Machine Learning Techniques Final Project

廖偉霖 B06502009 羅恩至 B06502027 吳泓昇 B06502057

1 Introduction and Learning Strategy

我們這組的 project 重點在於「在相同資料前處理的情況下，不同 Model 的表現差異，以及它們表現差異的可能原因」。在"is_canceled"方面，我們選用 *SVM*、*Random Forest*、*Logistic Regression* 這三種 Model；在"adr"方面，我們選用 *Ridge Regression*、*Random Forest Regressor*、*Neural Network* 這三種 Model。

我們之所以選用以上六種 Model，最大的原因就是上課教過，我們希望透過這次 final project 來跟我們這學期所學相互映證。以下是我們這次 final project 的進行流程：

圖一：final project 流程圖

2 Approaches

以下為上述流程圖的說明：

2.1 Label Function

利用 train.csv, train_label.csv 找出 feature 和 label 的關係，而我們得到的結論為：$\sum_{same\ date}[\![is\_canceled = 0]\!](total\ nights)(adr) mod 10000 = label$

2.2 Data Preprocessing

起初我們僅挑選 numerical column 並捨棄 categorical column 再分別預測"is_canceled"、"adr"，但準確率不高，因此開始嘗試將資料利用到最大化。

主要想法是不要隨意 drop 資料，且盡量避免用人工的方式篩選資料，畢竟人工篩選很主觀，雖然可以藉由 human knowledge 做處理，但我們很難看到藏在資料集中的 insight，只能從建模訓練完的各種指標去推斷該前處理是否合適。雖然資料集中包含許多 categorical column，且在 one-hot encoding 之後會產生大量(745)columns，但由於 train.csv 共有 91531 筆資料，相對是很足夠的。另外，在選擇這樣的想法之後，有比較不同的前處理例如在 one-hot encoding 之後：

(1) 把"agent"/"country"的 rare value(只出現過 1 次) merge 成一類

(2) 把"adults", "children", "babies"合成新的 column "total_guest"

(3) 把 12 個"month" merge 成 4 個不同"season"

…等等，而以下步驟是我們「認為」表現最好的前處理：

對 train.csv, test.csv 進行以下步驟

(1) Country 用眾數補值(PRT)

(2) Children 用眾數補值(0)

(3) 如果 agent 的值為 nan 且同 sample 的 company 有值就填進去

(4) agent 用眾數補值(9.0)

(5) Drop [“ID”, “company”, future fields columns]

(6) 合併兩個處理過的 DataFrame

(7) 對我們定義好的 categorical columns 做 one-hot encoding

(8) 將此 DataFrame 再分割回 training/testing data

2.3 Model Training

在使用不同 model，我們都固定從 training data 切出 20%當作 testing data，且設置相同 random state，確保訓練資料的統一性及整個流程的 reproductivity。另外我們不做 hyperparameter tuning，一方面避免 overfitting，二方面是在簡單實驗後發現差異不大，且我們「相信」各使用 model 的 default 參數就有一定的 generalization 的能力。

2.4 Cross Validation

我們使用 5-fold cross validation，其中選出 accuracy 最高（MSE 最低）的g^-作為我們預測 testing data 的 model，並且將 model 寫入 pickle 檔，之後便不需要再重新走過 2.1~2.4 的步驟，直接匯入 pickle 檔即可。

