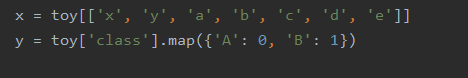
**Toy train**

**B083022053**

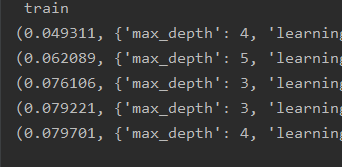
**黃啟桓**

**Garbage Train**

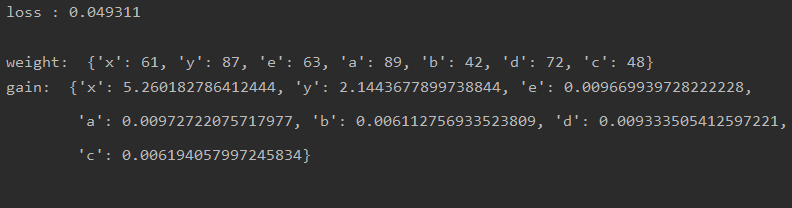
****

使用 有用的參數x, y ，以及無用的參數a, b, c, d, e來訓練。

以超參數組中，多次(50次)隨機選取超參數組合來訓練。並以validation loss 由小到大排序，並輸出validation loss最小的5個



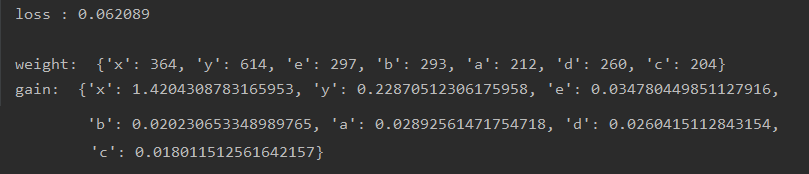
觀察第一組超參數模型下各變數的Weight、Gain



Gain中，x, y變數的平均增益遠大於a, b, c, d, e變數，可以推斷a, b, c, d, e變數為Garbage data。

============================================

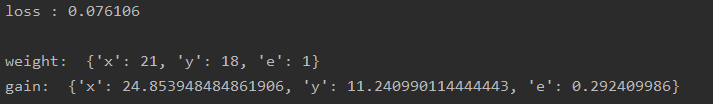
觀察第二組超參數模型下各變數的Weight、Gain



觀察Gain中，x, y變數的平均增益略大於a, b, c, d, e變數。

============================================

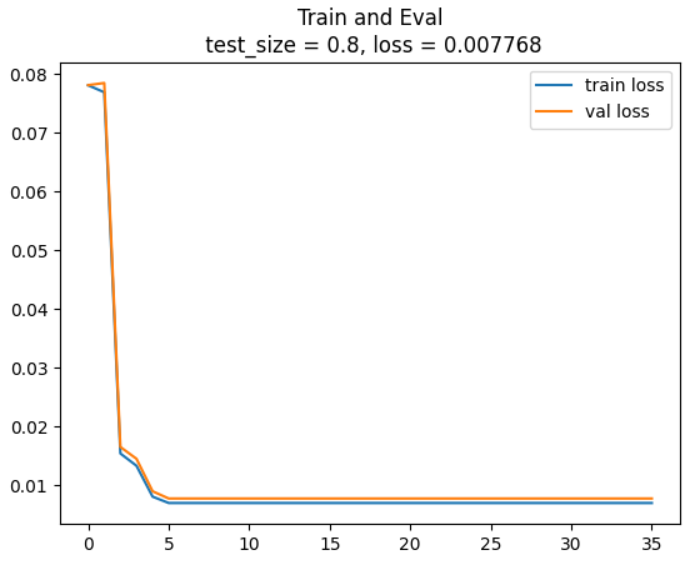
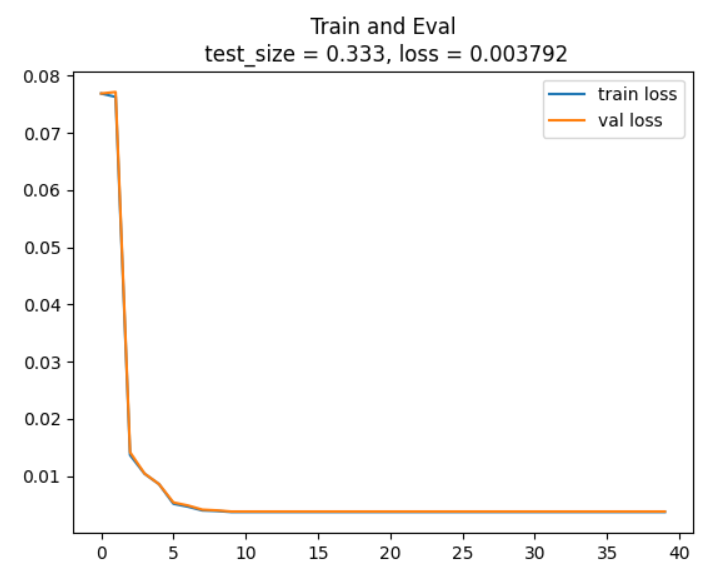
觀察第三組超參數模型下各變數的Weight、Gain



