機器學習技法Final Project

組別：PASS
組員：資管碩一R09725017謝欣珊、資管碩一R09725059吳昀蔚、資管碩二R08725011沈家妡

一、前言

本Project希望利用現有的飯店訂房資料去預測每天的營收，開始使用資料之前，本小組先進行資料觀察，發現訓練資料跟測試資料的欄位有所差異，is_canceled、adr、reservation_status以及reservation_status_date等四個欄位只出現在訓練資料集，並且訓練資料的label一共有10個類別，因此本小組最先想到的預測方法是忽略以上四個欄位，並且將取消的訂單資料去除，把每天的資料合併成一筆建立分類模型進行預測，然而這個方法沒有考量到測試資料也有取消的訂單，會使得營收被高估，因此本小組發想了方法二，計算每日非取消訂單佔每日所有訂單的營收比例平均值作為對於每日營收高估的調整比例，在預測每筆訂單的ADR並且計算每日的營收後，將每日的營收依照算出的比例進行調整，然而這樣的做法也只是粗略的估計沒有取消的訂單佔比，每筆訂單的情況可能不同，因此本小組利用Logistic regression、Random forest以及XGBoost進行訂單是否取消的預測，選擇表現最好的XGBoost，作為方法三所使用的演算法，在方法三先預測測試資料的訂單是否是取消訂單，再預測每筆訂單的ADR並且計算每日的營收，最後將營收以萬為單位紀錄，得到預測的label。

二、Algorthms & Packages 介紹

本小組使用了Logistic regression、Random forest、Linear regression以及XGBoost等四個演算法，其中Logistic regression、Random forest以及Linear regression是課堂上面教過的演算法，因此以下僅簡略介紹並寫出使用的Packages，會更著重在XGBoost的介紹：

1. Linear regression、Logistic regression、Random forest

   Linear regression是以最小化square error為目標的函式，對一個或多個自變數和應變數之間關係進行建模，希望找到一個hyperplane跟所有點的距離加總最小[1]，本小組將使用sklearn.linear_model.LinearRegression進行建模。

   Logistic Regression與Linear regression類似，差別在於Linear regression是針對連續變數，而Logistic Regression則是類別變數，計算成功機率與失敗機率的比值，作為勝算（Odds），觀察X變數對勝算的影響[2]，本小組將使用sklearn.linear_model.LogisticRegression()進行建模。

   Random Forest是基於Decision Tree的一種Ensemble Method，會隨機抽取樣本來建立決策樹，最後結合多棵決策樹的投票結果來進行分類[3]，本小組將使用sklearn.ensemble.RandomForestClassifier()。

2. XGBoost

   XGboost全名為eXtreme Gradient Boosting(極限梯度提升)是一個Boosting的演算法，以CART回歸樹模型作為樹模型，是將多個樹模型聚集在一起，形成一個強分類器的提升樹模型，目標函式有兩大部分，分別是計算真實跟預測之間的差距以及正規化項，透過正規化項來避免過度擬合的發生，也利用二階泰勒展開式來近似最小化目標函數以求得最佳解。XGboost的優點眾多，除了上述提到的可避免過度擬合、目標函數優化利用了損失函數關於待求函數的二階導數之外，還有對於稀疏數據的處理、進行交叉驗證、early stop以及設置樣本權重，透過調整權重可以去更加關注一些樣本，是一個精準度很高的演算法，也是目前在許多Kaggle競賽上面得到第一名的隊伍所使用的演算法[4]，本小組將使用XGBClassifier, XGBRegressor[5]。

三、方法及流程介紹

1. 方法一：直接預測label
   a. 流程
      i. 資料前處理
         1. 考量到test資料中沒有’is_canceled’，‘adr’，‘reservation_status’，‘reservation_status_date’等欄位，故train時不考慮這些欄位，同時不計is_cancel = 1之訂單
         2. 統整每天的訂單，並以當日訂單之統整為一筆training data。