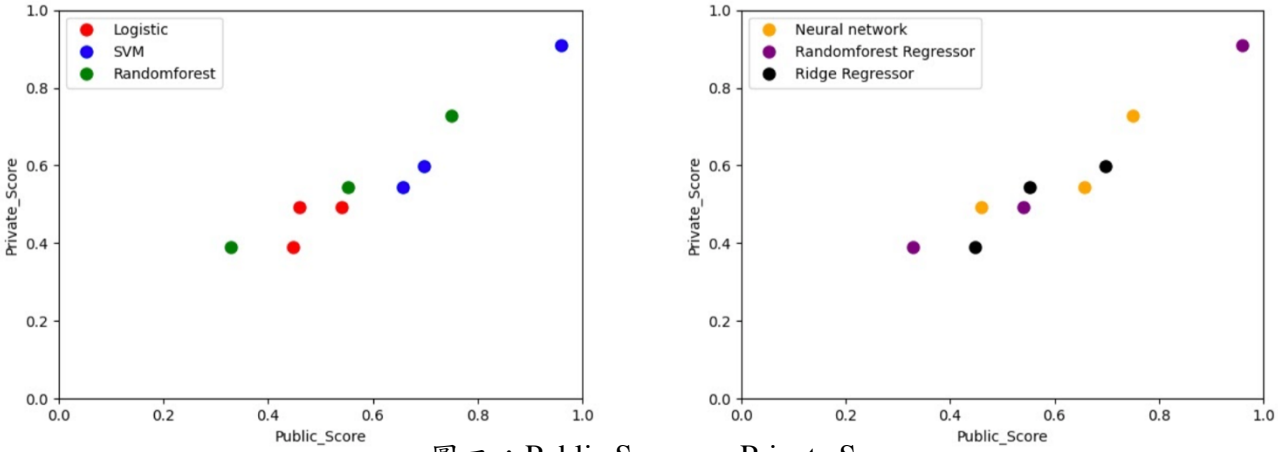
2.5 Combine Model

分別從預測”is_canceled”及”adr”中各挑選一個 model 出來預測 testing data 上的資料，再使用 2.1 中的 Label Function 得到 testing data 的 label。

3 Results and Comparison

以下為我們依照上述步驟得到 9 種不同模型組合的結果，圖二左右圖的每個散佈點均為一種模型組合結果，而右圖表示該散佈點在預測”is_canceled”所使用的 model，左圖則是代表預測”adr”所使用的 model，例如最左下角的點即為在”is_canceled”使用 Random Forest，而”adr”則使用 Random Forest Regressor。

3.1 Comparison of Public Score and Private Score



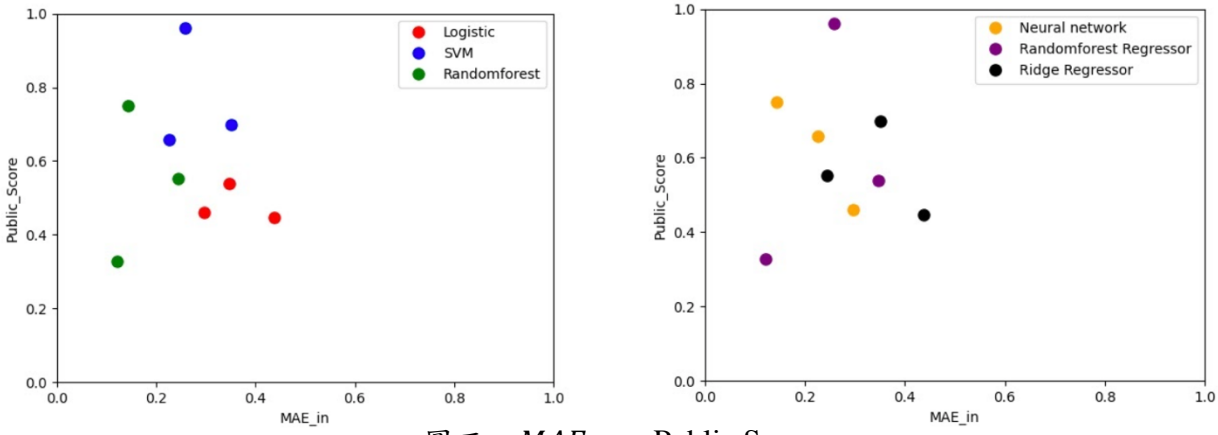
圖二：Public Score vs. Private Score

從圖二可以看出，使用 SVM + Random Forest Regressor 具有最高的 public/private score，而 Random Forest + Random Forest Regressor 最低。此外，雖然我們的散佈點並沒有在同一條直線上，但彼此相差均不大，因此可推測我們 overfit 到”public score”的程度不高，且有鑒於 testing data 樣本數不高，造成樣本變異數可能較高，因此 public/private score 出現些微差異也是合理的。在預

