EE655000 Machine Learning Final Project
Behavior Classification of Exposition Visitors

Team member: 周伯宇 呂宸漢 林子鵑

Abstract

本次 Final Project 主題為 Aidea 平台上的比賽 “Behavior Classification of Exposition Visitors”（馬拉松博覽會參訪動線類別預測），探討一場位在台北的馬拉松博覽會中，群眾的行為與人潮動線關係與預測，以人潮停留在每個攤位的數量種類以及時間作為特徵來預測人潮動線分類。

我們這組的做法是先由 feature 觀察出與 label 的關係，並對 training data 做些許修改，從原本給定的 3 個 feature 擴展成 22 個 feature，放入五種不同的 model 中進行預測，再選出其中三個表現較好的 model 做 voting，最後輸出每個類別的機率值，最終在 testing data 上以 logloss 0.0407222 獲得不錯的成績。

Introduction

本次題目提供四個 csv 檔，以及一個展場配置圖，說明如下：

1. train.csv：訓練所需的樣本數據資料(CSV)，有 3 個特徵，共 41,640 筆。
   mac_hash：樣本代號。
   sniffer_loc：樣本資料採集地點。
   created_time：樣本資料採集時間。
2. training-label.csv：訓練所需的樣本數據資料標記(CSV)。
3. test.csv：測試所需的樣本數據資料，共 20,419 筆。
4. submit_samples.csv：模型預測結果上傳範例格式檔。
5. marathon-map.png：展場群眾收集器配置圖。

比賽目標是要根據給定的 training data feature 做分析，並透過 model 預測屬於每個類別的機率值，越接近正確 label 的話得到的 logloss 越低，計算對數加總後取平均，公式如下：

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

Methodology

整體方法 Framework 如 Fig.1 所示，首先將 data load 進來以後，根據資料統計分析結果做 data preprocess 重新組合 feature，接著透過三種 model 進行預測，並將三個 model 的結果投票，最後以 postprocess 產出最後的 output 結果。

Fig.1: The proposed method

以下以五個階段介紹詳細做法，分別為 Data Load、Data Preprocess、Model、Voting、Probability Postprocess。

A. Data Load

第一階段讀取 data，讀入每一個參訪者的編號 mac_hash，通過的攤位站點存入 sniffer_loc_list，以及通過的時間點存入 created_time_list。

B. Data Preprocess

第二階段針對讀入的 data 做 feature 調整。

根據 training data 與 training label 可以做統計分析得到一些規則：

- 每個 label 平均停留站點數

Label	C0	C1	C2	C3	C4
Station Cnt	1.52583	5.31506	6.05625	12.29480	10.96780

- 每個 label 在七個分類站點（不同顏色攤位）停留的機率值：

	P1(1)	P2(2, 4, 5, 6)	P3(3)	P4(8)	P5(7, 13)	P6(9, 10, 11)	P7(12, 14)
C0	1.93	31.97	17.24	4.23	10.62	12.89	21.11
C1	3.06	51.2	17.11	5.05	4.76	11.68	7.13
C2	0.19	20.17	3.49	9.27	12.57	39.46	14.86
C3	7.75	31.12	7.69	7.76	13.24	22.79	9.64
C4	0	33.58	8.08	8.13	12.05	25.01	13.16

由以上的資訊分析之後，我們可以得到停留站點數是一個重要的特徵，若根據分類站點停留的資訊來做分類也是可以提升 prediction accuracy。

經過觀察以後，發現站點數，分類攤位，以及參訪日期都是和 label 有一定程度相關性的 feature。因此，我們將原本 input 提供的三個 feature 擴展成 22 個 feature，前 14 個 feature 表示的是參訪者是否有通過編號 1-14 的 sniffer_loc，沒通過為 0，有通過則為 1；第 15 個 feature 表示的是參訪者通過的攤位總數；第 16-19 個 feature 為分類站點的加權數，四項 feature 分別

為通過 P2(2, 4, 5, 6)、P5(7, 13)、P6(9, 10, 11)、P7(12,14)的數量；第 20-22 項則代表參訪者在三天博覽會中是哪幾天進行參訪。

