

國立交通大學

資訊科學系

碩士論文



資料探勘技術於病人疼痛自控裝置之應用與分析

Data Mining and Analysis in Patient Controlled Analgesia

研究生：吳欣儒

指導教授：胡毓志 教授

中華民國一百年七月

資料探勘技術於病人疼痛自控裝置之應用與分析

Data Mining and Analysis in Patient Controlled Analgesia

研究生：吳欣儒

Student : Shin-Ru Wu

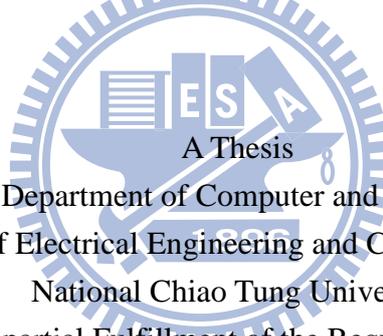
指導教授：胡毓志

Advisor : Yuh-Jyh Hu

國立交通大學

資訊科學系

碩士論文



Submitted to Department of Computer and Information Science
College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

資料探勘技術於病人疼痛自控裝置之應用與分析

學生：吳欣儒

指導教授：胡毓志教授

國立交通大學資訊科學系(研究所) 碩士班

摘 要

病患在接受手術後通常會發生疼痛的現象，嚴重的疼痛可能會影響到傷口的癒合。因此，如何有效地減緩術後疼痛現象是相當重要的。病人疼痛自控止痛法 (Patient-Controlled Analgesia ,PCA) 相較於傳統肌肉注射止痛法 (Intramuscular Injection ,IM) 更能有效且快速地減緩疼痛現象。本研究應用資料探勘技術，用以預測病患短期未來之內使用 PCA 裝置之情況。分析的資料來源為彰化基督教醫院，透過麻醉醫師的協助，我們收集了 1099 筆病患在術後使用 PCA 裝置的記錄。我們做了以下的預測分析：(1) 麻醉藥劑量預測，(2) PCA 裝置參數設定調整預測。將預測及屬性分析結果提供給麻醉醫師參考，藉此更了解病患使用 PCA 裝置之情況，而能更有效地為病患減輕術後疼痛。

關鍵字：病人疼痛自控、資料探勘、類別不平衡、資料清理

Data Mining and Analysis in Patient Controlled Analgesia

Student : Shin-Ru Wu

Advisor : Dr. Yuh-Jyh Hu

Department (Institute) of Computer Science
National Chiao Tung University

ABSTRACT

Effective pain control is particularly important after surgery, as pain can cause significant distress to patients, and affect wound healing. PCA (Patient-Controlled Analgesia) is more effectively and more quickly than IM (Intramuscular Injection) in the management of postoperative pain. By applying data mining techniques, this study aimed to predict the situation that patients using PCA devices in the short-term future. With the assistance of Changhua Christian Hospital, we collected 1099 PCA patient records. We concentrated on two prediction tasks in this study: (1) postoperative analgesic consumption, and (2) PCA setting readjustment. The result of prediction and feature analysis will be available to the anesthesiologist property reference, to better understand the situation of patients using PCA devices, and more effectively reduce postoperative pain for patients.

Keywords : Patient-Controlled-Analgesia · Data mining · Class imbalance · Data cleaning

目 錄

| | |
|---|-----|
| 摘 要 | i |
| ABSTRACT | ii |
| 目 錄 | iii |
| 表目錄 | v |
| 圖目錄 | vii |
| 一、 序論 | 1 |
| 1.1 背景介紹 | 1 |
| 1.2 研究動機 | 2 |
| 二、 相關方法研究 | 3 |
| 2.1 困難處 | 3 |
| 2.2 學習演算法(learning algorithms) | 4 |
| 2.2.1 簡單貝氏分類器(Naive Bayes Classifier) | 4 |
| 2.2.2 類神經網路(Artificial Neural Networks ,ANN) | 4 |
| 2.2.3 支援向量機(Support Vector Machines ,SVM) | 5 |
| 2.2.4 決策表(Decision Table) | 6 |
| 2.2.5 決策樹(Decision Tree) | 7 |
| 2.2.6 五種學習演算法之優缺點 | 8 |
| 2.3 整體學習演算法(ensemble learning algorithm) | 9 |
| 2.3.1 Bagging and Random Forest | 10 |
| 2.3.2 AdaBoost | 11 |
| 2.3.3 Stacking | 12 |
| 2.4 類別不平衡(class imbalance) | 14 |
| 2.4.1 減少多數類別取樣法(under sampling) | 17 |
| 2.4.2 增加少數類別取樣法(over sampling with replacement) | 18 |
| 2.4.3 DataBoost-IM | 18 |
| 2.4.4 資料清理(data cleaning) | 21 |
| 三、 實驗說明 | 23 |
| 3.1 資料收集 | 23 |
| 3.2 資料前處理 | 23 |
| 3.3 實驗 | 26 |
| 3.3.1 總劑量預測 | 26 |
| 3.3.2 PCA劑量預測 | 26 |
| 3.3.3 PCA裝置參數設定調整預測 | 27 |
| 3.4 評量準則 | 28 |
| 3.4.1 總劑量預測和PCA劑量預測 | 28 |
| 3.4.2 PCA裝置參數設定調整預測 | 29 |
| 3.4.3 檢定方法 | 30 |

| | |
|--------------------------|----|
| 四、 結果與討論 | 32 |
| 4.1 預測分析 | 32 |
| 4.2 實驗結果與討論 | 32 |
| 4.2.1 五種學習演算法之結果比較 | 32 |
| 4.2.2 整體學習演算法之結果比較 | 39 |
| 4.2.3 屬性分析 | 47 |
| 4.2.4 Lesion study..... | 62 |
| 五、 結論 | 72 |
| 參考文獻 | 75 |



表目錄

| | |
|--|----|
| 表 1. 五種學習演算法之優缺點比較..... | 9 |
| 表 2. 所有屬性之說明..... | 24 |
| 表 3. Confusion Matrix..... | 29 |
| 表 4. 總劑量預測的評量準則之說明..... | 29 |
| 表 5. Confusion Matrix..... | 30 |
| 表 6. PCA裝置參數設定調整預測的評量準則之說明..... | 30 |
| 表 7. 總劑量預測五種演算法之比較..... | 33 |
| 表 8. PCA劑量預測五種演算法之比較..... | 34 |
| 表 9. PCA裝置調整設定預測分析五種學習演算法之比較，資料未刪除雜訊..... | 34 |
| 表 10. PCA裝置調整設定預測分析五種學習演算法之比較，資料未刪除雜訊，搭配減少多數類別取樣法(under sampling)..... | 35 |
| 表 11. PCA裝置調整設定預測分析五種學習演算法之比較，資料未刪除雜訊，搭配增加少數類別取樣法(over sampling)..... | 36 |
| 表 12. PCA裝置調整設定預測分析五種學習演算法之比較，資料已刪除雜訊..... | 38 |
| 表 13. PCA裝置調整設定預測分析五種學習演算法之比較，資料已刪除雜訊，搭配減少多數類別取樣法(under sampling)..... | 39 |
| 表 14. PCA裝置調整設定預測分析五種學習演算法之比較，資料已刪除雜訊，搭配增加少數類別取樣法(over sampling)..... | 39 |
| 表 15. 總劑量預測分析之結果..... | 41 |
| 表 16. PCA劑量預測分析之結果..... | 42 |
| 表 17. PCA裝置調整設定預測分析之結果，資料尚未刪除雜訊..... | 45 |
| 表 18. PCA裝置調整設定預測分析之結果，資料已刪除雜訊..... | 46 |
| 表 19. 學習演算法和整體學習演算法表現較佳者之相互比較..... | 47 |
| 表 20. 總劑量預測分析，前五層中出現頻率為前 10 的屬性名稱及該屬性在該層之出現頻率..... | 54 |
| 表 21. 總劑量預測分析，十個最為關鍵之屬性與分數..... | 54 |
| 表 22. 總劑量預測分析，contidose屬性和總劑量分類結果之單因子變異數分析..... | 55 |
| 表 23. 總劑量預測分析，pcadose屬性和總劑量分類結果之單因子變異數分析..... | 56 |
| 表 24. 總劑量預測分析，生理資訊相關屬性與分類結果之相關性分析..... | 56 |
| 表 25. PCA劑量預測分析，前五層中出現頻率為前 10 的屬性名稱及該屬性在該層之出現頻率..... | 57 |
| 表 26. PCA劑量預測分析，十個最為關鍵之屬性與分數..... | 57 |
| 表 27. PCA劑量預測分析，pcadose屬性和PCA劑量分類結果之單因子變異數分析..... | 58 |
| 表 28. PCA劑量預測分析，p_timediff_mean屬性和PCA劑量分類結果之單因子變異數分析..... | 59 |
| 表 29. PCA劑量預測分析，生理資訊相關屬性與分類結果之相關性分析..... | 59 |
| 表 30. PCA裝置調整設定預測分析，資料已刪除雜訊，前五層中出現頻率為前 10 的屬 | |

| | |
|---|----|
| 性名稱及該屬性在該層之出現頻率 | 60 |
| 表 31. PCA裝置調整設定預測分析，十個最為關鍵之屬性與分數 | 60 |
| 表 32. PCA裝置調整設定預測分析，p_timediff_var屬性和是否調整PCA裝置參數之單因子變異數分析 | 61 |
| 表 33. PCA裝置調整設定預測分析，pcamode_set屬性和是否調整PCA裝置參數之單因子變異數分析 | 62 |
| 表 34. PCA裝置調整設定預測分析，生理資訊相關屬性與分類結果之相關性分析 | 62 |
| 表 35. 總劑量預測分析之結果，屬性使用病患之生理資訊及手術相關屬性 | 64 |
| 表 36. 總劑量預測分析之結果，屬性使用病患使用PCA裝置之相關屬性 | 64 |
| 表 37. PCA劑量預測分析之結果，屬性使用病患之生理資訊及手術相關屬性 | 66 |
| 表 38. PCA劑量預測分析之結果，屬性使用病患使用PCA裝置之相關屬性 | 66 |
| 表 39. PCA裝置調整設定預測分析之結果，資料尚未刪除雜訊，屬性使用病患之生理資訊及手術相關屬性 | 68 |
| 表 40. PCA裝置調整設定預測分析之結果，資料尚未刪除雜訊，屬性使用病患使用PCA裝置之相關屬性 | 69 |
| 表 41. PCA裝置調整設定預測分析之結果，資料已刪除雜訊，屬性使用病患之生理資訊及手術相關屬性 | 70 |
| 表 42. PCA裝置調整設定預測分析之結果，資料已刪除雜訊，屬性使用病患使用PCA裝置之相關屬性 | 71 |



圖目錄

| | |
|---|----|
| 圖 1. 具類別不平衡及資料鬆散特性之資料集範例 | 3 |
| 圖 2. 類神經網路之示意圖 | 5 |
| 圖 3. 支援向量機之示意圖 | 6 |
| 圖 4. 決策表之範例 | 7 |
| 圖 5. 根據天氣情況分類「是否適合打網球(PlayTennis)」之決策樹 | 7 |
| 圖 6. 「是否適合打網球(PlayTennis)」決策樹之if-else規則 | 8 |
| 圖 7. Bagging演算法之流程圖 | 10 |
| 圖 8. AdaBoost演算法 | 12 |
| 圖 9. Stacking演算法之流程圖 | 13 |
| 圖 10. Stacking演算法之簡單範例 | 14 |
| 圖 11. 類別不平衡及資料稀疏與學習困難度之相關性探討圖例 | 17 |
| 圖 12. DataBoost-IM演算法 | 20 |
| 圖 13. 資料清理演算法之示意圖例 | 22 |



一、序論

1.1 背景介紹

術後疼痛是一種必然的現象，當病患接受手術後，大多需要一段時間恢復，在這期間病患必須忍受因為手術造成的疼痛。至於如何讓病患減緩術後疼痛，目前手術後疼痛控制較好的方式有：(1) 傳統肌肉注射止痛(Intramuscular Injection ,IM)，(2) 病人疼痛自控(Patient-Controlled Analgesia ,PCA)。

傳統肌肉注射止痛(IM)的施用方式為當病患感到疼痛時，須按鈴呼叫麻醉醫師或護士，經由肌肉注射的方式注射麻醉藥，通常需要花費較長的時間才能達到止痛的效果。然而病人疼痛自控(PCA)，則是透過一台由微電腦控制的裝置。首先將麻醉藥注入藥袋中，再將藥袋放入 PCA 裝置裡，藉由該裝置控制麻醉藥的注射。當病患感到疼痛時，PCA 裝置可經由病患按壓按鈕的動作，將麻醉藥注射至病患體內，如此便可在較短的時間內，達到控制術後疼痛的效果。文獻裡提到，使用病人疼痛自控(PCA)比使用傳統肌肉注射止痛(IM)方式更能有效的達到控制術後疼痛[1,2]。

使用 PCA 裝置並非毫無限制，病患在使用該裝置之前，必須先由麻醉醫師依照病患的身體狀況，設定適合該病患的麻醉藥劑量和使用該裝置的安全範圍，其中安全範圍包括「鎖定時間(lockout)」以及「四小時限制(4 hour limit)」兩種設定。「鎖定時間(lockout)」控制在一定的時間內 PCA 裝置給藥的次數，例如，設定是 10 分鐘，即表示病患在 10 分鐘內，不管做了幾次的按壓動作，PCA 裝置只會給一次的麻醉藥。「四小時限制(4 hour limit)」則是控制在四個小時內，PCA 裝置給藥的總劑量，例如，當其設定是 10ml 時，即表示 PCA 裝置在四個小時內，只會允許病患被注射 10ml 以內的麻醉藥，若超過該範圍，PCA 裝置則不給藥。藉由這兩種安全範圍的設定，我們較可確保病患使用病人疼痛自控(PCA)裝置的安全性。因此，如何設定該裝置的參數以及預測其所影響的麻醉劑量就益顯重要。

當病患使用 PCA 裝置時，若產生噁心嘔吐的副作用或是無法減輕疼痛等不適的現象時，醫師必須為病患更改適合該病患的 PCA 裝置參數。為此我們希望透過資料探勘

的技術，在病患使用 PCA 裝置的一段時間(例：24 小時)之後，透過病患使用 PCA 的行為模式，再配合病患的生理資料以及手術相關資料，來預測病患在短期未來之內 (例：72 小時)所需的麻醉藥劑量為何，以及預測該病患在之後的 PCA 使用上(例：第 24 小時到第 72 小時)是否需要醫師調整 PCA 裝置的參數。

雖然 PCA 的使用上較方便，但也並非所有病患都適合使用病人疼痛自控止痛法，因為該方法病患必須透過自行使用 PCA 裝置，所以必須有體力且了解如何操作的病患才較為適合，像是有癱瘓病症的病患就無法使用。對於年紀較小的兒童，只要了解操作方式也是適合使用的，在臨床中已有七歲左右的兒童使用該方法改善術後疼痛[3]，不過還是需要麻醉醫師以及護士密切注意因藥物所帶來的副作用。

病人疼痛自控止痛法的目標就是希望透過病患自行控制，在更適合的時候使用麻醉藥來減輕術後疼痛，如此在術後恢復期間才可以得到更好的休息，且根據一些臨床經驗，使用病人疼痛自控止痛法的病患所需的麻醉藥劑量大多比使用傳統肌肉注射止痛(IM)法的所需劑量較少[3]。

1.2 研究動機

此研究的目的，是希望可以透過資料探勘的技術，讓麻醉醫師更能了解病患使用 PCA 裝置的情況，以期更有效的減輕病患的術後疼痛。目前我們完成以下三種預測的分析：(1)麻醉藥需求總劑量、(2)病患透過按壓 PCA 裝置被注射之麻醉藥需求劑量，和(3)當病患使用 PCA 裝置一段時間後，是否需要調整 PCA 裝置參數設定。

二、相關方法研究

2.1 困難處

PCA 裝置的按壓與否取決於病患感受到的疼痛程度，但因為疼痛是相當主觀的意識，因而每個人忍受疼痛的程度有所差異。即使是身體狀況類似的兩個人，對同樣的刺激也可能有不同疼痛程度的感受，因此，純粹從身體狀況來判斷該病患的 PCA 使用情況是相當困難的。此外，就 PCA 裝置參數設定調整的預測分析而言，只有少數的病患在使用該裝置一段時間，PCA 裝置的參數需要被調整，因此我們尚須面對類別不平衡(class imbalance)的問題，以本篇論文的資料來看，只有 18.9%的病患需要調整 PCA 裝置參數設定。另外，資料中有相當多的雜訊(noise)，我們將少數需要調整參數設定的病患以 1967 年 J. B. MacQueen 提出的 k-means 演算法[4]作分群，再以各群質心為圓心，該群所有資料中距質心最遠的資料之距離為半徑，發現多數不需要被調整參數設定的病患有 99.875%都在以此圓心半徑畫出來的邊界之內，如圖 1 所示，由此可看出我們的資料是相當稀疏的，這些因素更增加了預測工作的困難度。

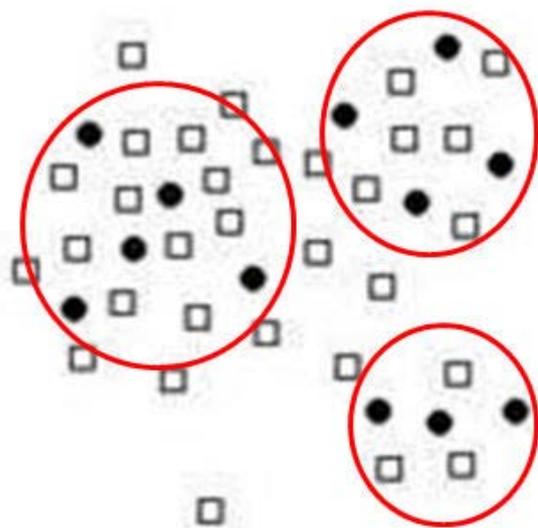


圖 1. 具類別不平衡及資料鬆散特性之資料集範例

白色的點為多數不需要調整參數設定的病患，黑色的點為少數需要調整參數設定的病患。將少數需要調整參數設定之病患即黑色點做分群，圓圈為分群後以各群質心為圓心，距質心最遠的資料之距離為半徑畫出之邊界。多數不需要調整參數設定的病患 99.875% 都在其邊界之內，可看出資料是相當稀疏的。

2.2 學習演算法(learning algorithms)

首先，我們使用幾個資料探勘常見的學習演算法，包括決策樹(Decision Tree)、貝氏分類器(Bayesian Classifier)、類神經網路(Artificial Neural Networks ,ANN)[7]、支援向量機(Support Vector Machines ,SVM)和法則歸納法(Rule Induction)。

決策樹使用 Ross Quinlan 發展的 C4.5 演算法[5]，貝氏分類器使用簡單貝氏分類器(Naive Bayes Classifier)[6]，支援向量機使用 Chih-Jen Lin 等人發展的 LIBSVM[8]，法則歸納法使用決策表(Decision Table)[9]。

2.2.1 簡單貝氏分類器(Naive Bayes Classifier)

貝氏分類器以貝氏定理(Bayesian Theorem)為基礎，是一種基於統計機率所發展的演算法，假設有 n 個屬性 X_1, X_2, \dots, X_n ， t 個類別值 c_1, c_2, \dots, c_t ，要判斷一筆資料 $x=(x_1, x_2, \dots, x_n)$ 的類別為何，是根據事後機率 $p(c_j|x)$ 來判定，而此機率的計算方式如下：

$$p(c_j|x) = \frac{p(x, c_j)}{p(x)} = \frac{p(x|c_j)p(c_j)}{p(x)} ; j = 1, 2, \dots, t \quad (2.1)$$

由式(2.1)，可以得到以下

$$p(c_j|x) \propto p(x|c_j)p(c_j) \quad (2.2)$$

若有一類別值 c_i 滿足

$$p(c_i|x) = \max_{j=1,2,\dots,t} p(c_j|x) \quad (2.3)$$

則 x 的類別值就會被判定為 c_i 。

而簡單貝氏分類器[6]，是假設在類別值給定的條件下，屬性間彼此互相獨立，在此假設下，可將式(2.2)轉變成

$$p(c_j|x) \propto p(x|c_j)p(c_j) = p(x_1, x_2, \dots, x_n|c_j)p(c_j) = p(c_j) \prod_{i=1}^n p(x_i|c_j) \quad (2.4)$$

再利用式(2.3)，就可以得到 x 的類別值。

2.2.2 類神經網路(Artificial Neural Networks ,ANN)

類神經網路[7]是模仿生物神經網路的能力。類神經元是生物神經元的簡單模擬，它

從外界環境或者其它類神經元取得資訊，並加以簡單的運算，再輸出其結果至外界環境或者其它類神經元。其主要結構是由神經元(neuron)、層(layer)和網路(network)三個部份所組成。分為輸入層(Input layer)、輸出層(Output layer)和隱藏層(Hidden layer)，如圖 2，三層連結型成一個神經網路，其中輸入層只從外部環境接收資訊，不做任何計算，只將資訊傳遞至下一層，而輸出層則是只將資訊輸出至外部環境。而分析計算部分均在隱藏層完成，隱藏層介於輸入層和輸出層之間，提供類神經網路處理神經元間的交互作用與問題的內在結構的能力。各層之間的各個神經元之間的連結都有其權重，例如，圖 2 的輸入層的 I_1 和隱藏層的 H_2 之間的連結，其權重記為 $w_{i1,h2}$ 。類神經網路即是希望透過簡單的函數計算所有連結之權重。當有未知資料須判定類別時，即可利用已計算出之權重得到輸出類別為何。

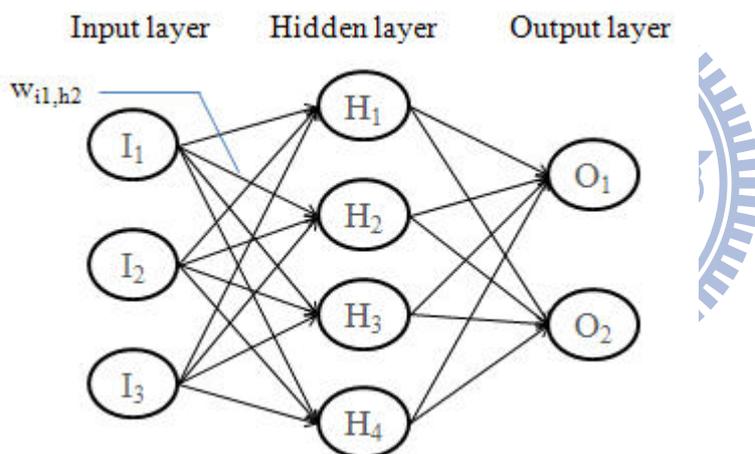


圖 2. 類神經網路之示意圖

2.2.3 支援向量機(Support Vector Machines ,SVM)

支援向量機[8]是以統計學習理論(Statistical Learning Theory)為基礎發展出來的分類方法。概念為在一群在多維空間中的資料，希望能在該空間中找出一個超平面(Hyperplane)，使得屬於兩個類別的資料可藉此超平面區分開來。超平面和兩個類別距離最近的點之間稱為邊界(Margin)，SVM 即是希望找出邊界最大之超平面，即為最佳超平面(Optimal Hyperplane)，如此可明確的區隔兩個類別的資料，如圖 3。而最靠近最佳超平面之資料稱之為支援向量(Support Vector)。當有未知資料須判定類別時，即是計算

該未知資料與最佳超平面之關係，以判定該資料之類別。

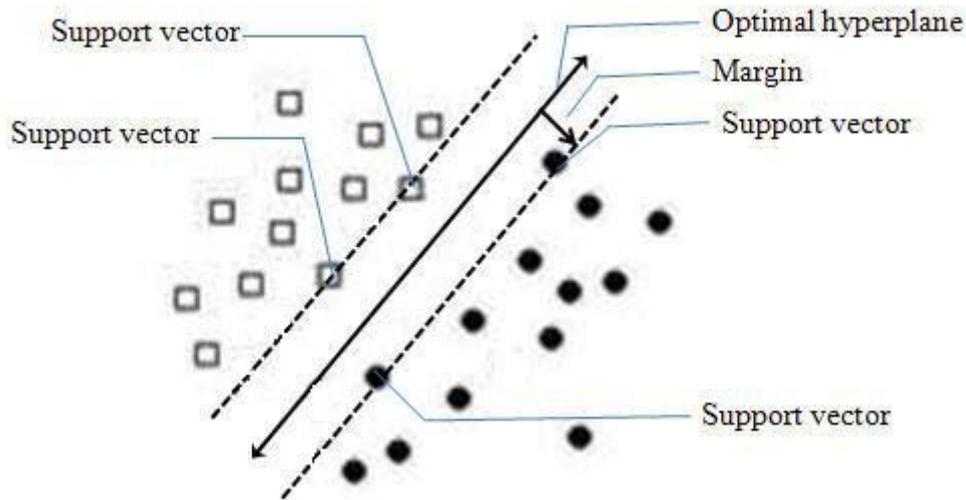


圖 3. 支援向量機之示意圖

2.2.4 決策表(Decision Table)