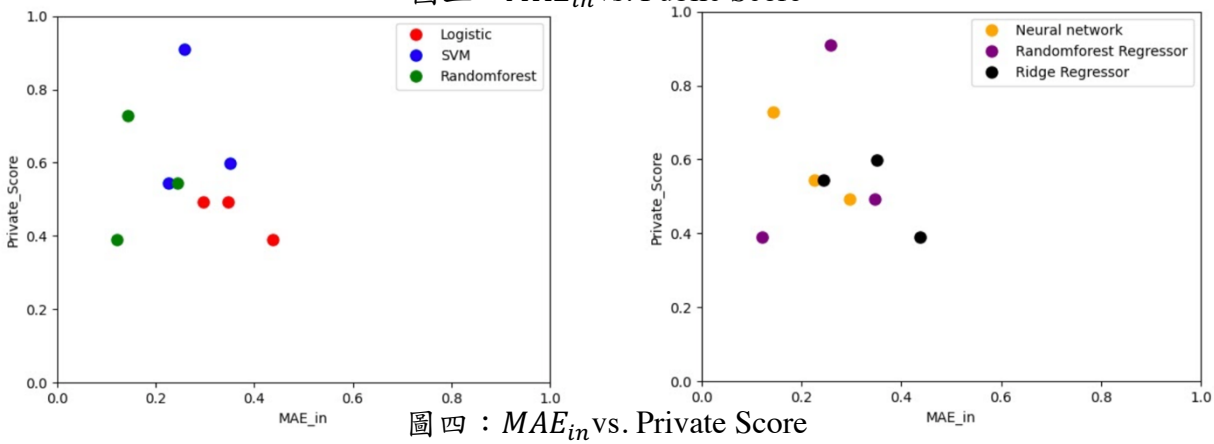
測”is_canceled”（右圖），SVM 會得到較高的值，Logistic Regression 的值則落在 0.5 附近；而在預測”adr”的部分（左圖），每種模型所預測出來的三個值都有不小之差異，難以判別出預測”adr”的 model 對 public/private score 的趨勢為何。因此推論，選擇使用何種模型去預測”is_canceled”，對於 public/private score 具有較大的影響。

3.2 Comparison of MAE_{in} and Public Score/Private Score (MAE_{out})

MAE_{in}為我們用這 9 種組合在 training data 上計算得到的 MAE。



圖三：MAE_{in}vs. Public Score



圖四：MAE_{in}vs. Private Score

從圖三、圖四可以看出，我們的散佈點符合$MAE_{in} < MAE_{out}$的趨勢，也可以在圖三、四左圖觀察到 Random Forest 有較低的MAE_{in}，其次是 SVM，符合 3.1 的結論–預測”is_canceled”對結果影響較大。在預測”is_canceled”的部分，SVM 的MAE_{out}相對較高，$|MAE_{out} - MAE_{in}|$最大；而 Logistic Regression 在$|MAE_{out} - MAE_{in}|$表現上最穩定；Random Forest 的MAE_{in}及MAE_{out}值皆為最低。另一方面，在預測”adr”的部分，雖然我們「認為」它對結果影響相對較小，但我們觀察到 Neural Network 的$|MAE_{out} - MAE_{in}|$相對較高，可能是因為它較容易 overfit。整體而言 SVM + RF Regressor 這組表現(public score)最差，Random Forest + RF Regressor 表現最佳，Logistic Regression + Ridge Regression 的$|MAE_{out} - MAE_{in}|$最小。而使用 RF Regressor 仍表現差，我們「粗略」推測是因為 tree based 和 distance based 的 model 運作原理不同，若將這兩種 model 組合預測，可能會 somehow 增加誤差，而這也可能是 Random Forest 只有在搭配 RF Regressor 的表現才會特別好，在搭配其他 distance based 的 model 表現就沒有那麼突出。

