國立中山大學 資訊工程學系
資料探勘期中書面報告

# 應用分類演算法進行糖尿病患者預測

**指導教授** 蔡崇煒 教授
第 13 組
B083040026 連宥鳴　　B093022044 許峻賢
B096060041 李彥德*　　B102010038 董宥弦

## 1. 摘要

本研究以 K-Nearest Neighbor(KNN) 及隨機森林 (random forest) 演算法，來判斷該病人是否罹患糖尿病。資料集分為實驗 A 和 B 兩資料集，其中皆包含訓練集和測試集，內容為數百位病人的各項測量值(如血液中葡萄糖/胰島素、舒張壓等)。資料經過清理去除異常值，再將資料分為是否進行類別平衡進行分類，每次實驗皆分為 3 組，原始實驗資料、Min-Max 正規化後的資料及 Z-score 標準化後的資料，最後再由平均及最高準確率，觀察 3 種演算法的結果。

## 2. 簡介

現今人們對於醫學健康問題愈來愈重視，近年來，隨著大數據和人工智慧的發展，利用演算法來預測各種疾病風險已經成為熱門的研究方向之一。其中，糖尿病是一種常見的慢性疾病，而其發病率在全球快速增加。因此，利用演算法來預測是否罹患糖尿病對於提高醫療水平和預防糖尿病的發生具有重要意義。本篇報告，我們將利用 KNN 演算法及隨機森林 (random forest) 來分析實驗 A 及 B 兩資料集，透過病人的各項測量值（如血液中葡萄糖/胰島素、舒張壓等），來判斷該病人是否罹患糖尿病。

## 3. 相關研究

### 3.1 資料前處理

#### A. 清理異常值資料

由於分類的準確度與資料的品質息息相關，因此在進行分類訓練前需要將資料進行清理，刪除含缺失值與異常值的資料。本研究所使用資料集中以下 6 項資料不應為 0，因此若其出現為 0 的數據，則刪除該筆訓練資料：

1. Glucose
2. BloodPressure
3. SkinThickness
4. Insulin
5. BMI
6. Age

#### B. Min-Max 正規化

Min-Max 正規化是一種常見的數據正規化方法，用於將數據範圍縮放到 [0,1] 或 [-1,1] 。Min-Max 正規化通常應用於機器學習和數據分析領域中，改善模型性能和準確率。

#### C. Z-score 標準化

Z-score 標準化是一種常見的數據標準化方法，它可以將數據轉換為標準常態分佈（均值為 0，標準差為 1）。

### 3.2 演算法

#### A. K-Nearest Neighbor (KNN)

KNN 演算法的基本概念是找到 k 個最接近的訓練樣本，若 k 太大，可能會造成欠擬合，使準確率過低；而 k 太小的話，會造成過度擬合，雖訓練集的準確率高，但測試集的準確率相對較低，這是因為過度擬合的模型在訓練集上學習了過多的特定細節和噪聲，而這些細節和噪聲對測試集上的預測效果沒有幫助，反而使得預測性能下降。根據距離遠近有不同的權重，常用的距離為曼哈頓距離、歐機里得距離。其中一種權重算法為 1/d (d 為距離) ，即距離愈近權重愈高。研究者們通常針對此演算法的優化和改進進行研究，例如透過改進距離度量方式、降低維度、使用適當的近鄰數量等方法來提高算法的準確性和效率。KNN 演算法也應用於多個領域，例如圖像識別、音頻識別、自然語言處理等。

#### B. 隨機森林

隨機森林是一個包含多個決策樹的分類器。研究者主要關注如何提高隨機森林的準確性和效率，例如通過優化決策樹的結構、選擇更明顯的特徵子集、增加樹的數量等方法。此外，隨機森林也應用於圖像分類、文本分類、股票預測等多個領域。

## 4.程式設計方法&討論

本研究分為三大部分，演算法包含 KNN、隨機森林與兩種方法；資料處理部分包含資料清理、資料平衡、正規化方法與繪圖顯示；分類器則是集合所有演算法一次比較所有演算法與方法的準確度差異。

