
     disgust       0.40      0.06      0.11     27892
        fear       0.88      0.13      0.22     12955
         joy       0.40      0.92      0.56    103089
     sadness       0.40      0.19      0.25     38835
    surprise       0.74      0.04      0.08      9750
       trust       0.65      0.05      0.09     40903

    accuracy                           0.42    291113
   macro avg       0.62      0.22      0.22    291113
weighted avg       0.51      0.42      0.34    291113
```

6. 嘗試與改進

- 增加最大特徵數：從 1000 增加到 3000，雖然準確率有所提升，但模型的計算成本也增加。
- **BOW**模型 **vs. TF-IDF**：詞袋模型的效果略差，TF-IDF 在區分情感詞上更有優勢。
- **hyperparameter**調整：調整 Multinomial Naive Bayes 的平滑參數 `alpha`，`alpha=0.5` 提升了穩定性。

7. 總結

本次情感分類模型基於清理後的推文文本，通過 TF-IDF 特徵提取和 Multinomial Naive Bayes 模型實現了高效的情感分類。模型性能尚有提升空間，可以嘗試：

- 引入情感詞典，進行特徵加權。
- 使用更複雜的深度學習模型（如 LSTM 或 BERT）。
- 分析數據的不均衡性，進一步提升小類別的分類效果。