觀察Gain中，x, y變數的平均增益遠大於 e變數且a, b, c, d變數沒有被使用到。

============================================

將train 和 test 分割比例調整



============================================

結論 : 分析前幾個較小validation loss超參數組合下的Gain，可以得出a, b, c, d, e變數確實是Garbage Data而非一次偶然。

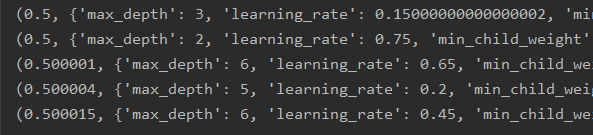
**Clean Train**

由上述結論下，為了得到好的訓練結果，在訓練模型時不應放置Garbage Data。

所以理所當然資料處理會是如下圖

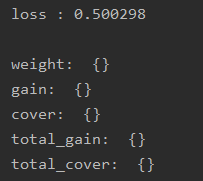
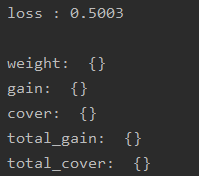
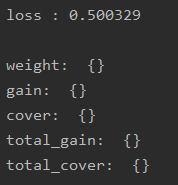


同樣隨機選取超參數多次(50次)，並使validation loss以小到大排序 並輸出前5個。



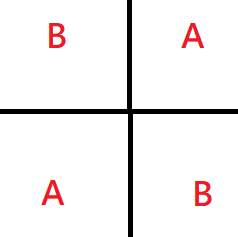
可以發現前5個最小validation loss 大概都是0.5 (50%)

觀察其weight及gain

所有的weight、gain都是空的，這其實是xgboost算法的問題:「xgboost使用貪心算法，即2個特徵變數的情況下，xgboost僅使用1個特徵變數作分割」。

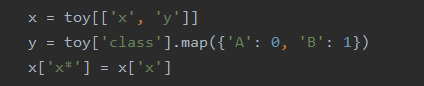
以坐標軸方式思考，假設各象限資料量均勻，且切割線通過資料質心(0, 0)，不管如何切割，其準確率都是50%。(可以推廣至and運算、or運算、甚至是多變數下。但我無法完全概括所有會導致此現象的情況)



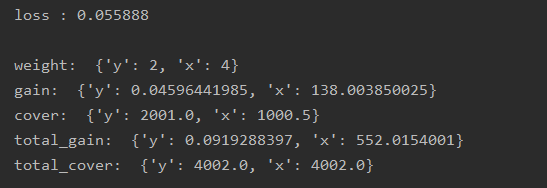
解決方法: 多一個沒有鑑別度的變數，例如:複製一個變數或新建一個所有資料皆為相同數值的變數。

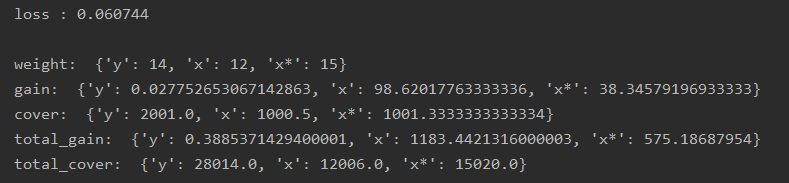
===========================================================