以下舉例其中一個參訪者的資訊經過 data preprocessing 之後的 feature，Fig.2 為其中一參訪者的原始 training data，經過 preprocess 得到新的 feature 如 Fig.3 所示。

```
00078611037990f7f36b722f22595fe7,3,2018-12-07 16:29:35
00078611037990f7f36b722f22595fe7,2,2018-12-07 16:30:41
00078611037990f7f36b722f22595fe7,4,2018-12-07 16:37:06
00078611037990f7f36b722f22595fe7,8,2018-12-07 16:37:07
00078611037990f7f36b722f22595fe7,6,2018-12-07 16:37:08
00078611037990f7f36b722f22595fe7,10,2018-12-07 16:38:10
00078611037990f7f36b722f22595fe7,11,2018-12-07 17:32:12
00078611037990f7f36b722f22595fe7,5,2018-12-08 17:30:38
```

Fig.2: Original training data with 3 features

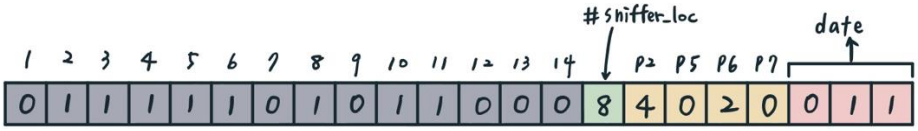


Fig.3: Modified training data with 22 features

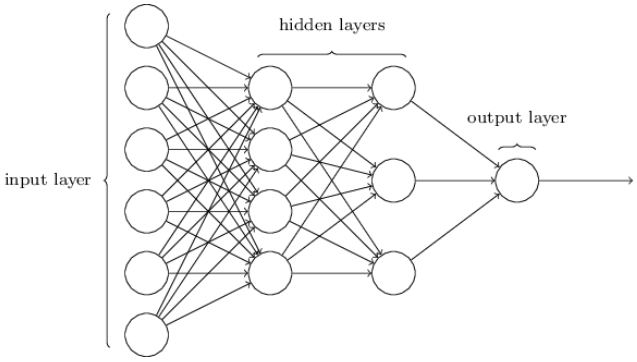
C. Model

我們所嘗試使用的 model 主要有五種：MLP、Decision Tree、Random Forest、Extreme Gradient Boosting (XGBoost) 以及 CatBoost。而透過觀察五種 model 的結果，loss 較低的有三種分別為 MLP、XGBoost 以及 CatBoost，最後會依據這三種 model predict 的結果進行 Voting 並決定最終類別機率。

以下分別介紹五種 model 的特性：

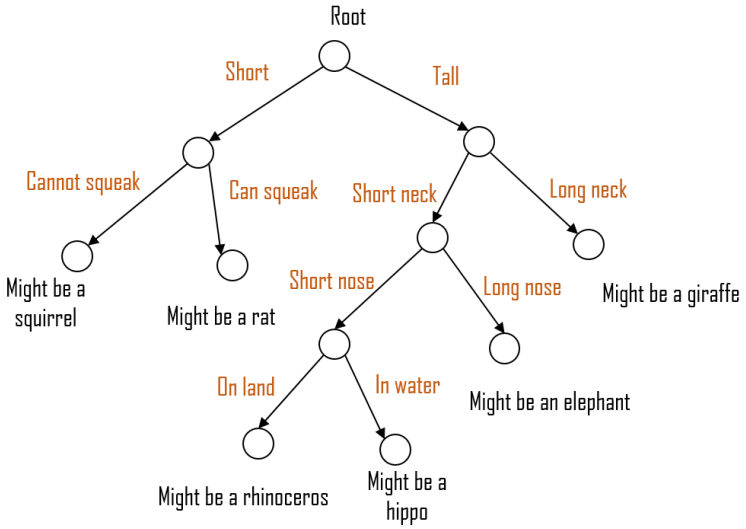
- Multilayer perceptron (MLP classifier)

