Machine Learning final report

學號：R07941023 姓名：呂彥穎

學號：R07323035 姓名：王嚴

學號：T08902109 姓名：賈成銷

A. Introduction and Motivation

此競賽為 Booz Allen Hamiltonooz 所舉辦，為了能了解孩童在學齡前遊戲上的表現會受到何因素所影響，還有重要的是教學影片對於孩童在學習上是否有提供幫助，所以透過這個比賽邀請各路好手一同解決和改善問題，為了吸引更多人參賽主辦方祭出了總計 160000 美元的獎金。

本次競賽的資料來源為 KIDS Measure Up! app 裡所保存的玩家遊玩紀錄，這個 App 是由 CPB-PBS Ready To Learn Initiative 所研發，這是一間由美國教育局所資助的公司 。競賽的參賽者需要預測玩家在某個 assessment 所得到的分數並分組，如果玩家第一次嘗試就通過 assessment 就會被分類為 3、第二次則為 2、三次以上則為 1、如果通過不了則為 0，所以也可以理解為這是一個類別有 4 種的分類問題，而最重要的是要找出教學影片和預測的結果之間的關係。

程式已包裝成把 test.py 的 scrip 直接複製貼上在 kaggle 上即可執行並得到結果.

lib:

- python=3.6.6
- numpy=1.16.4
- pandas=0.25.2
- matplotlib=3.0.3
- xgboost=0.90
- catboost=0.18
- shap=0.31.0
- scipy=1.2.1
- json=2.0.9
- tqdm=4.36.1
- sklearn=0.21.3
- lightgbm=2.3.0

B. Data preprocess&Feature engineering:

1.對不同類型 encode：

1. 建立一個新的由 title 和 event_code 所組成的 title_event_code 代表不同活動的事件列表
2. 對 event_code、 event_id、 world、title 做 encode
3. 處理 assessment 部分， 把 assessment 的 titile 抓出來
4. 將 Bird measurer(assessment)的 code 統一為 4110
5. timestamp 更改為 pd 格式的 datatime

2.創建 feature

feature 包括以下幾類

user_activities_count ：不同 type 的活動計數
accuracy_assement: 五種 assement 的正確率
event_code_count: event_code 計數
event_id_count: event_id 計數
title_event_code： title_event_code 計數
accumulated_correct_attempts： 正確嘗試次數
accumulated_uncorrect_attempts： 錯誤嘗試次數
duration_mean： 累計延時
accumulated_accuracy: 累計正確率
不同組別 group 次數 0-3
accumulated_accuracy_group： 累計正確組別
accumulated_actions： 總行為數

其他必要信息：

session_title： 尋找 label
installation_id:

total feature：890

3.得到處理後的 train 和 test

train_data、test_data :

1. 將至少有做過一次 assessment 的玩家抓出來
2. 去除 installation_id 和 accuracy_group

train_label: accuracy_group

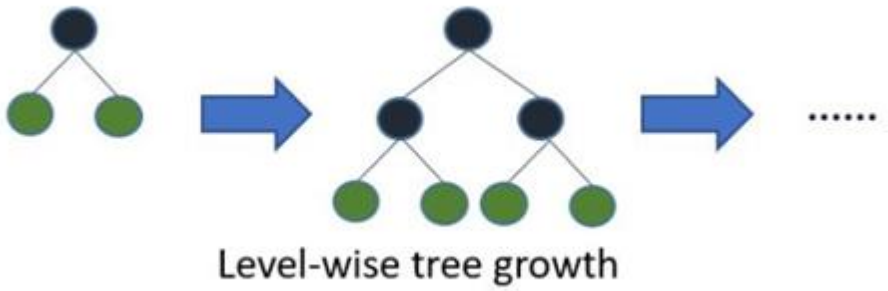
C. Model Description :

Boost 演算法是依序將 training data 放入不同的 weak classifier，而每一個 weak classifier 會分別調整 data 的 weight，其 Loss function 如下:

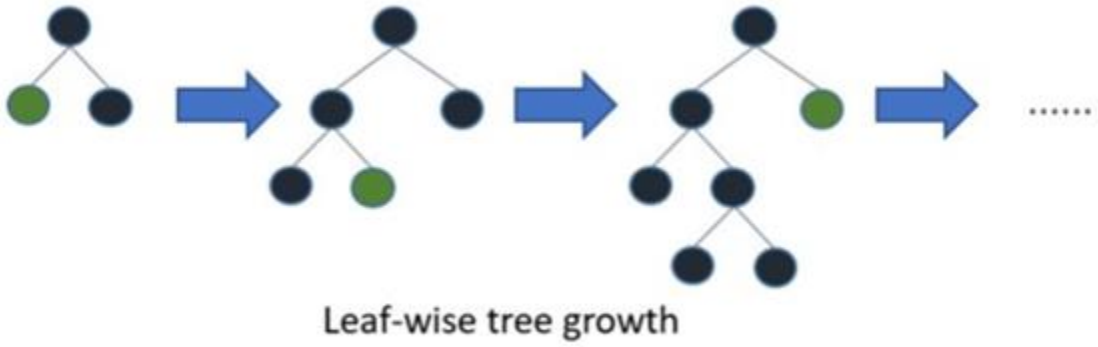
$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right)$$