3.3 Comparison of Models from Other Perspectives

Model (is_canceled)	E_{test}	Efficiency	Popularity	Interpretability	Scalability
Logistic Regression	0.1738	0.25 min/fold	High	Median	High
Linear Kernel SVM	0.1462	75 min/fold	Medium	High	High
Random Forest	0.094	0.44 min/fold	High	Median	Low
Model (adr)	MSE_{test}	**Efficiency**	**Popularity**	**Interpretability**	**Scalability**
Ridge Regression	802.8	0.024 min/fold	High	High	High
RF Regressor	159.38	3.08 min/fold	High	Median	Low
Neural Network	629.37	18.75 min/fold	High	Low	High

表一：Models with Different Perspectives

不同於 3.1、3.2 最後的組合表現，在 3.3 這部分我們只討論單一模型的表現，而表一中的E_{test}及MSE_{test}均為在 2.4 得到 modelg^-後，我們使用g^-預測在 2.3 切出的 testing data 的結果，作為我們比較不同 model 之間的一個指標。

我們可以看出 Random Forest 和 Random Forest Regressor 分別在各自的標的下表現得最好，且平均訓練每 fold 的時間也不會太久；相較下 SVM 跑得非常久，推估原因是 SVM 通常適合用在中小型資料，而我們的 data size 很大，同時在本身 feature 很多的情況下，SVM 的訓練效果可能也不彰，且根據 sklearn 官方文件：「The QP solver used by the libsvm-based implementation scales between $O(n_{feature} \times n_{samples}^2)$ and $O(n_{feature} \times n_{samples}^3)$」也能看出這些缺點；而 Ridge Regression 雖然訓練的極快，但由於我們的資料維度太高，讓他沒辦法表現得很好，比較適合用在當作其他 model 的初始權重；另外我們也試過 Linear Regression，同樣訓練很快，但因為沒有 regulator 的關係，結果接近發散狀態。

Popularity 的部分，Logistic Regression、Random Forest 在 classification 的 task 下都非常被使用，而 SVM 在近幾年已經不像以往那樣常用；而在 regression 的 task 下，Ridge Regression、Random Forest Regressor、Neural Network 也常被使用，尤其在低維度資料下 Linear Regression 就能表現的不錯，而 Ridge Regression 在加入 regulator 情況下能提高 generalization 的能力，且相對使用 L1 regularization 的 Lasso Regression 更容易做 optimization。

Interpretability 的部分，在 classification 的 task 下我們認為 Tree-base 的 model 會有比較好的 Interpretability，而 Random Forest 是 Decision Tree 經過 Bagging 算法獲得的，因此雖然無法像 Decision Tree 可以從單一個 tree 解釋分類結果，且訓練過程具一定 randomness，但經過 vote 得到結果也具有一定的 Interpretability；而 Logistic Regression 是 Linear Regression 在 classification task 下的 extension，同

樣是輸出一個定值，但輸出為機率，因此相較 Linear Regression 能直接透過 weight coefficient 解釋輸出，Logistic Regression 無法很直觀的解釋，但仍有一定的 Interpretability；而 SVM 因為搭配 Kernel 函數，轉換到更複雜的空間，一般被認為 Interpretability 較差，但我們使用的是 Linear Kernel，weight coefficient 即是在原本的空間找到一個 hyperplane 區分不同 class，因此具有良好 Interpretability。在 regression task 下，Ridge Regression 為 Linear Regression 加上 L2 regulator，因此具有良好的 Interpretability；Random Forest Regressor 和其 Classifier 同樣奠基在 Bagging 算法，僅是在計算 impurity 時使用 square error，而最後再取每棵 Decision Tree 結果平均，具有一定 Interpretability 但不及 Ridge Regression；Neural Network 由於具有 hidden layer，因此被形容為 black box，描述其低 Interpretability 的特性。

Scalability 的部分，我們將 Scalability 定義為 model 在 data size 增大時保持 Performance（Efficiency, Memory, Accuracy…）的能力。使用 Stochastic Gradient Descent(SGD)優化的 model 具有較佳的 Scalability，如："Linear" SVM, Logistic Regression；另外 Neural Network 在 training 過程可使用平行計算，因此 Scalability 同樣較佳。Decision Tree 在 data size 增大時長出的 tree 同樣也會增大，而 Random Forest 為多個 Decision Tree 的 aggregation，因此具有低 Scalability。