  由一層 input layer、一層 output layer 和多層 hidden layer 組成的神經網路，每個 node 有 activation function 使其能達到非線性分類。MLP 使用 weight 存儲數據，並使用 back propagation 來調整 weight 並減少 training loss。其主要優勢在於其快速解決複雜問題的能力。



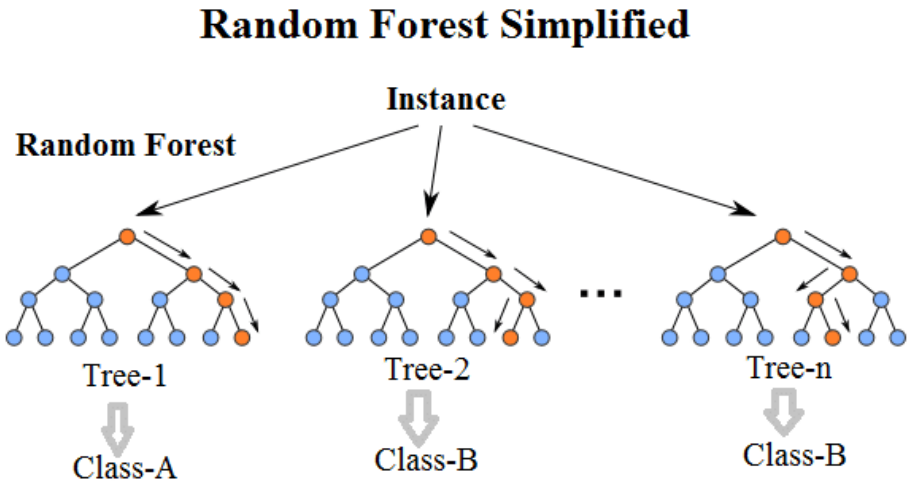
- Decision Tree

  根據訓練資料產生一棵樹，用以做 classification 以及 regression。透過算 entropy 或 gini index 來決定分支切點，差別在於 gini index 只能將決策一分為二，而 entropy 可產生多個分支。決策數的樹葉節點即為最終分類或預測結果。



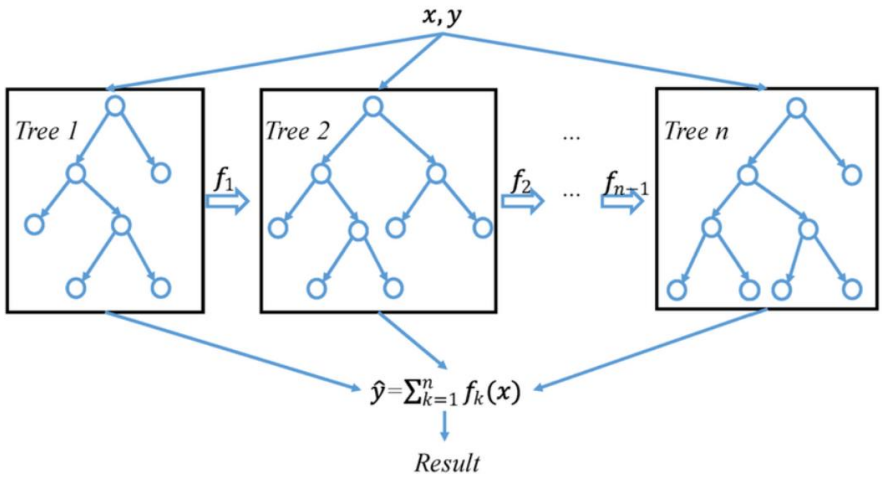
- Random forest

  隨機森林就是進階版的決策樹，而森林就是由很多棵決策樹組成。隨機森林是使用 Bagging 加上隨機特徵採樣的方法所產生出來的 ensemble algorithm。隨機森林藉由多棵不同樹的概念所組成，讓結果比較不容易 overfitting，並使得預測能力更提升。最終預測為每個決策樹多數決或是取平均的結果。