我們將採用一下幾種不同的 boosting 演算法來訓練 data。

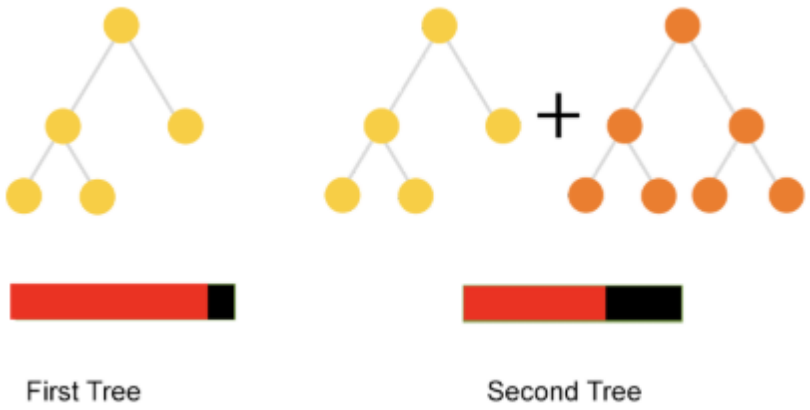
1. XGBoost: 最早在 2014 年被提出，是一種基於 decision tree 的 boosting 演算法，其中使用 level-wise 來進行分裂。



2. Light GBM: 最早在 2017 年被提出，是一種基於 decision tree 的 boosting 演算法，其中使用 GOSS (Gradient-based One-Side Sampling) 來實現 leaf-wise (最大增益)的分裂成長。



3. CatBoost (categorical boosting): 最早在 2017 年被提出，是利用 feature 的統計和 category 出現的頻率來進行疊代每一次的結構，來防止 overfitting。



D. Experiment and Discussion:

參考 Light GMB 的 document，我們認為在犧牲效率的情提下要提升準確度可以試著調整以下參數：

1. 提高 max_depth 和 num_leaves，且 num_leaves < (2*max_depth)
2. 提高 max_bin
3. 提高 num_iterations 和降低 learning_rate

其中第一點可能會造成 over fitting，第二點會讓速度降低。

以下為實驗結果：

max_depth	num_leaves	max_bin	learning_rate	num_iterations	kaggle
15	29	255	0.01	5000	0.54
20	29	255	0.01	5000	0.542
25	29	255	0.01	5000	0.542
25	49	255	0.01	5000	0.537
25	29	510	0.01	5000	0.536
25	29	255	0.001	6000	0.532

1. 增加 max_depth 可使準確度上升
2. num_leaves 增加並沒有幫助
3. max_bin 增加並沒有幫助
4. 調降 learning_rate 和增加 num_iteartions 並沒有幫助

XGBoost (weight)	Light GBM (weight)	CatBoost (weight)	Kaggle
1	0	0	0.517
0	1	0	0.542
0	0	1	0.533
0.6	0.2	0.2	0.541

1. xgboost 的 level-wise 會將增益較低的 leaf 一起分裂，相較於 lightGBM 的 leaf-wise，不僅耗時且會得到增益低部分影響的誤差。
2. 於只每次只分裂最大增益的 leaf，很可能會讓其中幾個增益大的 leaf 不斷的加深導致 overfitting，所以設置 max depth=35 的 constrain，讓分數最高。
3. catboost 的疊代成長會使訓練時間增加，成為這些 boosting 演算法中最久的一個。
4. catboost 的隨機數設定對準確率有一定的影響，沒能成為最佳者。
5. ensemble 的權重需要設置的恰當才能有好的結果。

E.Conclusion:

可以發現在三個模型中，LGB 的表現是最好的而 ensemble 也沒有帶來比較好的結果 ，所以最終我們決定只使用 LGB 作為我們的預測模型，不過我們並沒有特別調整其他兩個模型的參數，或許在參數上做更多的調整後在用 ensemble 會有比較好的結果。

由於在模型調整上沒有明顯的進步，要能得到顯著的進步或許要在對 feature 做更多處理，因為本次競賽的 data 存在序列關係，如果模型上加入 lstm 可能也會有比較好的結果，另外在 event_data 中有關於玩家點擊螢幕的座標，直覺上這是一個有用的資訊，不過要如何把這項資訊加入模型中是項挑戰，在討論區中也有人提出這個看法，但是沒有人提出具體該如何操作。

總結來說，如果要提升準確度可以從兩個方面著手，第一就是結合其他模型，像是 lstm、xgb 等等，第二就是增加新的 feature 和精簡現有的 feature，在比賽結束後會嘗試更多的想法，希望能把這學期所學靈活的運用在比賽中。

F.Reference:

https://catboost.ai/news/catboost-enables-fast-gradient-boosting-on-decision-trees-using-gpus

https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html

https://www.kaggle.com/braquino/890-features