4 Recommendation

我們推薦使用 Random Forest + RF Regressor。

4.1 Pros

(1) 準確度高
(2) 常被使用
(3) 容易建模

在競賽初期我們便使用這個組合，因此這也是我們能在競賽前期維持較低 Public Score 的原因；而雖然我們在競賽中沒有刻意調整參數，但若有需求，此組合的兩個 model 在參數調整上並沒有太多參數需要調整，相較不少 model 算是非常容易使用。

4.2 Cons

(1) 訓練時間稍久
(2) 可解釋性稍低
(3) 可規模性低

由於兩個 model 都需要數個 Decision Tree 才能做預測，因此需要訓練需要較長的時間，且可規模性也因它是 tree based 而低；而如同 3.3 所述，兩者都是透過 Bagging 算法，同時訓練過程都具有 randomness，因此資料可解釋度相對較低。

4.3 Conclusion

雖然在 Efficiency 兩者都不是最佳的，但兩者分別都在準確度上有更好的結果，在組合上也具有最好的結果。同樣地，雖然資料可解釋性較差，但由於他

的 performance 讓它常被使用。綜合以上，若處理與本次 project 相似的 dataset 類型，我們認為「Random Forest」搭配「Random Forest Regressor」是值得推薦的，另外儘管我們沒有調整參數，但經查詢資料後發現若要避免 Random Forest 的 overfitting 還是需要適當的調整參數並在調參期間做好 Validation。

5 Other Experiment and Discussion

5.1 Prediction Target

我們這組在一開始看到題目時曾對我們學習的策略產生分歧，一方認為我們應該先單獨預測每筆 data 的"is_canceled"和"adr"，然後透過某種事先 train 好的 label scaling function 轉換成 label；另一方認為我們可以把一筆筆單獨的下單資訊特徵，透過某種轉換過程轉換成每天下單資訊的特徵，然後直接做 Multiple Classification。但是，我們發現如果要做 Multiple Classification，在中間的轉換過程會使我們的 data 數從本來大約十萬筆降到剩 640 筆資料，這會導致整體學習成果沒那麼高，所以我們最後選擇照著前者的思路來完成這次 project。

5.2 Error Discussion

如同在 3.1 提及 testing data 的 variance 較大，因此過度追逐 public score 的意義不大，一來很可能是 overfit 到 public score，二來也可能純粹是運氣較差，得到的 public score 低了些，但其實整體是不錯的。從競賽後期看到前 40%的組別 public score 其實差不到 0.1，但最後的 private score 卻有高有低也是印證了上述的說法；不過同樣也是 sample size 的問題，我們的 private score 只奠基在 76 筆 data，因此也沒辦法提供一個「精準」的指標告訴我們預測表現如何。另外，由於$E(E_{in}) < E(E_{out})$，我們的 public score 本質上就很難超越我們在 training set 的表現，且資料及本身也具有 noise，就算有能力在 training data 上排除 noise，但只要 testing data 上仍存在 noise，把E_{out}降低的難度又會提高許多。而經觀察後，我們這次使用的資料集確實存在 noise，例如：duplicated data 具有不同 label、以及部分 feature 具有大量空值。

6 Collaboration and Work Loads

廖偉霖負責做 Random Forest + RF Regressor，羅恩至負責做 Logistic Regression + Neural Network，吳泓昇負責做 SVM + Ridge Regression，其他部分包括 project 初期策略、報告撰寫等等為共同討論分工。

7 Reference

[1] Christoph Molnar. 2021. Interpretable Machine Learning *A Guide for Making Black Box Models Explainable*

[2] Matthias Döring. 2018. KDnugget Supervised Learning: Model Popularity from Past to Present (web post)

[3] Mikio Braun. 2014. Big Data Zone: What is Scalable Machine Learning? (web post)

[4] Scikit-learn.org: 1.4. Support Vector Machines (official document)