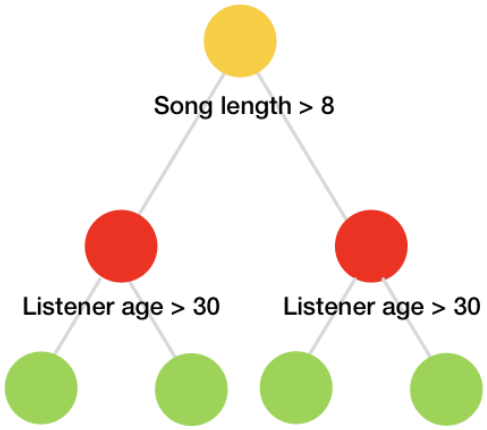
- XGBoost

  全名為 eXtreme Gradient Boosting，是一種 gradient boosting decision tree，每一棵樹是互相關聯的，目標是希望後面生成的樹能夠修正前面一棵樹犯錯的地方，同時也結合了 bagging 的概念，使每一個決策樹可以併行計算，樹的生長方式為 deep level wise，使分類或預測過程更有效率。



- CatBoost

  CatBoost 名稱源於 Category 和 Boost 兩個單字，同樣是基於 Gradient Boosting Tree 的梯度提升樹模型框架，其中他最特別的地方是能夠處理非數值型態的資料，也就是無需對數據特徵進行任何的預處理就可以將類別轉換為數字。訓練過程中允許沒有編碼的類別特徵，透過分類和數字特徵組合的各種統計量為類別型的特徵做編碼。不過在訓練前必須確保該特徵中無缺失值。其訓練資料若有缺失值 CatBoost 預設會將數值型的資料補上最小值，在效能上比 XGBoost 和 LightGBM 更加優化，同時支援 CPU 和 GPU 運算。



D. Voting

第四階段是將第三階段五種 model 中表現較好的其中三種：MLP、XGBoost 以及 CatBoost 進行 Voting Classification，進一步得到更準確的預測結果，而這裡使用的是 soft voting classification，將所有模型預測樣本為某一類別機率的平均作為標準，機率最高的對應類別為最終的預測結果，另外三種 model 的權重值分別為 1, 0.9, 1。

E. Probability Postprocess

第五階段則是將其中一類別機率大於等於 0.99 的樣本預測結果修改為 1 和 0，以降低 logloss 並提升整體 Accuracy。

Experiment Result

針對 data preprocess 的做法進行了以下不同 feature 組合的測試，總共 22 項 feature 如前面章節介紹，而我們抓出了 4 種 feature 組合，分別為 1-14 項（只有 sniffer_loc 資訊）、1-15 項（加入通過 sniffer loc 總數量）、1-19 項（加入 group sniffer_loc 權重），以及 1-22 項（加入參訪日期資訊）。

另外我們將以上組合還分成兩種來探討，一種是以時間資訊來產生 feature，一種是忽略時間資訊產生 feature，以下會介紹兩種 22 個 feature 分別如何產生：

- Feature with time (Fig.4)

  第 1-14 項（sniffer_loc）：以參訪者在每個 sniffer_loc 停留的時間計算。

  第 15 項：以參訪者在每個 sniffer_loc 停留的時間加總計算。

  第 16-19 項：分別在各 group 中 sniffer_loc 的時間加總計算。

  第 20-22 項：參訪者是哪一天參訪。

- Feature without time (Fig.5)

