資料探勘HW2

AI三 B1228005 胡樂麒

4.Report also your heuristics to find such a prototype set

一、輸出結果比較：

1. **KNN**
    - 使用所有訓練樣本並使用交叉驗證，平均準確率為 82.6%。
2. **X_broken**
    - 模擬特徵縮放差異後，準確率降至 73.8%。
    - 說明 KNN 對特徵尺度極度敏感。
3. **Pipeline + MinMaxScaler**
    - 使用正規化修復特徵差異，準確率回升至 82.9%。
4. **1-NN**
    - 使用 Condensed Nearest Neighbor 方法來壓縮樣本從 263 筆訓練資料中，挑出 196 筆樣本。
    - 雖然壓縮後模型在測試集上達到 93.18% 的準確率，但其 10 折交叉驗證的平均準確率為 80.1%，略低於原始模型的 82.6%。這

顯示資料壓縮雖然有效降低樣本數、提升預測效率，卻在某種程度上犧牲了模型的泛化能力與整體穩定性。

```
PS C:\Users\User\Downloads\胡樂麒 資料探勘hw2> python -u "c:\U
The accuracy is 86.4%
The average accuracy is 82.6%
The original average accuracy for is82.6%
The 'broken' average accuracy for is73.8%
The average accuracy for is 82.9%
The pipeline scored an average accuracy for is 82.9%
第三題 condensed prototypes 訓練 1-NN：
原始訓練樣本數：263
Condensed 選出樣本數：196
測試集準確率：93.18%
Condensed set 上交叉驗證平均準確率： 80.10526315789474
```

二、Condensed Nearest Neighbor

在第 3 題中，為了找到最小且可以維持準確度的壓縮樣本，我採用了 Condensed Nearest Neighbor (CNN) 演算法。

CNN 透過 1-NN 分類器不斷檢查訓練樣本是否能被正確分類。若有樣本被誤判，就代表現有樣本集無法準確覆蓋該類型的決策區域，因此會將此樣本加入 condensed set 中。保留對分類結果較為重要的樣本點，濃縮資料並保留分類能力。

CNN 步驟

1. **初始化**

    - 從每個類別中各挑選一筆樣本作為初始 condensed set，確保起始集合中至少有所有類別的代表，避免模型一開始無法判斷特定類別。

2. **建立分類器（1-NN）**

    - 訓練一個 1-NN 模型，對整個訓練資料集進行預測，以檢查目前模型是否能正確分類所有樣本。

3. **動態樣本補充**

    - 對每筆訓練樣本，若模型分類錯誤，代表目前 condensed set 缺少該類型的特徵資訊；則將這筆樣本加入 condensed set。

    - 透過補充被錯誤分類的樣本，逐步強化模型的判斷能力與準確率。

4. **收斂條件**

    - 重複以上步驟，直到整個訓練集都能被目前的 1-NN 模型正確分類為止。

三、結論與反思

在本次作業中，我透過 Condensed Nearest Neighbor（CNN）作為壓縮樣本的選擇策略，從原始 263 筆訓練資料挑選出 196 筆代表性樣本。藉由CNN 的核心概念是只保留位於分類邊界上的關鍵資訊

點，透過不斷測試與補入被錯誤分類的樣本，逐步逼近決策邊界，達到壓縮資料與維持分類能力的平衡。

雖然 CNN 模型在交叉驗證下的平均準確率為 80.1%，略低於原始模型的 82.6%，但在測試集的準確率有 93.18%，代表在特定分布下依然有優秀的分類能力。

綜上所述， CNN 不僅能顯著減少樣本數量，在不顯著犧牲準確率的情況下，進行有效分類。