法則歸納法(rule induction)是一種由多個 if-else 規則對資料進行細分的技術，在一系列有意義的規則中，顯示資料蘊含的關聯性。例如，零售業者可以瞭解百分之七十的顧客於購買真皮牛仔褲後，均購買塑膠皮帶，因此這兩者間有其一定之關連性，可稱之為法則。瞭解法則之存在，有助於知道顧客與產品之特質，然而並非每一項法則均有其實用價值，資料之實用性與否仍須視企業性質而定。

在實際應用上，如何界定該法則是否有實用價值是最大的問題，通常會先將資料中發生次數太少的項目先予以刪除。利用法則試圖找出最佳特徵值，將包含此特徵值的資料全部找出來並加以聚合形成一個子集合，反覆執行直到該子集合中的每一個記錄都屬於同一類別為止，即為所謂的法則歸納法。

而決策表[9]為法則歸納法中其中一種演算法，是透過表格的方式檢視所有條件以及動作，以找出有實用價值之法則，圖 4 為一決策表之範例，此為印表機故障時的處理方法決策表，表格上方為印表機故障的條件，下方為排除故障之動作，例如，當印表機無法列印(Printer dose not print)且紅色警示燈閃爍(A red light is flashing)時，可檢查或更換墨水匣(check/replace ink)和檢查紙匣(check for paper jam)。

| Printer troubleshooter | | Rules | | | | | | | |
|------------------------|--------------------------------------|-------|---|---|---|---|---|---|---|
| Conditions | Printer does not print | Y | Y | Y | Y | N | N | N | N |
| | A red light is flashing | Y | Y | N | N | Y | Y | N | N |
| | Printer is unrecognised | Y | N | Y | N | Y | N | Y | N |
| Actions | Check the power cable | | | X | | | | | |
| | Check the printer-computer cable | X | | X | | | | | |
| | Ensure printer software is installed | X | | X | | X | | X | |
| | Check/replace ink | X | X | | | X | X | | |
| | Check for paper jam | | X | | X | | | | |

圖 4. 決策表之範例

2.2.5 決策樹(Decision Tree)

在這五種學習演算法中，其中決策樹[5]是以樹狀結構為基礎的分類方法，其優點在於可以轉換成易於了解的決策法則，相較於支援向量機、類神經網路等其它學習演算法，決策樹具有較高的可讀性，並且可表示為多個 if-else 的規則，圖 5 為一決策樹之範例，這棵決策樹根據天氣情況分類「是否適合打網球(PlayTennis)」，例如：< Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong >，會被這棵決策樹判定為 PlayTennis=No，此一決策樹的 if-else 規則如圖 6。決策樹在文獻中已經被證明在實務應用上是可以運作的，例如：1993 年加州理工學院噴氣推進實驗室與天文學家合作開發 SKICAT 系統，成功幫助天文學家發現遙遠的星體[10]以及 1995 年 Lehnert 等人利用決策樹來做醫療系統上的文字分類(text classification)[11]。

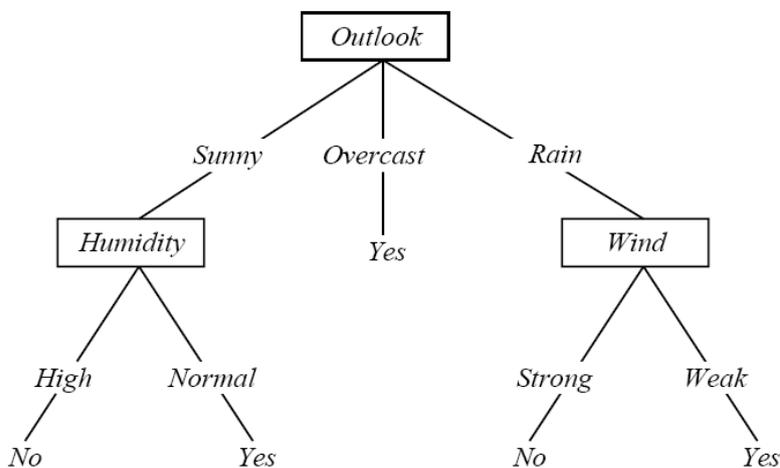


圖 5. 根據天氣情況分類「是否適合打網球(PlayTennis)」之決策樹

```

If Outlook = Sunny Then
  If Humidity = High Then
    Play Tennis = No
  Else If Humidity = Normal Then
    Play Tennis = Yes
  End If
Else If Outlook = Overcast Then
  Play Tennis = Yes
Else If Outlook = Rain Then
  If Wind = Strong Then
    Play Tennis = No
  Else If Wind = Weak Then
    Play Tennis = Yes
  End If
End If

```

圖 6. 「是否適合打網球(PlayTennis)」決策樹之 if-else 規則

2.2.6 五種學習演算法之優缺點

表 1 為以上五種演算法之優缺點比較，這五種演算法均有分類速度快之優點，簡單貝氏分類器的訓練速度快，可是該演算法須符合屬性彼此之間需為獨立關係之假設，但在實務應用上此假設通常不存在，且該演算法無法直接處理連續值之屬性，需假設該屬性符合常態分配(normal distribution)或是對該連續值之屬性做分類，轉成離散值才可處理。類神經網路的應用相當廣泛，但訓練時是利用迭代的方式，所以計算量相當大，在訓練上需要花費較多的時間，而且在訓練時，無法預知需要幾層隱藏層以及多少神經元為最佳。支援向量機在訓練和分類速度上都很快，但它對雜訊相當的敏感，在求解時會限制所有訓練資料一定能找出超平面予以切開，因此在執行前使用者需先給定一誤差參數放寬該限制，此參數對結果通常會有很大的影響。決策表可整理出易懂的 if-else 規則，有較高之可讀性，但在眾多法則中，如何界定何者才為實用之法則是有其困難的，且該演算法需要較長的訓練時間，且該演算法也不易處理連續值之屬性。決策樹的優勢在於有相當高的可讀性，可轉成易懂的 if-else 規則，也可從決策樹中挑選出較為關鍵之屬性，且訓練和分類速度都快，但是當類別過多時，決策樹的準確度會降低。

因此，為保持預測準確度與預測結果可讀性之間的平衡，我們採用以決策樹為基礎

的學習演算法，作為資料探勘的工具。

表 1. 五種學習演算法之優缺點比較

| 學習演算法 | 訓練速度 | 可讀性 | 處理連續值之屬性 | 表達屬性間之相互關係 | 缺點 |
|---------|------|-----|----------|------------|-----------------|
| 簡單貝氏分類器 | 快 | 低 | 需作離散化處理 | 無 | 屬性間需為獨立關係。 |
| 類神經網路 | 慢 | 低 | 易 | 可 | 隱藏層和神經元數目須先設定。 |
| 支援向量機 | 快 | 低 | 易 | 無 | 需先給定一誤差參數。 |
| 決策表 | 慢 | 高 | 需作離散化處理 | 可 | 界定何者為實用法則有其困難處。 |
| 決策樹 | 快 | 高 | 需作離散化處理 | 無 | 類別過多時分類準確度下降。 |

2.3 整體學習演算法(ensemble learning algorithm)

整體學習演算法(ensemble learning algorithm)的運作是透過多次執行弱學習演算法，並且針對每次產生的分類規則進行投票，最後整合投票的結果，透過多次執行可以整合多個弱學習演算法，使其等價於一個強學習演算法。

1999年，Bauer & Kohavi 的實驗研究[12]中，說明了使用以決策樹為基礎的整體學習演算法會比使用單一的弱學習演算法來的準確，其中兩種常見的整體學習演算法有 bagging[13]和 boosting[14]兩種，其中 bagging 是利用不同的訓練資料集合產生不同的分類規則，最後以投票的方式整合不同的分類規則；而 boosting 則是在每次的迭代中，更新所有資料的權重，提高分類錯誤的資料權重，反之降低分類正確的資料權重，最後則以有權重的投票方式整合每次迭代中產生的分類規則。

2.3.1 Bagging and Random Forest

1996 年 Breiman 提出 bagging(bootstrap aggregating)方法[13]，它是採用投票表決的方式結合多個分類規則，每個分類規則雖使用相同的學習演算法，但均使用不同的訓練資料集合，所以每個分類規則彼此之間都有著些許差異。此演算法的流程圖如圖 7，利用重複取樣(sampling with replacement)的方式產生多組訓練資料集合。例如，原始訓練資料(training data)若有 n 筆資料，即對該原始訓練資料做 n 次的重複取樣，如此就可產生一組訓練資料(training data 1)，重複此取樣動作 m 次，最後會產生 m 組訓練資料(training data 1~training data m)，其中每一組訓練資料都可以訓練出一棵決策樹，如此就可產生 m 棵決策樹(decision tree 1~decision tree m)，最後，將測試資料(testing data)透過 m 棵決策樹產生 m 個分類結果，再以多數決投票的方式得到最終分類結果。

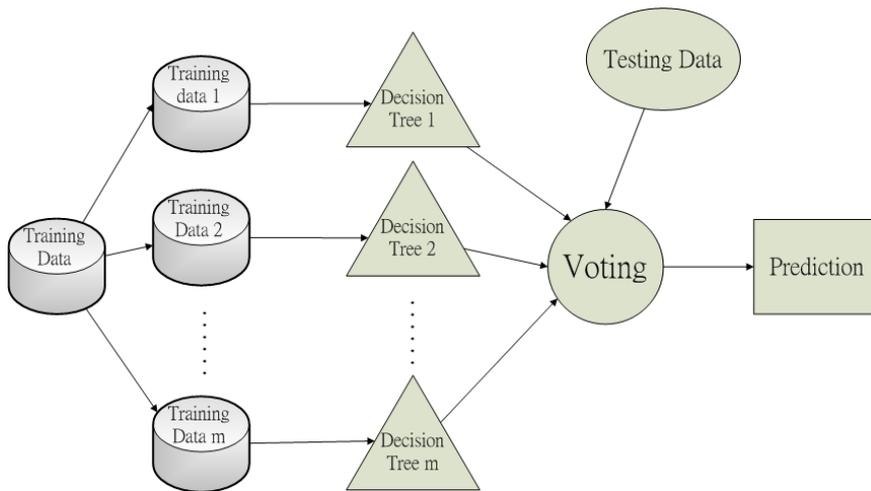


圖 7. Bagging 演算法之流程圖

2002 年，Breiman 提出 random forest[15]，其與 bagging[13] 差異之處在於訓練方式的不同，在利用訓練資料產生決策樹時，bagging 使用的是全部的屬性，而 random forest 則隨機挑選出 M 個屬性，換言之，每一棵決策樹除了使用的訓練資料集合不同之外，用以訓練決策樹的屬性也不盡相同，根據 Breiman 的建議， M 的值應為屬性數量的平方根[16]。由於決策樹具高度敏感性，只要訓練資料有些微的不同，其所產生的決策樹就會有很大的差異。因此，在產生一個決策樹的集合時，我們希望這些決策樹的相異性越大越好，藉此，該決策樹的集合才能覆蓋包含更多的資訊，就此 random forest 除了強調訓練資料的差異外，再加上屬性選擇上的差異，以期得到差異性更大的決策樹集合，讓

學習演算法可以涵蓋更多的資訊。

2.3.2 AdaBoost

1996 年 Freund 和 Schapire 提出 AdaBoost[17]演算法，AdaBoost 為 boosting 方法中常被使用的演算法。Boosting 為整體學習演算法中常見的方式之一，boosting 的想法是透過不斷的迭代執行之後，將一個原本屬於弱學習的演算法，使用有權重的投票機制輸出結果，透過這樣的機制，不僅可以改善分類的準確度，並可以等價於一個強學習的演算法。AdaBoost 演算法中，每一筆資料都有其權重，在每一次的迭代(iteration)執行之後，會更新所有資料的權重，其中，分類錯誤的資料權重會增加，反之分類正確的資料則會降低其權重，透過這樣的方式，讓學習演算法更重視那些被分類錯誤的資料，以提高分類的準確度。

AdaBoost 的演算法如圖 8，輸入為 m 筆訓練資料，一個弱學習演算法(WeakLearn)和最多迭代次數 T ，每一筆訓練資料均被預設有相同之權重，進入第一次的迭代使用弱學習演算法，計算該次分類錯誤率，若錯誤率大於 $1/2$ 則離開迴圈，反之則需利用 $\beta_t = \epsilon_t / (1 - \epsilon_t)$ 計算 β_t ，若 β_t 越大，表示該次分類錯誤的程度越高，並利用該值和此次該筆資料分類正確與否來更新所有資料之權重， β_t 為一小於 1 之數值，因此透過

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$$

此公式更新完權重後，會降低分類正確的資料權重，分類錯誤的資料則會被提高權重。更新完所有資料的權重後，即進入第二次的迭代，會依照目前的資料權重使用弱學習演算法，因此，弱學習演算法會較為重視在之前的迭代中被分類錯誤的資料，因為在之前的迭代中被分類錯誤的資料權重會較分類正確的資料高，直到此次迭代的錯誤率大於 $1/2$ 或是迭代次數等於輸入的最多迭代次數 T 為止即離開迴圈，最終分類結果是由每一次迭代的分類結果來投票，但每次分類結果的投票權重並非相等，若該次迭代的分類錯誤率越低，即有越高的投票權重，最後透過有權重的投票機制得到最終分類結果。

Algorithm AdaBoost

Input : sequence of m examples $\{(x_1, y_1), \dots, (x_m, y_m)\}$ with labels $y_i \in Y = \{1, \dots, k\}$

Weak learning algorithm **WeakLearn**

integer T specifying number of iterations

Initialize $D_1(i) = 1/m$ for all i .

Do for $t=1, 2, \dots, T$:

1. Call **WeakLearn**, providing it with the distribution D_t .
2. Get back a hypothesis $h_t : X \rightarrow Y$.

3. Calculate the error of h_t :
$$\epsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$$

If $\epsilon_t > 1/2$, then set $T=t-1$ and abort loop.

4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$

5. Update distribution D_t :

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$$

, where Z_t is a normalization constant (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis :

$$h_{\text{fin}}(x) = \operatorname{argmax}_{y \in Y} \sum_{t: h_t(x)=y} \log \frac{1}{\beta_t}$$

圖 8. AdaBoost 演算法

2.3.3 Stacking

Stacking[18]和 bagging、boosting 不同的地方在於，stacking 可以另一方式整合不同的學習演算法(例如：決策樹、支援向量機等)的預測結果，圖 9 為 stacking 演算法之流程圖。以決策樹為例，在產生 m 棵決策樹後，再利用此 m 棵決策樹對訓練資料所做的預測結果，加以整理完成新的訓練資料(stacking training data)。

我們以圖 10 作為一個 stacking 的簡單範例，此範例為根據天氣情況分類「是否適合打網球(PlayTennis)」，(a)有七筆訓練資料和(b)兩筆測試資料，利用重複取樣的方式產生五棵決策樹，訓練資料分別利用這五棵決策樹產生五個分類結果。(c)接著則以此分類結果當作新的訓練資料(stacking training data)之屬性，類別則為原始訓練資料之類別，以 D1 為例，五棵決策樹的分類結果分別為 Yes、Yes、Yes、No 和 No，則這五個分類結果即為新的訓練資料之屬性，而 D1 原始的類別為 No，所以該資料之類別即為 No；(d)新的測試資料(stacking testing data)亦然，利用五棵決策樹產生的五個分類結果為屬性，而其類別即為我們要預測之值。

與 bagging 或 boosting 相較，stacking 的優勢在於其利用兩階段的模式整合不同的預測結果。舉一較極端之範例，若原使用一階段的整體學習演算法的分類結果和資料正確類別完全相反，即所有 positive 之資料預測為 negative，且所有 negative 之資料預測為 positive，以整體準確度(overall accuracy)來說，這樣的結果是 0%，但若是透過 stacking 演算法，就能在第二階段的學習演算法中修正此結果，達到 100%的準確度。

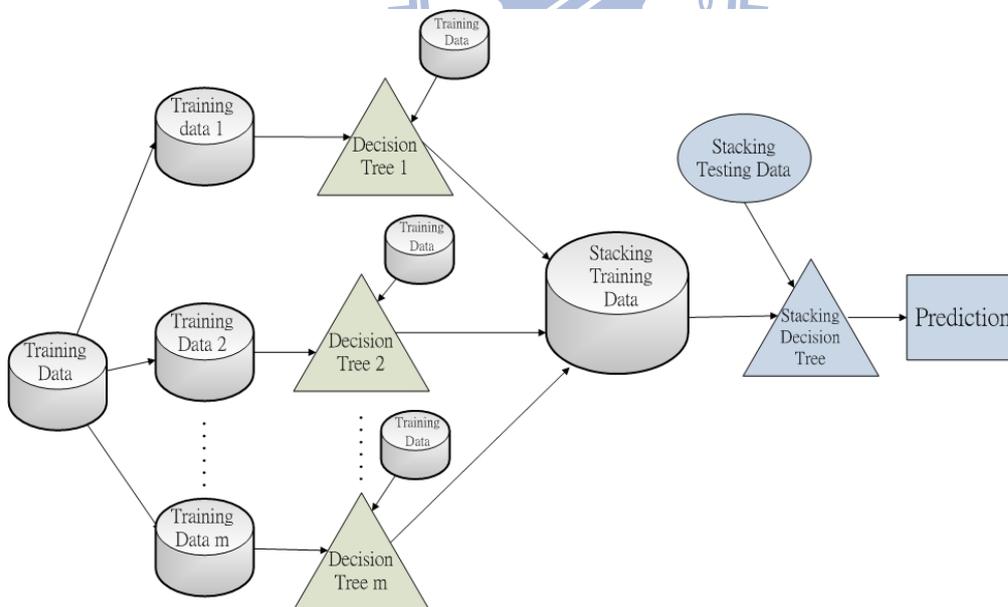


圖 9. Stacking 演算法之流程圖

(a) 訓練資料(Training data)

| Day | Outlook | Temperature | Humidity | Wind | Play Tennis |
|-----|----------|-------------|----------|--------|-------------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |

(b) 測試資料(testing data)

| Day | Outlook | Temperature | Humidity | Wind | Play Tennis |
|-----|----------|-------------|----------|------|-------------|
| D7 | Overcast | Cool | Normal | Weak | Yes |
| D8 | Sunny | Mild | High | Weak | No |

(c) 新的訓練資料(stacking training data)

| Day | Decision Tree 1 | Decision Tree 2 | Decision Tree 3 | Decision Tree 4 | Decision Tree 5 | Play Tennis |
|-----|-----------------|-----------------|-----------------|-----------------|-----------------|-------------|
| D1 | Yes | Yes | Yes | No | No | No |
| D2 | Yes | No | Yes | Yes | Yes | No |
| D3 | No | No | No | No | Yes | Yes |
| D4 | No | No | Yes | No | No | Yes |
| D5 | Yes | No | No | No | No | Yes |
| D6 | Yes | No | No | Yes | Yes | No |

(d) 新的測試資料(stacking testing data)

| Day | Decision Tree 1 | Decision Tree 2 | Decision Tree 3 | Decision Tree 4 | Decision Tree 5 | Play Tennis |
|-----|-----------------|-----------------|-----------------|-----------------|-----------------|-------------|
| D7 | Yes | No | Yes | No | No | Yes |
| D8 | Yes | Yes | No | Yes | Yes | No |

圖 10. Stacking 演算法之簡單範例

2.4 類別不平衡(class imbalance)

資料探勘在實務的應用上，經常會出現類別不平衡(class imbalance)[19]的問題，此問題指的是大多數的資料都屬於某一類別，稱之為多數類別(majority class)，其數量遠大於另一類別，此則稱為少數類別(minority class)。然而在日常生活中，我們比較關注的通常是比較少發生的現象。這些比較少發生的現象大多是比較有趣或重要的，例如：詐欺偵測(fraud detection)[20]以及漏油偵測(oil spill detection)[21]等兩種類別不平衡之實務上都是在偵測少數發生的現象。

在傳統的學習演算法中，對於在學習類別不平衡的資料時，通常會傾向預測為多數類別(majority class)而忽略了少數類別(minority class)，這樣的預測傾向，雖然可以有很高的多數類別準確度，但少數類別的準確度往往會嚴重降低。而且在大多數的領域中，少數類別通常是比較被關注的，因此在遇到類別不平衡的資料時，我們希望可以透過一些方法來改善少數類別的準確度。

過去的文獻中，已提出一些方法來解決類別不平衡的問題，像是 1997 年 Kubat 等人提出的減少多數類別取樣法(under-sampled examples of the majority class)[22]、1998 年 Ling 等人提出的增加少數類別取樣法(over-sampled examples of the minority class)[23]、2002 年 Chawla 等人提出的結合減少多數類別和增加少數類別取樣法(over-sampled the minority class and under-sampled the majority class)[24]、1997 年 Cardie 和 Howe 提出的增加少數類別之資料權重，使其學習演算法傾向於少數類別[25]、2002 年 Joshi 等人利用 boosting 演算法分類少數類別[26]和 2004 年 Guo 和 Viktor 發展的 DataBoost-IM 演算法，其為整合 boosting 以及產生虛擬資料來提升少數類別的分類準確度[27]等方法。而在此研究中，我們採用了減少多數類別取樣法、增加少數類別取樣法和 DataBoost-IM。

在一般類別不平衡的資料中，若資料分佈如圖 11(a)，少數類別(minority class)的資料以黑色點表示，白色點則代表多數類別(majority class)之資料，雖然有類別不平衡之問題，不過兩種類別的資料可被明顯區隔，使用減少多數類別取樣法和增加少數類別取樣法即可提高少數類別的準確率。但若是資料除了有類別不平衡的問題之外，資料的分佈還相當的稀疏，如圖 11(b)所示，資料分佈相當稀疏，長方框為其少數類別資料之判定區間(decision region)，可看出框內亦有多數類別之資料，兩種類別之間的邊界相當模糊。使用傳統解決類別不平衡問題之方法，減少多數類別取樣法和增加少數類別取樣法後，平衡兩種類別的資料量，希望藉此使得兩種類別彼此的邊界更為明顯，但在分佈相當稀疏的資料中，圖 11(c)為使用增加少數類別取樣法之示意圖，可以看出在取樣後，少數類別之判定區間會較為縮小，區間內的多數類別資料也會減少，但文獻中提到，在使用決策樹學習演算法時，此法會造成少數類別被分割成較多之區塊[24]，對決策樹而言，將資料分割成越多區塊，會造成決策樹的深度越深而有越多的葉子(leaf node)，因

而產生過度適應(overfitting)的問題。而使用減少多數類別取樣法如圖 11(d)，挑選多數類別的資料直到數量和少數類別資料量相同為止，但因為挑選多數類別的資料是採用隨機挑選的方式，從圖 11(d)可看出，兩種類別的資料數量雖然相同，但兩種類別之間還是無法被明顯的區隔開來，少數類別資料的判定區間內依舊會有多數類別之資料。

除了上述的取樣方法外，在文獻中，亦有使用 k-nearest-neighbor 演算法來做例外偵測(outlier detection)[28,29]，而且如 2.1 所述，因為本篇論文所使用的資料雜訊(noise)過多，兩種類別的資料是相當稀疏且重疊的，因此我們利用 k-nearest-neighbor 來做資料清理(data cleaning)，刪除部分多數類別的資料，除了可以平衡兩種類別的資料數目外，也希望類別彼此的邊界更為明顯，讓學習演算法更能區分兩種類別。

而在我們的研究中，大多數的病患在使用 PCA 裝置設定後，都是不需要再調整參數設定的，只有少部分的病患會有噁心嘔吐等副作用或是無法減輕疼痛的狀況而需要調整參數，以本篇論文的資料來看，只有 18.9%的病患需要被調整參數設定，因此，在 PCA 裝置參數設定調整的預測分析上，我們會面臨到類別不平衡(class imbalance)的問題，希望透過減少多數類別取樣法、增加少數類別取樣法或資料清理的方式藉以改善此問題。



