團隊測驗報告

報名序號:112096 (報名序號(格式:112XXX)已寄至隊長email)

團隊名稱: Brute Force Buddy

註1:請用本PowerPoint 文件撰寫團隊程式說明, **請轉成PDF檔案繳交**。
註2:依據競賽須知第七條, 第4項規定:

測試報告之簡報資料不得出現企業、學校系所標誌、提及企業名稱、學校系所、教授姓名及任何可供辨識參賽團隊組織或個人身分的資料或資訊, 違者取消參賽資格或由評審會議決議處理方式。

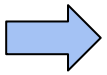
一、資料前處理(說明資料前處理過程)

我們欲將anomaly_train.csv、cooler.csv、power.csv裡頭的資料合併成一個大表格，以利後續的分析。

我們先瀏覽過anomaly_train.csv中的每筆資料，發現有date為2055年的錯誤，將他改為2022。

由於原先的資料在lamp_id欄位中，會出現一列有2或3個lamp在同個儲存格的情況，因此，我們**運用python的.split()函式把每筆資料的lamp_id以_做為分界，使每lamp成為每一筆資料的基本單位**，並重新整理表格。如下：

	date	oven_id	layer_id	lamp_id	anomaly_accumulation_hour	anomaly_total_number
0	2021/12/27	1B0	5	26_49	5116	2
1	2021/12/27	1C0	3	45_91	4699	2
2	2021/12/27	1D0	14	64	3241	1
3	2021/12/27	1.00E+00	1	96	4138	1
4	2021/12/27	1.00E+00	8	51	3818	1
...	...	...	...	...	...	...
1442	2023/2/4	2C0	7	72_80_100	10146	3
1443	2023/2/4	2C0	19	18_23_31	9454	3
1444	2023/2/6	2B0	15	67_90_92	8276	3
1445	2023/2/6	2B0	16	94	9507	1
1446	2023/2/6	2C0	12	4_41_79	9795	3



	date	oven_id	layer_id	lamp_id	anomaly_accumulation_hour	anomaly_accumulation_hour_divided
0	2021/12/27	1B0	5	26	5116	2558.000000
1	2021/12/27	1B0	5	49	5116	2558.000000
2	2021/12/27	1C0	3	45	4699	2349.500000
3	2021/12/27	1C0	3	91	4699	2349.500000
4	2021/12/27	1D0	14	64	3241	3241.000000
...	...	...	...	...	...	...
3399	2023/2/6	2B0	15	92	8276	2758.666667
3400	2023/2/6	2B0	16	94	9507	9507.000000
3401	2023/2/6	2C0	12	4	9795	3265.000000
3402	2023/2/6	2C0	12	41	9795	3265.000000
3403	2023/2/6	2C0	12	79	9795	3265.000000

一、資料前處理(說明資料前處理過程)

將cooler.csv的資料，分別配對到anomaly_train.csv的layer_id和oven_id，整理成一張大表格，也就是在anomaly_train.csv新增上下層進水流量(water_volumne_lower和water_volumne_upper)、水冷板迴水溫度(in_temp和out_temp)、水冷板A點B點溫度(a_temp和b_temp)，6個feature

	date	oven_id	layer_id	lamp_id	anomaly_accumulation_hour	water_volume_lower	water_volume_upper	in_temp	out_temp	a_temp	b_temp
0	2021/12/27	1B0	5	26	5116	11.0	11.0	20.3	24.8	24.2	25.2
1	2021/12/27	1B0	5	49	5116	11.0	11.0	20.3	24.8	24.2	25.2
2	2021/12/27	1C0	3	45	4699	11.0	11.0	22.8	25.5	23.1	23.0
3	2021/12/27	1C0	3	91	4699	11.0	11.0	22.8	25.5	23.1	23.0
4	2021/12/27	1D0	14	64	3241	11.0	11.0	19.5	23.6	22.9	23.8
...	...	...	...	...	...	...	...	...	...	...	...
3399	2023/2/6	2B0	15	92	8276	11.0	11.0	21.0	24.5	22.6	22.6
3400	2023/2/6	2B0	16	94	9507	11.0	11.0	21.0	24.5	22.6	22.6
3401	2023/2/6	2C0	12	4	9795	10.0	10.0	19.5	25.1	23.5	25.3
3402	2023/2/6	2C0	12	41	9795	10.0	10.0	19.5	25.1	23.5	25.3
3403	2023/2/6	2C0	12	79	9795	10.0	10.0	19.5	25.1	23.5	25.3

