一、HW4

* Method description

使用Kaggle上的資料及做為訓練，目標是預測出機器在未來一天/兩天內會發生錯誤的機率是多少。而這次使用到的工作有libsvm和matlab。

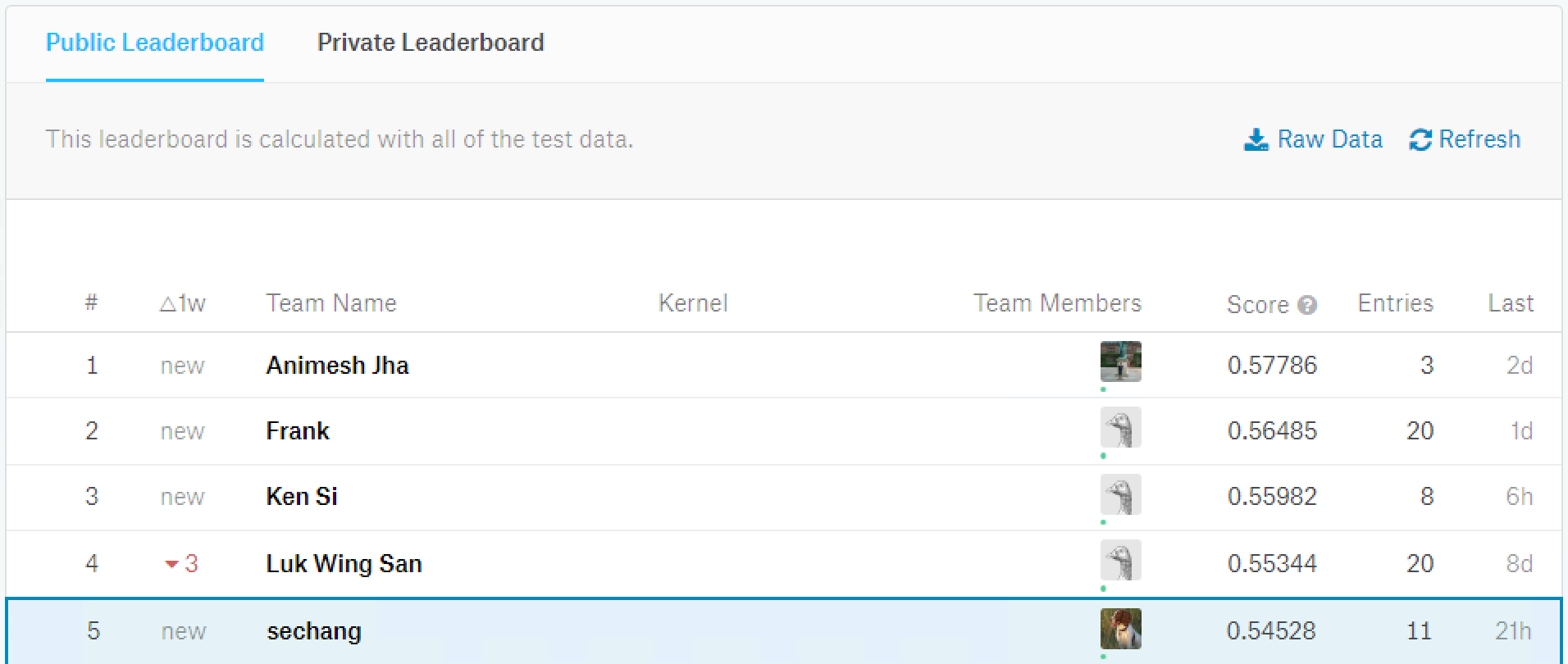
libsvm是一個可以用svm來做機器學習的工具，裡面已經寫好了許多不同語言中能使用的工具(如：matlab、python、java…等)。而這次使用裡面已經編好的matlab來寫。之所以挑選matlab的原因是在資料處理上會簡單蠻多的，不像python還要重頭開始讀資料、編碼等。

那用matlab的作法是先把下載下來的資料，所有不相關的header都拿掉，每行都只保留features。另外，因為有一些column的label是空白的，我們無法知道究竟其是true或是false，為了保持資料正確性，那筆資料就乾脆不使用了，所以還需要把這些資料挑出來刪除。最後把label放到另一個檔案中，並把false設為0、true設為1。而test data也是用一樣的方法完成處理。最後即可使用csvread和svmtrain來完成作業。而參數上的設定是-b 1 -t 3 -m 10000。我發現kernel不要用預設的kernel，改成使用sigmoid function後分數會上升滿多的。而這兩組資料，我自己實驗發現都是用OW1這組資料train出來有最好的效果。