3. 變數部分依照原變數為分類資料、數值資料有不同處理。
    a. 取當天訂單之平均: ‘lead_time’, ‘arrival_date_week_number’, ‘stays_in_weekend_nights’, ‘stays_in_week_nights’, ‘adults’, ‘children’, ‘babies’, ‘previous_cancellations’, ‘previous_bookings_not_canceled’, ‘booking_changes’, ‘required_car_parking_spaces’
    b. 取當日各類別之個數: ‘distribution_channel', ‘is_repeated_guest’, ‘reserved_room_type’, ‘assigned_room_type’, ‘deposit_type’, ‘agent’, ‘company’, ‘customer_type’
    c. 取當日數值區間之個數: ‘days_in_waiting_list’, ‘total_of_special_requests’

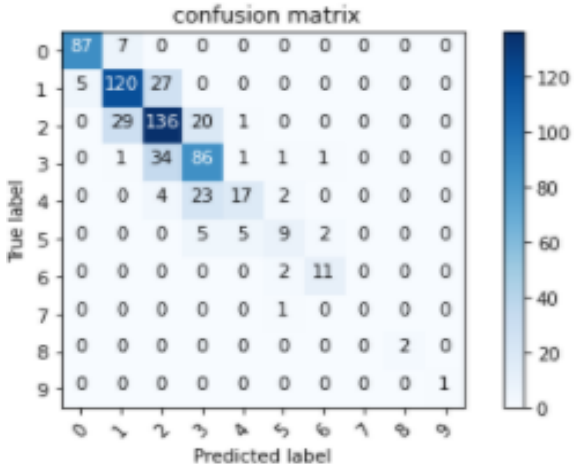
ii. 利用前處理過的training dataset建立預測模型，去預測testing dataset的label

b. 使用方法與package：sklearn.linear_model.LogisticRegression()

c. 結果

i. 以此方法預測 test label，並上傳競賽結果為：
- Public score = 1.065789
- Private Score=0.961039

ii. 針對training dataset的預測如下:
- MAE= 0.2875
- accuracy= 0.77124
- 右圖是confusion matrix



2. 方法二：計算非取消訂單的佔比以及預測ADR

a. 流程

i. 資料前處理
1. 將訓練資料中’is_canceled’、’adr’獨立出來以便計算revenue以及預測ADR
2. 去除測試資料中沒有的變數：’reservation_status’以及’reservation_status_date’
3. 進行日期轉換：將’arrival_date_year’、’arrival_date_month’、’arrival_date_day_of_month’合併（格式yyyy-mm-dd）
4. 去除不使用的變數：’country’、’agent’以及’company’
5. 將類別變數（'hotel', 'meal', 'market_segment', 'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type'）轉換成dummy variable
6. 補齊training dataset有但testing dataset在dummy之後沒有的欄位：’market_segment_Undefined’、’distribution_channel_Undefined’、’reserved_room_type_L’、’assigned_room_type_L’、’customer_type_Transient-Party’

ii. 計算每日非取消訂單的revenue在所有訂單的revenue佔比平均值
1. 計算每筆訂單的revenue：
住宿天數 = stays_in_weekend_nights + stays_in_week_nights
revenue = ADR*住宿天數
2. 加總每日所有訂單的revenue以及非取消訂單的revenue
3. 計算每日非取消訂單的revenue佔所有訂單的revenue佔比：
每日非取消佔比 = 非取消訂單的revenue加總／所有訂單的revenue加總
4. 計算每日非取消佔比平均值（每日佔比加總／天數）