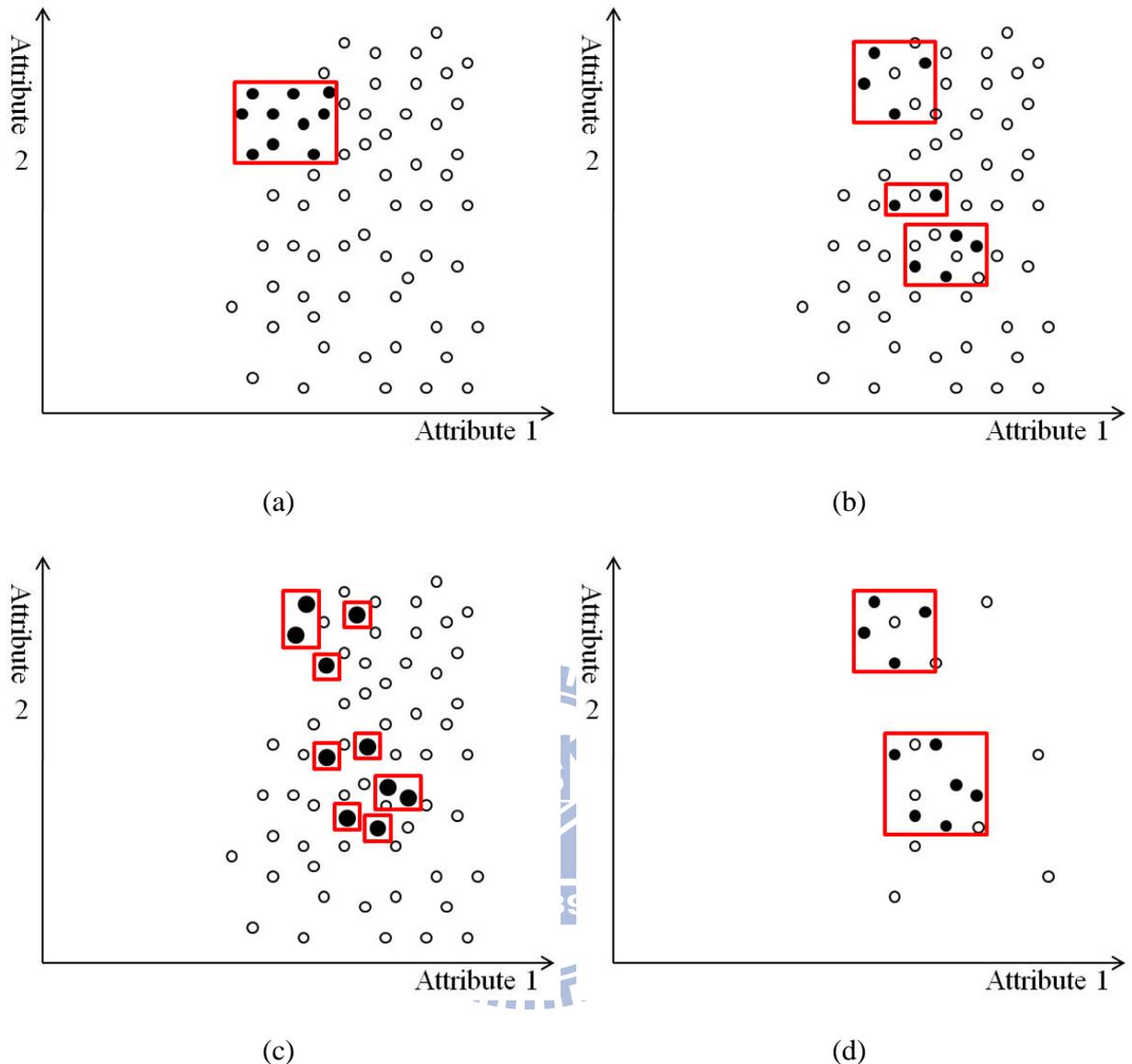


圖 11. 類別不平衡及資料稀疏與學習困難度之相關性探討圖例

說明：白色點為多數類別(majority class)，黑色點為少數類別(minority class)，長方框為少數類別資料之判定區間(decision region)。(a) 兩種類別之間有明顯邊界。(b) 兩種類別之間的邊界模糊。(c) 對(b)進行增加少數類別取樣法(over sampling)，較大的黑色點表示該資料被重複取樣。(d) 對(b)進行減少多數類別取樣法(under sampling)。

2.4.1 減少多數類別取樣法(under sampling)

減少多數類別取樣法(under-sampled examples of the majority class)[22]的演算法，在做重複取樣產生訓練資料集合時，分別對兩種類別作取樣，利用降低取樣多數類別的次數，來減少多數類別的資料數量，以期平均兩種類別的資料數目。例如，多數類別(majority class)的資料有 80 筆，而少數類別(minority class)有 20 筆資料，在使用減少多數類別取

樣法時，降低取樣多數類別的次數，僅對多數類別取樣 20 次，取樣完成後，多數類別和少數類別的資料筆數會是平衡的狀態。

2.4.2 增加少數類別取樣法(over sampling with replacement)

增加少數類別取樣法(over-sampled examples of the minority class)[23]，在做重複取樣產生訓練資料集合時，則是透過重覆取樣的方式提高少數類別的取樣量來增加少數類別的資料數量，以此平衡兩種類別的資料數量。例如，多數類別(majority class)的資料有 80 筆，而少數類別(minority class)有 20 筆資料，若使用增加少數類別取樣法，重覆少數類別的取樣次數到 80 次，在取樣完成後，多數類別和少數類別的資料數量則可達到相同。

減少多數類別取樣法和增加少數類別取樣法不同的地方在於，前者為降低取樣多數類別的次數，而後者則是提高少數類別的取樣次數。除此之外，兩種演算法的取樣方式也略為不同，減少多數類別取樣法的取樣方式為取出不放回(sampling without replacement)，即取樣過的資料不可放回被再次取樣，表示同一筆資料最多只會被取樣一次，而增加少數類別取樣法的取樣方式則是取出放回(sampling with replacement)，表示資料即使已被取樣過，下次取樣依舊是從全部的資料進行取樣，表示同一筆資料有可能被取樣兩次以上。兩種方式都是希望可以透過改變取樣次數的方式，平衡兩種類別的資料個數，讓學習演算法可以由等量的兩種類別資料中學習，以期提高分類的準確度。

2.4.3 DataBoost-IM

2004 年 Guo 和 Viktor 提出 DataBoost-IM[27]演算法，DataBoost-IM 為一整合 boosting 以及產生虛擬資料(data generation)之演算法，其 boosting 的部分和 2.3.2 提到的 AdaBoost 相同，每一筆資料都有其權重，在每一次的迭代(iteration)執行之後，會更新所有資料的權重，其中，分類錯誤的資料權重會增加，反之分類正確的資料則會降低其權重。除了針對分類正確與否更新資料的權重外，DataBoost-IM 也結合了產生虛擬資料(data generation)的想法，透過產生虛擬資料來平衡兩種類別的資料數量，並且是針對資料權重較高者來產生，也就是說，產生的虛擬資料之資料分佈，會和在之前的迭代中，被分

類錯誤的資料類似。希望透過這樣的方式，除了平衡兩種類別的資料量外，也可以讓學習演算法更專注在那些被分類錯誤的資料上。

DataBoost-IM 的演算法如圖 12，輸入為 m 筆訓練資料，一個弱學習演算法 (WeakLearn) 和最多迭代次數 T ，每一筆訓練資料均被預設有相同之權重，進入迭代後，分別判斷各個類別的 hard examples，即為資料中權重較高者，意指先前迭代中被分類錯誤之資料。各類別以各自的 hard examples 為基礎，分別產生虛擬資料 (synthetic data) 以平衡各個類別之資料量。虛擬資料 (synthetic data) 加上原始的訓練資料 (original training set) 產生新的訓練資料 (new training data set)。分別更新各個類別的資料權重，更新後各個類別的資料權重總合均為相等，目的是希望學習演算法不會傾向預測任一類別。針對新的訓練資料 (new training data set) 和更新後的資料權重使用弱學習演算法 (WeakLearn)，計算該次分類錯誤率，若錯誤率大於 $1/2$ 則離開迴圈，反之則需利用

$\beta_t = \epsilon_t / (1 - \epsilon_t)$ 計算 β_t ，若 β_t 越大，表示該次分類錯誤的程度越高，並利用該值和此次該筆資料分類正確與否來更新所有資料之權重， β_t 為一小於 1 之數值，因此透過

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$$

此公式更新完權重後，會降低分類正確的資料權重，分類錯誤的資料則會被提高權重。更新完所有資料的權重後，即進入下一次的迭代，會依照目前的資料權重新新判斷 hard examples 以產生虛擬資料。因此，弱學習演算法會較為重視在之前的迭代中被分類錯誤的資料，因為在之前的迭代中被分類錯誤的資料權重會較分類正確的資料高，直到此次迭代的錯誤率大於 $1/2$ 或是迭代次數等於輸入的最多迭代次數 T 為止即離開迴圈，最終分類結果是由每一次迭代的分類結果來投票，但每次分類結果的投票權重並非相等，若該次迭代的分類錯誤率越低，即有越高的投票權重，最後透過有權重的投票機制得到最終分類結果。

Algorithm DataBoost-IM

Input : sequence of m examples $\{(x_1, y_1), \dots, (x_m, y_m)\}$ with labels $y_i \in Y = \{1, \dots, k\}$

Weak learning algorithm **WeakLearn**

integer **T** specifying number of iterations

Initialize $D_1(i) = 1/m$ for all i .

Do for $t=1, 2, \dots, T$:

1. Identify hard examples from the original data set for different classes.
2. Generate synthetic data to balance the training knowledge of different classes.
3. Add synthetic data to the original training set to form a new training data set.
4. Update and balance the total weights of the different classes in the new training data set.
5. Call **WeakLearn**, providing it with the new training set with synthetic data and rebalanced weights.
6. Get back a hypothesis $h_t : X \rightarrow Y$.
7. Calculate the error of h_t : $\epsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$.

If $\epsilon_t > 1/2$, then set $T=t-1$ and abort loop.

8. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$

9. Update distribution D_t :

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$$

, where Z_t is a normalization constant (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis :

$$h_{\text{fin}}(x) = \operatorname{argmax}_{y \in Y} \sum_{t: h_t(x)=y} \log \frac{1}{\beta_t}$$

圖 12. DataBoost-IM 演算法

2.4.4 資料清理(data cleaning)

有鑑於相同類別的資料通常較為靠近，我們使用 k-nearest-neighbor 演算法來做資料清理(data cleaning)，若是某一筆資料附近的資料類別都與其類別相異的話，該筆資料極可能為雜訊(noise)。同時因為 PCA 資料有類別不平衡的問題，所以我們以少數類別(minority class)的資料為基準，找出離每個少數類別資料最近之 k 筆資料，其距離乃是使用歐幾里得距離(Euclidean distance)，在這 k 筆資料中，類別若為多數類別(majority class)的資料即視為雜訊，在檢查過所有較少類別的最近之 k 筆資料後，我們將所有被視為雜訊之資料予以刪除。透過這樣的方式，兩種類別之間的邊界會更為明顯，就學習演算法而言，將更能區分兩種類別，以提高分類之準確度。

圖 13 為該演算法之範例，(a)為未做資料清理的原始資料，白色點為多數類別而黑色點為少數類別之資料，長方框為少數類別資料之判定區間(decision region)，可看出有不少的多數類別資料在區間內，(b)可以透過縮小判定區間的方式，降低區間內多數類別的資料，但這會造成過度適應(overfitting)的問題，(c)因此，我們利用 k-nearest-neighbor 演算法，當 k=3 時，我們以所有少數類別的資料為基準，找出距離該資料最近之 3 筆資料，(d)在這 3 筆資料中，若是類別為多數類別者則以 x 記號被標記為雜訊，(e)為刪除所有被標記為雜訊後之資料。從圖 13(a)及(e)的比較可以看出，在原始資料中，兩種類別的資料交錯無法明顯區隔，但在刪除雜訊後，兩種類別資料之間的邊界已較為明顯。我們並且比較圖 13(b)及(e)，兩者少數類別的判定區間內都僅有少數的多數類別資料，但(e)的判定區間明顯比(b)來得大，因此，可以降低造成過度適應(overfitting)的可能性。

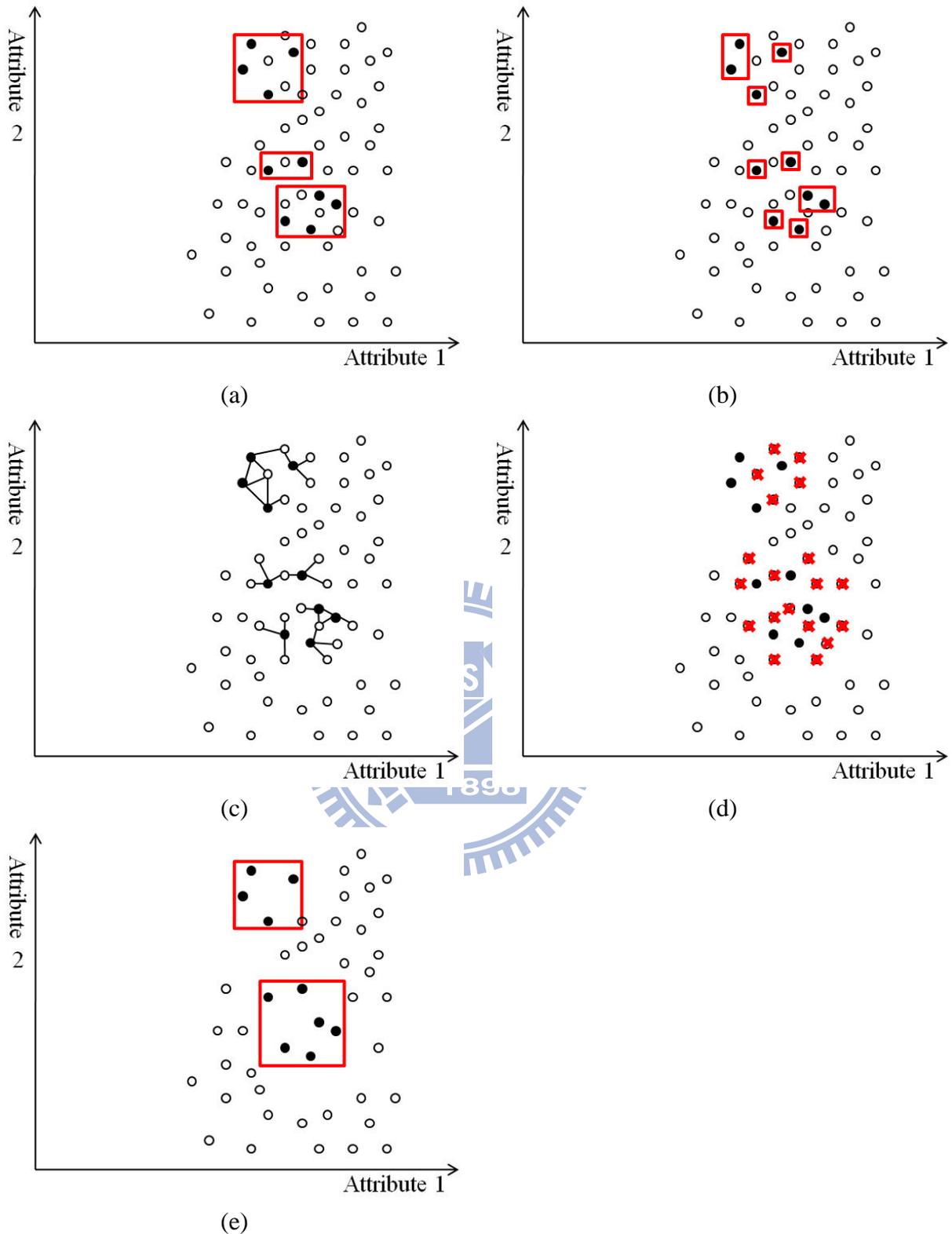


圖 13. 資料清理演算法之示意圖例

說明：白色點為多數類別(majority class)，黑色點為少數類別(minority class)，長方框為少數類別資料之判定區間(decision region)。(a)原始資料。(b) 縮小少數類別之判定區間。(c)以所有少數類別的資料為基準，找出距離該點最近之 k 筆資料， $k=3$ 。(d) 這 k 筆資料中，類別為多數類別的資料即為雜訊，劃 x 者為所有被標記為雜訊之資料。(e) 刪除所有被定義為雜訊的資料。

三、實驗說明

實驗分成三個部分，包括總劑量預測、PCA 劑量預測和 PCA 裝置參數設定調整預測。

針對總劑量預測的部分，我們目前是參考病患前 24 小時的 PCA 使用行為及其基本生理資料與病歷，用以預測該病患於 72 小時後，其使用的麻醉藥總劑量，這其中包括了病患透過按壓 PCA 裝置所注射的麻醉藥以及經由點滴被注射的麻醉藥。我們將總劑量之多寡概分為三類：低劑量(low dose)、一般劑量(medium dose)、高劑量(high dose)。而 PCA 劑量預測一樣是參考病患前 24 小時的 PCA 使用行為及其基本生理資料與病歷，但僅僅是預測病患於 72 小時後，透過按壓 PCA 裝置所注射的麻醉藥劑量，相同地，我們依照劑量之多寡概分為三類：低劑量(low dose)、一般劑量(medium dose)、高劑量(high dose)。至於 PCA 裝置參數設定調整預測，則是以病患前 24 小時的 PCA 使用行為及其基本生理資料與病歷等，預測病患在爾後的 48 小時內，其 PCA 裝置參數的設定是否應該調整。

3.1 資料收集

此篇論文所使用的資料庫來自於彰化基督教醫院,Changhwa Christian Hospital (CCH)，2005~2010 的病患使用 PCA 裝置的資料。

3.2 資料前處理

彰化基督教醫院 2005~2010 使用 PCA 裝置總共有 5432 位病患，刪除因 PCA 裝置暫存空間不足而毀損之檔案後剩餘 4632 位病患，之後再刪除資料庫中無病歷資料之病患後剩餘 2586 位病患，此外若該病患的病歷資料不完整的比例太高者也將其刪除，最後可用資料總共有 2207 位病患，其中使用超過 24 小時的病患有 1655 位，而有 1099 位病患使用 PCA 裝置超過 72 小時。

每一位病患使用 PCA 裝置的狀況不盡相同，所以實驗需要病患的性別和年齡等生理資料、手術類別和手術時間等手術相關屬性、單位小時成功按壓次數和單位小時失敗按壓次數等使用 PCA 裝置的情況以及 PCA 裝置參數的設定，像是單位小時的鎖定時間

(lockout)設定等做為預測的依據。從彰化基督教醫院的資料庫中，我們根據病患的病歷和使用 PCA 裝置的記錄檔整理出以下的屬性。

每一位病患總共會有 280 個屬性，其中包含了

- a. 病患生理資料：BP_HIGH, BP_LOW, PULSE, WEIGHT, SEX, age, DM, HT, AMI，共 9 個屬性。
- b. 手術相關屬性：OP_CLASS, ansTimediff, ASA_TYPE1, ASA_TYPE2, M1_ANS_WAY, APS，共 6 個屬性。
- c. 使用 PCA 情況相關屬性：success_p_1hr~24hr, failure_p_1hr~24hr, pcadose_1hr~24hr, change_times_1hr~24hr, p_timediff_mean_1hr~24hr, p_timediff_var_1hr~24hr 共 144 個屬性。
- d. PCA 設定相關屬性：loadingdose, contidose_1hr~24hr, pamode_set_1hr~24hr, pcadose_set_1hr~24hr, lockout_set_1hr~24hr, 4hrlimit_set_1hr~24hr，共 121 個屬性。

表 2 為所有屬性之詳細說明。

表 2. 所有屬性之說明

| 屬性名稱 | 屬性說明 |
|-------------|--|
| BP_HIGH | 收縮壓 |
| BP_LOW | 舒張壓 |
| PULSE | 心跳 |
| WEIGHT | 體重 |
| OP_CLASS | 手術的分類 (例如：開腔手術、上腹手術、下腹手術、泌尿系統手術、主要關節手術、皮膚手術等) |
| ansTimeDiff | 手術進行時間 |
| ASA_TYPE1 | 1：健康 2：輕度全身性疾病 3：重度全身性疾病 4：危及生命的狀況或疾病 5：死亡機率極高之狀況或疾病 6：器官移植 1: healthy 2: mild systemic disease. 3: major systemic disease. 4. life threatening disease or condition. 5. not expect to survive. 6. donor |

| | |
|-----------------|---|
| | (transplation). |
| ASA_TYPE2 | 手術類型 E：緊急手術、R：一般手術 E(emergency surgery), R(regular surgery) |
| M1_ANS_WAY | 手術麻醉方式 SA：半身、GA：全身、LE：腰椎硬脊膜外麻醉、NB：神經阻斷 |
| APS | 是否在術前接受 PCA |
| SEX | 性別 |
| age | 年齡 |
| DM | 是否患有糖尿病 |
| HT | 是否患有高血壓 |
| AMI | 是否心肌梗塞 |
| loadingdose | 病患在手術後但使用 PCA 裝置之前所被注射的麻醉藥劑量 |
| success_p | 單位小時成功按壓次數 成功按壓意指病患該次按壓動作為可被接受的，PCA 裝置會替病患注射麻醉藥。 |
| failure_p | 單位小時失敗按壓次數 失敗按壓意指病患該次按壓動作為不可接受的，可能違反「lockout」或是「4hrlimit」，PCA 裝置判定不替病患注射麻醉藥。 |
| pcadose | 單位小時病人經由按壓 PCA 裝置方式被注射的麻醉藥劑量 |
| contidose | 單位小時病人經由點滴方式被注射的麻醉藥劑量 |
| change_times | 單位小時 PCA 裝置設定調整次數 |
| p_timediff_mean | 單位小時按壓時間差的平均 |
| p_timediff_var | 單位小時按壓時間差的變異數 |
| pcamode_set | 單位小時 pca mode 設定 PCA：僅透過按壓 PCA 裝置方式被注射麻醉藥。 CONTINUOUS：僅透過點滴方式被注射麻醉藥。 CONTINUOUS+PCA：可透過按壓 PCA 裝置和點滴方式被注射麻醉藥。 |
| pcadose_set | 單位小時 pca dose 設定 pca dose 設定意指病患每次按壓 PCA 裝置，若被判定為成功按壓，PCA 裝置會替病患注射的麻醉藥劑量。 |
| lockout_set | 單位小時 lockout 設定 lockout 意指病患該次按壓與下次按壓需間隔時間。 |
| 4hrlimit_set | 單位小時 4hrlimit 設定 4hrlimit 意指病患在四小時之內僅可被注射之總麻醉藥劑量。 |

3.3 實驗

3.3.1 總劑量預測

總劑量預測的主要目的為病患使用 PCA 裝置一段時間之後，透過病患使用 PCA 的行為模式，再配合病患的生理資料等，來預測病患在短期未來之內所需麻醉藥劑量的多寡程度。我們想要預測病患使用 PCA 裝置的 72 小時期間所注射的麻醉藥總劑量，其中包括了透過按壓 PCA 裝置以及經由點滴方式所注射的麻醉藥。在該預測分析上，總計有 1099 筆可用資料，由於要預測的內容為病患使用 PCA 裝置 72 小時的麻醉藥總劑量，因此病患需要使用 PCA 裝置超過 72 小時，該資料才可用於此預測分析。我們依照總劑量多寡將所有資料分為三個區間，各區間麻醉藥總劑量的標準差類似，表示三種類別的資料離散程度相似，將其分為低劑量(low dose)、一般劑量(medium dose)和高劑量(high dose)，其中低劑量的資料數量為 399 筆，而一般劑量為 551 筆，高劑量的資料數量較少，只有 149 筆。而在屬性的選擇上，我們使用了病患的生理資料以及手術相關屬性。除此之外，由於疼痛是相當主觀的，即使是相同身體狀況的兩人在疼痛的感受上也會有所差異，因此單純的從身體狀況來進行預測分析是相當困難的，所以除了上述屬性外，我們也將病患使用 PCA 裝置 24 小時期間的單位小時使用 PCA 情況和 PCA 設定相關屬性用於此預測分析。學習演算法的部分，我們會比較決策樹(C4.5)、簡單貝氏分類器(Naive Bayes Classifier)、類神經網路(Artificial Neural Network ,ANN)、支援向量機(Support Vector Machine ,SVM)和決策表(Decision Table)五種學習演算法，其中除了決策樹(C4.5)以外，均是使用 WEKA[30]做為實驗的工具。而在整體學習演算法(ensemble learning algorithm)的部分，基於分類規則可讀性、學習速度以及預測準確度之考量下，我們以決策樹(C4.5)為基礎演算法，搭配 bagging、random forest、bagging + stacking、random forest + stacking 和 Adaboost 共五種整體學習演算法，並加以比較分析。

3.3.2 PCA 劑量預測

PCA 劑量預測和總劑量預測類似，一樣在病患使用 PCA 裝置一段時間之後，透過病患使用 PCA 的行為模式和病患的生理資料等，預測病患在短期未來之內會經由 PCA

裝置注射的麻醉藥劑量多寡程度，和總劑量預測不同的地方在於，總劑量預測是預測病患經由點滴方式和透過按壓 PCA 裝置注射的總麻醉藥劑量多寡，而 PCA 劑量預測僅針對病患透過按壓 PCA 裝置注射的麻醉藥劑量做預測。我們利用病患的生理資料和手術相關屬性以及病患使用 PCA 裝置 24 小時期間的單位小時使用 PCA 情況和 PCA 設定相關屬性，用以預測病患使用 PCA 裝置 72 小時的 PCA 劑量多寡程度。因此，病患需使用 PCA 裝置超過 72 小時，該資料才為可用資料，因此和總劑量預測相同，總計會有 1099 筆可用資料。我們依照 PCA 劑量多寡將所有資料分為三個區間，各區間麻醉藥總劑量的標準差類似，表示三種類別的資料離散程度相似，將其分為低劑量(low dose)、一般劑量(medium dose)和高劑量(high dose)，其中低劑量的資料數量為 580 筆，而一般劑量為 373 筆，高劑量的資料數量較少，只有 146 筆。

學習演算法的部分，和總劑量預測相同，我們先比較決策樹(C4.5)、簡單貝氏分類器(Naive Bayes Classifier)、類神經網路(Artificial Neural Network ,ANN)、支援向量機(Support Vector Machine ,SVM)和決策表(Decision Table)五種學習演算法，其中除了決策樹(C4.5)以外，均是使用 WEKA[30]做為實驗的工具。而整體學習演算法(ensemble learning algorithm)的部分，以決策樹(C4.5)為基礎演算法，搭配 bagging、random forest、bagging + stacking、random forest + stacking 和 Adaboost 共五種整體學習演算法，並加以比較分析。