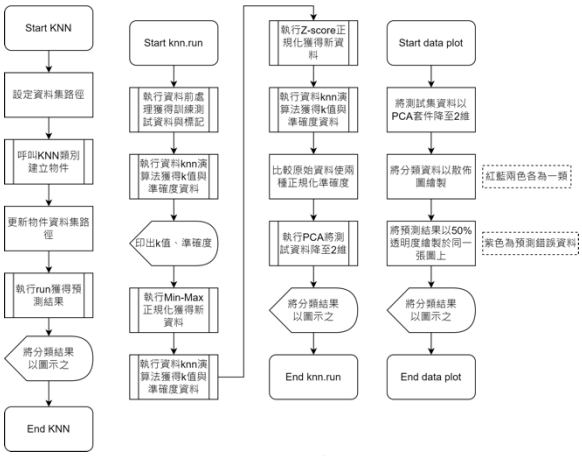
### 4.1 演算法

#### 4.1.1 KNN 演算法

本研究之演算法皆使用 python 語言撰寫，於 KNN 演算法中僅使用 numpy 建立演算法運算，未使用任何演算法套件進行分類。演算法建立在 class (KNearestNeighbor) 類別中，並繼承至 class (KNN) 中，本研究在演算法中特別設計尋找最佳 K 值的流程，預設在 K=1~30 中尋找最佳值。執行 KNN 演算法的流程如下圖，在設定資料集路徑後可建立 KNN 物件，並更新演算法的資料集路徑，呼叫物件中 run 函式可獲得預測結果，最後將結果以圖示

之。

分類結果是以主成分分析 (Principal Component Analysis, PCA) 方法將資料降維至 2 維，以散佈圖的方式呈現，圖中藍色與紅色圓點各為一類，而顯示為紫色的類別為預測錯誤之結果。



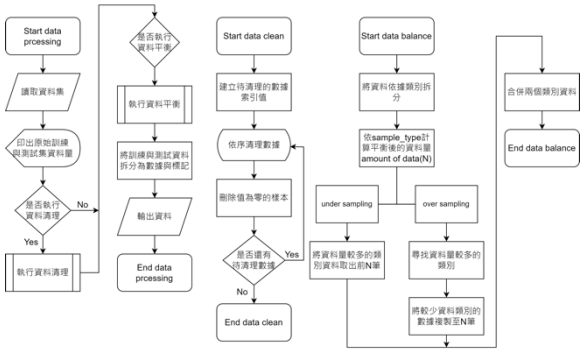

圖一、KNN 演算法流程圖

### 4.1.2 隨機森林演算法

演算法使用 python 語言撰寫，以 sklearn 套件建立演算法物件，森林樹木設定為 100，將資料集輸入進行預測，將分類結果以圖示之。

## 4.2 資料處理

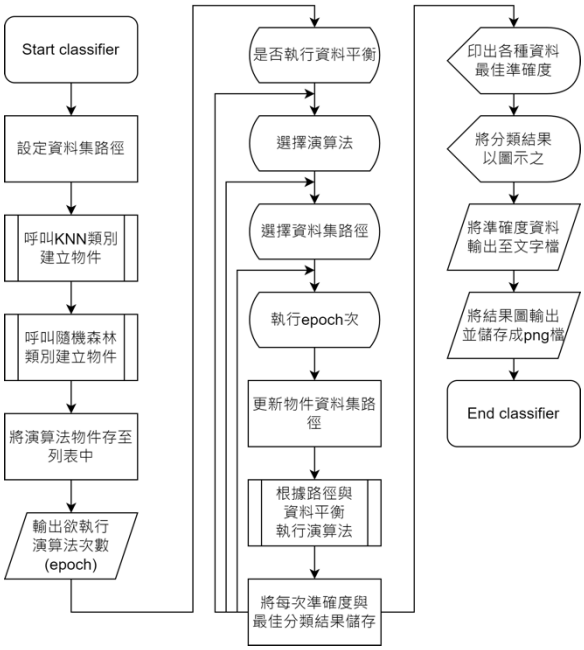
資料前處理包含資料清理資料類別平衡，本研究資料處理流程圖如下。首先讀取資料集，將原始資料量印出後執行資料清理，將 3.1A 所述 6 樣加入待清理的索引值，並刪除缺失值與為 0 的樣本資料；當資料的兩種類別資料量差異過大時，會導致分類結果不佳，而資料平衡常見的方法有 over sampling 與 under sampling 兩種，在進行資料分類時會希望訓練資料越多越好，over sampling 是一種將較少的資料類別進行複製，使各類別的資料量平衡；而 under sampling 則是將較多的資料類別減少樣本使資料量平衡，實現方法如下圖所示。