iii. 使用Linear regression建立預測模型，預測testing dataset中每筆訂單的ADR

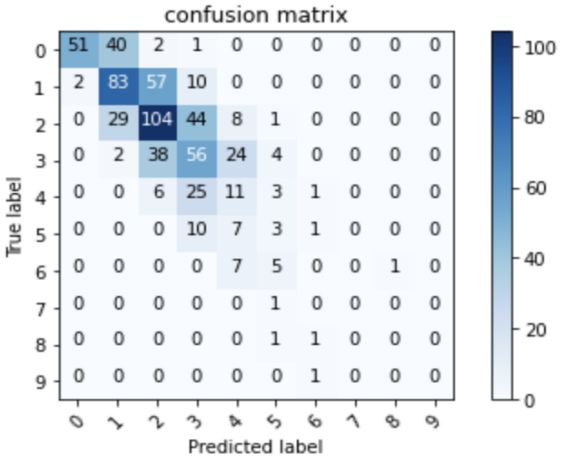
iv. 計算testing dataset每筆訂單的revenue：

住宿天數 = stays_in_weekend_nights + stays_in_week_nights
revenue = ADR*住宿天數

v. 加總testing dataset每日所有訂單的revenue
vi. 將每日的revenue利用前面算出的每日佔比平均值進行調整：
調整過的每日revenue = 每日revenue*佔比平均值
vii. 將調整過的每日revenue除以10,000取商，即為預測的label

b. 使用方法與package：sklearn.linear_model.LinearRegression()
c. 結果
i. 預測 test label，並上傳競賽結果為：
- Public score = 0.986842
- Private Score=1.090909

ii. 每日非取消佔比平均值 = 0.6649460334
iii. 預測每筆訂單 ADR 的模型：
MSE(training data) = 1489.457063698
iv. 用預測的結果計算 training data 的 label
- MAE= 0.6140625
- Accuracy=0.48125
- 右圖是confusion matrix



d. 針對取消訂單的處理改善：
利用每日佔比的平均去計算的誤差會相當大，因此本小組嘗試用分類模型去對testing dataset資料進行是否取消的預測，以下列出利用三種演算法進行cross validation的平均準確率結果：

指標＼演算法	Logistic regression	Random forest	XGBoost
平均準確度	0.810796	0.813080	0.859784

由上表可以看到XGBoost的表現較好，因此在方法三將採用XGBoost進行訂單是否取消的預測。

3. 方法三：預測取消與預測ADR
a. 流程
i. 資料前處理:
1. 去除不使用的變數: 'country'
2. 類別變數('hotel', 'meal', 'market_segment', 'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type')轉換為 dummy variable

ii. 使用前處理過的所有 training data 來訓練模型
iii. 使用 XGBoost 建立預測模型，預測訂單是否會被取消
iv. 使用 XGBoost 建立預測模型，預測單筆訂單的 ADR
- 計算每筆訂單的 revenue：
住宿天數 = stays_in_weekend_nights + stays_in_week_nights
revenue = ADR*住宿天數
- 將每天預測沒被取消的訂單的 revenue 加總，並除以10,000取商，即為預測的label

b. 使用方法與package：XGBoost[5]（官方文件連結在最後一頁的參考資料）
i. 參數設定:

使用 sklearn.model_selection 的 GridSearchCV 調整兩個預測模型的參數。預測訂單是否會被取消的模型參數，及預測每筆訂單 ADR 的模型參數。
GridSearchCV用於自動調整參數，用法為寫一堆循環，藉由自己設定的參數列表，一個一個試，找到最合適的參數。另外，因使用全部 training data 來 train model，所以參數 ‘subsample’ 設定為 1。
而兩種模型須調整的參數如下:

參數名	意義	取值範圍
booster	決定選用哪種 booster	gbtree 或 gblinear
n_estimators	關係到模型複雜度，決定樹的個數	[400, 500, 600, 700, 800]
max_depth	決定樹最大的深度	3~10 之間的整數
min_child_weight	最小權重總和，用於控制 overfiit	[1, 2, 3, 4, 5, 6]
gamma	指定進行拆分所需的最小損失減少量	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]
colsample_bytree	每個樹隨機樣本列的樹	[0.6, 0.7, 0.8, 0.9]
reg_alpha	關於權重的L1正則化項，以在非常高的維度的情況下使用，以便算法在實現時運行得更快	[0.05, 0.1, 1, 2, 3]
learning_rate	通過縮小每一步的權重，使模型更加穩固	[0.01, 0.05, 0.07, 0.1, 0.2]