3.3.3 PCA 裝置參數設定調整預測

PCA 裝置參數設定調整預測的目的則是在病患使用 PCA 裝置一段時間之後，透過病患使用 PCA 的行為模式，再配合病患的生理資料等，來預測病患在之後的 PCA 使用上是否需要醫師調整 PCA 裝置的參數。我們使用了病患的生理資料和手術相關屬性，以及病患使用 PCA 裝置 24 小時內的單位小時的使用 PCA 情況和 PCA 設定相關屬性，來預測病患在使用 PCA 裝置第 25 小時到第 72 小時之間，PCA 裝置參數設定是否被調整。和上述三種預測分析相同，因為是預測病患使用 PCA 裝置 72 小時的情況，所以病患需使用 PCA 裝置超過 72 小時才可用於此預測分析，可用資料有 1099 筆。我們依照

PCA 裝置是否被調整將全部資料分為兩類，其中「未做調整」(negative)的資料有 891 筆，而「有做調整」(positive)的資料僅 208 筆，「未做調整」之資料遠大於「有做調整」之資料，兩者資料的比例為 0.79：0.21，因此在該預測分析上，有類別不平衡(class imbalance)之問題。

同於上述三種預測分析，我們比較決策樹(C4.5)、簡單貝氏分類器(Naive Bayes Classifier)、類神經網路(Artificial Neural Network ,ANN)、支援向量機(Support Vector Machine ,SVM)和決策表(Decision Table)五種學習演算法，其中除了決策樹(C4.5)以外，均是使用 WEKA[30]做為實驗的工具。在此預測分析因有類別不平衡的問題，因此我們以減少多數類別取樣法(under sampling)和增加少數類別取樣法(over sampling)兩種改變取樣次數之演算法，搭配上上述五種學習演算法。除此之外，我們會利用 k-nearest-neighbor 演算法刪除資料中的雜訊(noise)，對資料進行資料清理(data cleaning)，藉此方式平衡兩種類別的資料數量，以提昇分類的準確度。

而整體學習演算法(ensemble learning algorithm)的部分，以決策樹為基礎，使用了 bagging、random forest、bagging + stacking、random forest + stacking 和 AdaBoost 和 DataBoost-IM 五種整體學習演算法，因為在該預設分析上有類別不平衡的問題，所以搭配了減少多數類別取樣法(under sampling)和增加少數類別取樣法(over sampling)，因此除上述五種整體學習演算法外，還使用了 bagging + over sampling、bagging + under sampling、AdaBoost + over sampling、AdaBoost + under sampling 和 DataBoost-IM 五種方法，共十種整體學習演算法。除此之外，也使用 k-nearest-neighbor 演算法對資料進行資料清理(data cleaning)，以刪除資料中的雜訊，藉此改善類別不平衡之問題。

3.4 評量準則

3.4.1 總劑量預測和 PCA 劑量預測

總劑量和 PCA 劑量兩種預測分析均使用了 1099 位病患，都是依據麻醉藥劑量的多寡程度將所有病患分為三個區間，各區間麻醉藥總劑量的標準差類似，意即三類資料的麻醉藥總劑量離散程度相似，將其分為低劑量(low dose)、一般劑量(medium dose)和高劑

量(high dose)。在這兩種預測分析中，我們使用整體準確度(overall accuracy)做為主要評量準則，除此之外，也以各類別的 recall 和 precision 做為評量準則。整體準確度(overall accuracy)為所有資料中，預測正確的比率。各類別的 recall 和 precision 的計算公式如表 3 和表 4，recall 意指在該類別中預測正確的比率，例如，” low dose” Recall 表示類別為低劑量(low dose)的資料中，有多少比率的資料被預測正確；而 precision 表示被預測為該類別的資料中，原始類別確實為該類別的比率，例如，” medium dose” Precision 表示被預測為一般劑量(medium dose)的資料中，其中原始類別亦為一般劑量的資料比率為何。

表 3. Confusion Matrix

| | low dose | medium dose | high dose |
|--------------------------|----------|-------------|-----------|
| low dose (prediction) | ① | ② | ③ |
| medium dose (prediction) | ④ | ⑤ | ⑥ |
| high dose (prediction) | ⑦ | ⑧ | ⑨ |

表 4. 總劑量預測的評量準則之說明

| 評量準則 | 說明 |
|-------------------------|-----------------------------------|
| “low dose” Recall | $\frac{①}{①+④+⑦}$ |
| “medium dose” Recall | $\frac{⑤}{②+⑤+⑧}$ |
| “high dose” Recall | $\frac{⑨}{③+⑥+⑨}$ |
| “low dose” Precision | $\frac{①}{①+②+③}$ |
| “medium dose” Precision | $\frac{⑤}{④+⑤+⑥}$ |
| “high dose” Precision | $\frac{⑨}{⑦+⑧+⑨}$ |
| Overall Accuracy | $\frac{①+⑤+⑨}{①+②+③+④+⑤+⑥+⑦+⑧+⑨}$ |

3.4.2 PCA 裝置參數設定調整預測

在 PCA 裝置設定調整的預測分析中，由於大多數病患在 PCA 的初次設定之後不再需要重新調整，因此容易產生預測類別不平衡的問題，例如，在過去的文獻中[31,32]指出，當資料有類別不平衡的問題時，若只使用整體準確度(overall accuracy)來做為評量準則是不合理的。例如，多數類別(majority class)的資料筆數佔全數的 95%，而少數類別(minority class)僅佔全數的 5%，倘若學習演算法將全數預測為多數類別，則其準確度可高達 95%，然而在此情況之下，所有少數類別的資料都被分類錯誤。在一個分類相當不平衡的資料中，我們有時更為關注少數類別的資料，因為它們通常俱備特殊意義，因此

在有類別不平衡問題的資料中，少數類別的預測準確度才是我們想要改善的重點。就此，我們除了使用整體準確度(overall accuracy)以及各類別的 recall 和 precision 作為參考之外，在該預測分析上，我們增加了幾何平均(geometric mean)[33]和 F-score[34]作為評量準則，其中幾何平均為 positive recall 和 negative recall 的幾何平均。各評量準則之詳細計算方式如表 5 和表 6。

表 5. Confusion Matrix

| | Positive | Negative |
|----------------------|--------------------|--------------------|
| Positive(prediction) | True Positive(TP) | False Positive(FP) |
| Negative(prediction) | False Negative(FN) | True Negative(TN) |

表 6. PCA 裝置參數設定調整預測的評量準則之說明

| 評量準則 | 說明 |
|--------------------|---|
| Positive Recall | $TP/(TP+FN)$ |
| Positive Precision | $TP/(TP+FP)$ |
| Positive F-score | $2 \times \frac{\text{Positive Precision} \times \text{Positive Recall}}{\text{Positive Precision} + \text{Positive Recall}}$ |
| Negative Recall | $TN/(FP+TN)$ |
| Negative Precision | $TN/(TN+FN)$ |
| Negative F-score | $2 \times \frac{\text{Negative Precision} \times \text{Negative Recall}}{\text{Negative Precision} + \text{Negative Recall}}$ |
| Geometric Mean | $\sqrt{\text{Positive Recall} \times \text{Negative Recall}}$ |
| Overall Accuracy | $(TP+TN)/(TP+FP+FN+TN)$ |

3.4.3 檢定方法

在上述三種預測分析中，總劑量預測和 PCA 劑量預測是以整體準確度(overall accuracy)為主要評量準則，PCA 裝置參數設定調整預測因為有類別不平衡(class imbalance)之問題，若使用整體準確度為主要評量準則會有不合理之情況，應以少數類別的預測準確值為其主要改善之重點，而在 PCA 資料中，僅少數病患在 PCA 初始設定後，需要重新調整設定，因此「有做調整」(positive)為少數類別(minority class)，故此預測分析是以 positive F-score 為其主要評量準則。

在總劑量預測、PCA 劑量預測和 PCA 裝置參數設定調整預測上，三種預測分析皆會比較決策樹(C4.5)、簡單貝氏分類器(Naive Bayes Classifier)、類神經網路(Artificial

Neural Network ,ANN)、支援向量機(Support Vector Machine ,SVM)和決策表(Decision Table)五種學習演算法，比較方式是以五種學習演算法中表現最好之演算法為基準，使用 paired t-test 檢定，比較該演算法和其餘四種學習演算法之差異，若其差異之 p-value 小於 0.05，即認為兩者演算法之準確值有顯著差異，以此方式比較五種學習演算法。而整體學習演算法(ensemble learning algorithm)的部分，一樣會以所有整體學習演算法中表現最好之演算法為基準，與其餘之整體學習演算法以 paired t-test 檢定，藉此得知個別演算法之準確值彼此間的差異性。

而在 PCA 裝置調整預測分析上，我們以 k-nearest-neighbor 演算法刪除雜訊(noise)，對資料進行資料清理(data cleaning)，希望藉此方式改善類別不平衡的問題。為檢測資料清理對個別演算法預測類別不平衡之資料是否有幫助，我們會針對相同的演算法，包含五種學習演算法以及十種整體學習演算法，以 paired t-test 檢定，比較未作資料清理和已作資料清理之結果，若已作資料清理之表現優於未作資料清理之結果，且其差異之 p-value 小於 0.05，則表示資料清理對該演算法而言，確實可達到改善之效果。



四、結果與討論

4.1 預測分析

我們希望可以透過資料探勘的技術，讓麻醉醫師更能了解病患使用 PCA 裝置的情況，以期更有效的減輕病患的術後疼痛。因此透過總劑量預測、PCA 劑量預測以及 PCA 裝置參數設定調整預測三種預測分析，希望可以提供給麻醉醫師更多病患使用 PCA 裝置的資訊，譬如，若是預測病患的麻醉藥需求為高劑量，但目前使用的劑量設定較低，可能表示未來該病患會有較多的麻醉藥需求，或是若病患被預測為需要做 PCA 設定的調整，表示該病患在未來可能會有噁心嘔吐等副作用或是無法減輕疼痛等不適的現象，可以提醒麻醉醫師需要特別注意該病患的狀況。

4.2 實驗結果與討論

實驗結果的部分，都是採用 10-fold 交叉驗證(cross validation)法，評估各方法的優劣，而在 bagging 和 random forest 兩種整體學習演算法(ensemble learning algorithm)或是以這兩者搭配其他演算法(stacking、under sampling 和 over sampling)之方法時，例如，bagging + stacking 或 random forest + under sampling 整體學習演算法時，在每次實驗中我們會透過重複取樣以產生 200 組訓練資料，並做十次的 10-fold 交叉驗證求其平均。

而在評量準則的選擇上，總劑量預測分析和 PCA 劑量預測分析是以整體準確度(overall accuracy)為主要評量準則。在 PCA 裝置參數設定調整預測分析上，由於多數病患不需要參數調整，因此有類別不平衡(class imbalance)的問題，不適合使用整體準確度作為主要評量準則，其中我們以 positive class 作為少數類別(minority class)，意即 positive class 的病患需要 PCA 參數的調整。因此，我們以 positive F-score 和幾何平均(geometric mean)做為主要的評量準則。

4.2.1 五種學習演算法之結果比較

我們針對獨立的學習演算法，使用決策樹(C4.5)、簡單貝氏分類器(Naive Bayes Classifier)、類神經網路(Artificial Neural Network ,ANN)、支援向量機(Support Vector

Machine ,SVM)和決策表(Decision Table)五種學習演算法，比較在兩種預測分析上，各學習演算法在預測準確值上之表現，並加以分析結果。

4.2.1.1 總劑量預測

總劑量預測五種學習演算之結果如表 7，其中表現最好之演算法為決策表(Decision Table)，我們將決策表和其餘四種學習演算法的結果作 paired t-test 檢定，若其差異之 p-value 小於 0.05，則在表 7 該演算法上以星號“*”標記之，而 p-value 小於 0.001 者，會於表 7 該演算法上標記“ $\overline{\text{—}}$ ”，表示該演算法和決策表之結果有顯著差異。

表 7. 總劑量預測五種演算法之比較

| 總劑量預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|-------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------|
| “low dose” Recall | 77.4* | 79.0 $\overline{\text{—}}$ | 69.8 | 8.0 $\overline{\text{—}}$ | 74.9 |
| “medium dose” Recall | 72.8 $\overline{\text{—}}$ | 67.7 $\overline{\text{—}}$ | 79.6 | 96.1 $\overline{\text{—}}$ | 78.9 |
| “high dose” Recall | 60.6 $\overline{\text{—}}$ | 38.0 $\overline{\text{—}}$ | 21.6 $\overline{\text{—}}$ | 0.0 $\overline{\text{—}}$ | 48.3 |
| “low dose” Precision | 76.3 | 71.8 $\overline{\text{—}}$ | 80.2* | 59.4 $\overline{\text{—}}$ | 75.6 |
| “medium dose” Precision | 73.5* | 70.2 $\overline{\text{—}}$ | 66.1 $\overline{\text{—}}$ | 50.6 $\overline{\text{—}}$ | 72.4 |
| “high dose” Precision | 62.8 $\overline{\text{—}}$ | 46.3 $\overline{\text{—}}$ | 56.9* | 0.0 $\overline{\text{—}}$ | 73.4 |
| Overall Accuracy | 72.8 | 67.9 $\overline{\text{—}}$ | 68.5 $\overline{\text{—}}$ | 50.7 $\overline{\text{—}}$ | 73.3 |

註：與 Decision Table 之比較：* 表示 p-value < 0.05， $\overline{\text{—}}$ 表示 p-value < 0.001。

4.2.1.2 PCA 劑量預測

PCA 劑量預測五種學習演算之結果如表 8，簡單貝氏分類器(Naive Bayes)為其中表現最好之演算法，我們將簡單貝氏分類器和其餘四種學習演算法的結果作 paired t-test 檢定，以星號“*” 在表 8 上標記差異之 p-value 小於 0.05 之演算法，並以“ $\overline{\text{—}}$ ”標記差異之 p-value 小於 0.001 之演算法，表示該演算法和簡單貝氏分類器之結果有顯著差異。

表 8. PCA 劑量預測五種演算法之比較

| PCA 劑量預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|-------------------------|-------------------|-------------|-------------------------|-------------------------|-------------------------|
| “low dose” Recall | 76.9 [⎯] | 81.4 | 89.2 [*] | 99.9 [⎯] | 82.8 [*] |
| “medium dose” Recall | 51.5 [⎯] | 48.1 | 19.1 [⎯] | 0.0 [⎯] | 42.3 [⎯] |
| “high dose” Recall | 45.4 [⎯] | 50.8 | 8.4 [⎯] | 0.0 [⎯] | 26.8 [⎯] |
| “low dose” Precision | 76.1 | 75.4 | 62.0 [⎯] | 52.8 [⎯] | 68.4 [⎯] |
| “medium dose” Precision | 51.2 [⎯] | 55.6 | 20.7 [⎯] | 0.0 [⎯] | 49.1 [⎯] |
| “high dose” Precision | 49.3 [*] | 51.2 | 22.8 [⎯] | 0.0 [⎯] | 54.4 |
| Overall Accuracy | 64.1 | 65.4 | 54.7[⎯] | 52.7[⎯] | 61.1[⎯] |

註：與 Naive Bayes 之比較：* 表示 p-value < 0.05，[⎯]表示 p-value < 0.001。

4.2.1.3 PCA 裝置參數設定調整預測

表 9 為五種學習演算法針對 PCA 裝置參數設定調整預測分析之結果，其中表現最好的為簡單貝氏分類器(Naive Bayes)，我們利用 paired t-test 檢定比較簡單貝氏分類器(Naive Bayes)和其餘四種學習演算法結果之差異，若其差異之 p-value 小於 0.05，則在表 9 該演算法上以星號“*”標記之若小於 0.001 則以“[⎯]”標記，表示該演算法和簡單貝氏分類器(Naive Bayes)的結果有顯著差異。

從表 9 可以看出，單一的學習演算法在該預測分析上表現大多不佳，明顯受到類別不平衡(class imbalance)問題之影響，預測結果大多傾向於預測多數類別(negative class)，其中類神經網路(ANN)和支援向量機(SVM)在少數類別(positive class)的表現最差，幾乎是全數預測為多數類別(negative class)。

表 9. PCA 裝置調整設定預測分析五種學習演算法之比較，資料未刪除雜訊

| PCA 裝置調整設定預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|--------------------|-------------------------|-------------|------------------------|------------------------|------------------------|
| Positive Recall | 25.5 [⎯] | 65.8 | 0.0 [⎯] | 0.0 [⎯] | 0.8 [⎯] |
| Positive Precision | 25.4 [*] | 23.6 | 0.0 [⎯] | 1.0 [⎯] | 4.8 [⎯] |
| Positive F-score | 25.2[⎯] | 34.5 | 0.0[⎯] | 0.1[⎯] | 1.3[⎯] |
| Negative Recall | 82.5 [⎯] | 49.7 | 100.0 [⎯] | 100.0 [⎯] | 99.0 [⎯] |
| Negative Precision | 82.6 [⎯] | 86.2 | 81.1 [⎯] | 81.1 [⎯] | 81.0 [⎯] |
| Negative F-score | 82.5 [⎯] | 62.7 | 89.5 [⎯] | 89.5 [⎯] | 89.1 [⎯] |
| Geometric Mean | 45.8[⎯] | 56.6 | 0.0[⎯] | 0.2[⎯] | 3.2[⎯] |
| Overall Accuracy | 71.7 [⎯] | 52.7 | 81.1 [⎯] | 81.1 [⎯] | 78.9 [⎯] |

註：與 Naive Bayes 之比較：* 表示 p-value < 0.05，[⎯]表示 p-value < 0.001。

為解決該預測分析類別不平衡之問題，我們搭配了減少多數類別取樣法(under

sampling)和增加少數類別取樣法(over sampling)，利用改變取樣次數以平衡兩種類別資料量。

表 10 為搭配減少多數類別取樣法(under sampling)之結果，其中表現最好之演算法為決策表(Decision Table)，使用 paired t-test 比較決策表(Decision Table)與其餘四種演算法之差異，若其差異之 p-value 小於 0.05，則用星號”*” 標記表 10 該演算法，並以” $\overline{\text{—}}$ ” 標記 p-value 小於 0.001 之該演算法。。

表 10. PCA 裝置調整設定預測分析五種學習演算法之比較，資料未刪除雜訊，搭配減少多數類別取樣法(under sampling)

| PCA 裝置調整設定預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|--------------------|---|-------------|---|--|----------------|
| Positive Recall | 46.8* | 54.9 | 71.4 $\overline{\text{—}}$ | 2.2 $\overline{\text{—}}$ | 52.6 |
| Positive Precision | 22.3 $\overline{\text{—}}$ | 24.9 | 21.2 $\overline{\text{—}}$ | 20.3 | 25.6 |
| Positive F-score | 30.1$\overline{\text{—}}$ | 33.7 | 31.0* | 3.8$\overline{\text{—}}$ | 34.0 |
| Negative Recall | 62.0 | 60.2 | 39.6 $\overline{\text{—}}$ | 98.2 $\overline{\text{—}}$ | 63.3 |
| Negative Precision | 83.3 $\overline{\text{—}}$ | 85.1 | 86.4 | 81.1 $\overline{\text{—}}$ | 85.2 |
| Negative F-score | 70.9* | 69.7 | 47.6 $\overline{\text{—}}$ | 88.8 $\overline{\text{—}}$ | 72.2 |
| Geometric Mean | 53.8* | 56.2 | 40.2$\overline{\text{—}}$ | 8.8$\overline{\text{—}}$ | 56.9 |
| Overall Accuracy | 59.1* | 59.1 | 45.7 $\overline{\text{—}}$ | 80.0 $\overline{\text{—}}$ | 61.3 |

註：與 Decision Table 之比較：* 表示 p-value < 0.05， $\overline{\text{—}}$ 表示 p-value < 0.001。

搭配增加少數類別取樣法(over sampling)之結果如表 11，其中類神經網路(ANN)為其中表現最好之結果，與上述相同，利用 paired t-test 比較類神經網路(ANN)與其餘演算法之差異，並且用星號”*” 在表 11 上標記其差異之 p-value 小於 0.05 之演算法，而 p-value 小於 0.001 之演算法，則以” $\overline{\text{—}}$ ” 標記之。

表 11. PCA 裝置調整設定預測分析五種學習演算法之比較，資料未刪除雜訊，搭配增加少數類別取樣法 (over sampling)

| PCA 裝置調整設定預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|--------------------|-------------------------|-------------------------|-------------|------------------------|-------------------------|
| Positive Recall | 31.6 [⎯] | 62.1 [*] | 67.8 | 1.9 [⎯] | 36.5 [⎯] |
| Positive Precision | 22.0 [*] | 24.2 | 24.8 | 19.3 | 24.4 |
| Positive F-score | 25.3[⎯] | 34.3 | 34.7 | 3.4[⎯] | 28.7[⎯] |
| Negative Recall | 73.7 [⎯] | 53.4 | 48.4 | 98.8 [⎯] | 73.1 [⎯] |
| Negative Precision | 82.2 | 85.9 | 84.2 | 81.2 [*] | 83.2 |
| Negative F-score | 77.6 [⎯] | 65.1 [*] | 57.7 | 89.1 [⎯] | 77.6 [⎯] |
| Geometric Mean | 48.2 | 56.3[*] | 50.2 | 7.5[⎯] | 50.6 |
| Overall Accuracy | 65.7 [⎯] | 55.0 | 52.1 | 80.4 [⎯] | 66.2 [⎯] |

註：與 ANN 之比較：* 表示 p-value < 0.05，[⎯]表示 p-value < 0.001。

除了上述兩種方式外，我們也對資料進行資料清理 (data cleaning)，使用 k-nearest-neighbor 演算法刪除資料中的雜訊 (noise)，希望可以藉此方式平衡兩種類別的個數，以提昇分類的準確度。在資料清理後，兩種類別的資料比例，從 81% (negative class) : 19% (positive class) 降為 65% (negative class) : 35% (positive class)，這表示資料清理能降低類別不平衡的程度。資料清理後，再搭配減少多數類別取樣法 (under sampling) 和增加少數類別取樣法 (over sampling)，以期改善類別不平之問題。

表 12 為資料清理 (data cleaning) 後之結果，表 13 為資料清理後，再搭配減少多數類別取樣法 (under sampling) 之結果，表 14 則是資料清理後，再搭配增加少數類別取樣法 (over sampling) 之結果。其中表現最好之演算法均為簡單貝氏分類器 (Naive Bayes)，同樣地，我們利用 paired t-test 檢定分別比較簡單貝氏分類器 (Naive Bayes) 和其餘四種學習演算法結果之差異，若其差異之 p-value 小於 0.05，則在表 12、表 13 和表 14 該演算法上以星號 “*” 標記之，並以 “[⎯]” 標記 p-value 小於 0.001 之演算法，表示該演算法和簡單貝氏分類器 (Naive Bayes) 的結果有顯著差異。

另外，為檢測資料清理 (data cleaning) 對上述演算法而言，是否對提升預測準確值有其效果，我們針對相同的演算法，將未作資料清理和已做資料清理的結果作 paired t-test 檢定，若其差異之 p-value 小於 0.05，則在表 12、表 13 和表 14 三者已作資料清理之結果，在該演算法的結果上以井字號 “#” 標記之，並以 “※” 標記 p-value 小於 0.001

者，表示使用資料清理與否對該演算法而言有顯著差異。從表 9 和表 12 可以看出，資料清理後五種學習演算法中，決策樹(C4.5)、支援向量機(SVM)和決策表(Decision Table)三種學習演算法在 positive F-score 和幾何平均的表現上都有顯著提升，表示資料清理對這三種演算法而言都是有幫助的。而在類神經網路(ANN)上，即使做了資料清理，對該演算法依舊沒有幫助。此外，只有簡單貝氏分類器(Naive Bayes)資料清理後，預測準確值是些微下降的。而搭配減少多數類別演算法(under sampling)的部分，從表 10 和表 13 比較之下得知，決策樹(C4.5)、類神經網路(ANN)和支援向量機(SVM)三種學習演算法資料清理後的表現是優於未作資料清理之結果，但其中只有支援向量機(SVM)是有顯著提升的，而簡單貝氏分類器(Naive Bayes)和決策表(Decision Table)兩種演算法資料清理後，預測準確值略微降低。此外，搭配增加少數類別演算法(over sampling)的部分，可以比較表 11 和表 14，發現在決策樹(C4.5)、支援向量機(SVM)和決策表(Decision Table)三者的表現上，資料清理後皆可顯著提升預測準確值，但資料清理對簡單貝氏分類器(Naive Bayes)和類神經網路(ANN)，似乎沒有明顯的改善。

就上述而言，利用 k-nearest-neighbor 演算法對資料進行資料清理(data cleaning)，對大多數的演算法而言，都有顯著之改善，其中以決策樹(C4.5)和支援向量機(SVM)最為明顯，在資料清理後，配合減少多數類別取樣法(under sampling)和增加多數類別取樣法(over sampling)，與未作資料清理之結果相較準確度均有顯著提升。