圖二、資料前處理流程圖

## 4.3 演算法整合分類器

為了方便觀察兩種演算法在資料平衡條件與不同資料集下的結果，本研究設計分類器方便一次性操作，程式流程圖如下，首先設定所有資料集路徑，建立演算法物件，並決定欲執行次數 (epoch) ，依是否執行資料平衡 (yes/no) 、演算法選擇 (KNN/隨機森林) 、資料集路徑 (實驗 A/實驗 B) ，執行 epoch 次尋找每一種組合準確度的平均與最高，並將最佳分類結果儲存於圖檔中。




圖三、分類器流程圖

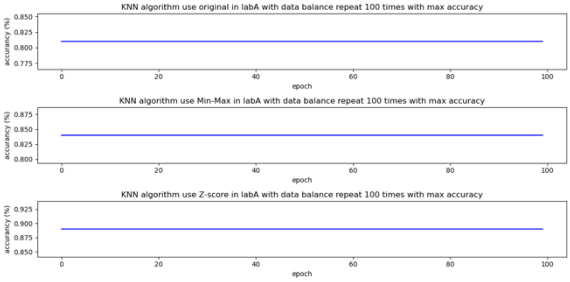
## 4.2 實驗結果

本次實驗以 k=1~30，隨機森林樹木為 100，執行次數 epoch 為 100 次進行預測。

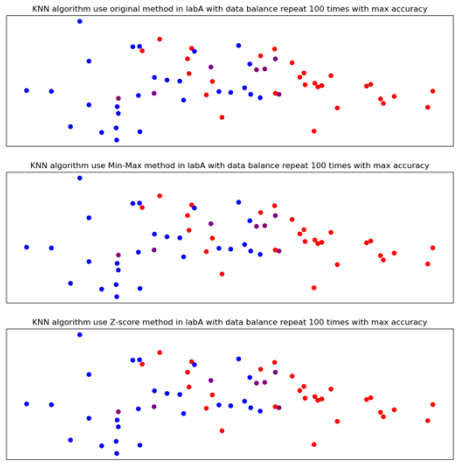
總共 8 種組合且各有 3 種正規化資料進行預測，其中 KNN 演算法於實驗 A 資料集，進行資料平衡後的 Z-score 標準化有著最高的準確度 88.71%。

表一、演算法與資料處理後之最高準確度（單位：%）

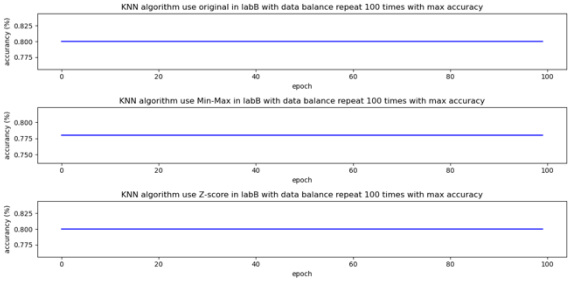
| | data | With data balance | | Without data balance | |
|---|---|---|---|---|---|
| | | KNN | Random Forest | KNN | Random Forest |
| Lab A | original | 80.65 | 88.24 | 81.82 | 86.87 |
| | Min-Max | 83.87 | 80.15 | 84.85 | 78.79 |
| | Z-score | 88.71 | 87.50 | 84.85 | 85.86 |
| Lab B | original | 80.43 | 82.76 | 76.92 | 80.77 |
| | Min-Max | 78.26 | 84.48 | 78.85 | 86.54 |
| | Z-score | 80.43 | 82.76 | 75.00 | 80.77 |



