第二組 HW3 REPORT

H24081341 耿皓畇 H24081163 陳詠翰 B14081027 劉恩兆

1) Introduction

這次的作業目的是要找出一個適用於不同資料集的模型 (generalized model)，此模型包含填補遺失值，以及利用填補後的資料來預測精準的結果。而我們總共有 7 個資料集，但是各資料集的訓練集和測試集都有部分的遺失值，因此先填補適當的遺失值，能夠幫助幫助我們追求更高的預測準確率。我們在這次作業中嘗試了許多填補遺失值的方法和不同預測模型，並在這些方法和預測模型的組合中尋找效果最好的組合。

2) Methodology

遺失值補值方法我們嘗試了三種方法，分別為 fancyimpute 套件裡的 KNN、MICE 還有 LGBMImputer :

- KNN：在這個方法中嘗試了四種不同的 K 值，包括 k = 3, 5, 7, 9
- MICE：在這個方法中嘗試了兩種不同的 iteration 次數，max_iter=10 & 100
- LGBMImputer：這個方法中使用的 iteration 次數為 n_iter=500

另外，在填補遺失值時我們也使用兩種不同的方式填補 :

1. 將補值模型 fit 到 train 資料集，再將學習好的補值模型 transform 到 test 資料集
2. 直接將 train 和 test 資料合併，利用補值模型 fit 和 transform

會想要嘗試將 train 和 test 資料合併的主要原因是我們認為將資料合併能增加觀測值數量，有機會使補值模型訓練效果更好。

遺失值補值完後，我們使用 Standard Scaler 進行資料前處理，將特徵值進行標準化

- Train 和 Test 資料分開 : 先對 train 資料進行 fit ，再將訓練完的 Standard Scaler transform 到 test 資料集
- Train 和 Test 資料合併 : 直接把合併的資料集使用 Standard Scaler fit 和 transform

訓練的模型我們使用了三種訓練模型，分別為 XGBoostRegressor, LGBMRegressor 和 Neural Network。由於每種遺失值補值方法所搭配的訓練模型參數有些許不同，因此我們將各種模型的參數組合以 A~F 代表，我們將 A~F 分別代表的參數組合列在表格(一)。

表格(一) 方法總整理

方法	遺失值填補	Train & Test set 是否分開填補	預測模型	模型設定
1	fancyimpute KNN(k=3)	No	XGBoostRegressor	D
2	fancyimpute KNN(k=3)	No	LGBMRegressor	C
3	fancyimpute KNN(k=5)	No	XGBoostRegressor	D
4	fancyimpute KNN(k=7)	No	XGBoostRegressor	D
5	fancyimpute KNN(k=9)	No	XGBoostRegressor	D
6	MICE (default)	Yes	LGBMRegressor	A
7	MICE (max_iter=100)	Yes	LGBMRegressor	A
8	MICE (max_iter=100)	No	LGBMRegressor	B
9	MICE (default)	No	XGBoostRegressor	D
10	MICE (max_iter=100)	No	Neural Network	E
11	LGBMImputer	No	XGBoostRegressor	D
12	LGBMImputer	No	LGBMRegressor	C
13	LGBMImputer	No	Neural Network	F
14	None 【註】	None 【註】	XGBoostRegressor 【註】	D

【註】未做遺失值填補，直接使用 XGBoost 模型內建處理遺失值方法預測結果

Model Setting

LightGBM

A. LGBMRegressor(learning_rate=0.01, n_estimators=3000)
B. LGBMRegressor(learning_rate=0.01, n_estimators=550)
C. LGBMRegressor(boosting_type='gbdt',num_leaves=31, max_depth=-1, learning_rate=0.01, n_estimators=100, objective='regression', metric='mae')

XGBoost

D. XGBRegressor(verbosity=0, learning_rate=0.1, n_estimators=550, max_depth=4, min_child_weight=5, gamma=0.1)

Neural Network

E. Loss function = MSE
F. Loss function = MAE

- 4 層隱藏層，節點分別為 24、12、8、1 個
- Optimizer = 'Adam', activation function = 'relu'
- 訓練模型時，epoch=200，batch size=20

3) Experimental Analysis

表格(二) 資料筆數

Dataset		number of rows	number of columns (未扣除 target column)
Data 1	train	765	9
	test	309	9
Data 2	train	574	9
	test	230	9
Data 3	train	371	14
	test	151	14
Data 4	train	6064	9
	test	2457	9
Data 5	train	7077	5
	test	2870	5
Data 6	train	1187	12
	test	479	12
Data 7	train	231	7
	test	92	7