4.2.1.4 五種學習演算法之結果討論

在總劑量預測分析中，五種學習演算法表現最好的是決策表(Decision Table)，整體準確度(overall accuracy)為 73.3%，而決策樹(C4.5)之結果僅略差於決策表，整體準確度為 72.8%。PCA 劑量預測分析中，則是以簡單貝氏分類器(Naive Bayes)為表現最佳之演算法，整體準確度為 65.4%，決策樹(C4.5)的整體準確度則為 64.1%，兩者僅些微的落差。而在 PCA 裝置參數設定調整預測的部分，全數表現最好之結果為類神經網路(ANN)搭配增加少數類別取樣法(over sampling)，positive F-score 為 34.7%，第二之結果為簡單貝氏分類器(Naive Bayes)搭配增加少數類別取樣法(over sampling)，positive F-score 為

34.3%，而決策樹(C4.5)最好的結果，為搭配資料清理和增加少數類別取樣法(over sampling)，positive F-score 為 31.2%。

雖然在三種預測分析中，決策樹(C4.5)不是表現最好之演算法。但是，決策表(Decision Table)在訓練資料時，比起決策樹需要花費較長的時間。而類神經網路(ANN)無法像決策樹一樣提供可見之分類規則，亦無法從分類的過程中得知何者為較為重要之屬性。簡單貝氏分類器(Naive Bayes)則沒有辦法直接處理連續值之屬性，一般需假設該屬性符合常態分配(normal distribution)或是先對該連續值之屬性做分類，將其轉成離散值才可處理。

決策樹在可讀性上有相當高的優勢，除了可轉成易讀的 if-else 規則，我們還可從決策樹中找出較為重要之屬性，將此資訊提供給麻醉醫師參考，而且屬性不管為離散值或連續值均可直接運算，學習和分類的速度也較其他演算法來的快。因此，即使決策樹(C4.5)並非表現最好之結果，但在可讀性以及預測準確度之平衡下，我們還是以決策樹為基礎演算法，發展整體學習演算法，希望透過整體學習演算法可提升決策樹(C4.5)之準確度，並且從其中找出較為關鍵之屬性。

表 12. PCA 裝置調整設定預測分析五種學習演算法之比較，資料已刪除雜訊

| PCA 裝置調整設定預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|--------------------|---------------------------|-------------------------|------------------------|---------------------------|---------------------------|
| Positive Recall | 43.5 ^{‡**} | 69.7 [*] | 0.0 [‡] | 3.5 ^{‡**} | 28.2 ^{‡**} |
| Positive Precision | 23.2 [#] | 22.0 [*] | 0.0 [‡] | 27.8 ^{**} | 27.9 ^{‡**} |
| Positive F-score | 30.2^{‡**} | 33.4[*] | 0.0[‡] | 6.1^{‡**} | 27.2^{‡**} |
| Negative Recall | 66.3 ^{‡**} | 42.1 [*] | 100.0 [‡] | 98.3 ^{‡**} | 82.8 ^{‡**} |
| Negative Precision | 83.4 ^{‡#} | 85.7 [#] | 81.1 [‡] | 81.4 ^{‡**} | 83.2 ^{‡**} |
| Negative F-score | 73.8 ^{‡**} | 56.2 [*] | 89.5 [‡] | 89.0 ^{‡**} | 82.9 ^{‡**} |
| Geometric Mean | 53.7[*] | 53.9[*] | 0.0[‡] | 13.1^{‡**} | 46.9^{‡**} |
| Overall Accuracy | 62.0 ^{‡**} | 46.6 [*] | 81.1 [‡] | 80.4 ^{‡**} | 72.4 ^{‡**} |

註 1：與 Naive Bayes 之比較：* 表示 p-value < 0.05，[‡]表示 p-value < 0.001。

註 2：雜訊清理前後之比較：# 表示 p-value < 0.05，^{*}表示 p-value < 0.001。

表 13. PCA 裝置調整設定預測分析五種學習演算法之比較，資料已刪除雜訊，搭配減少多數類別取樣法 (under sampling)

| PCA 裝置調整設定預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|--------------------|-------------------------|-------------------------|-------------------------|---------------------------|--------------------|
| Positive Recall | 52.9 ^{†#} | 67.0 [#] | 72.5 [*] | 6.5 ^{†**} | 53.4 [†] |
| Positive Precision | 22.0 | 22.6 ^{**} | 22.7 | 23.6 | 24.6 ^{†#} |
| Positive F-score | 31.0[*] | 33.6[#] | 32.4 | 9.7^{†**} | 33.2 |
| Negative Recall | 56.2 ^{†**} | 45.6 ^{**} | 39.2 [*] | 94.9 ^{†**} | 61.1 ^{†#} |
| Negative Precision | 83.7 [*] | 85.4 | 85.8 | 81.3 ^{†#} | 85.0 |
| Negative F-score | 67.1 ^{†**} | 58.9 ^{**} | 48.8 [†] | 87.6 ^{†**} | 70.4 ^{†#} |
| Geometric Mean | 54.5 | 54.6[#] | 44.5[†] | 20.9^{†**} | 55.8 |
| Overall Accuracy | 55.6 ^{†**} | 49.7 [*] | 45.6 [*] | 78.2 ^{†**} | 59.6 ^{†#} |

註 1：與 Naive Bayes 之比較：* 表示 p-value < 0.05，[†]表示 p-value < 0.001。

註 2：雜訊清理前後之比較：# 表示 p-value < 0.05，**表示 p-value < 0.001。

表 14. PCA 裝置調整設定預測分析五種學習演算法之比較，資料已刪除雜訊，搭配增加少數類別取樣法 (over sampling)

| PCA 裝置調整設定預測(%) | C4.5 | Naive Bayes | ANN | SVM | Decision Table |
|--------------------|---------------------------|-------------------------|---------------------------|---------------------------|-------------------------|
| Positive Recall | 53.0 ^{†**} | 68.8 [#] | 64.7 | 5.8 ^{†**} | 51.4 ^{†**} |
| Positive Precision | 22.8 | 22.2 ^{**} | 19.5 ^{**} | 21.9 | 24.0 [*] |
| Positive F-score | 31.2^{†**} | 33.5[#] | 26.1^{†**} | 8.9^{†**} | 32.5[#] |
| Negative Recall | 58.0 ^{†**} | 43.6 ^{**} | 42.4 | 95.2 ^{†**} | 61.7 ^{†**} |
| Negative Precision | 84.2 ^{**#} | 85.7 | 84.8 | 81.3 [†] | 84.5 ^{**#} |
| Negative F-score | 68.5 ^{†**} | 57.4 ^{**} | 45.9 ^{**#} | 87.7 ^{†**} | 70.9 ^{†**} |
| Geometric Mean | 55.4[*] | 54.3[*] | 29.2^{†**} | 19.7^{†**} | 55.5[*] |
| Overall Accuracy | 57.1 ^{†**} | 48.3 ^{**} | 46.6 [#] | 78.3 ^{†**} | 59.7 ^{†**} |

註 1：與 Naive Bayes 之比較：* 表示 p-value < 0.05，[†]表示 p-value < 0.001。

註 2：雜訊清理前後之比較：# 表示 p-value < 0.05，**表示 p-value < 0.001。

4.2.2 整體學習演算法之結果比較

我們以決策樹為基礎演算法，使用 bagging、bagging + stacking、random forest、random forest + stacking 和 AdaBoost 五種整體學習演算法(ensemble learning algorithm)。另外，因為在 PCA 裝置參數設定調整預測分析上，有類別不平衡(class imbalance)的問題，所以還搭配了減少多數類別取樣法(under sampling)和增加少數類別取樣法(over sampling)，因此除上述五種整體學習演算法外，還使用了 bagging + over sampling、bagging + under sampling、AdaBoost + over sampling、AdaBoost + under sampling 和

表 15. 總劑量預測分析之結果

| 總劑量預測(%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost |
|-------------------------|-------------------------|--------------|-------------------------|-------------------------|-----------------------------|-------------------------|
| “low dose” Recall | 77.4 [¯] | 84.3 | 84.7 | 80.1 [¯] | 80.3 [¯] | 79.2 [¯] |
| “medium dose” Recall | 72.8 [¯] | 83.5 | 79.9 [¯] | 83.6 | 77.6 [¯] | 75.8* |
| “high dose” Recall | 60.6* | 62.4 | 66.8 [¯] | 47.4 [¯] | 58.1* | 61.2 |
| “low dose” Precision | 76.3 [¯] | 84.3 | 82.5 [¯] | 81.4 [¯] | 78.3 [¯] | 78.8 [¯] |
| “medium dose” Precision | 73.5 [¯] | 79.7 | 80.3* | 74.8 [¯] | 75.9 [¯] | 75.3 [¯] |
| “high dose” Precision | 62.8 [¯] | 78.5 | 73.2 [¯] | 80.4* | 71.1 [¯] | 66.0 [¯] |
| Overall Accuracy | 72.8[¯] | 80.9 | 79.9[¯] | 77.4[¯] | 75.9[¯] | 75.1[¯] |

註：與 Bagging 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

4.2.2.2 PCA 劑量預測

使用五種整體學習演算法(ensemble learning algorithm)對 PCA 劑量作預測之結果如表 16。在該預測分析中，C4.5 再配合 bagging 為其中表現最好的整體學習演算法。我們利用 paired t-test 比較 bagging 和其餘方法的準確度，在表 16 上以星號“*”標記差異之 p-value 小於 0.05 之該演算法，若其 p-value 小於 0.001 則以“[¯]”標記之。從表 16 可看出，在整體準確度的表現上，C4.5 配合 bagging 的確優於其餘的整體學習演算法，而且都有顯著差異。

表 16. PCA 劑量預測分析之結果

| PCA 劑量預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost |
|-------------------------|-------------------------|--------------|-----------------------|-------------------------|-----------------------------|-------------------------|
| “low dose” Recall | 76.9 [⌘] | 84.3 | 81.0* | 95.4 [⌘] | 87.6 [⌘] | 79.0 [⌘] |
| “medium dose” Recall | 51.5 [⌘] | 65.8 | 63.8* | 47.2 [⌘] | 54.8 [⌘] | 54.3 [⌘] |
| “high dose” Recall | 45.4* | 47.5 | 49.4* | 31.8 [⌘] | 33.8 [⌘] | 45.1* |
| “low dose” Precision | 76.1 [⌘] | 81.7 | 82.1* | 73.2 [⌘] | 75.8 [⌘] | 77.4 [⌘] |
| “medium dose” Precision | 51.2 [⌘] | 60.7 | 59.0 [⌘] | 61.4* | 57.9 [⌘] | 53.6 [⌘] |
| “high dose” Precision | 49.3 [⌘] | 75.1 | 67.6 [⌘] | 85.7 [⌘] | 69.8 [⌘] | 52.3 [⌘] |
| Overall Accuracy | 64.1[⌘] | 73.1 | 71.0* | 70.6[⌘] | 69.4[⌘] | 66.1[⌘] |

註：與 Bagging 之比較：* 表示 p-value < 0.05，[⌘]表示 p-value < 0.001。

4.2.2.3 PCA 裝置參數設定調整預測

表 17 為 PCA 裝置參數設定調整使用整體學習演算法(ensemble learning algorithm)之預測結果，除了 bagging、bagging + stacking、random forest、random forest + stacking 和 AdaBoost 五種整體學習演算法，還使用了 bagging + over sampling、bagging + under sampling、AdaBoost + over sampling、AdaBoost + under sampling 和 DataBoost-IM 五種方法，共十種整體學習演算法。其中 bagging + under sampling 演算法在 positive F-score 和幾何平均(geometric mean)的表現上都是最好的。我們利用 paired t-test 比較 bagging + under sampling 和其餘演算法，假若 bagging + under sampling 與其他演算法的預測準確值有明顯差異，且其 p-value 小於 0.05，我們以星號“*”標記之，而 p-value 小於 0.001 者，則以“[⌘]”標記之。從表 17 可看出，bagging + under sampling 與其餘演算法相較，皆有顯著差異，表示 bagging + under sampling 演算法在該預測分析上，確實是比其餘演算法來得好。

在該預測分析中，因有類別不平衡(class imbalance)之問題，我們使用 k-nearest-neighbor 演算法刪除資料中的雜訊(noise)，對資料進行資料清理(data

cleaning)。資料清理後，兩種類別的資料比例，從 81%(negative class) : 19%(positive class) 降為 65%(negative class) : 35%(positive class)，表示資料清理可改善類別不平衡之程度。表 18 為資料清理後十種整體學習演算法的預測結果，從表 17 和表 18 可以看出，所有的整體學習演算法資料清理後在 positive F-score 和幾何平均的表現上都優於未做資料清理的結果。針對相同的演算法，我們將未做資料清理和已做資料清理的結果做 paired t-test 檢定，若其差異之 p-value 小於 0.05，則在表 18 該演算法的結果上標記”#”，並以”※”標記 p-value 小於 0.001 之該演算法。從表 18 得知，所有的方法中，除了 AdaBoost + under sampling 外，其餘的演算法在採用資料清理之後，其預測準確值皆顯著提升。我們的實驗證明，利用 k-nearest-neighbor 演算法執行資料清理(data cleaning)，對提高分類的準確度確實是有幫助的。

此外，在使用過資料清理的所有演算法中，針對 positive F-score 和幾何平均 (geometric mean)，其中表現最好的演算法為 bagging + under sampling，我們同樣使用 paired t-test，將其與其他方法做比較，於表 18 中，以星號”*”標記 p-value 小於 0.05 之比較結果，並以”⎯”標記 p-value 小於 0.001 之結果，表 18 之結果顯示，bagging + under sampling 演算法，在使用過資料清理(data cleaning)之後，其預測分析的表現上確實優於其餘的演算法。

在 4.2.1.4 中提到，五種單一的學習演算法中，表現最好之演算法為 ANN(類神經網路) + over sampling 以及 Naive Bayes(簡單貝氏分類器) + over sampling，我們將這兩種方法的結果和整體演算法中表現最好之 bagging + under sampling 演算法之結果相互比較。比較結果如表 19，同樣以 paired t-test 檢定做比較，且在表 19 中，以星號”*”標記 p-value 小於 0.05 之結果，同樣地，以”⎯”標記 p-value 小於 0.001 之結果，其中 bagging + under sampling 搭配資料清理，在 positive F-score 和幾何平均(geometric mean)的表現皆優於 ANN + over sampling 和 Naive Bayes + over sampling，雖然 positive F-score 僅些微的獲勝，但我們從 negative F-score 的角度來看的話，可以發現 bagging + under sampling 搭配資料清理，明顯優於另外兩種演算法，表示 ANN + over sampling 和 Naive Bayes + over sampling 在提升少數類別(positive class)預測準確度的同時，犧牲了較多的多數類別

(negative class)之預測準確度。大多數在解決類別不平衡的問題時，為了改善少數類別 (minority class)的預測準確度，多數類別(majority class)通常會因此被犧牲而降低其預測準確度，雖然在此問題中，少數類別較受到關注，其準確度為我們想要改善之重點，但也不希望在提升少數類別的預測準確度的同時，犧牲掉太多的多數類別之預測準確度。然而，bagging + under sampling 搭配資料清理除了可提升少數類別的預測準確度外，在多數類別的準確度的部分，相較於 ANN + over sampling 和 Naive Bayes + over sampling 也並未降低太多。

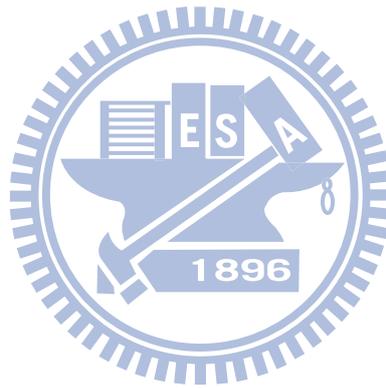


表 17. PCA 裝置調整設定預測分析之結果，資料尚未刪除雜訊

| PCA 裝置調整設定預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Bagging Over Sampling | C4.5 Bagging Under Sampling | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost | C4.5 AdaBoost Over Sampling | C4.5 AdaBoost Under Sampling | C4.5 DataBoost-IM |
|--------------------|-------------------------|-------------------------|-------------------------|----------------------------|-----------------------------|-------------------------|-----------------------------|-------------------------|-----------------------------|------------------------------|-------------------------|
| Positive Recall | 25.5 [¯] | 4.3 [¯] | 9.4 [¯] | 16.1 [¯] | 41.5 | 2.6 [¯] | 25.9 [¯] | 19.6 [¯] | 32.8 [¯] | 47.3 [¯] | 19.7 [¯] |
| Positive Precision | 25.4 [*] | 39.3 [*] | 32.7 [*] | 28.2 | 27.5 | 38.2 [*] | 28.0 | 27.9 | 24.6 [*] | 23.1 [¯] | 23.7 [*] |
| Positive F-score | 25.2[¯] | 7.5[¯] | 13.8[¯] | 20.2[¯] | 32.9 | 4.7[¯] | 26.6[¯] | 22.5[¯] | 27.9[¯] | 30.9[*] | 20.7[¯] |
| Negative Recall | 82.5 [¯] | 98.7 [¯] | 94.1 [¯] | 90.4 [¯] | 74.4 | 99.6 [¯] | 84.5 [¯] | 88.2 [¯] | 76.5 [*] | 62.9 [¯] | 85.4 [¯] |
| Negative Precision | 82.6 [¯] | 81.6 [¯] | 81.0 [*] | 82.2 [¯] | 84.5 | 81.4 [¯] | 83.0 [¯] | 82.5 [¯] | 83.0 [¯] | 83.7 [*] | 82.0 [¯] |
| Negative F-score | 82.5 [¯] | 89.3 [¯] | 87.0 [¯] | 86.1 [¯] | 79.0 | 89.6 [¯] | 83.7 [¯] | 85.2 [¯] | 79.5 | 71.7 [¯] | 83.6 [¯] |
| Geometric Mean | 45.8[¯] | 20.4[¯] | 29.6[¯] | 38.1[¯] | 55.5 | 15.9[¯] | 46.6[¯] | 41.4[¯] | 50.0[¯] | 54.5 | 40.9[¯] |
| Overall Accuracy | 71.7 [¯] | 80.9 [¯] | 78.1 [¯] | 76.4 [¯] | 68.1 | 81.2 [¯] | 73.4 [¯] | 75.2 [¯] | 68.2 | 60.0 [¯] | 73.0 [¯] |

註：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

表 18. PCA 裝置調整設定預測分析之結果，資料已刪除雜訊

| PCA 裝置 調整設定預 測(%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Bagging Over Sampling | C4.5 Bagging Under Sampling | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost | C4.5 AdaBoost Over Sampling | C4.5 AdaBoost Under Sampling | C4.5 DataBoost-IM |
|-------------------------|---------------------|---------------------|-----------------------------|-------------------------------------|--------------------------------------|--------------------------|--------------------------------------|---------------------|--------------------------------------|---------------------------------------|----------------------|
| Positive Recall | 43.5 ^{¯**} | 40.7 ^{¯**} | 42.1 ^{¯**} | 51.1 ^{¯**} | 54.4 [*] | 31.8 ^{¯**} | 35.6 ^{¯**} | 42.5 ^{¯**} | 55.4 [*] | 54.0 [#] | 49.6 ^{¯**} |
| Positive Precision | 23.2 ^{¯#} | 29.5 ^{¯#} | 29.6 [¯] | 26.1 [#] | 26.2 [#] | 30.2 ^{¯#} | 28.6 [¯] | 24.5 ^{*#} | 23.6 [¯] | 22.3 [¯] | 24.3 [¯] |
| Positive F-score | 30.2 ^{¯**} | 33.9 ^{**} | 34.5 ^{**} | 33.7 ^{**} | 35.2 [*] | 30.5 ^{¯**} | 31.3 ^{¯#} | 30.9 ^{¯**} | 33.0 ^{¯**} | 31.4 [¯] | 31.5 ^{¯**} |
| Negative Recall | 66.3 ^{¯**} | 77.2 ^{¯**} | 76.5 ^{¯**} | 66.0 ^{¯**} | 64.0 [*] | 82.7 ^{¯**} | 79.2 ^{¯**} | 69.5 ^{¯**} | 57.9 ^{¯**} | 55.9 ^{¯**} | 63.8 [*] |
| Negative Precision | 83.4 ^{¯#} | 84.8 ^{¯**} | 85.0 ^{¯**} | 85.3 ^{**} | 85.7 [*] | 83.9 ^{¯**} | 84.1 ^{¯#} | 83.9 ^{¯**} | 84.8 ^{**} | 84.0 [¯] | 84.5 ^{¯**} |
| Negative F-score | 73.8 [*] | 80.8 ^{¯**} | 80.5 ^{¯**} | 74.3 ^{¯**} | 73.1 [*] | 83.2 ^{¯**} | 81.5 ^{¯**} | 75.9 ^{¯**} | 68.6 ^{¯**} | 66.9 ^{¯**} | 72.5 [*] |
| Geometric Mean | 53.7 ^{¯**} | 56.0 ^{¯**} | 56.7 ^{¯**} | 58.1 ^{**} | 59.0 [*] | 51.2 ^{¯**} | 53.1 ^{¯**} | 54.3 ^{¯**} | 56.6 ^{¯**} | 54.9 [¯] | 56.2 ^{¯**} |
| Overall Accuracy | 62.0 [*] | 70.3 ^{¯**} | 70.0 ^{¯**} | 63.2 ^{**} | 62.1 [*] | 73.0 ^{¯**} | 71.0 ^{¯**} | 64.4 ^{¯**} | 57.4 ^{¯**} | 55.6 ^{¯**} | 61.1 ^{**} |

註 1：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

註 2：雜訊清理前後之比較：# 表示 p-value < 0.05，**表示 p-value < 0.001。

表 19. 學習演算法和整體學習演算法表現較佳者之相互比較

| PCA 裝置調整設定預測(%) | C4.5 Bagging Under Sampling | Naive Bayes Over sampling | ANN Over Sampling |
|--------------------|--------------------------------|------------------------------|-------------------------|
| Positive Recall | 54.4 | 62.1 [*] | 67.8 [⌣] |
| Positive Precision | 26.2 | 24.2 [*] | 24.8 |
| Positive F-score | 35.2 | 34.3[*] | 34.7 |
| Negative Recall | 64.0 | 53.4 [⌣] | 48.4 [⌣] |
| Negative Precision | 85.7 | 85.9 | 84.2 |
| Negative F-score | 73.1 | 65.1 [⌣] | 57.7 [⌣] |
| Geometric Mean | 59.0 | 56.3[⌣] | 50.2[⌣] |
| Overall Accuracy | 62.1 | 55.0 [⌣] | 52.1 [*] |

註：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，[⌣]表示 p-value < 0.001。

4.2.3 屬性分析

由於決策樹學習演算法採行資料分割(data partition)的方法，佐以遞迴的模式，不斷分割現有的資料集，以期搜尋最佳的分類決策樹。因此，在根部(root)所挑選的屬性，及其他距離根部較近的節點(node)所挑選之屬性，它們所能參考的資料相較於距離葉子(leaf)較近的節點而言，其資料量相對較多。換言之，距根部較近(且含根部)的屬性具有較高之重要性。若以根部為第一層，我們想要統計在決策樹中，在前五層中各個屬性被挑選為該階層屬性的次數，例如：age 在決策樹集合中，被挑選為第一層屬性的頻率為多少。希望藉此方式，找出對該預測分析而言，在 280 個屬性中，哪些屬性才是較為關鍵之屬性。

4.2.3.1 總劑量預測

在總劑量預測分析中，bagging 演算法在整體準確度(overall accuracy)的表現上是最好的。而利用 bagging 演算法，在 10-fold 交叉驗證中，因為每一次驗證都會產生 200 組訓練資料集合，由這 200 個資料集合，我們可以產生 200 棵決策樹，因此，10 個 folds 總共會產生 2000 棵決策樹，以根部(root)為第一層，我們統計第一到第五層中，被決策樹挑選為節點(node)的頻率為前十名之屬性，表 20 為統計結果，數字表示該屬性在該層中的出現頻率，例如：contidose_24hr 在這 2000 棵決策樹中，被挑選為根部的頻率有 74.8%，為出現次數最高之屬性，而出現頻率第二名的屬性則是 contidose_23hr，有 18.4%。

從表 20 可以看出，出現頻率較高之屬性大多為使用 PCA 情況相關屬性，例如：contidose_24hr、p_timediff_var_17hr 和 pcadose_21hr 等。此外，我們還發現，根部挑選的屬性全數都是 contidose 屬性，但在第二層到五層，contidose 被挑選的次數就降低許多。然而在第二層到第五層，被挑選到的屬性大多是 pcadose 居多。以這樣的結果看起來，contidose 屬性是比 pcadose 屬性來得更為重要，這似乎和直覺有些差異，因為這是一個使用 PCA 裝置的總劑量預測分析，照理說經由 PCA 裝置所注射的麻醉藥劑量應該較為重要才對。因此，我們統計了所有的病患麻醉藥的來源比例，發現有 78.1% 的病患經由點滴方式被注射的麻醉藥劑量是高於經由 PCA 裝置被注射的劑量的，我們推測這就是 contidose 屬性大多在根部被挑選到的原因。