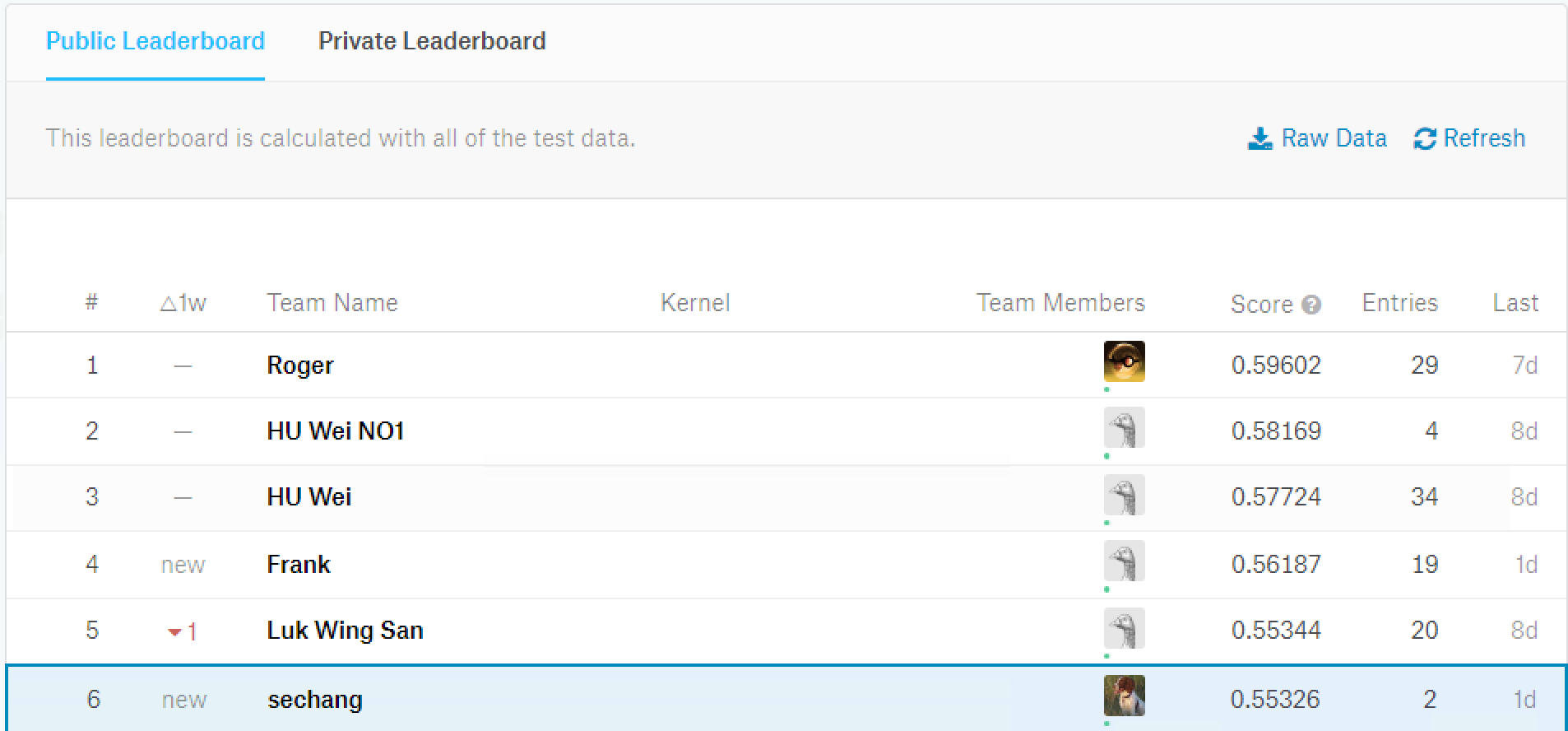
比較python來說，還需要有嚴格的格式要求，會比較麻煩一點，因為這次的deadline比較短一點，就選擇比較快完成的方式來做。

* Experimental results

Full 1： Score 0.54528



Full 2： Score 0.55326



* Discussion of difficulty or problem encountered

其實在剛開始時也沒想到可以用matlab寫，是後來聽了學姊的寫法後，發現這種方法不用花太多時間在資料處理上真的會方便很多。其實機器學習一直以來都有很多不統的工具、不同的操作方法，有的時候真的不太需要拘泥在某種方法上，多方嘗試會有不同的體驗。

另外剛開始寫的時候，其實搞不懂這麼多檔案各是在幹嘛的，上傳kaggle的結果他也跟我說格式不符，後來才發現原來他只要上傳OW16的預測結果就好。光是在搞懂題目其實就花了蠻多的時間了。

最後一切都順利後，就是要開始想辦法改善分數了，原本使用預設的參數，使用不同的training set都沒辦法有效提高分數，後來稍微改了一下不同的kernel，不只訓練速度快很多，就連分數也衝了蠻多的。但是這次並沒有使用到libsvm提供的調整參數工具，因為用實驗室的電腦沒權限安裝gnuplot，下次或許可以試看看在自己的電腦上做看看，或許如此一來可以試出更好的參數！