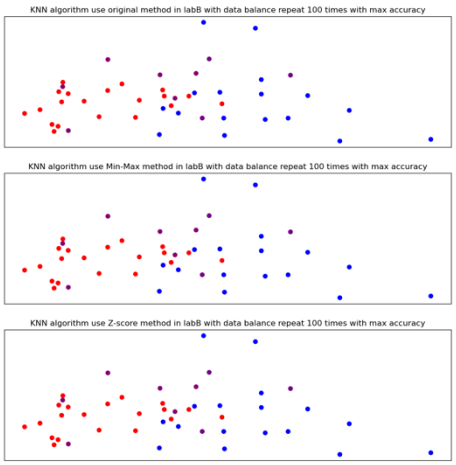

圖四、實驗 A 經資料平衡後利用 KNN 演算法迭代 100 次之準確度變化



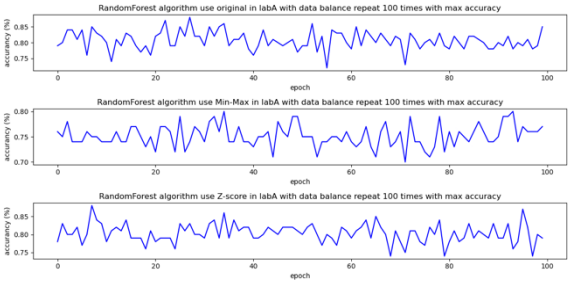

圖五、實驗 A 經資料平衡後利用 KNN 演算法與不同標準化的分類結果



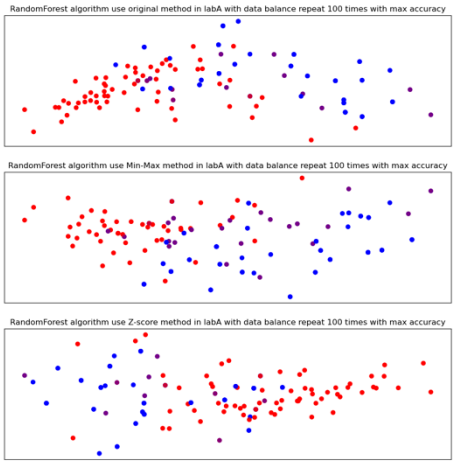

圖六、實驗 B 經資料平衡後利用 KNN 演算法迭代 100 次之準確度變化




圖七、實驗 B 經資料平衡後利用 KNN 演算法與不同標準化的分類結果

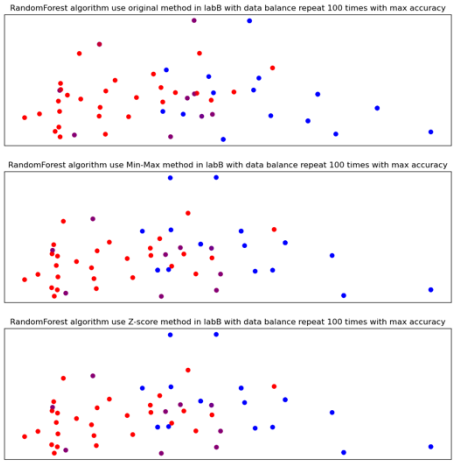



圖八、實驗 A 經資料平衡後利用隨機森林演算法迭代 100 次之準確度變化

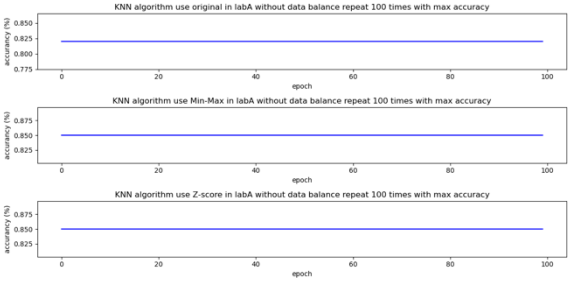



圖九、實驗 A 經資料平衡後利用隨機森林演算法與不同標準化的分類結果

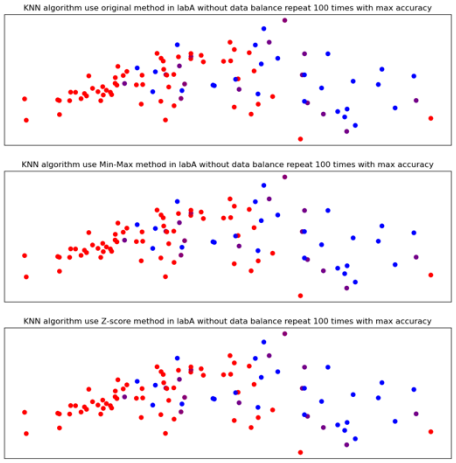

圖十、實驗 B 經資料平衡後利用隨機森林演算法迭代 100 次之準確度變化



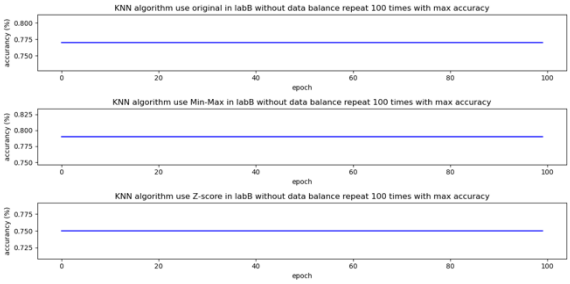

圖十一、實驗 B 經資料平衡後利用隨機森林演算法與不同標準化的分類結果



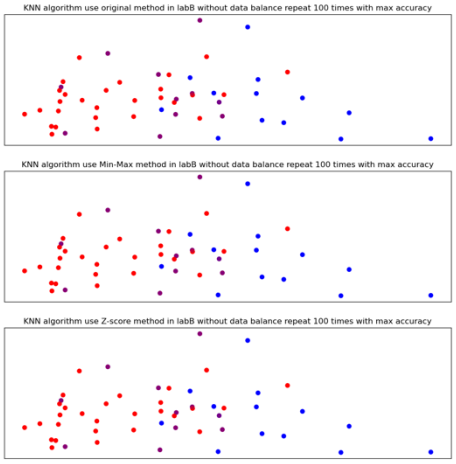

圖十二、實驗 A 未經資料平衡利用 KNN 演算法迭代 100 次之準確度變化




圖十三、實驗 A 未經資料平衡利用 KNN 演算法與不同標準化的分類結果

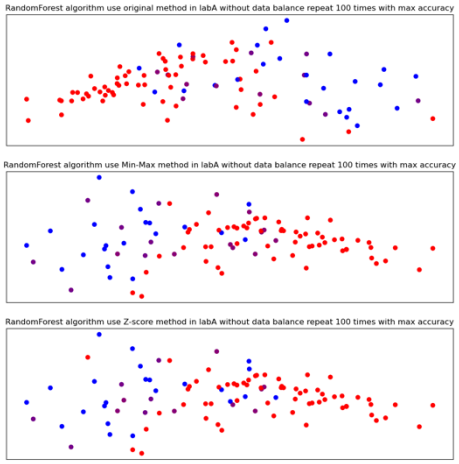



圖十四、實驗 B 未經資料平衡利用 KNN 演算法迭代 100 次之準確度變化



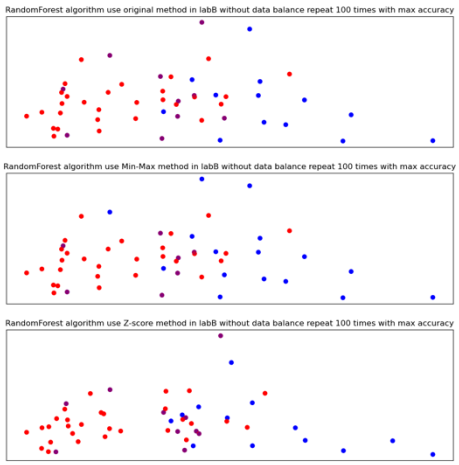