我們再利用表 20 作進一步的運算，使第一層有較高的權重，反之第五層的權重最低，以此方式計算所有出現在第一層到第五層中的出現頻率為前十高之所有屬性之重要性，例如，contidose_24hr 為第一層出現頻率最高之屬性，出現頻率為 74.8%，而它在第二層到第五層也均為前十高屬性之一，出現頻率分別為第二層的 4.5%、第三層的 12.9%、第四層的 11.6% 和第五層的 8.6%，第一層的出現頻率有最高之權重，反之第五層最低，因此，運算方式為， $74.8*5+4.5*4+12.9*3+11.6*2+8.6*1$ ，結果為 462.5，即以此作為 contidose_24hr 屬性的分數，對表 20 所有有被挑選到的屬性計算該分數，再以此分數排名，最後挑選出十個分數最高之屬性，為該預測分析中，我們覺得最為關鍵之屬性，表 21 即為最後挑選之結果，可以看出，這十個較為關鍵之屬性大多為 contidose(病患經由點滴所注射之單位小時麻醉藥劑量)和 pcadose(病患經由點滴所注射之單位小時麻醉藥劑量)兩種屬性。

針對 contidose 的部分，我們利用單因子變異數分析(One-way ANOVA)，分析 contidose 所有的時間點(contidose_1hr~contidose_24hr)和總劑量分類結果之相關性，計算結果如表 22，” $-\log_{10}(p\text{-value})$ ” 越高者即表示利用單因子變異數分析，該屬性為較為重要之屬性。而從表 21 得知，決策樹中被挑選到的 contidose 屬性有 contidose_24hr、contidose_23hr 和 contidose_22hr 三個時間點，計算這三個屬性的” $-\log_{10}(p\text{-value})$ ” (如表 22)之平均值為 152.4。此外，我們再利用蒙地卡羅法(Monte Carlo method)，從

contidose_1hr~contidose_24hr中隨機挑選出三個時間點，計算隨機挑選出的時間點之” $-\log_{10}(\text{p-value})$ ”之平均，結果為 113.1。可以證實從決策樹中挑選出的較為重要的三個時間點的屬性，並非隨機挑選的結果，而是確實有較高之重要性。

我們再計算contidose_1hr~contidose_24hr的” $-\log_{10}(\text{p-value})$ ”之平均，發現超過該平均值之間點共有 17 個，分別是 6hr、7hr、9hr、11hr~24hr的contidose，而從決策樹中挑選到的三個關鍵之屬性為 22hr、23hr和 24hr，發現從單因子變異數分析中挑選到的重要之屬性有 17.6%的比例也為從決策樹中挑選到的關鍵之屬性，而從決策樹中挑選到的屬性則 100%均為單因子變異數中挑選到之屬性，以此計算出的F-score為 30%。

而pcadose的部分，我們一樣利用單因子變異數分析(One-way ANOVA)，分析pcadose所有的時間點和總劑量分類結果之相關性，表 23 為其結果。決策樹中被挑選到的pcadose屬性則有pcadose_9hr、pcadose_21hr、pcadose_3hr、pcadose_19hr和pcadose_2hr五個時間點的pcadose，計算這五個屬性的” $-\log_{10}(\text{p-value})$ ”之平均為 11.4。利用蒙地卡羅法，從 24 個時間點中隨機挑選出五個時間點，計算隨機挑選出的時間點之” $-\log_{10}(\text{p-value})$ ”之平均，結果為 10.9。因此，從決策樹中挑選出較為重要的五個時間點之pcadose屬性，可以證實並非隨機挑選的結果。

和contidose相同，計算pcadose_1hr~pcadose_24hr的” $-\log_{10}(\text{p-value})$ ”之平均，發現有 10 個時間點超過該平均值，分別是 2hr、3hr、6hr、9hr、11hr、14hr、17hr、18hr、20hr和 22hr的pcadose，而被決策樹中挑選到的五個關鍵之屬性為 2hr、3hr、9hr、19hr和 21hr，發現從單因子變異數分析中挑選到的重要之屬性有 30%的比例也為從決策樹中挑選到的關鍵之屬性，而從決策樹中挑選到的屬性則 60%均為單因子變異數中挑選到之屬性，以此計算出的F-score為 40%。

在過去的文獻中[35-40]，都是以病患的生理資訊，例如，年齡、體重等，來做預測分析，那在我們的研究中，我們除了使用病患的生理資訊外，還利用病患使用 PCA 裝置的相關屬性用以預測分析，而從表 20 也可以看出，在總劑量預測分析中，這些病患使用 PCA 裝置的相關屬性，的確也被決策樹挑選為較為重要之屬性。而在過去的研究中，年齡[35-37]、性別[38-39]和體重[40]常被挑選為較為關鍵之屬性，因此，我們將這

三個屬性(age、sex 和 weight)和另外三個在過去文獻中並未被說明為較為重要之屬性(BP_HIGH、BP_LOW 和 PULSE)，分別和總劑量分類結果做相關性分析，結果如表 23，可以看出年齡、性別和體重的確是有較高之相關性。而在我們的研究中，因為我們考量了過去文獻中未被考慮之屬性，即病患使用 PCA 之相關情形，因此被決策樹挑選的關鍵屬性均以此為主，但是過去文獻中提到的性別、年齡和體重和其他的生理資訊相比，依舊為較為重要之屬性。

4.2.3.2 PCA 劑量預測

PCA 劑量預測分析的部分，所有演算法中在整體準確度(overall accuracy)的表現上以 bagging 演算法為最佳，我們收集 bagging 演算法在 10-fold 交叉驗證時產生的 2000 棵決策樹，以根部(root)為第一層，統計 280 個屬性，被決策樹挑選為第一層到第五層的節點(node)，挑選次數為前十名之屬性，統計結果如表 25，數字表示被挑選為該層屬性之頻率，例如，pcadose_9hr 在這 2000 棵決策樹中，被挑選為根部的頻率為 24.4%，為出現頻率最高之屬性，而出現頻率第二名的屬性則是 pcadose_19hr，為 17.3%。

和總劑量預測分析相同，我們對表 25 作進一步的運算，以計算出的分數排名，挑選十個分數最高之屬性如表 26，發現這十個較為關鍵之屬性大多為pcadose(病患經由點滴所注射之單位小時麻醉藥劑量)屬性和p_timediff_mean(單位小時病患按壓PCA裝置時間差之平均)屬性。因此我們利用單因子變異數分析(One-way ANOVA)，分析pcadose所有的時間點(pcadose_1hr~pcadose_24hr)和PCA劑量分類結果之相關性，表 27 為分析之結果。而該預測分析中，被決策樹挑選到的pcadose屬性有pcadose_9hr、pcadose_6hr、pcadose_19hr、pcadose_14hr 和 pcadose_11hr 五個時間點，這五個屬性的” $-\log_{10}(\text{p-value})$ ”之平均值為 44.9。一樣利用蒙地卡羅法，從pcadose_1hr~pcadose_24hr 中隨機挑選出五個時間點，計算隨機挑選出的時間點之” $-\log_{10}(\text{p-value})$ ”之平均，結果為 38.4。這說明了利用決策樹挑選出的較為關鍵之五個時間點的屬性，和隨機挑選的結果相比，決策樹挑選的屬性確實有較高之重要性。

同樣地，我們計算pcadose_1hr~pcadose_24hr的” $-\log_{10}(\text{p-value})$ ”之平均，發現超

過該平均值之間點共有 13 個，分別是 2hr~6hr、9hr、11hr、12hr、14hr、17hr、19hr、20hr和 24hr的pcadose，而從決策樹中挑選到的五個關鍵之屬性為 6hr、9hr、11hr、14hr 和 19hr，從單因子變異數分析中挑選到的重要之屬性有 38.5%的比例也為從決策樹中挑選到的關鍵之屬性，而從決策樹中挑選到的屬性則 100%均為單因子變異數中挑選到之屬性，以此計算出的F-score為 55.6%。

而 p_timediff_mean 屬性亦同，利用單因子變異數分析(One-way ANOVA)，分析 p_timediff_mean 所有的時間點和PCA劑量分類結果之相關性，表 28 為其結果。決策樹中被挑選到的 p_timediff_mean 屬性有 p_timediff_mean_22hr、p_timediff_mean_14hr、p_timediff_mean_9hr和p_timediff_mean_17hr四個時間點的 p_timediff_mean，這四個屬性的” $-\log_{10}(\text{p-value})$ ” 之平均為 14.3。利用蒙地卡羅法，從 24 個時間點中隨機挑選出五個時間點，計算隨機挑選出的時間點之” $-\log_{10}(\text{p-value})$ ” 之平均，結果為 10.54。因此，從決策樹中挑選出較為重要的四個時間點之 p_timediff_mean 屬性，可以證實並非隨機挑選的結果。

同樣地，我們計算 p_timediff_mean_1hr~p_timediff_mean_24hr的” $-\log_{10}(\text{p-value})$ ” 之平均，發現有 11 個時間點超過該平均值，分別是 4hr~6hr、9hr、11hr、12hr、14hr~17hr、和 20hr的 p_timediff_mean，而被決策樹中挑選到的四個關鍵之屬性為 9hr、14hr、17hr、和 22hr，發現從單因子變異數分析中挑選到的重要之屬性有 27.3%的比例也為從決策樹中挑選到的關鍵之屬性，而從決策樹中挑選到的屬性則 75%均為單因子變異數中挑選到之屬性，以此計算出的F-score為 40%。

和總劑量預測分析相同，我們利用過去文獻中找出的關鍵之三個屬性(age、sex 和 weight)和另外三個並未被說明為較為重要之屬性(BP_HIGH、BP_LOW 和 PULSE)，分別和 PCA 劑量分類結果做相關性分析，結果如表 28，得知年齡、性別和體重的確是有較高之相關性。在此研究中，我們考量了過去文獻中未被考慮的病患使用 PCA 裝置之相關屬性，而被決策樹挑選到的屬性也大多為此屬性，但是過去文獻中提到的性別、年齡和體重和其他的生理資訊相比，依舊為較為重要之屬性。

4.2.3.3 PCA 裝置調整設定調整預測

而 PCA 裝置參數設定調整預測分析，以 bagging + under sampling 再搭配資料清理的結果，在 positive F-score 和幾何平均(geometric mean)的表現上是最佳的。而利用 bagging + under sampling，在 10-fold 交叉驗證中，一樣會產生 2000 棵決策樹，我們統計 bagging + under sampling 搭配資料清理，在這 2000 棵決策樹中，所有 280 個屬性，以根部(root)為第一層，統計第一到第五層中，被決策樹挑選為節點(node)的頻率前十高的屬性，表 30 為統計結果，數字表示該屬性在該層中的出現次數，例如：p_timediff_var_3hr 在這 2000 棵決策樹中，被挑選為根部的頻率為 36.3%，為出現次數最高之屬性，而出現頻率第二高的屬性則是 BP_HIGH，出現頻率為 12.7%。

和上述兩種預測分析相同，以表 30 為基礎，計算各個屬性之分數並加以排名，十個分數最高之屬性如表 31，其中分數最高的屬性為 p_timediff_var_3hr。因此，我們對 p_timediff_var 作和之前相同的單因子變異數分析(One-way ANOVA)，分析 24 個時間點的 p_timediff_var 和是否調整 PCA 裝置參數之相關性，表 32 為分析之結果。其中，被決策樹中被挑選到的 p_timediff_var 屬性有 p_timediff_var_3hr 和 p_timediff_var_8hr 兩個時間點，其” $-\log_{10}(\text{p-value})$ ” 之平均值為 0.9。再利用蒙地卡羅法，從 24 個時間點中隨機挑選出兩個時間點，計算其” $-\log_{10}(\text{p-value})$ ” 之平均，結果為 0.4。證實被決策樹挑選為較為關鍵之兩個時間點確實有較高之重要性。

同樣地，我們計算 p_timediff_var_1hr~p_timediff_var_24hr 的” $-\log_{10}(\text{p-value})$ ” 之平均，發現超過該平均值之時間點共有 7 個，分別是 1hr、3hr、6hr~8hr、12hr、17hr 和 22hr 的 p_timediff_var 屬性，而 3hr 和 8hr 則是從決策樹中挑選的關鍵之屬性，從單因子變異數分析中挑選到的重要之屬性有 28.6% 的比例也為從決策樹中挑選到的關鍵之屬性，而從決策樹中挑選到的屬性則 100% 均為單因子變異數中挑選到之屬性，以此計算出的 F-score 為 44.4%。

而 pcamode_set 屬性在表 31 中，也有兩個時間點被挑選到，分別是 pcamode_set_24hr 和 pcamode_set_14hr，因此，我們一樣將 pcamode_set 屬性 24 個時間點和是否調整 PCA

裝置參數作單因子變異數分析(One-way ANOVA)，觀察其相關性，結果如表 33。其中，被決策樹中被挑選到的pcamode_set屬性其” $-\log_{10}(\text{p-value})$ ” 之平均值為 2.7。而利用蒙地卡羅法，從 24 個時間點中隨機挑選出兩個時間點，計算其” $-\log_{10}(\text{p-value})$ ” 之平均，結果為 3.1。我們推測是因為大部分的時間點，其p-value小於 0.05，也就是大部分的時間點都和是否調整參數有高度的相關性。

那計算pcamode_set_1hr~pcamode_set_24hr的” $-\log_{10}(\text{p-value})$ ” 之平均，發現超過該平均值之時間點共有 12 個，分別是 1hr、11hr和 24hr的pcamode_set屬性，而 14hr和 24hr則是從決策樹中挑選的關鍵之屬性，從單因子變異數分析中挑選到的重要之屬性有 8.3%的比例也為從決策樹中挑選到的關鍵之屬性，而從決策樹中挑選到的屬性則 50%均為單因子變異數中挑選到之屬性，以此計算出的F-score為 14.3%。

另外，表 31 中挑選到較為關鍵的生理資訊相關屬性為 BP_HIGH 和 PULSE，我們一樣利用單因子變異數分析，比較這兩個屬性和其它未被挑選或是重要性相對較低之屬性(weight 和 BP_LOW)，表 34 為其分析結果，可以看出 BP_HIGH 和 PULSE 均有較低之 p-value，且其 p-value 小於 0.01，而 weight 和 BP_LOW 之 p-value 均大於 0.1，以此說明被決策樹挑選出的生理資訊相關屬性具有較高之相關性。

其中 BP_HIGH(收縮壓)的部分，我們推測是因為患有高血壓(Hypertension)或是血壓較高之病患在使用麻醉藥時，風險比一般未患有高血壓的病患來的高，在 Braunwald's Heart Disease 中的第 85 章(Anesthesia and Noncardiac Surgery in Patients with Heart Disease)中說明，已有臨床上的證據證實高血壓病患在術後會有瀕臨或急性的器官損傷，會造成心肌梗塞或腦血管意外(腦中風)的風險[41]。1979 年 Prys-Roberts 提出，高血壓的病患和一般病患相較之下，在手術前需要提高評估之標準，而在手術中和手術後比起一般病患也需要更仔細的監督，以避免其手術引發之併發症[42]。1978 年 Steen 等人證實，高血壓是造成病患在手術後心肌梗塞的危險因素之一[43]。因此在術後恢復期間，我們覺得患有高血壓之病患會比一般未患高血壓之病患較容易有不適之狀況，因此在使用 PCA 裝置時，會比一般病患更有可能需要做 PCA 參數之調整，因此 BP_HIGH(收縮壓)才會是 PCA 裝置參數設定調整預測分析中較為重要之屬性。

表 20. 總劑量預測分析，前五層中出現頻率為前 10 的屬性名稱及該屬性在該層之出現頻率

| 出現頻率(%) | Level 1 | | Level 2 | | Level 3 | | Level 4 | | Level 5 | |
|--------------|----------------|------|----------------------|------|---------------------|------|------------------|------|---------------------|------|
| Top1 | contidose_24hr | 74.8 | p_timediff_var_17hr | 24.3 | pcamode_set_24hr | 16.6 | pcamode_set_24hr | 16.2 | pcadose_3hr | 21.1 |
| Top2 | contidose_23hr | 18.4 | pcadose_21hr | 16.3 | pcadose_9hr | 15.9 | pcadose_3hr | 15.0 | pcadose_9hr | 16.2 |
| Top3 | contidose_22hr | 6.6 | pcadose_9hr | 11.9 | contidose_24hr | 12.9 | pcadose_9hr | 13.1 | contidose_23hr | 10.4 |
| Top4 | contidose_20hr | 0.1 | pcamode_set_24hr | 9.5 | pcadose_3hr | 12.0 | contidose_24hr | 11.6 | pcadose_19hr | 10.0 |
| Top5 | | | pcadose_17hr | 8.5 | pcadose_21hr | 9.7 | contidose_23hr | 11.4 | contidose_24hr | 8.6 |
| Top6 | | | p_timediff_mean_19hr | 8.3 | contidose_23hr | 9.3 | pcadose_2hr | 7.6 | pcamode_set_24hr | 8.1 |
| Top7 | | | pcadose_19hr | 6.1 | contidose_22hr | 7.1 | pcadose_21hr | 7.0 | p_timediff_mean_5hr | 6.7 |
| Top8 | | | pcadose_3hr | 5.8 | pcadose_19hr | 6.2 | pcadose_19hr | 6.6 | pcadose_7hr | 6.7 |
| Top9 | | | pcadose_14hr | 4.8 | pcadose_2hr | 5.6 | contidose_22hr | 6.3 | pcadose_2hr | 6.3 |
| Top10 | | | contidose_24hr | 4.5 | p_timediff_var_17hr | 4.6 | pcadose_7hr | 5.3 | pcadose_21hr | 5.9 |

表 21. 總劑量預測分析，十個最為關鍵之屬性與分數

| | 屬性名稱 | 分數 | $-\log_{10}(\text{p-value})$ |
|--------------|---------------------|-------|---|
| Top1 | contidose_24hr | 462.5 | 156.2 |
| Top2 | contidose_23hr | 153.1 | 153.5 |
| Top3 | pcadose_9hr | 137.7 | 11.1 |
| Top4 | pcamode_set_24hr | 128.3 | ∞ (It's extremely significant, so p-value is almost zero.) |
| Top5 | pcadose_21hr | 114.2 | 9.3 |
| Top6 | p_timediff_var_17hr | 111 | 3.9 |
| Top7 | pcadose_3hr | 110.3 | 15.7 |
| Top8 | contidose_22hr | 66.9 | 147.6 |
| Top9 | pcadose_19hr | 66.2 | 6.3 |
| Top10 | pcadose_2hr | 38.3 | 14.4 |

表 22. 總劑量預測分析，contidose 屬性和總劑量分類結果之單因子變異數分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|----------------|-------------|------------------------------|
| contidose_1hr | 9.75611E-71 | 70.0 |
| contidose_2hr | 1.72612E-63 | 62.8 |
| contidose_3hr | 6.34474E-62 | 61.2 |
| contidose_4hr | 4.1778E-110 | 109.4 |
| contidose_5hr | 1.69196E-69 | 68.8 |
| contidose_6hr | 9.1398E-119 | 118.0 |
| contidose_7hr | 1.1112E-115 | 115.0 |
| contidose_8hr | 3.27798E-69 | 68.5 |
| contidose_9hr | 3.8047E-118 | 117.4 |
| contidose_10hr | 1.7134E-68 | 67.8 |
| contidose_11hr | 2.2228E-121 | 120.7 |
| contidose_12hr | 5.6013E-125 | 124.3 |
| contidose_13hr | 5.6464E-121 | 120.2 |
| contidose_14hr | 9.702E-125 | 124.0 |
| contidose_15hr | 2.0139E-126 | 125.7 |
| contidose_16hr | 4.2869E-123 | 122.4 |
| contidose_17hr | 1.3425E-125 | 124.9 |
| contidose_18hr | 2.298E-129 | 128.6 |
| contidose_19hr | 8.0482E-133 | 132.1 |
| contidose_20hr | 9.4393E-147 | 146.0 |
| contidose_21hr | 1.4871E-135 | 134.8 |
| contidose_22hr | 2.7203E-148 | 147.6 |
| contidose_23hr | 3.3139E-154 | 153.5 |
| contidose_24hr | 6.6656E-157 | 156.2 |

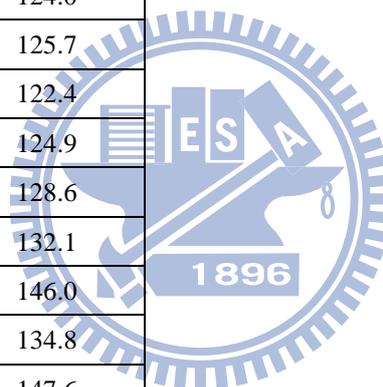


表 23. 總劑量預測分析，pcadose 屬性和總劑量分類結果之單因子變異數分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|--------------|-------------|------------------------------|
| pcadose_1hr | 1.59324E-09 | 8.8 |
| pcadose_2hr | 3.81689E-15 | 14.4 |
| pcadose_3hr | 1.92548E-16 | 15.7 |
| pcadose_4hr | 1.28439E-06 | 5.9 |
| pcadose_5hr | 2.8256E-10 | 9.5 |
| pcadose_6hr | 1.7311E-13 | 12.8 |
| pcadose_7hr | 1.86484E-10 | 9.7 |
| pcadose_8hr | 2.93792E-09 | 8.5 |
| pcadose_9hr | 8.54355E-12 | 11.1 |
| pcadose_10hr | 2.58171E-10 | 9.6 |
| pcadose_11hr | 3.85629E-14 | 13.4 |
| pcadose_12hr | 8.8052E-10 | 9.1 |
| pcadose_13hr | 1.23557E-11 | 10.9 |
| pcadose_14hr | 1.30171E-14 | 13.9 |
| pcadose_15hr | 1.43281E-10 | 9.8 |
| pcadose_16hr | 5.46394E-11 | 10.3 |
| pcadose_17hr | 9.34115E-16 | 15.0 |
| pcadose_18hr | 5.563E-12 | 11.3 |
| pcadose_19hr | 4.55083E-07 | 6.3 |
| pcadose_20hr | 1.26678E-13 | 12.9 |
| pcadose_21hr | 5.47524E-10 | 9.3 |
| pcadose_22hr | 1.1497E-19 | 18.9 |
| pcadose_23hr | 2.80102E-09 | 8.6 |
| pcadose_24hr | 5.54573E-08 | 7.3 |

表 24. 總劑量預測分析，生理資訊相關屬性與分類結果之相關性分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|---------|-------------|------------------------------|
| age | 2.80551E-18 | 17.6 |
| sex | 5.43466E-10 | 9.3 |
| weight | 6.5477E-22 | 21.2 |
| BP_HIGH | 0.768876541 | 0.1 |
| BP_LOW | 0.193921887 | 0.7 |
| PULSE | 0.093969073 | 1.0 |

表 25. PCA 劑量預測分析，前五層中出現頻率為前 10 的屬性名稱及該屬性在該層之出現頻率

| 頻率(%) | Level 1 | | Level 2 | | Level 3 | | Level 4 | | Level 5 | |
|--------------|----------------------|------|----------------------|------|----------------------|------|----------------------|------|----------------------|------|
| Top1 | pcadose_9hr | 24.4 | pcadose_9hr | 25.7 | pcadose_9hr | 18.6 | pcadose_6hr | 15.7 | pcadose_24hr | 15.4 |
| Top2 | pcadose_19hr | 17.3 | pcadose_6hr | 14.1 | p_timediff_mean_22hr | 16.6 | p_timediff_mean_9hr | 15.3 | failure_p_15hr | 13.6 |
| Top3 | pcadose_14hr | 12.7 | p_timediff_mean_22hr | 12.0 | pcadose_6hr | 16.2 | p_timediff_mean_17hr | 11.7 | p_timediff_mean_9hr | 13.4 |
| Top4 | p_timediff_var_19hr | 11.2 | pcadose_19hr | 8.8 | p_timediff_mean_9hr | 11.6 | p_timediff_mean_22hr | 11.4 | p_timediff_mean_17hr | 11.5 |
| Top5 | pcadose_11hr | 10.0 | p_timediff_mean_14hr | 8.1 | p_timediff_mean_17hr | 8.4 | failure_p_15hr | 10.6 | pcadose_6hr | 8.7 |
| Top6 | p_timediff_mean_14hr | 9.2 | p_timediff_mean_9hr | 7.9 | p_timediff_mean_14hr | 6.9 | pcadose_9hr | 9.5 | pcadose_23hr | 8.2 |
| Top7 | pcadose_6hr | 6.7 | p_timediff_mean_17hr | 6.5 | pcadose_14hr | 6.5 | pcadose_24hr | 7.2 | pcadose_9hr | 7.4 |
| Top8 | pcadose_12hr | 3.4 | pcadose_14hr | 6.5 | pcadose_13hr | 5.2 | p_timediff_mean_14hr | 7.0 | pcadose_7hr | 7.3 |
| Top9 | p_timediff_var_11hr | 2.7 | pcadose_11hr | 5.7 | failure_p_15hr | 5.2 | pcadose_23hr | 5.9 | pcadose_19hr | 7.3 |
| Top10 | pcadose_22hr | 2.4 | p_timediff_var_15hr | 4.6 | pcadose_24hr | 4.9 | pcadose_13hr | 5.7 | pcadose_13hr | 7.1 |