KNN

在 KNN 遺失值填補中，我們分別使用 K=3, 5, 7, 9 來對遺失值進行填補，並從繳交的歷史紀錄中試著找出洞見。而在預測模型的選擇上，我們先挑選了兩種模型，分別是 LGBMRegressor 及 XGBRegressor，並在比較過後挑選出預測表現較好的模型為 XGBRegressor。

其使用的參數如下：

XGBRegressor(verbosity=0, learning_rate=0.1, n_estimators=550, max_depth=4, min_child_weight=5, gamma=0.1)

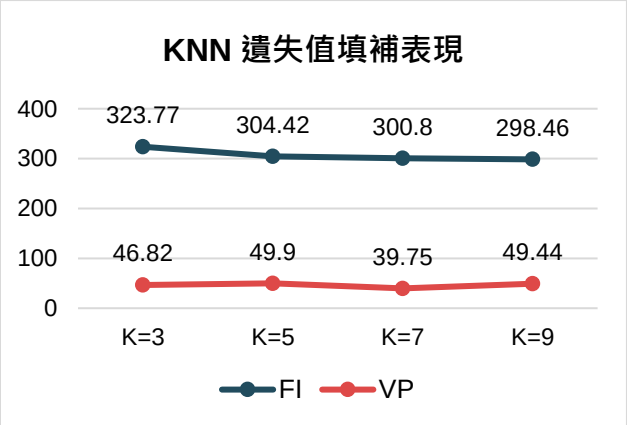
因此，在表格(三)中所呈現的數據皆使用 XGBRegressor 作為預測模型。

表格(三) KNN 填補遺失值比較

KNN	K=3		K=5		K=7		K=9	
History / Dataset	FI	VP	FI	VP	FI	VP	FI	VP
Data 1	789.20	108.02	695.58	98.88	697.43	98.37	714.73	102.22
Data 2	294.53	18.37	259.83	6.50	237.98	6.45	215.87	11.07
Data 3	392.14	49.39	386.75	51.93	382.13	50.50	371.34	51.74
Data 4	0.41	0.05	0.39	0.05	0.38	0.05	0.38	0.05
Data 5	759.19	61.39	758.25	60.9	757.70	58.32	757.42	57.97
Data 6	30.80	0.50	30.02	0.50	29.82	0.51	29.32	0.51
Data 7	0.10	90.01	0.15	130.49	0.17	64.04	0.18	122.50
Average	323.77	46.82	304.42	49.90	300.80	39.75	298.46	49.44

【註：各組中，平均表現最差的 FI、VP 使用紅色標示，表現最佳的則使用綠色標示】

由右圖所示，隨著 K 值增大，遺失值填補的表現也越來越好，但從 K=5 後就表現就逐漸趨緩；若是更加仔細地檢視各 dataset，可以發現相較於筆數較少的資料集(如：Data 2、Data 3)，筆數較多的資料集(如：Data 4、Data 5)對於不同 K 值所造成的遺漏值填補差異小上許多。



至於模型預測的準確率，從右圖可以發現於 K=7 時表現最好，同時 K=7 時填補遺漏值表現也為最佳，但是當從 K=3 至 K=5 的變化中和 K=9 的數據來看，遺漏值填補雖然更加準確，但預測準確率反而卻下降。因此，我們推得在 KNN 的遺失值填補方法中，遺失值與預測準確度兩者間並無直接的因果關係。

MICE 比較

針對 MICE 遺失值填補，我們嘗試了兩種方式，一個是 iteration 次數為 10，另一個是 100。會想要嘗試這兩種方法的主要原因是因為 MICE 是透過多次重複的補值，以達到回歸線趨於一致。若是 iteration 次數過少，可能導致回歸線沒有趨於一致。以下表格的結果使用的模型皆為 LGBMRegressor(learning_rate=0.01, n_estimators=550)。

表格(四) MICE 填補遺失值比較

	Max_iter=10		Max_iter=100	
	FI	**VP**	**FI**	**VP**
Data 1	863.16	150.89	849.74	113.76
Data 2	63.66	9.64	63.66	8.86
Data 3	351.52	48.01	527.94	52.06
Data 4	0.25	0.05	0.25	0.05
Data 5	20.42	39.41	20.42	39.46
Data 6	25.48	0.50	25.34	0.53
Data 7	0.09	48.99	0.10	78.94
Average	189.23	42.50	212.49	41.95