3404 rows × 11 columns

一、資料前處理(說明資料前處理過程)

接著我們**以一週為單位合併每一列, 以一週的單位去看一週在某爐某層總共壞了幾支燈管** ，並新增time_index、start和end欄位，分別代表週次、該週開始時間和該週結束時間。

因此我們將**刪除lamp_id並改以壞掉燈管支數來呈現（total_count）**，此外，我們觀察到power裡頭有部分層數的設定是不同的，因此我們也把total_count欄位做細分，**增加count和special_count兩欄**，後者為燈管編號為 1,2,60,61,62,63,121,122壞掉的支數，前者則是其他燈管壞掉的支數。

若該爐該曾在一週 內有壞掉多次, 則合併後的 accumulation_hour將取該週該爐該層的異常紀錄的最大 值 。完成後則輸出資料

	time_index	start	end	oven_id	layer_id	accumulation_hour	count	special_count	water_volume_lower	water_volume_upper	in_temp	out_temp	a_temp	b_temp	total_count
0	0	21/12/27	22/01/03	1B0	5	5116	2	0	11.0	11.0	20.3	24.8	24.2	25.2	2
1	0	21/12/27	22/01/03	1C0	3	4699	2	0	11.0	11.0	22.8	25.5	23.1	23.0	2
2	0	21/12/27	22/01/03	1D0	14	3241	1	0	11.0	11.0	19.5	23.6	22.9	23.8	1
3	0	21/12/27	22/01/03	1E0	1	4138	1	0	12.0	12.0	19.7	24.0	24.0	24.0	1
4	0	21/12/27	22/01/03	1E0	8	3818	1	0	12.0	11.0	19.7	24.0	24.0	24.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1313	45	23/2/1	23/02/08	2B0	19	9695	3	0	11.0	11.0	21.0	24.5	22.6	22.6	3
1314	45	23/2/1	23/02/08	2C0	7	10146	3	0	11.0	11.0	19.5	24.4	23.5	25.3	3
1315	45	23/2/1	23/02/08	2C0	19	9454	3	0	10.0	10.0	19.5	25.1	23.5	25.3	3
1316	45	23/2/1	23/02/08	2B0	15	8276	3	0	11.0	11.0	21.0	24.5	22.6	22.6	3
1317	45	23/2/1	23/02/08	2C0	12	9795	3	0	10.0	10.0	19.5	25.1	23.5	25.3	3

1318 rows × 15 columns

二、演算法和模型介紹(介紹方法細節):多模型ensemble

我們採用**Ensemble**的方法, 使用了四種模型, 包含Linear Regression, SARIMA(時間序列), Random Forest, Decision Tree

最後根據4個模型做驗證時輸出的結果, 來選擇預測方法

二、演算法和模型介紹(介紹方法細節):Linear Regression

欲透過accumlation_hour.csv中的特徵，來預測在某月時，壞掉的燈管支數為多少。

產線一和產線二分別有獨立的Regression model各自訓練，並以accumulation_hour.csv為主要的資料集，將前處理ouput_new.csv中的time_index轉換成絕對時間後，加入accumlation_hour.csv中，並根據accumlation_hour.csv中的date來往前推1個月，結合前處理過的資料來計算該月的壞掉支數，作為訓練資料集的y。

資料集:

- Attributes為layer_id(層別), accumulation_hour(某爐某層累計使用時間), oven_num(爐別), time_index(當天的絕對時間)
- Answer為count (壞掉的支數)