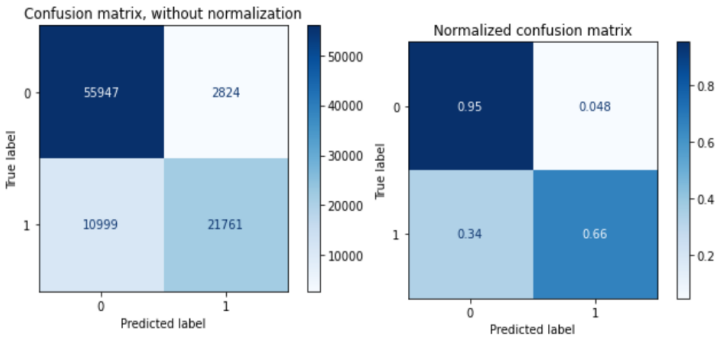
→ 預測訂單是否會取消的模型:
- booster 使用 XGBClassifier()，為 gbtree，基於 tree 的模型進行 boosting 計算
- 以 f1 作為 GridSearchCV 挑選參數的 score 基準
- 其他參數設定:
  {'learning_rate': 0.07, 'n_estimators': 400, 'max_depth': 3, 'min_child_weight': 5, subsample': 1, 'colsample_bytree': 0.8, 'gamma': 0, 'reg_alpha': 3}

→預測每筆訂單 ADR 的模型
- booster 使用 XGBRegressor()，為 gblinear，，基於 linear 的模型進行 boosting 計算
- 以 MAE 作為 GridSearchCV 挑選參數的 score 基準
- 其他參數設定:
  {'learning_rate': 0.1, 'n_estimators': 400, 'max_depth': 3, 'min_child_weight': 5,'subsample': 1, 'colsample_bytree': 0.8, 'gamma': 0.1, 'reg_alpha': 0.05}

c. 結果
以此方法預測 test label，並上傳競賽結果為：
Public score = 0.394737、Private Score=0.441558
另外，以下為此方法預測 training data 的結果:

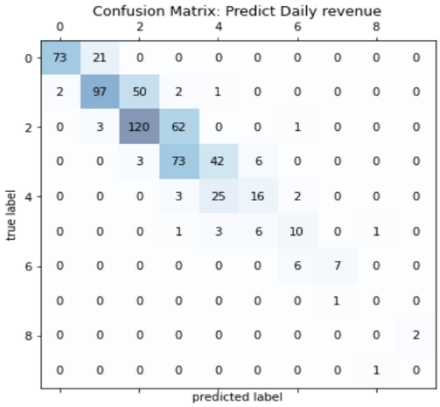
i. 預測訂單是否會取消的模型
- F1 score(training data) = 0.758950
- Accuracy(training data) = 0.848980

- 下圖是normalize前後的confusion matrix



ii. 預測每筆訂單 ADR 的模型
- MAE(training data) = 15.574717
- MSE(training data) = 782.571287

iii. 用預測的結果計算 training data 的 label
- MAE(training data) = 0.401562
- Accuracy(training data) = 0.621875
- 右圖是confusion matrix