  第 1-14 項（sniffer_loc）：以參訪者是否經過每個 sniffer_loc，若經過該 feature 為 1，未經過 feature 則為 0。

  第 15 項：以參訪者在每個 sniffer_loc 停留的站點數量加總計算。

  第 16-19 項：分別在各 group 中 sniffer_loc 的站點數量加總計算。

  第 20-22 項：參訪者是哪一天參訪。

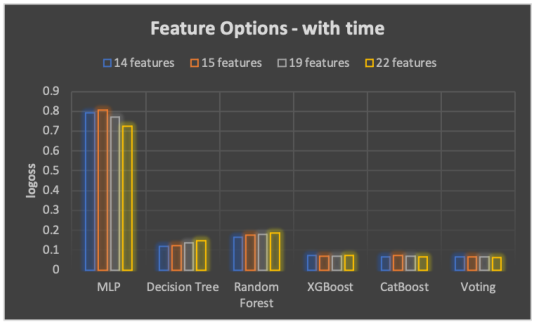


Fig.4: Feature with time

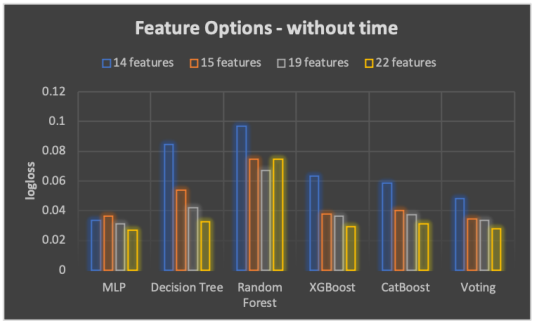


Fig.5: Feature without time

以下表格顯示五種 model 在 validation data 上的預測結果，可以從表格中（Tab.1 & Tab.2）發現有時間資訊所產出的 feature 反而導致訓練結果變差，因此最後選擇使用 without time 的 feature 產生方式，model 是採用 MLP、XGBoost 以及 CatBoost 來做 Voting，並得到最終預測結果。

	MLP			decision tree			random forest		
without time	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**
14 features	0.990661	0.036946	0.033722	0.951868	0.084999	0.0848	0.970546	0.097102	0.096627
15 features	0.987787	0.039307	0.036212	0.970546	0.053942	0.053743	0.986351	0.075009	0.074496
19 features	0.989224	0.037479	0.031061	0.97773	0.04238	0.042181	0.987787	0.067771	0.067272
22 features	0.987787	0.030198	0.026762	0.98204	0.032767	0.032767	0.986351	0.07509	0.074661
with time	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**
14 features	0.728448	0.866346	0.792089	0.926006	0.119089	0.119089	0.963362	0.166197	0.165612
15 features	0.729167	0.855589	0.805905	0.923132	0.123714	0.123714	0.961925	0.174565	0.173908
19 features	0.724138	0.83412	0.772163	0.914511	0.137588	0.137588	0.95546	0.181326	0.180719
22 features	0.739224	0.778513	0.72458	0.908046	0.147994	0.147994	0.95977	0.187949	0.187422

Tab.1: Validation result on MLP, decision tree and random forest

	xgboost			catboost			voting		
without time	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**
14 features	0.978448	0.064072	0.063355	0.979167	0.059202	0.058363	0.987787	0.048933	0.048149
15 features	0.987069	0.040821	0.037934	0.987069	0.043833	0.040355	0.989943	0.037399	0.034668
19 features	0.986351	0.039284	0.036133	0.987069	0.040934	0.037215	0.991379	0.036535	0.033327
22 features	0.987787	0.031896	0.029086	0.987069	0.034429	0.031379	0.992098	0.028443	0.02782
with time	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**	**accuracy**	**log-loss**	**post log-loss**
14 features	0.974138	0.084219	0.071359	0.981322	0.073608	0.06662	0.980603	0.072926	0.065319
15 features	0.968391	0.083068	0.069018	0.97342	0.077106	0.073133	0.977011	0.073193	0.064333
19 features	0.969109	0.08418	0.071001	0.978448	0.076714	0.067952	0.972701	0.07454	0.067345
22 features	0.968391	0.07763	0.072832	0.977011	0.065661	0.064777	0.972701	0.066532	0.063382

Tab.2: Validation result on XGBoost, CatBoost and voting

Conclusion

我們在這次 Final Project 中以五種 model 嘗試對 training data 做訓練，忽略 training data 中駐留時間的資訊，並觀察 data 的特性，提出有效率的 feature 產生方式，並且使用 voting 機制取出最好的預測結果，accuracy 達到 99.2%，最終在 Aidea 平台上獲得最好的 logloss 為 0.0407222。