ML Project

資管三 陳品妤 B07705020
資管三 黃心 B07705013
資管三 李旻叡 B07705037

前言

這次 final project 的主題是關於飯店訂單與收益之間的關係。由資料集提供 2015/01 ~2017/03 之間每筆訂單的詳細資料、平均每日收益、訂單是否被取消等資訊，並提供每日的實際收益標籤，希望藉由這些資訊訓練出適當的模型，用以預測往後數個月的訂單帶來的每日收益標籤。我們將整個流程分成兩個部分：

第一部分為每日收益的預測，我們嘗試利用 Bagging、GradientBoost、XGBoost 及 Random Forest 等方法，先透過每筆訂單的資料預測其平均每日收益（adr）、訂單是否被取消（is_canceled），再透過預測出的資訊計算出預測每日收益。

第二部分為每日收益標籤的預測，我們原本也打算利用 machine learning 的方法，利用實際每日收益及實際每日收益標籤來訓練模型，但我們在過程中發現兩者之間有一個簡單的數學關係：

$$label = min([Revenue \ / \ 10000], \quad 10)$$

因此我們捨棄原本的方法，改為直接使用此關係進行每日收益及每日標籤之間的轉換。

結合兩個部分我們先預測出每日受益，再透過轉換得到其每日收益標籤。

在下個小節，我們將簡單介紹我們用來預測的三種方法及在預測過程中所做的處裡。

模型建立

1. Random forest

Preprocessing

1. 獨立出 train data 中應被預測的欄位：獨立 adr、is_canceled
2. 將 test data 中沒有的欄位從 train data 中切除：切除 reservation_status 及 reservation_status_date，確保 train data 與 test data 中的欄位相同
3. 切割 validation data：因為 test data 的日期是緊接在 train data 的日期後面，因此我們直接將 train data 中，後三個月的資料切割作為 validation data。

4. 處理 nominal data：將 nominal data 轉換成 dummy variable。
5. 處理 nan：將所有 interval data 中的 nan 轉換成 0 (因為通過觀察，有 nan 的欄位中，0 的比例皆高於 90%，因此平均值都十分接近 0)

模型部分

預測 adr：利用 python 中的 sklearn 執行 random forest 的預測，使用的函數為 RandomForestRegressor。將 train data 依照 7:3 的比例切割成 train data 及 test data (此處的 test data 為 adr 的 test data，並非 label 的 test data)，訓練出模型並計算出 score 及 E-out (使用 mean square error)。最後使用整筆 train data (沒有切割過的) 訓練出模型。

預測 is_canceled：利用 python 中的 sklearn 執行 random forest 的預測，使用的函數為 RandomForestClassifier。將 train data 依照 7:3 的比例切割成 train data 及 test data (此處的 test data 為 is_cenceled 的 test data，並非 label 的 test data)，訓練出模型並計算出 score 及 E-out (使用 0/1 error)。最後使用整筆 train data (沒有切割過的) 訓練出模型。

以上預測使用的參數為：決策數棵數 200 棵，每個 leaf points 最小資料量 5 筆。

Feature selection

使用 backward elimination 總計刪掉了三個 feature。
首先是 ID 這個 column，我們認為這個 column 提供的資訊量為 0 ，因為他作為一筆 nominal data，每筆交易的 ID 皆不相同，因此才被選為第一個刪除的 feature。
再來是 agent 及 company 這兩個 columns。同樣是 nominal data，這兩筆資料的分散程度也非常大，而且空值的數量非常多，提供的資訊量十分有限，因此才被選為接下來刪除的兩個 feature。

最佳結果

預測 is_canceled 時 validation 的 accuracy 為 0.90；預測 adr時 validation 的 square error 為 384.52。最後將這個最佳模型以整個 training data 訓練，產生的 test label 誤差為 0.34。

2. XGBoost

介紹

XGBoost 是一種 Gradient Boosted Tree (GBDT)，每一次保留原來的模型不變，並且加入一個新的函數至模型中，修正上一棵樹的錯誤，以提升整體的模型。

Preprocessing

1. 獨立出 train data 中應被預測的欄位：獨立 adr、is_canceled
2. 將 test data 中沒有的欄位從 train data 中切除：切除 reservation_status 及 reservation_status_date，確保 train data 與 test data 中的欄位相同