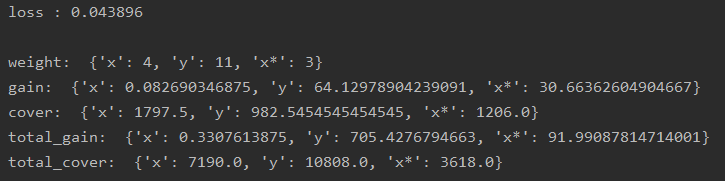
1. 複製一個變數並使用相同超參數結果:



x\* 這個變數複製x



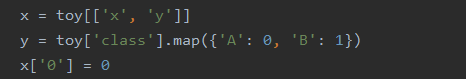


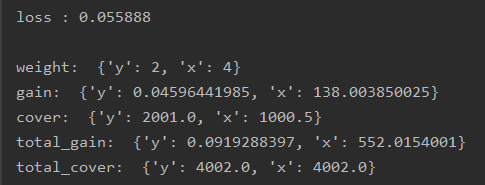


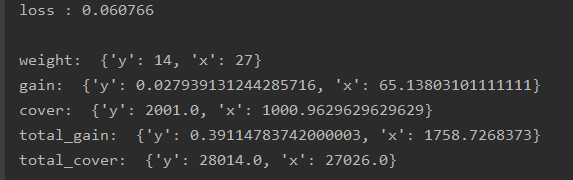
可以發現即使是相同超參數下，其validation loss 會大幅下降。

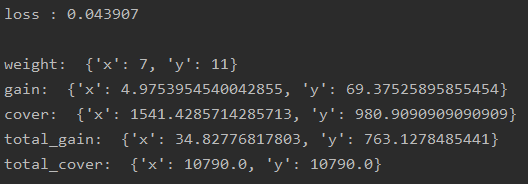
1. 新建一個所有資料皆為相同數值的變數，並使用相同超參數的 :

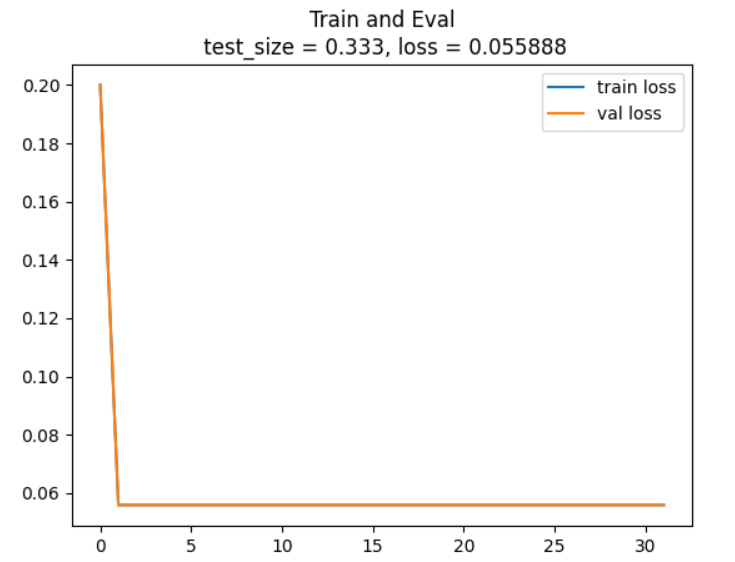
如下圖 : 新建一個資料全為0的變數







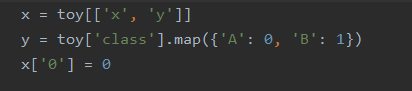




可以發現即使是相同超參數下，其validation loss 會大幅下降。並且與「複製一個變數並使用相同超參數結果」差異不大，其原理算是利用xgboost算法的問題(「xgboost使用貪心算法，即2個特徵變數的情況下，xgboost僅使用1個特徵變數作分割」)，將他擴展到「xgboost使用貪心算法，即n個特徵變數的情況下，xgboost僅使用n-1個特徵變數作分割」。已知toy\_train需要在座標軸上需要切割2刀才能完美切割4個象限，所以需要3個特徵變數。