四、方法比較

項目/方法		方法一	方法二	方法三
簡介		將同一天的訂單，以比例、平均等方法總合成一筆data。 直接用各天data，預測當天的y。	預測各訂單adr， 並將當日revenue乘以(非取消訂單revenue/總訂單revenue)比例，避免高估revenue。	預測各訂單的取消與否以及adr。 再利用revenue公視加總當天訂單並換算成y。
使用模型		Logistic regression	Linear regression	XGBoost
效率	資料預處理	資料須經過較多預處理，最慢	將類別資料轉成dummy，並計算取消之revenue比例之平均	只需將類別資料轉成dummy，最快
	模型	與資料維度成線性關係，較快。 加上 n 較其他二者小(總天數)，模型訓練較快。	與資料維度成線性關係，較快。	原理為gradient boosting decision tree，計算較慢。 但有通過平行處理、分散計算等方式加速。
擴充性(匯入新資料訓練)		資料預處理較繁瑣，匯入新訓練資料最麻煩。 但新資料訓練快速。	預處理居中。 新資料訓練快速。	預處理最容易。 但因為非線性模型，受訓練資料影響較大，應較常更新。 然而訓練速度最慢。
實際	解釋性	適用於各類別和X為線性關係。	適用於數值結果和X為線性關係。	可畫出決策樹，看出資料預測過程。

應用		較易解釋，並較易看出各變數影響(顯著性)。	較易解釋，並較易看出各變數影響(顯著性)。	但較難解釋決策樹如何選擇變數並建立。
	效果	訓練相對快，但不適用於非線性關係資料。	訓練相對快，但不適用於非線性關係資料。	通常預測較準確，但也較容易overfit。
結果	score	1.065789	0.986842	0.394737
	推測	因為沒有考慮被cancel的訂單，導致test的revenue被高估。 可能X,y之間非線性關係，logistic預測較差。	不能直接用過去平均的取消結果來預測未來各天狀況。 X,adr的關係可能非線性，導致預測較差	預測cancel時，XGBoost表現較logistic好很多，故可推測X,y應為非線性關係。 adr的預測亦然，應為非線性關係。

五、Final Recommendation

我們推薦使用方法三用來預測 revenue，因為以預測結果而言，方法三的結果是最為準確的。以下為方法三的優劣分析：

- 優點：
  1. 透過個別預測訂單是否被取消及訂單的 ADR，可以更準確估計 daily revenue
  2. 方法三所使用的 Package XGBoost 原理就是 gradient boosting decision tree，通常gradient boosting 的準確度較高，但是計算速度會比較慢。但因 XGBoost 有做Parallelization、Distributed Computing、Out-of-Core Computing的設計，所以在於計算速度上相對於其他 gradient boosting decision tree 的package 快，且模型表現也不錯
- 缺點：
  1. 因為此方法對每筆訂單同時做了是否會被取消及 ADR 的預測，但 ADR 的準確度並不高，因而也連帶影響最後計算daily revenue label的準確度
  2. 因為資料前處理對類別特徵做 dummy 的轉換，造成 feature sparse，會降低 learning 的效率

六、工作分配表:

姓名	工作內容
吳昀蔚	方法一、方法比較
沈家妡	方法二、前言、Algorthms & Packages 介紹
謝欣珊	方法三、Final Recommendation

七、參考資料：

1. Linear regression – 維基百科。線性迴歸。https://zh.wikipedia.org/wiki/%E7%B7%9A%E6%80%A7%E5%9B%9E%E6%AD%B8
2. Logistic regression – 永析統計及論文諮詢顧問。羅吉斯迴歸分析(Logistic regression, logit model)-統計說明與SPSS操作 https://www.yongxi-stat.com/logistic-regression/。
3. Random forest – Chung-Yi。ML入門（十七）隨機森林(Random Forest)。https://medium.com/chung-yi/ml%E5%85%A5%E9%96%80-%E5%8D%81%E4%B8%83-%E9%9A%A8%E6%A9%9F%E6%A3%AE%E6%9E%97-random-forest-6afc24871857
4. XGBoost – 人工智能遇見磐創。一文讀懂機器學習大殺器XGBoost原理。https://kknews.cc/zh-tw/news/grejk5m.html。
5. XGBoost – 官方文件。https://xgboost.readthedocs.io/en/latest/python/python_intro.html#install-xgboost