圖十五、實驗 B 未經資料平衡利用 KNN 演算法與不同標準化的分類結果


圖十六、實驗 A 未經資料平衡利用隨機森林演算法迭代 100 次之準確度變化




圖十七、實驗 A 未經資料平衡利用隨機森林演算法與不同標準化的分類結果


圖十八、實驗 B 未經資料平衡利用隨機森林演算法迭代 100 次之準確度變化




圖十九、實驗 B 未經資料平衡利用隨機森林演算法與不同標準化的分類結果

## 5.結論

Z-score標準化對KNN及隨機森林演算法有較好的準確率，Min-Max正規化對KNN有較好的準確率，但對Random Forest大多則較差。因此資料處理的方法不同、使用的演算法不同，對準確率可能都會造成正面或負面的影響，因此需要透過不斷的實驗及比較，才能找到最適合的配對，使準確率提高。

## 6.參考文獻

[1]. ML Zhang, ZH Zhou, “ML-KNN: A lazy learning approach to multi-label learning’’ Pattern recognition, 2007.

[2]. S Zhang, X Li, M Zong, X Zhu, D Cheng, “Learning k for KNN Classification’’ dl.acm.org, 2017.

[3]. S Patro, KK Sahu, “Normalization: A preprocessing stage’’ arXiv preprint arXiv:1503.06462, 2015.

[4]. IB Mohamad, D Usman, “Standardization and its effects on K-means clustering algorithm’’ Research Journal of Applied Sciences, 2013.

[5]. SJ Rigatti, “Random forest’’ Journal of Insurance Medicine, 2017