表 26. PCA 劑量預測分析，十個最為關鍵之屬性與分數

| | 屬性名稱 | 分數 | $-\log_{10}(\text{p-value})$ |
|--------------|----------------------|-------|------------------------------|
| Top1 | pcadose_9hr | 307 | 40.1 |
| Top2 | pcadose_6hr | 178.6 | 46.6 |
| Top3 | pcadose_19hr | 129 | 41.6 |
| Top4 | p_timediff_mean_22hr | 120.6 | 10.4 |
| Top5 | p_timediff_mean_14hr | 113.1 | 16.9 |
| Top6 | p_timediff_mean_9hr | 110.4 | 18.5 |
| Top7 | pcadose_14hr | 109 | 47.0 |
| Top8 | p_timediff_mean_17hr | 86.1 | 11.3 |
| Top9 | pcadose_11hr | 72.8 | 49.0 |
| Top10 | p_timediff_var_19hr | 56 | 5.7 |

表 27. PCA 劑量預測分析，pcadose 屬性和 PCA 劑量分類結果之單因子變異數分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|--------------|-------------|------------------------------|
| pcadose_1hr | 3.86183E-30 | 29.4 |
| pcadose_2hr | 4.00393E-43 | 42.4 |
| pcadose_3hr | 1.50566E-42 | 41.8 |
| pcadose_4hr | 4.18381E-41 | 40.4 |
| pcadose_5hr | 2.02602E-51 | 50.7 |
| pcadose_6hr | 2.33807E-47 | 46.6 |
| pcadose_7hr | 1.17008E-29 | 28.9 |
| pcadose_8hr | 6.39116E-31 | 30.2 |
| pcadose_9hr | 8.53315E-41 | 40.1 |
| pcadose_10hr | 1.16715E-30 | 29.9 |
| pcadose_11hr | 8.95398E-50 | 49.0 |
| pcadose_12hr | 8.2681E-43 | 42.1 |
| pcadose_13hr | 2.4256E-33 | 32.6 |
| pcadose_14hr | 9.91982E-48 | 47.0 |
| pcadose_15hr | 1.05048E-34 | 34.0 |
| pcadose_16hr | 1.31569E-38 | 37.9 |
| pcadose_17hr | 2.4323E-48 | 47.6 |
| pcadose_18hr | 5.89723E-38 | 37.2 |
| pcadose_19hr | 2.25211E-42 | 41.6 |
| pcadose_20hr | 4.21549E-43 | 42.4 |
| pcadose_21hr | 2.34963E-24 | 23.6 |
| pcadose_22hr | 2.54557E-44 | 43.6 |
| pcadose_23hr | 6.23482E-29 | 28.2 |
| pcadose_24hr | 1.70192E-35 | 34.8 |



表 28. PCA 劑量預測分析，p_timediff_mean 屬性和 PCA 劑量分類結果之單因子變異數分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|----------------------|-------------|------------------------------|
| p_timediff_mean_1hr | 0.004173426 | 2.4 |
| p_timediff_mean_2hr | 2.45764E-05 | 4.6 |
| p_timediff_mean_3hr | 1.30791E-09 | 8.9 |
| p_timediff_mean_4hr | 2.03545E-12 | 11.7 |
| p_timediff_mean_5hr | 1.46532E-16 | 15.8 |
| p_timediff_mean_6hr | 1.77981E-16 | 15.7 |
| p_timediff_mean_7hr | 3.1689E-08 | 7.5 |
| p_timediff_mean_8hr | 1.11156E-08 | 8.0 |
| p_timediff_mean_9hr | 2.89482E-19 | 18.5 |
| p_timediff_mean_10hr | 1.71224E-05 | 4.8 |
| p_timediff_mean_11hr | 2.69839E-17 | 16.6 |
| p_timediff_mean_12hr | 1.11185E-14 | 14.0 |
| p_timediff_mean_13hr | 8.11241E-11 | 10.1 |
| p_timediff_mean_14hr | 1.2487E-17 | 16.9 |
| p_timediff_mean_15hr | 3.56527E-15 | 14.4 |
| p_timediff_mean_16hr | 2.57382E-12 | 11.6 |
| p_timediff_mean_17hr | 4.82334E-12 | 11.3 |
| p_timediff_mean_18hr | 3.77859E-11 | 10.4 |
| p_timediff_mean_19hr | 7.5296E-11 | 10.1 |
| p_timediff_mean_20hr | 1.51029E-11 | 10.8 |
| p_timediff_mean_21hr | 3.03677E-05 | 4.5 |
| p_timediff_mean_22hr | 3.83554E-11 | 10.4 |
| p_timediff_mean_23hr | 6.05972E-05 | 4.2 |
| p_timediff_mean_24hr | 4.30352E-10 | 9.4 |

表 29. PCA 劑量預測分析，生理資訊相關屬性與分類結果之相關性分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|---------|-------------|------------------------------|
| age | 0.030054587 | 1.5 |
| sex | 0.05039 | 1.3 |
| weight | 0.000340414 | 3.5 |
| BP_HIGH | 0.374544425 | 0.4 |
| BP_LOW | 0.956868575 | 0.0 |
| PULSE | 0.982779231 | 0.0 |

表 30. PCA 裝置調整設定預測分析，資料已刪除雜訊，前五層中出現頻率為前 10 的屬性名稱及該屬性在該層之出現頻率

| 頻率(%) | Level 1 | | Level 2 | | Level 3 | | Level 4 | | Level 5 | |
|--------------|----------------------|------|----------------------|------|----------------------|------|----------------------|------|---------------------|------|
| Top1 | p_timediff_var_3hr | 36.3 | BP_HIGH | 21.1 | BP_HIGH | 19.6 | BP_HIGH | 21.4 | BP_HIGH | 24.2 |
| Top2 | BP_HIGH | 12.7 | p_timediff_var_3hr | 18.4 | PULSE | 12.5 | PULSE | 17.1 | PULSE | 12.6 |
| Top3 | PULSE | 8.4 | PULSE | 13.1 | p_timediff_var_3hr | 9.6 | WEIGHT | 9.3 | ansTimeDiff | 10.5 |
| Top4 | contidose_24hr | 7.6 | p_timediff_mean_17hr | 8.5 | p_timediff_mean_17hr | 9.4 | ansTimeDiff | 9.1 | failure_p_21hr | 9.6 |
| Top5 | p_timediff_var_8hr | 7.2 | contidose_24hr | 8.0 | contidose_24hr | 9.2 | p_timediff_var_3hr | 8.5 | p_timediff_var_1hr | 8.4 |
| Top6 | pcamode_set_24hr | 7.2 | p_timediff_var_8hr | 6.9 | p_timediff_var_8hr | 9.1 | p_timediff_mean_17hr | 8.2 | WEIGHT | 8.0 |
| Top7 | pcamode_set_20hr | 6.1 | failure_p_11hr | 6.2 | WEIGHT | 8.7 | p_timediff_var_1hr | 7.4 | p_timediff_mean_8hr | 7.5 |
| Top8 | pcamode_set_14hr | 6.0 | pcamode_set_24hr | 6.2 | ansTimeDiff | 8.1 | pcadose_17hr | 6.3 | p_timediff_var_3hr | 6.6 |
| Top9 | p_timediff_mean_17hr | 4.5 | age | 5.9 | p_timediff_mean_24hr | 7.2 | failure_p_21hr | 6.3 | contidose_24hr | 6.4 |
| Top10 | age | 4.1 | pcamode_set_14hr | 5.8 | failure_p_15hr | 6.6 | pcamode_set_24hr | 6.3 | failure_p_1hr | 6.2 |

表 31. PCA 裝置調整設定預測分析，十個最為關鍵之屬性與分數

| | 屬性名稱 | 分數 | $-\log_{10}(p\text{-value})$ |
|--------------|----------------------|-------|------------------------------|
| Top1 | p_timediff_var_3hr | 307.5 | 1.3 |
| Top2 | BP_HIGH | 273.7 | 2.8 |
| Top3 | PULSE | 178.7 | 2.1 |
| Top4 | contidose_24hr | 104 | 12.0 |
| Top5 | p_timediff_mean_17hr | 101.1 | 0.5 |
| Top6 | p_timediff_var_8hr | 90.9 | 0.5 |
| Top7 | pcamode_set_24hr | 73.4 | 3.2 |
| Top8 | pcamode_set_14hr | 53.2 | 2.1 |
| Top9 | ansTimeDiff | 53 | 1.2 |
| Top10 | WEIGHT | 52.7 | 1.0 |

表 32. PCA 裝置調整設定預測分析，p_timediff_var 屬性和是否調整 PCA 裝置參數之單因子變異數分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|---------------------|-------------|------------------------------|
| p_timediff_var_1hr | 0.29361717 | 0.5 |
| p_timediff_var_2hr | 0.910682212 | 0.0 |
| p_timediff_var_3hr | 0.050896126 | 1.3 |
| p_timediff_var_4hr | 0.761665778 | 0.1 |
| p_timediff_var_5hr | 0.578374664 | 0.2 |
| p_timediff_var_6hr | 0.020603257 | 1.7 |
| p_timediff_var_7hr | 0.348946269 | 0.5 |
| p_timediff_var_8hr | 0.306984252 | 0.5 |
| p_timediff_var_9hr | 0.979320895 | 0.0 |
| p_timediff_var_10hr | 0.51432813 | 0.3 |
| p_timediff_var_11hr | 0.574531304 | 0.2 |
| p_timediff_var_12hr | 0.718330118 | 0.1 |
| p_timediff_var_13hr | 0.691513961 | 0.2 |
| p_timediff_var_14hr | 0.866282344 | 0.1 |
| p_timediff_var_15hr | 0.590917587 | 0.2 |
| p_timediff_var_16hr | 0.759547896 | 0.1 |
| p_timediff_var_17hr | 0.127544126 | 0.9 |
| p_timediff_var_18hr | 0.942378033 | 0.0 |
| p_timediff_var_19hr | 0.525110253 | 0.3 |
| p_timediff_var_20hr | 0.968413686 | 0.0 |
| p_timediff_var_21hr | 0.789912599 | 0.1 |
| p_timediff_var_22hr | 0.009572045 | 2.0 |
| p_timediff_var_23hr | 0.91271761 | 0.0 |
| p_timediff_var_24hr | 0.739990876 | 0.1 |

表 33. PCA 裝置調整設定預測分析，pcamode_set 屬性和是否調整 PCA 裝置參數之單因子變異數分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|------------------|-------------|------------------------------|
| pcamode_set_1hr | 0.000146789 | 3.8 |
| pcamode_set_2hr | 0.000214518 | 3.7 |
| pcamode_set_3hr | 0.000214518 | 3.7 |
| pcamode_set_4hr | 0.00021 | 3.7 |
| pcamode_set_5hr | 0.00008 | 4.1 |
| pcamode_set_6hr | 0.00008 | 4.1 |
| pcamode_set_7hr | 0.00008 | 4.1 |
| pcamode_set_8hr | 0.00017 | 3.8 |
| pcamode_set_9hr | 0.00024 | 3.6 |
| pcamode_set_10hr | 0.00024 | 3.6 |
| pcamode_set_11hr | 0.00066 | 3.2 |
| pcamode_set_12hr | 0.0009 | 3.0 |
| pcamode_set_13hr | 0.0009 | 3.0 |
| pcamode_set_14hr | 0.007254763 | 2.1 |
| pcamode_set_15hr | 0.00164 | 2.8 |
| pcamode_set_16hr | 0.004456233 | 2.4 |
| pcamode_set_17hr | 0.0135 | 1.9 |
| pcamode_set_18hr | 0.00271 | 2.6 |
| pcamode_set_19hr | 0.00654 | 2.2 |
| pcamode_set_20hr | 0.00207 | 2.7 |
| pcamode_set_21hr | 0.00604 | 2.2 |
| pcamode_set_22hr | 0.01272 | 1.9 |
| pcamode_set_23hr | 0.014242791 | 1.8 |
| pcamode_set_24hr | 0.000577082 | 3.2 |

表 34. PCA 裝置調整設定預測分析，生理資訊相關屬性與分類結果之相關性分析

| 屬性名稱 | p-value | $-\log_{10}(\text{p-value})$ |
|---------|----------|------------------------------|
| BP_HIGH | 0.001527 | 2.8 |
| PULSE | 0.00854 | 2.1 |
| weight | 0.10552 | 1.0 |
| BP_LOW | 0.574053 | 0.2 |

4.2.4 Lesion study

在 4.2.3 屬性分析中，提到在過去的文獻中，都是以病患的生理資訊，例如，年齡、體重等，來做預測分析。而在我們的研究中，我們除了使用病患的生理資訊外，也利用

病患使用 PCA 裝置的相關屬性用以預測分析，而從屬性分析中，病患使用 PCA 裝置的相關屬性確實有較高之相關性。因此，我們推測加入病患使用 PCA 裝置的相關屬性對我們的預測工作是有幫助的。為了進一步證實該想法，我們將所有屬性分為兩大部分，第一部分為病患的生理資訊以及手術相關屬性等，即過去文獻中所使用的屬性；第二部分則為病患使用 PCA 裝置的相關屬性，包括單位小時病患按壓 PCA 裝置的次數以及病患使用 PCA 裝置的單位小時參數設定等，即在我們的研究中，新加入用來預測分析的屬性。我們分別利用這兩大部份的屬性來進行總劑量預測、PCA 劑量預測和 PCA 裝置參數設定調整預測三種預測分析，並且會和 4.2.2 利用完整屬性來作預測分析之結果加以比較。

4.2.4.1 總劑量預測

表 35 為總劑量預測分析，僅使用病患之生理資訊及手術相關屬性來作預測的結果，而表 36 則是以病患使用 PCA 裝置之相關屬性來作預測。從表 35 和表 36 比較之下可以發現，在整體準確度(Overall Accuracy)的表現上，利用病患使用 PCA 裝置之相關屬性來作預測明顯比只利用病患之生理資訊及手術相關屬性的預測結果好的相當多。我們再將表 35 和表 36 分別與表 15 也就是使用全部的屬性的預測結果相互比較，可以看出，僅使用病患之生理資訊以及手術相關屬性的預測結果明顯差於使用全部的屬性之預測結果。而僅利用病患使用 PCA 裝置之相關屬性之預測結果，和使用全部的屬性之結果相比，兩者之間大多並無顯著差異。由此則可說明，病患使用 PCA 裝置之相關屬性，在該預測分析中，的確是扮演比較重要的角色，而病患之生理資訊以及手術相關屬性，相較之下就略顯不重要。以此可以說明，在我們的研究中，加入病患使用 PCA 裝置的相關屬性，對預測總劑量來說，確實是有幫助的。

表 35. 總劑量預測分析之結果，屬性使用病患之生理資訊及手術相關屬性

| 總劑量預測(%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost |
|-------------------------|-------------------|-------------------|-----------------------|--------------------|-----------------------------|-------------------|
| “low dose” Recall | 42.1 [¯] | 39.2 [¯] | 45.8 [¯] | 35.0 | 43.0 [¯] | 43.1 [¯] |
| “medium dose” Recall | 60.5 [¯] | 73.8 [¯] | 60.6 [¯] | 80.0 | 65.4 [¯] | 60.7 [¯] |
| “high dose” Recall | 16.7 [¯] | 10.5 [¯] | 19.2 [¯] | 4.4 | 14.2 [¯] | 16.9 [¯] |
| “low dose” Precision | 46.5 [¯] | 51.6 [¯] | 48.1 [¯] | 54.7 | 50.2 [¯] | 46.2 [¯] |
| “medium dose” Precision | 53.1 | 53.7* | 53.8 | 53.2 | 53.4 | 52.9 |
| “high dose” Precision | 22.8 [¯] | 39.9* | 30.7* | 34.6 | 26.2* | 27.3 [¯] |
| Overall Accuracy | 47.9 [¯] | 52.7* | 49.6 [¯] | 53.4 | 50.3 [¯] | 48.4 [¯] |

註：與 Random Forest 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

表 36. 總劑量預測分析之結果，屬性使用病患使用 PCA 裝置之相關屬性

| 總劑量預測(%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost |
|-------------------------|-------------------|--------------|-----------------------|--------------------|-----------------------------|-------------------|
| “low dose” Recall | 77.1 [¯] | 83.8 | 82.6* | 81.3 [¯] | 79.3 [¯] | 79.1 [¯] |
| “medium dose” Recall | 73.2 [¯] | 84.0 | 80.4 [¯] | 83.2 | 76.9 [¯] | 76.1 [¯] |
| “high dose” Recall | 60.2 [¯] | 63.8 | 67.1 [¯] | 48.9 [¯] | 60.8* | 61.8* |
| “low dose” Precision | 77.0 [¯] | 84.8 | 83.1 [¯] | 81.5 [¯] | 78.3 [¯] | 78.7 [¯] |
| “medium dose” Precision | 73.2 [¯] | 79.8 | 79.2* | 75.6 [¯] | 75.6 [¯] | 75.4 [¯] |
| “high dose” Precision | 62.5 [¯] | 78.5 | 71.9 [¯] | 80.2* | 69.8 [¯] | 67.4 [¯] |
| Overall Accuracy | 72.9 [¯] | 81.2 | 79.4 [¯] | 77.9 [¯] | 75.6 [¯] | 75.2 [¯] |

註：與 Bagging 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

4.2.4.2 PCA 劑量預測

PCA 劑量預測分析中，以病患之生理資訊及手術相關屬性來作預測分析的結果如表

37，而表 38 則是以病患使用 PCA 裝置之相關屬性來作預測的結果。表 37 和表 38 相互比較之下，一樣可以看出，利用病患使用 PCA 裝置之相關屬性來作預測的結果，整體準確度(Overall Accuracy)明顯優於僅使用病患之生理資訊及手術相關屬性的結果。同樣地，我們也將表 37 和表 38 分別與使用全部的屬性來作預測的結果表 16 相互比較，和總劑量預測分析類似，僅使用病患之生理資訊以及手術相關屬性的預測結果明顯差於使用全部的屬性之預測結果。而以病患使用 PCA 裝置之相關屬性之預測結果，和使用全部的屬性之預測結果相比，兩者之間大多並無顯著差異。由此得知，在 PCA 劑量預測分析中，病患之生理資訊以及手術相關屬性，和病患使用 PCA 裝置之相關屬性相比，使用 PCA 裝置之相關屬性較為重要的多。也可說明，我們加入病患使用 PCA 裝置之相關屬性櫃該預測工作是有幫助的。

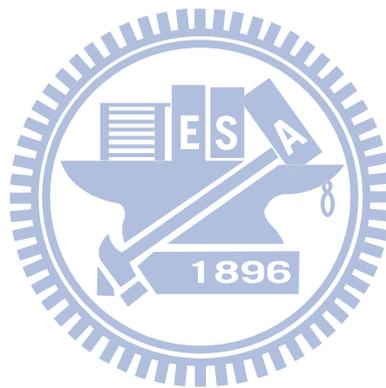


表 37. PCA 劑量預測分析之結果，屬性使用病患之生理資訊及手術相關屬性

| PCA 劑量預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost |
|-------------------------|-------------------|-------------------|-----------------------|--------------------|-----------------------------|-------------------|
| “low dose” Recall | 60.2 [¯] | 78.7 [¯] | 63.4 [¯] | 86.8 | 69.3 [¯] | 62.1 [¯] |
| “medium dose” Recall | 31.2 [¯] | 20.7 [¯] | 28.6 [¯] | 13.1 | 24.2 [¯] | 31.4 [¯] |
| “high dose” Recall | 10.8 [¯] | 1.2 [¯] | 8.6 [¯] | 0.0 | 7.2 [¯] | 8.1 [¯] |
| “low dose” Precision | 53.6 | 53.2 | 53.0 | 53.2 | 53.3 | 53.9* |
| “medium dose” Precision | 34.7* | 34.5* | 33.7 | 32.3 | 33.3* | 34.8* |
| “high dose” Precision | 14.2 [¯] | 7.5* | 13.5 [¯] | 0.0 | 14.3 [¯] | 12.6 [¯] |
| Overall Accuracy | 43.8 [¯] | 48.7 [¯] | 44.3 [¯] | 50.3 | 45.8 [¯] | 44.5 [¯] |

註：與 Random Forest 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

表 38. PCA 劑量預測分析之結果，屬性使用病患使用 PCA 裝置之相關屬性

| PCA 劑量預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost |
|-------------------------|-------------------|--------------|-----------------------|--------------------|-----------------------------|-------------------|
| “low dose” Recall | 77.7 [¯] | 84.0 | 81.1 [¯] | 95.4 [¯] | 85.4* | 80.1 [¯] |
| “medium dose” Recall | 51.2 [¯] | 65.7 | 64.5* | 47.5 [¯] | 56.1 [¯] | 54.3 [¯] |
| “high dose” Recall | 45.0 | 47.4 | 51.0 [¯] | 34.1 [¯] | 36.4 [¯] | 46.3 |
| “low dose” Precision | 75.9 [¯] | 81.9 | 82.3* | 73.6 [¯] | 76.2 [¯] | 77.5 [¯] |
| “medium dose” Precision | 51.2 [¯] | 60.3 | 58.6 [¯] | 61.9* | 56.7 [¯] | 54.3 [¯] |
| “high dose” Precision | 50.6 [¯] | 74.2 | 67.0 [¯] | 86.0 [¯] | 71.6 | 54.4 [¯] |
| Overall Accuracy | 64.3 [¯] | 72.9 | 71.5 [¯] | 71.0 [¯] | 68.9 [¯] | 66.8 [¯] |

註：與 Bagging 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

4.2.4.3 PCA 裝置參數設定調整預測

表 39 為 PCA 裝置參數設定調整預測分析，僅使用病患之生理資訊及手術相關屬性來作預測的結果，而表 40 則是以病患使用 PCA 裝置之相關屬性來作預測，上述均為未作資料清理(data cleaning)的結果，表 41 和表 42 則是以使用 k-nearest-neighbor 演算法進行資料清理後的結果。

在過去的文獻中，大多只有探討麻醉藥劑量的部分，在 PCA 裝置的參數調整與否的討論較少，而我們在 4.2.3 的屬性分析中，覺得加入病患使用 PCA 裝置之相關屬性，有其幫助的預測也為總劑量預測和 PCA 劑量預測的部分。那在 PCA 裝置參數設定調整預測分析中，在 4.2.3 中，從其中挑選出較為關鍵的屬性，也並非均是病患使用 PCA 裝置之相關屬性，其中也包含了 BP_HIGH 和 PULSE 等病患之生理資訊。那從表 39 和表 40 以及表 41 和表 42 之間的比較下可以發現，在該預測分析中，以病患使用 PCA 裝置之相關屬性來作預測的結果，在 Positive F-score 和幾何平均(Geometric Mean)的表現上，並未優於使用病患之生理資訊以及手術相關屬性的預測結果。

那如果比較表 39 和表 41 以及表 40 和表 42，可以發現，在利用 k-nearest-neighbor 演算法對資料進行資料清理(data cleaning)以刪除雜訊(noise)後，Positive F-score 和幾何平均(Geometric Mean)大多數都有顯著的提升。因此，在該預測分析中，我們覺得使用資料清理(data cleaning)，對預測工作是最有幫助的，在預測的準確度能有所改善。

表 39. PCA 裝置調整設定預測分析之結果，資料尚未刪除雜訊，屬性使用病患之生理資訊及手術相關屬性

| PCA 裝置調整設定預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Bagging Over Sampling | C4.5 Bagging Under Sampling | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost | C4.5 AdaBoost Over Sampling | C4.5 AdaBoost Under Sampling | C4.5 DataBoost-IM |
|--------------------|-------------------|-------------------|-----------------------|----------------------------|-----------------------------|--------------------|-----------------------------|-------------------|-----------------------------|------------------------------|-------------------|
| Positive Recall | 13.6 [¯] | 2.3 [¯] | 19.0 [¯] | 25.7 [¯] | 41.7 | 0.7 [¯] | 24.0 [¯] | 13.2 [¯] | 36.3 [¯] | 47.7* | 18.5 [¯] |
| Positive Precision | 22.1* | 22.6 | 23.3 | 29.6 [¯] | 24.9 | 12.0 [¯] | 24.1 | 24.2 | 25.0 | 22.3* | 22.8 [¯] |
| Positive F-score | 16.4 [¯] | 4.0 [¯] | 20.4 [¯] | 27.1 [¯] | 31.0 | 1.3 [¯] | 23.7 [¯] | 16.8 [¯] | 29.4* | 30.3 | 20.0 [¯] |
| Negative Recall | 88.8 [¯] | 98.4 [¯] | 85.1 [¯] | 85.5 [¯] | 70.4 | 99.8 [¯] | 81.9 [¯] | 90.1 [¯] | 74.4 [¯] | 61.1 [¯] | 85.3 [¯] |
| Negative Precision | 81.5 [¯] | 81.2 [¯] | 81.8 [¯] | 83.1* | 83.8 | 81.1 [¯] | 82.2 [¯] | 81.6 [¯] | 83.4* | 83.4 | 81.8 [¯] |
| Negative F-score | 85.0 [¯] | 88.9 [¯] | 83.4 [¯] | 84.2 [¯] | 76.5 | 89.5 [¯] | 82.0 [¯] | 85.6 [¯] | 78.6 [¯] | 70.3 [¯] | 83.5 [¯] |
| Geometric Mean | 34.6 [¯] | 14.5 [¯] | 40.2 [¯] | 46.8 [¯] | 54.2 | 7.7 [¯] | 44.3 [¯] | 34.4 [¯] | 51.9* | 53.9 | 39.7 [¯] |
| Overall Accuracy | 74.6 [¯] | 80.2 [¯] | 72.6 [¯] | 74.1 [¯] | 65.0 | 81.0 [¯] | 71.0 [¯] | 75.5 [¯] | 67.2 [¯] | 58.6 [¯] | 72.7 [¯] |