3. 處理 nominal data：將 nominal data 轉換成 dummy variable。
4. 處理 nan：將所有 interval data 中的 nan 轉換成 0 (因為通過觀察，有 nan 的欄位中，0 的比例皆高於 90%，因此平均值都十分接近 0)

模型部分

預測 is_canceled：利用 python 中的 sklearn 執行 logistic regression 以及 XGBoost 的預測，其函式為 LogisticRegression 和 XGBClassifier。將給定的 training data 以 9 : 1 的比例劃分成 subtraining 和 validation，利用 accuracy 來挑選最佳的模型，最後利用整筆 training data 來訓練模型。

預測 adr：利用 python 中的 sklearn 執行 XGBoost 的預測，其函式為 XGBRegressor。將給定的 training data 以 9 : 1 的比例劃分成 subtraining 和 validation，利用 score 來挑選最佳的模型，最後利用整筆 training data 來訓練模型。

Feature selection

針對預測 is_canceled 的模型，我們使用三種 feature 的組合，第一種是只取我們認定的相關feature，即為 previous_cancellations、previous_bookings_not_canceled 和booking_changes，這是為了避免模型把不相關的 feature 也納入考量，可能造成預測不夠準確；第二種是取除了 country、arrival_date_year 和 arrival_date_day_of_month
外的所有 feature，因為我們發現這些 feature 的 feature importance 偏低；第三種則是 arrival_date_year 和 arrival_date_day_of_month 以外全部的 feature。

針對預測 adr 的模型，我們使用兩種 feature 的組合，第一種是取除了 ID、country、arrival_date_year 和 arrival_date_day_of_month
外的所有 feature，因為我們發現 country 的 feature importance 偏低；第二種則是 arrival_date_year 和 arrival_date_day_of_month 以外全部的 feature。

在預測 is_canceled 和 adr 時，我們調整了四個參數，分別為 max_depth、learning_rate、n_estimators 以及 subsample。當預測 is_canceled 的時候，我們發現我們限制樹的最大深度為10、learning rate 為 0.1、n_estimators 是 800 且 subsample 等於 0.95 時，能夠有最佳的結果；當預測 adr 的時候，我們發現我們限制樹的最大深度為10、learning rate 為 0.05、n_estimators 是 1000 且 subsample 等於 0.85 時，能夠有最佳的結果。

最佳結果

預測 is_canceled 時，使用 XGBClassifier 並選取第二種 feature selection 的方法，如此 validation 的 accuracy 為 0.87；預測 adr 時，使用XGBRegressor並選取第二種 feature selection 的方法，所產生的 validation 的 score 為 0.88。最後將這個最佳模型以整個 training data 訓練，產生的 test label 誤差為0.5。

3. Bagging

Preprocessing

1. 獨立出 train data 中應被預測的欄位：獨立 adr、is_canceled
2. 將 test data 中沒有的欄位從 train data 中切除：切除 reservation_status 及 reservation_status_date，確保 train data 與 test data 中的欄位相同
3. 處理 nominal data：將 nominal data 轉換成 dummy variable。
4. 處理 nan：將所有 interval data 中的 nan 轉換成 0 (因為通過觀察，有 nan 的欄位中，0 的比例皆高於 90%，因此平均值都十分接近 0)

模型部分

預測 is_canceled：利用 python 中的 sklearn 執行 Bagging 的預測，使用的函數為 BaggingClassifier，其 base algorithm 使用 Decision Tree，共100棵，為 sklearn 的 DecisionTreeClassifier，整體概念類似 Random forest，但每棵樹的權重是相同的。將 train data 中抽取十分之一進行 Validation，訓練出模型並計算出 score 及 E-out (使用 0/1 error)。最後使用整筆 train data (沒有切割過的) 訓練出模型。

預測 adr：利用 python 中的 sklearn 執行 Bagging 的預測，使用的函數為 BaggingRegressor其 base algorithm 使用 Decision Tree，共100棵，為 sklearn 的 DecisionTreeRegressor，整體概念類似 Random forest，但每棵樹的權重是相同的。將 train data 中抽取十分之一進行 Validation，訓練出模型並計算出 score 及 E-out (使用 mean square error)。最後使用整筆 train data (沒有切割過的) 訓練出模型。

