Hw6 心得

林家同

這次的競賽中，主要利用 Scikit-learn 的機器學習工具包，我們試過一些主流演算法像是 KNN、ExtraTreesClassifier、SVM 等等，只要有機器學習的基本架構和 python 基礎能力，coding 的部分並不難完成。而這次作業有兩個比較困難的地方，第一是如何選擇適當地模型，這取決於資料的類型和我們要解決的問題，很多演算法因為是第一次接觸，所以在使用時還不是非常了解背後的數學式，以至於會不太有根據地調整參數。第二個我覺得是除了 row data 外，我們的 domain knowledge 還不夠，如果對題目指定的特定領域有更深入了解，我相信能更容易找出資料裡的特徵。總結來說，雖然我們這組的競賽結果不是特別突出，但在過程中還是嘗試了不少提升模型準確度的方式，也詢問過很多同學的意見，相信這次經驗可以成為我往後在資料科學精進的養分。

李培倫

這次的競賽將課堂上介紹到的各種模型實做, 我自己也有去 Scikit-learn 官方看了一些資料，了解每個參數在模型中代表的意義, 為了取得最佳的模型，我手刻一些模組化的副程式(在資料夾 Control_version 中)，如資料預處理的部分整理成 DataPrepocess.py，這樣在使用各個演算法可以直接 import 取得 data(DP.func())，又如 Algorithm_controlPanel，我將幾個常用的演算法一起進行測試，並取得其機器學習的評估指標，這樣的模組化的好處可以幫我們在分析問題時從大至小，更有組織地進行模型訓練以及參數微調。另外，因為測試過大多演算法還是無法提升成績(我們的排名長時間處於 15 左右)，所以我們最後找到了一個類似猜數字的方法可以提升 public leaderboard，但這方法似乎在 private leaderboard 不可行。

莊上緣

先談使用眾多演算法後排名卻無法顯著提升的原因，我覺得很大一部分是不知道要選取那些特徵當作 train 的資料，也不太確定如何去預處理資料才是最好的，這部分我覺得可以藉由更多實作經驗去提升對資料的敏感度。除了機器學習的演算法，我們中間嘗試過類似 randomforest 交叉比對投票的方式，對各個演算法最佳化的結果進行投票來做出最終的 csv，這樣的作法讓我們一度拿到第 15 中間左右的名次。在最後幾天，我們決定放手一搏，想到可以利用區間猜數字(把 2000 筆資料切割成很多份，去預測各區段中 1 和 0 的機率分佈)的方法，進而提升模型的評估指標，再把這些額外過濾的資料塞回 train data 來訓練模型，但很可惜的是最後沒有成功(因此造成 public leaderboard precision=1, 但 private 排名倒數的慘案)。