註：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

表 40. PCA 裝置調整設定預測分析之結果，資料尚未刪除雜訊，屬性使用病患使用 PCA 裝置之相關屬性

| PCA 裝置調整設定預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Bagging Over Sampling | C4.5 Bagging Under Sampling | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost | C4.5 AdaBoost Over Sampling | C4.5 AdaBoost Under Sampling | C4.5 DataBoost-IM |
|--------------------|-------------------|-------------------|-----------------------|----------------------------|-----------------------------|--------------------|-----------------------------|-------------------|-----------------------------|------------------------------|-------------------|
| Positive Recall | 26.1 [¯] | 5.2 [¯] | 10.6 [¯] | 16.7 [¯] | 39.4 | 2.8 [¯] | 23.3 [¯] | 20.3 [¯] | 31.5 [¯] | 43.9* | 20.0 [¯] |
| Positive Precision | 25.1 | 39.7 [¯] | 29.1* | 26.1* | 24.5 | 36.3 [¯] | 23.4 | 25.7 | 22.8* | 21.5 [¯] | 23.8 |
| Positive F-score | 25.3 [¯] | 8.9 [¯] | 14.9 [¯] | 20.1 [¯] | 30.1 | 5.2 [¯] | 23.0 [¯] | 22.3 [¯] | 26.3 [¯] | 28.7* | 21.4 [¯] |
| Negative Recall | 82.1 [¯] | 98.4 [¯] | 93.8 [¯] | 89.2 [¯] | 71.8 | 99.4 [¯] | 82.1 [¯] | 86.8 [¯] | 74.9 [¯] | 62.4 [¯] | 85.1 [¯] |
| Negative Precision | 82.7 [¯] | 81.6 [¯] | 81.8 [¯] | 82.1 [¯] | 83.6 | 81.4 [¯] | 82.1 [¯] | 82.4 [¯] | 82.4 [¯] | 82.7* | 82.0 [¯] |
| Negative F-score | 82.3 [¯] | 89.2 [¯] | 87.4 [¯] | 85.5 [¯] | 77.1 | 89.5 [¯] | 82.1 [¯] | 84.5 [¯] | 78.4 [¯] | 71.0 [¯] | 83.5 [¯] |
| Geometric Mean | 46.2 [¯] | 22.4 [¯] | 31.5 [¯] | 38.6 [¯] | 53.2 | 16.8 [¯] | 43.7 [¯] | 41.9 [¯] | 48.6 [¯] | 52.3 | 41.1 [¯] |
| Overall Accuracy | 71.5 [¯] | 80.7 [¯] | 78.1 [¯] | 75.5 [¯] | 65.6 | 81.1 [¯] | 71.0 [¯] | 74.2 [¯] | 66.7* | 58.9 [¯] | 72.7 [¯] |

註：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，[¯]表示 p-value < 0.001。

表 41. PCA 裝置調整設定預測分析之結果，資料已刪除雜訊，屬性使用病患之生理資訊及手術相關屬性

| PCA 裝置調整設定預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Bagging Over Sampling | C4.5 Bagging Under Sampling | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost | C4.5 AdaBoost Over Sampling | C4.5 AdaBoost Under Sampling | C4.5 DataBoost-IM |
|--------------------|---------------------|---------------------|-----------------------|----------------------------|-----------------------------|---------------------|-----------------------------|---------------------|-----------------------------|------------------------------|---------------------|
| Positive Recall | 48.3 ^{†**} | 48.1 ^{†**} | 51.5 ^{†**} | 58.3 ^{**} | 60.5 ^{**} | 42.9 ^{†**} | 48.0 ^{†**} | 47.9 ^{†**} | 55.8 ^{**} | 57.2 ^{*#} | 53.4 ^{†**} |
| Positive Precision | 22.3 [†] | 26.1 [*] | 25.1 [#] | 24.5 ^{**} | 24.6 | 26.5 ^{†**} | 25.9 ^{†#} | 23.0 [†] | 22.2 ^{†#} | 22.6 [†] | 22.8 [†] |
| Positive F-score | 30.4 ^{†**} | 33.6 ^{**} | 33.6 ^{**} | 34.4 [*] | 34.9 [*] | 32.5 ^{†**} | 33.4 ^{†**} | 30.9 ^{†**} | 31.7 ^{†#} | 32.3 ^{†#} | 31.8 ^{†**} |
| Negative Recall | 60.6 ^{†**} | 68.0 ^{†**} | 63.9 ^{†**} | 58.0 ^{**} | 56.6 [*] | 72.1 ^{†**} | 67.9 ^{†**} | 62.3 ^{†**} | 54.4 ^{†**} | 53.7 ^{†**} | 57.6 [*] |
| Negative Precision | 83.4 ^{†**} | 84.9 ^{†**} | 84.9 ^{†**} | 85.7 [*] | 86.0 [*] | 84.5 ^{†**} | 84.9 ^{†**} | 83.7 ^{†**} | 84.1 [†] | 84.3 [†] | 84.2 ^{†**} |
| Negative F-score | 70.1 ^{**} | 75.4 ^{†**} | 72.8 ^{†**} | 69.0 ^{**} | 68.1 [*] | 77.7 ^{†**} | 75.3 ^{†**} | 71.3 ^{†**} | 65.9 ^{†**} | 65.4 ^{†**} | 68.2 [*] |
| Geometric Mean | 54.1 ^{†**} | 57.2 ^{**} | 57.3 ^{**} | 58.1 [*] | 58.5 [*] | 55.6 ^{†**} | 57.0 ^{†**} | 54.6 ^{†**} | 55.1 ^{†#} | 55.4 [†] | 55.5 ^{†**} |
| Overall Accuracy | 58.3 ^{**} | 64.2 ^{†**} | 61.5 ^{†**} | 58.0 ^{**} | 57.3 [#] | 66.6 ^{†**} | 64.1 ^{†**} | 59.6 ^{†**} | 54.6 ^{†**} | 54.4 ^{†**} | 56.8 [*] |

註 1：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，[†]表示 p-value < 0.001。

註 2：雜訊清理前後之比較：# 表示 p-value < 0.05，※表示 p-value < 0.001。

表 42. PCA 裝置調整設定預測分析之結果，資料已刪除雜訊，屬性使用病患使用 PCA 裝置之相關屬性

| PCA 裝置調整設定預測 (%) | C4.5 | C4.5 Bagging | C4.5 Bagging Stacking | C4.5 Bagging Over Sampling | C4.5 Bagging Under Sampling | C4.5 Random Forest | C4.5 Random Forest Stacking | C4.5 AdaBoost | C4.5 AdaBoost Over Sampling | C4.5 AdaBoost Under Sampling | C4.5 DataBoost-IM |
|--------------------|--------------------|---------------------|-----------------------|----------------------------|-----------------------------|---------------------|-----------------------------|---------------------|-----------------------------|------------------------------|---------------------|
| Positive Recall | 44.9 ^{**} | 40.7 ^{⎯**} | 38.6 ^{⎯**} | 52.6 ^{**} | 53.5 ^{**} | 36.5 ^{⎯**} | 41.2 ^{⎯**} | 43.4 ^{⎯**} | 53.4 ^{**} | 54.8 ^{**} | 47.4 ^{**} |
| Positive Precision | 22.0 [#] | 25.5 ^{**} | 25.7 [#] | 23.9 [#] | 23.9 | 28.0 ^{⎯#} | 26.5 ^{⎯#} | 22.5 [#] | 22.8 [*] | 22.0 [*] | 22.0 [#] |
| Positive F-score | 29.4 [#] | 31.2 ^{**} | 29.4 ^{**} | 32.7 ^{**} | 32.8 ^{**} | 30.4 ^{**} | 31.9 ^{**} | 29.5 ^{**} | 31.9 ^{**} | 31.3 [#] | 29.9 ^{⎯**} |
| Negative Recall | 62.7 ^{**} | 72.2 ^{⎯**} | 73.6 ^{⎯**} | 60.9 ^{**} | 60.2 ^{**} | 78.0 ^{⎯**} | 73.4 ^{⎯**} | 64.9 ^{**} | 57.6 ^{**} | 54.6 ^{**} | 60.6 ^{**} |
| Negative Precision | 83.0 [*] | 83.9 ^{⎯**} | 83.8 ^{⎯**} | 84.7 ^{**} | 84.8 ^{**} | 84.1 ^{**} | 84.3 ^{**} | 83.1 ^{⎯#} | 84.1 [#] | 83.8 [#] | 83.2 ^{⎯#} |
| Negative F-score | 71.3 ^{**} | 77.6 ^{⎯**} | 78.0 ^{⎯**} | 70.7 ^{**} | 70.2 ^{**} | 80.7 ^{⎯**} | 78.3 ^{⎯**} | 72.7 ^{**} | 68.3 ^{**} | 65.9 ^{⎯**} | 70.0 ^{**} |
| Geometric Mean | 53.0 ^{**} | 54.2 ^{**} | 51.8 ^{**} | 56.6 ^{**} | 56.5 ^{**} | 52.5 ^{**} | 54.7 ^{**} | 53.1 ^{**} | 55.4 ^{**} | 54.6 [#] | 53.6 ^{⎯**} |
| Overall Accuracy | 59.3 ^{**} | 66.3 ^{⎯**} | 66.9 ^{⎯**} | 59.3 ^{**} | 58.9 ^{**} | 70.1 ^{⎯**} | 67.3 ^{⎯**} | 60.8 ^{**} | 56.8 ^{**} | 54.6 ^{⎯**} | 58.1 ^{**} |

註 1：與 Bagging Under Sampling 之比較：* 表示 p-value < 0.05，⎯表示 p-value < 0.001。

註 2：雜訊清理前後之比較：# 表示 p-value < 0.05，**表示 p-value < 0.001。

五、結論

從過去的研究中可以得知[44-45]，病人疼痛自控止痛法(Patient-Controlled Analgesia ,PCA)，在手術後疼痛的控制上，與傳統肌肉注射止痛法(Intramuscular Injection ,IM)相比，控制術後疼痛的效果較佳且所需時間較短。使用 PCA 裝置需先由麻醉醫師，依據病患的身體狀況，設定適合該病患的麻醉藥劑量和使用該裝置的安全範圍，其中安全範圍包括「鎖定時間(lockout)」和「四小時限制(4 hour limit)」。

本篇論文將資料探勘技術應用於實務經驗，用以預測分析病患使用 PCA 裝置的情形，其中包括預測病患在使用 PCA 裝置一段時間後，在短期未來之內所需之總麻醉藥劑量和病患經由按壓 PCA 裝置所注射之麻醉藥劑量，以及預測病患在之後的 PCA 使用上，是否需要醫師調整 PCA 裝置的設定。在 PCA 裝置參數設定調整預測分析中，因為大多數的病患使用 PCA 裝置時，在初始設定後大多無需重新設定參數，因此，在此預測分析中會遭遇類別不平衡(class imbalance)之問題，傳統的學習演算法在學習類別不平衡的資料時，通常會傾向預測多數類別(majority class)而忽略了少數類別(minority class)，造成「未做調整」類別的預測準確度很高，「有做調整」類別的資料卻大多被預測錯誤，而對於類別不平衡的資料而言，大多數會較為關注少數類別，因此，「有做調整」類別的預測準確度為我們希望改善之重點。此外，我們透過分群的方式，對「有做調整」類別的資料作分群，以各群質心為圓心，該群中距質心最遠之資料與質心距離為半徑，發現 99.875% 「未做調整」之資料都在此圓邊界內，由此得知，我們的資料除了有類別不平衡的問題以外，兩種類別的資料相當稀疏且重疊，其中有相當多的雜訊。這些問題，在資料探勘領域中，是很具備挑戰性且困難的。

在總劑量預測、PCA 劑量預測和 PCA 裝置參數設定調整預測三種預測分析中，我們嘗試了單一的學習演算法，包括決策樹(C4.5)、簡單貝氏分類器(Naïve Bayes Classifier)、類神經網路(Artificial Neural Network ,ANN)、支援向量機(Support Vector Machine ,SVM)和決策表(Decision Table)五種學習演算法，再以決策樹為基礎演算法，發展 bagging、bagging + stacking、random forest、random forest + stacking 和 AdaBoost 五

種整體學習演算法(ensemble learning algorithm)，而在 PCA 裝置參數設定調整預測中，為解決類別不平衡的問題，搭配減少多數類別取樣法(under sampling)和增加少數類別取樣法(over sampling)，因此在此預測分析中，還使用了 bagging + under sampling、bagging + over sampling、AdaBoost + under sampling、AdaBoost + over sampling 和 DataBoost-IM 五種方法，共十種整體學習演算法。此外，我們也使用 k-nearest-neighbor 演算法對資料進行資料清理(data cleaning)，藉此刪除資料中的雜訊(noise)。

總劑量預測分析的實驗結果，bagging 為表現最佳之演算法，整體準確度為 80.9%，而所有的演算法中，高劑量(high dose)的 recall 和 precision 的結果都差於一般劑量(medium dose)和低劑量(low dose)的表現，我們推測是因為高劑量的資料數量較另外兩種類別的資料數量來的少。而屬性分析的部分，我們發現 contidose(病人經由點滴方式被注射的麻醉藥劑量)為較為重要之屬性，推測是因為有 78.1%的病患，麻醉藥的來源比例，經由點滴方式被注射的麻醉藥劑量高於經由 PCA 裝置被注射的劑量。

PCA 劑量預測分析的部分，表現最佳的演算法為 bagging，有 73.1%的整體準確度，並且經由 paired t-test 檢定，C4.5 搭配 bagging 均優於其他之演算法，且均有顯著差異。我們也從屬性分析中得知，在該預測分析中，被決策樹挑選為較為重要之屬性，均與病患按壓 PCA 裝置相關，例如，pcadose、p_timediff_var 等。而其中以 pcadose_9hr、pcadose_6hr 和 pcadose_19hr 為出現頻率較高之屬性，我們利用單因子變異數分析(One-way ANOVA)，分析上述三個屬性以及鄰近時間之屬性與 PCA 劑量分類結果之相關性，發現這幾個時間點之屬性與其他出現頻率非前十高之屬性相比，確實與 PCA 劑量分類結果有較高之相關性。

在 PCA 裝置調整預測分析中，以 bagging + under sampling 搭配資料清理的表現最佳，positive F-score 為 35.2%。而且大多數的學習演算法和整體學習演算法，在資料清理後，positive F-score 和幾何平均(geometric mean)都顯著提升，顯示資料清理對大多數的演算而言，在預測工作上都是有幫助的。我們以 bagging + under sampling 搭配資料清理的結果和單一的學習演算法中，表現較好的 ANN + over sampling 和 Naïve Bayes + over sampling 的表現作比較，可以看出 bagging + under sampling 搭配資料清理的結果，

不僅可提升「有做調整」類別的準確度，在「未做調整」類別的準確度的部分，和 ANN + over sampling 比較之下，也並未降低太多。屬性分析的部分，我們找到 BP_HIGH 和 PULSE 兩種較為關鍵之屬性。

在我們的研究中，我們使用決策樹(C4.5)作為主要的資料探勘工具，而實驗結果也證實，決策樹可以被應用於此研究上，在搭配整體學習演算法後，準確度的表現上有不錯的表現外，決策樹也提供了較佳的可讀性。而在面對類別不平衡以及資料相當稀疏的問題，我們利用 k-nearest-neighbor 演算法對資料進行資料清理，幾乎對所有的演算來說，該方法均確實顯著提升了預測準確度，對我們的預測工作上是有幫助的。

我們利用決策樹的特性，挑選出在各個預測分析中，較為關鍵之屬性，並利用單因子變異數分析(One-way ANOVA)，證實這些屬性確實和各個預測分析的分類結果有較高之相關性。和過去的文獻不同的地方在於，我們除了考量病患生理資訊的相關屬性外，還加入病患使用 PCA 裝置的相關資訊作為預測工作之屬性，而在劑量的預測分析中，經由決策樹挑選出的較為關鍵之屬性也大多為病患使用 PCA 裝置之相關資訊，然而文獻中提出的年齡、性別和體重，和其他生理資訊屬性相比，確實還是具有較高之相關性。而在最後的 lesion study 中，我們將屬性分為兩大部分，一部分為病患之生理資訊以及手術相關屬性，也就是在過去的文獻中所使用之屬性，另一部份則是病患使用 PCA 裝置之相關屬性，即在我們的研究中加入用來預測的屬性，發現僅使用病患之生理資訊以及手術相關屬性的預測結果大多比以病患使用 PCA 裝置之相關屬性來作預測的結果來得差，得知在我們的研究中，加入這些病患使用 PCA 裝置之相關屬性，對總劑量預測和 PCA 裝置調整預測確實是有幫助的。

並且在過去的研究中，較少人去探討病患在使用 PCA 裝置期間，其參數是否需要被調整，而我們在此問題中，找出 BP_HIGH 為較為關鍵之屬性，推測是因為患有高血壓或是血壓較高者的病患，在使用 PCA 裝置時，會較一般未患有高血壓之病患有較高之危險性，因此，也更容易會有不適的情況而需要調整參數。

參考文獻

1. B. Walder, M. Schafer, H. Henzi, "Efficacy and safety of patient-controlled opioid analgesia for acute postoperative pain," *Acta Anaesthesiol Scand*, pp. 795-804, 2001.
2. S.J. Dolin, J.N. Cashman, J.M. Bland, "Effectiveness of acute postoperative pain management: evidence from published data," *Br J Anaesth*, pp. 409-423, 2002.
3. <http://www.surgeryencyclopedia.com/La-Pa/Patient-Controlled-Analgesia.html>
(Patient-controlled analgesia)
4. J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1, pp.281-297, 1967.
5. J.R. Quinlan, *C4.5: Programs for Empirical Learning*, Morgan Kaufmann, San Francisco, CA. USA, 1993.
6. I.J. Good, "The Estimation of Probabilities; an Essay on Modern Bayesian Methods," M.I.T. Press, 1965.
7. D. Paterson, "Artificial Neural Networks," Prentice Hall, Singapore, 1996.
8. R.E. Fan, P.H. Chen, C.J. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines," *Journal of Machine Learning Research*, 6, pp. 1889-1918, 2005.
9. R. Kohavi, "The power of decision tables," in *European Conference on Machine Learning*, 1995.
10. U.M. Fayyad, N. Weir, S. Djorgovski, "SKICAT: A machine learning system for automated cataloging of large scale sky survey," in *Proc. 10th International Conference on Machine Learning*, pp. 112-119, 1993.
11. W. Lehnert, S. Soderland, D. Aronow, F. Feng, A. Shmueli, "Inductive Text Classification

- for Medical Applications,” J Experimental and Theoretical Artificial Intelligence 7(1), pp. 271-302, 1995.
12. E. Bauer, R. Kohavi, “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants,” Machine Learning, 36, pp. 105-139, 1999.
 13. L. Breiman, “Bagging predictors,” Machine Learning Journal, 24, pp.123-140, 1996.
 14. R. Schapire, “The strength of weak learnability,” Machine Learning, 5 (2), pp. 197-227, 1990.
 15. L. Breiman, “Random forests,” Machine Learning Journal, 45, pp.5-32, 2001.
 16. L. Breiman, “Looking Inside the Black Box,” Wald Lecture II, Department of Statistics, California University, 2002.
 17. Y. Freund, R. Schapire, “Experiments with a new boosting algorithm,” In Proc. 13th International Conference on Machine Learning, pp. 148-156, 1996.
 18. D. Wolpert, “Stacked generalization,” Neural Networks, 5, pp. 241-259, 1992.
 19. N. Japkowics, S. Stephen, “The class imbalance problem: A systematic study,” Intelligent Data Analysis, 6(5), pp. 203-231, 2002.
 20. C. Phua, D. Alahakoon, V. Lee, “Minority Report in Fraud Detection: Classification of Skewed Data,” SIGKDD Explorations, 6(1), pp. 50-59, 2004.
 21. M. Kubat, R.C. Holte, S. Matwin,” Machine learning for the detection of oil spills in satellite radar images,” Machine Learning, 30, pp. 195–215, 1998.
 22. M. Kubat, S. Matwin. “Addressing the curse of imbalanced training sets : One-sided selection,” In Proc. 14th International Conference on Machine Learning, pp. 179-186, 1997.
 23. C. Ling, C. Li, “Data Mining for Direct Marketing Problems and Solutions,” In Proc. of 4th International Conference on Knowledge Discovery and Data Mining (KDD-98), 1998.
 24. N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, “SMOTE: Synthetic Minority

- Over-sampling Technique,” *J. Artificial Intelligence Research*, 16, pp. 321-357, 2002.
25. C. Cardie, N. Howe, “Improving minority class prediction using case-specific feature weights,” in *Proc. 14th International Conference on Machine Learning*, pp. 57-65, 1997.
 26. M.V. Joshi, R.C. Agarwal, V. Kumar, “Predicting rare classes: can boosting make any weak learner strong?” In *Proc. 8th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 297-306, 2002.
 27. H. Guo, H.L. Viktor, “Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach,” *Sigkdd Explorations*, 6(1), pp. 30–39, 2004.
 28. E.M. Knorr, R.T. Ng, V. Tucakov, “Distance-based outliers: algorithms and applications,” *The VLDB Journal*, 8, pp. 237-253, 2000.
 29. F. Angiulli, C. Pizzuti, “Fast outlier detection in high dimensional spaces,” In *Proc. 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 15-26, 2002.
 30. G. Batista, R.C. Prati, M.C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *SIGKDD Explorations*, 6(1), pp. 20-29, 2004
 31. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: An update,” *SIGKDD Explorations*, 11(1):10–18, 2009.
 32. F. Provost, T. Fawcett, “Robust classification for imprecise environments,” *Machine Learning*, 42, pp. 203-231, 2001.
 33. M. Kubat, R. Horte, S. Matwin, “Learning when negative examples abound,” In *Proc. 9th European Conference on Machine Learning*, pp. 146-153, 1997.
 34. D.D. Lewis, W. A. Gale, “A sequential algorithm for training text classifiers,” In *Proc. 17th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, pp. 3–12, 1994.
 35. P. E. Macintyre, D. A. Jarvis, “Age is the best predictor of postoperative morphine requirements,” *Pain*, 64, pp. 357–364, 1995.
 36. H. Y. Ip, A. Abrishami, P. W. Peng, J. Wong, F. Chung, “Predictors of postoperative pain

- and analgesic consumption : A qualitative systematic review,” *Anesthesiology*, 111, pp. 657 – 77, 2009.
37. A. Abrishami, J. Chan, F. Chung, J. Wong, ”Preoperative Pain Sensitivity and Correlation with Analgesic Consumption,” *Anesthesiology*, 114, pp.445-457, 2011.
38. Y. Y. Chia, L. H. Chow, C. C. Hung, K. Liu, L. P. Ger, P. N. Wang,” Gender and pain upon movement are associated with the requirements for postoperative patientcontrolled iv analgesia: a prospective survey of 2,298 Chinese patients,” *ANESTH*, 49(3), pp.249-255, 2002.
39. B. Ng, J. E. Dimsdale, J.D. Rollnik and H. Shapiro, “The effect of ethnicity on prescriptions for patient-controlled analgesia for post-operative pain,” *Pain*,66, pp. 9–12, 1996.
40. P. H. Pan, R. Coghill, T. T. Houle, M. H. Seid, W. M. Lindel and R. L. Parker et al., “Multifactorial preoperative predictors for postcesarean section pain and analgesic requirement,” *Anesthesiology*,104, pp. 417–425, 2006.
41. L. A. Fleisher, J. Beckman, ”Anesthesia and Noncardiac Surgery in Patients with Heart Disease,” In: R. O. Bonow, D. L. Mann, D. P. Zipes, P. Libby, Libby P, “Braunwald’s Heart Disease. 9th ed,” Philadelphia: Elsevier, pp. 1811-1827, 2012.
42. C. Prys-Roberts, “Hypertension and anesthesia—fifty years on,” *Anesthesiology*,50,pp. 281-284, 1979.
43. P.A. Steen, J.H. Tinker, S. Tarhan, “Myocardial reinfarction after anesthesia and surgery,” *JAMA*, 239, pp. 2566–2570, 1978.
44. V. Thomas, M. Heath, D. Rose, P. Flory, ”Psychological characteristics and the effectiveness of patient-controlled analgesia,” *British Journal of Anaesthesia*, 74, pp. 271-276, 1995.
45. A.M. Chang, W.Y. Ip, T.H. Cheung, “Patient-controlled analgesia versus conventional intramuscular injection: a cost effectiveness analysis,” *J Adv Nurs*, 46(5), pp. 531–541,

2004.