Feature selection

針對預測 is_canceled 以及 adr 的模型，我們使用 Forward selection，逐步增加使用的 feature，最終選用了 previous_cancellations, previous_bookings_not_canceled, lead_time, stays_in_weekend_nights, stays_in_week_nights, booking_changes, deposit_type, market_segment, reserved_room_type, customer_type, meal, hotel, arrival_date_month, total_of_special_requests, country 共14項 feature，並對其中屬於 nominal data 的部分進行 one hot encoding。

最佳結果

預測 is_canceled 時 validation 的 accuracy 為 0.88；預測 adr 時 validation 的 square error 為 467.48。最後將這個最佳模型以整個 training data 訓練，產生的 test label 誤差為 0.434211。

4. GradientBoost

Preprocessing

1. 獨立出 train data 中應被預測的欄位：獨立 adr、is_canceled

2. 將 test data 中沒有的欄位從 train data 中切除：切除 reservation_status 及 reservation_status_date，確保 train data 與 test data 中的欄位相同
3. 處理 nominal data：將 nominal data 轉換成 dummy variable。
4. 處理 nan：將所有 interval data 中的 nan 轉換成 0 (因為通過觀察，有 nan 的欄位中，0 的比例皆高於 90%，因此平均值都十分接近 0)

模型部分

預測 is_canceled：利用 python 中的 sklearn 執行 GradientBoost 的預測，使用的函數為 GradientBoostingClassifier，其 base algorithm 使用 Decision Tree，共100棵。將 train data 中抽取十分之一進行 Validation，訓練出模型並計算出 score 及 E-out (使用 0/1 error)。最後使用整筆 train data 訓練出模型。

預測 adr：利用 python 中的 sklearn 執行 GradientBoost 的預測，使用的函數為 GradientBoostingRegressor，其 base algorithm 使用 Decision Tree，共100棵。將 train data 中抽取十分之一進行 Validation，訓練出模型並計算出 score 及 E-out (使用 mean square error)。最後使用整筆 train data 訓練出模型。

Feature selection

針對預測 is_canceled 以及 adr 的模型，我們使用 Forward selection，逐步增加使用的 feature，最終選用了 previous_cancellations, previous_bookings_not_canceled, lead_time, stays_in_weekend_nights, stays_in_week_nights, booking_changes, deposit_type, market_segment, reserved_room_type, customer_type, meal, hotel, arrival_date_month, total_of_special_requests, country 共14項 feature，並對其中屬於 nominal data 的部分進行 one hot encoding。

此外，經過測試後我們發現 is_canceled的模型上，其 base algorithm 的 Decision Tree，最大層數限定為 20，adr 限定為10有較佳的預測效果。

最佳結果

預測 is_canceled 時 validation 的 accuracy 為 0.88；預測 adr 時 validation 的 square error 為 480.99。最後將這個最佳模型以整個 training data 訓練，產生的 test label 誤差為 0.513158。

結果比較

比較以上四種模型的結果，最終 label 的預測準確度以 Random forest 最佳，Bagging 次之，XGBoost、GradientBoost 較差。而耗時方面，由於最終選用的 Feature 數目不同導致 column 數有所差異，以 GradientBoost 及 Bagging 最快（228個，約十分鐘），Random forest 次之（325個，約二十分鐘）， XGBoost 最慢（293個，約兩個小時）。

針對我們選出的最佳 Random forest 模型，其優點如下：
1. 訓練可以並行化，訓練上較為快速，樣本較大的情形下能夠節省許多時間。
2. 由於選擇 feature 時隨機選擇，因此在樣本維度較高的時候，仍然具有比較高的訓練效能。
3. 由於具有隨機抽樣，訓練出來的模型 square error 小，應用範圍廣。

其缺點如下
1. 在某些 noise 較大的 feature 上，模型容易 overfitting。
2. 值種類較多的 feature 容易 overfitting。

分工
1. Random Forest ：黃心
2. XGBoost：陳品妤
3. Bagging 和 GradientBoost ：李旻叡