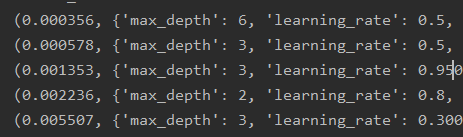
了解到xgboost貪心算法的問題，我們重新探討Clean train

資料處理如下:



使用 有用的參數x, y ，以及無特徵的參數 0 來訓練。

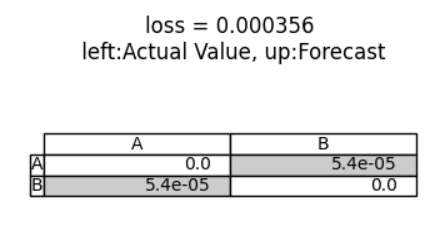
以超參數組中，多次(300次)隨機選取超參數組合來訓練。並以validation loss 由小到大排序，並輸出validation loss最小的5個



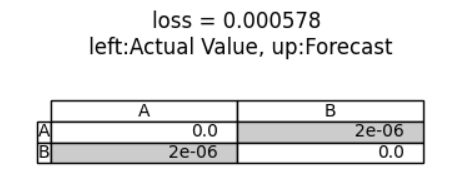
validation loss皆趨近於0，效果令人滿意。

分析這5組超參數對於不同class的精確度，實驗方法為使用新的資料作為測試，並拆分class為A以及class為B。

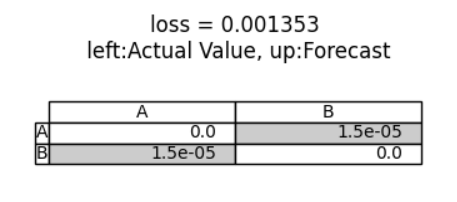
第一組超參數 :



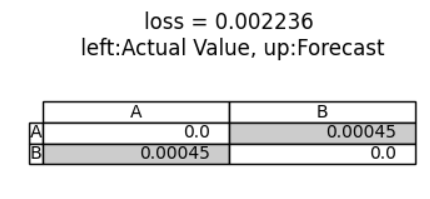
第二組超參數 :



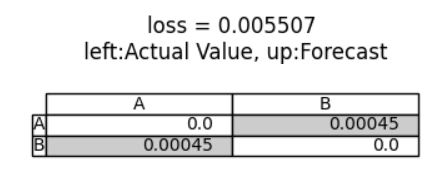
第三組超參數 :



第四組超參數 :

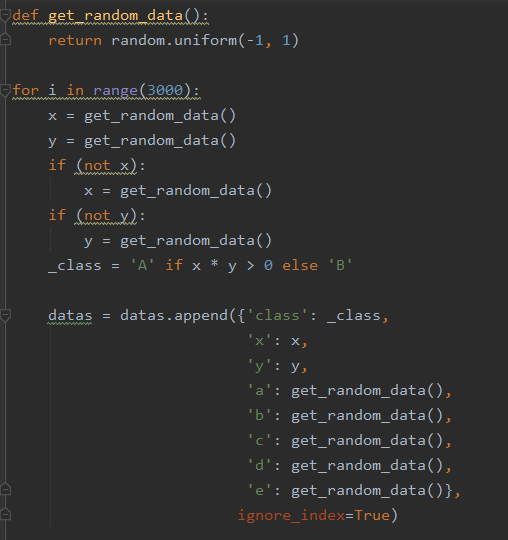


第五組超參數 :



可以發現訓練時的loss越小，其精確度越高。但並不是絕對的，loss小到某種程度下，一點點波動都會影響精確度，精確度已經99%以上了，所以對於小數點幾位並不太重要

**資料**

****

**3000筆，x, y, a, b, c, d, e都是隨機**

**而class在只有在x y 同號時為A，異號時為B。而其中的if (not x): x = get\_random\_data()，只是為了排除x等於0的情況，y同理。**

**以實現同號和異號的判斷。**