【註：兩組平均表現較差的 FI、VP 使用紅色標示】

從表格(四)可以發現，雖然 Max_iter 調到 100 後平均 VP 有略為下降，但是平均 FI 的部分反而上升許多。仔細地查看各資料及的數據後，發現雖然有些資料集的 FI 有下降，但是像 Data 3 在 Max_iter=100 的情況下，FI 卻大於 Max_iter=10 的。思考背後原理後，發現隨著 MICE iteration 次數的增加，確實會趨於一致，但是會圍繞著某個定值上下波動，所以才會發生就算 iteration 次數多，FI 仍可能會略大於 iteration 次數少的情況。由此可知，使用 MICE 不一定需要把 iteration 調那麼高，而是可以視資料集而定，聽常調到 20~30 次就會趨於一致了；當 iteration 次數設定太高時，不僅會增加計算量，模型表現也不一定會更好。

Train 和 Test 分開與合併比較

我們在這次作業中，也比較了當 train 和 test 分開填補遺失值，與 train 和 test 合併填補遺失值時，哪一個方法的效果會比較好，結果如下。

為控制變因，以下使用的兩種遺失值填補方法皆為 MICE(max_iter=100)，預測模型皆為 LGBMRegressor(learning_rate=0.01, n_estimators=550)。

表格(五) Train 和 Test 分開與合併比較

History / Dataset	Train 和 Test 分開		Train 和 Test 合併	
	FI	**VP**	**FI**	**VP**
Data 1	849.74	113.76	787.58	116.11
Data 2	63.66	8.86	70.12	37.00
Data 3	527.94	52.06	320.50	70.57
Data 4	0.25	0.05	0.25	0.05
Data 5	20.42	39.46	20.56	37.92
Data 6	25.34	0.53	24.81	0.47
Data 7	0.10	78.94	0.11	74.35
Average	212.49	41.95	174.85	48.07

【註：兩組平均表現較差的 FI、VP 使用紅色標示】

從表格 (五) 中可以發現，在同樣的遺失值填補和預測模型下，雖然在平均 VP 表現上，train 和 test 分開填補，表現比 train 和 test 填補合併還好，但是在平均 FI 的表現上，train 和 test 分開填補卻，表現卻遠差於 train 和 test 合併。尤其在 data 1 和 data 3 中，FI 的下降程度最大。我們推測這跟這兩個資料 train 資料集的觀測值數量有關。data 1 和 data 3 的 train 資料集觀測值數分別有 765 和 371 筆，相對起 data 4 和 data 5 的上千筆觀測值相比少了許多。藉由 train 和 test 資料集的合併，增加資料筆數使 MICE 訓練更完全，確實可以讓原先資料筆數較少的 data 在遺失值填補上有更好的效果。

壞模型檢討

Neural Network : 以下各點是我們在使用 Neural Network 時，嘗試的各種方法，但是最後模型表現都不盡理想。

a. Drop out : 在每個隱藏層後面加上 drop out layer ，使 NN 模型可以去除部分 neuron 之間的連結，使每個 neuron 能各自學習，減少雜訊。我們把 drop out rate 調成 0.3、0.4 和 0.5。但是最後模型預測表現都很差。推測是因為有些資料集本身資料很少，本身資訊就不多了，若再進行 drop out 的話，反而會導致 under fitting。
b. BatchNormalization : 在每個隱藏層前加入 BatchNormalization layer，對每個 batch 進行標準化，但是效果也同樣不好。推測可能是因為設定的 batch size 不夠大，導致標準化後得出的平均數和變異數不夠 robust。

Best Performed Methods

因為每個資料集的特性皆不盡相同，所以每個資料集表現最好的模型都不一樣。由於這次作業的目的是要訓練出一個泛化的模型，因此這邊我們就不列出各個資料集表現最好的模型。
而整體表現最好的模型是使用 LGBMImputer 對遺漏值進行填補，再使用 LGBMRegressor 進行預測；7 個資料集的平均 FI 為 123.06，平均 VP 為 43.42。這個模型表現最好的最大原因是遺失值填補方法是我們試的所有方法裡最成功的，平均 FI 為所有模型裡最低的，先前試的遺失值填補方法不論預測模型再怎麼調參數，得出的變現都無法與其比擬，直到將遺失值填補方法換成 LGBMImputer 才有所進展，因此也可以從這裡發現遺失值填補的重要性。

