Machine Learning final report

學號：R07941023  姓名：呂彥穎

學號：R07323035  姓名：王嚴

學號：T08902109  姓名：賈成銪

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

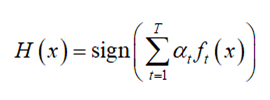
train\_data、test\_data :

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

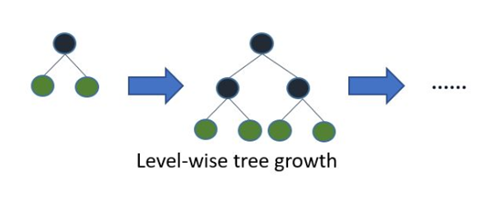
C. Model Description :

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:

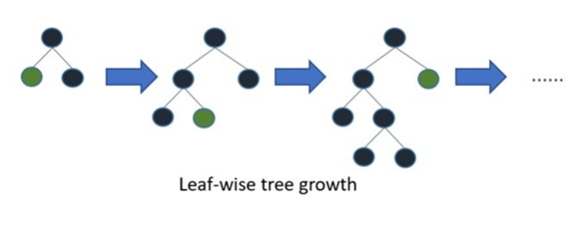


我們將採用一下幾種不同的boosting演算法來訓練data。

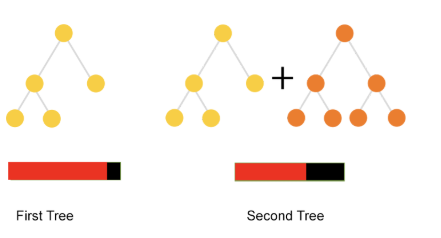
1. XGBoost: 最早在2014年被提出，是一種基於decision tree的boosting演算法，其中使用level-wise 來進行分裂。



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>