4) Conclusions

Summarize the findings

- 根據我們做出的成果，遺失值的填補方法效果最好的是 LGBMImputer ，其次是 MICE，最後才是 KNN。
- 預測模型的部分，LGBMRegressor 和 XGBoostRegressor 做出的成效其實差不多，但是 LGBMRegressor 略勝一籌，且計算速度更快。比較令人意外的是 Neural Network 的

效果差強人意。原先以為運用深度學習，表現應該會勝於機器學習模型，但是結果竟然比較差，推測可能是因為這次有些資料集的資料量比較少，使神經網路無法發揮它的長處。

- 針對資料數較少的資料集，可以將 train 和 test 合併一起進行補值，可以讓遺失值補值的模型訓練效果更好。
- MICE 並不是我們想像中 iteration 次數越大，就會更趨於一致。通常 iteration 到某個次數會就會在某個定值上下波動，因此把 iteration 次數刻意用很大也不會提升效果，只會徒增計算量而已。
- KNN 補值法在資料筆數足夠時，K 值影響不大；而在我們測試的資料集中，遺漏值填補雖然更加準確，但與預測準確率無直接相關。

Point out how to improve in the future

我們在進行模型預測時，把所有的資料的特徵視為連續型變數，並使用 StandardScaler 進行資料前處理，最後我們發現儘管 7 個資料集的特徵都是數字，但有些特徵可能只有三種數字的表達方式，應該要視為類別型資料並用 one-hot encoding 等其他方式進行處理。這個發現讓我們審視到 Explorative data analysis 的重要性，不能盲目的套用方法，而是要根據資料的特性進行相對應的處理；未來有機會的話應該要先判斷資料集哪些變數是連續型變數、哪些是類別型變數，再分別以 StandardScaler 和 one-hot encoding 處理，相信這種處理方式所得出的效果一定會比原本來的好。

5) 作業心得

恩兆：在這次的作業中，很開心能以競賽得方式來讓我們清楚且充分地練習了遺失值填補的各式方法，從一開始認為遺失值填補不是很重要，且只會使用平均值、中位數等最基礎的方法填補遺失值的我，到後來了解了 KNN、MICE 等較為進階的遺失值填補原理後，看著排行榜的名次逐步爬升，這帶給我了很大的成就感，也讓我更有動力去找尋更多更好的遺失值填補方法。雖然途中嘗試了許多套件，都沒有成功，像是照著助教課中使用 GRAPE 套件失敗了數次，但也另闢蹊徑，找到了 LGBMImputer 的方法填補遺失值，並且出乎意料地獲得了滿不錯的成績。這次的作業小競賽，不但讓我學習到了許多以前未曾接觸過的遺失值填補方法，更是能從交出去的成果

檢視不同資料集適合甚麼填補方法，又該搭配什麼模型；當然，這是個範疇極大的議題，但我相信能透過這次練習的經驗，幫助未來在學習新的遺失值填補方法時能夠更快上手、獲得更多洞見。

詠翰：這次的 HW3 遺失值處理讓我收穫滿滿，原本我打算用助教課所講的 GRAPE 當作 HW3 的模型。但可惜的是 GRAPE 的原本使用方式是把完整的資料自行選擇部分的值做覆蓋，而不是像 HW3 給的那七個資料集本來就是不完整的資料。總而言之，沒辦法直接套用 HW3 資料集進行填補遺失值和預測。因此後來我們在轉而用 MICE 與其他填補遺失值的方法並選擇其他模型做預測，很感謝這次我和我的隊友們花了不少時間嘗試不同的模型，總共我們上傳了超過 100 次，雖然效果不是所有隊伍裡面最好的，但已經很滿意了。

皓昀 : 以前對於遺失值的填補都沒有甚麼概念，常常不是填補平均數和中位數，不然就是把有遺失值的觀測值刪除。但是經過這次的競賽，我學到更多其他更有效的遺失值填補方法，包括使用機器學習的 KNN、MICE ，甚至還有深度學習的 GRAPE。這次作業最困難的大概就是要同時處理 7 個資料，迫使我們要找一個泛化的模型。這次這次我負責使用 MICE 的方法填補，雖然效果不錯，但是到最後仍然沒辦法用到更好。我認為我們如果有使用 GRAPE 的話，一定能再更進步，可惜最終仍然沒有成功理解 GRAPE 的使用方法。很高興能透過這次的競賽磨練自己，希望以後遇到同樣的問題能夠更得心應手。

6) 繳交檔案註明

這次繳交的程式碼有兩份，其主要內容如檔名所述。