根據我們在做資料分析時的觀察(如第四章、補充說明)，我們發現G0的爐只有極微稀少的異常次數，因此我們**訂定1G0和2G0的預測結果應為0，若模型預測這兩爐的結果不為0，則表示模型出現bias，因此預測的結果會再減去G0的預測結果，來矯正模型的偏差**

二、演算法和模型介紹(介紹方法細節):Linear Regression

產線一之訓練資料(4, 5, 6 ,7月):

	date	oven_id	layer_id	accumulation_hour	oven_num	count	time_index
0	2022/5/4	1B0	1	7731	0.0	3.0	1651622400
1	2022/5/4	1B0	2	6388	0.0	2.0	1651622400
2	2022/5/4	1B0	3	7792	0.0	2.0	1651622400
3	2022/5/4	1B0	4	6942	0.0	4.0	1651622400
4	2022/5/4	1B0	5	7361	0.0	3.0	1651622400
...	...	...	...	...	...	...	...
375	2022/7/30	1G0	15	235	4.0	0.0	1659139200
376	2022/7/30	1G0	16	427	4.0	0.0	1659139200
377	2022/7/30	1G0	17	377	4.0	0.0	1659139200
378	2022/7/30	1G0	18	346	4.0	0.0	1659139200
379	2022/7/30	1G0	19	771	4.0	0.0	1659139200

380 rows × 7 columns

產線一之驗證資料集(8月):

	date	oven_id	layer_id	accumulation_hour	oven_num	time_index
0	2022/8/15	1B0	1	9584	0.0	1661817600
1	2022/8/15	1B0	2	8328	0.0	1661817600
2	2022/8/15	1B0	3	9621	0.0	1661817600
3	2022/8/15	1B0	4	8624	0.0	1661817600
4	2022/8/15	1B0	5	9232	0.0	1661817600
...	...	...	...	...	...	...
90	2022/8/15	1G0	15	630	4.0	1661817600
91	2022/8/15	1G0	16	819	4.0	1661817600
92	2022/8/15	1G0	17	772	4.0	1661817600
93	2022/8/15	1G0	18	734	4.0	1661817600
94	2022/8/15	1G0	19	1131	4.0	1661817600

95 rows × 6 columns

產線一之預測資料集(9月):

	date	oven_id	layer_id	accumulation_hour	oven_num	time_index
0	2022/9/1	1G0	1	5840	4.0	1664409600
1	2022/9/1	1G0	2	5679	4.0	1664409600
2	2022/9/1	1G0	3	5567	4.0	1664409600
3	2022/9/1	1G0	4	5379	4.0	1664409600
4	2022/9/1	1G0	5	5118	4.0	1664409600
5	2022/9/1	1G0	6	4859	4.0	1664409600
6	2022/9/1	1G0	7	4533	4.0	1664409600
7	2022/9/1	1G0	8	4227	4.0	1664409600
8	2022/9/1	1G0	9	3868	4.0	1664409600
9	2022/9/1	1G0	10	3510	4.0	1664409600
10	2022/9/1	1G0	11	3268	4.0	1664409600
11	2022/9/1	1G0	12	2876	4.0	1664409600
12	2022/9/1	1G0	13	3119	4.0	1664409600
13	2022/9/1	1G0	14	2350	4.0	1664409600
14	2022/9/1	1G0	15	943	4.0	1664409600
15	2022/9/1	1G0	16	1132	4.0	1664409600
16	2022/9/1	1G0	17	1084	4.0	1664409600
17	2022/9/1	1G0	18	1046	4.0	1664409600
18	2022/9/1	1G0	19	1443	4.0	1664409600

產線二之訓練資料(4~12月):

	date	oven_id	layer_id	accumulation_hour	oven_num	count	time_index
0	2022/5/4	2B0	1	5835	0.0	0.0	1651622400
1	2022/5/4	2B0	2	5833	0.0	1.0	1651622400
2	2022/5/4	2B0	3	5849	0.0	2.0	1651622400
3	2022/5/4	2B0	4	5762	0.0	1.0	1651622400
4	2022/5/4	2B0	5	5808	0.0	0.0	1651622400
...	...	...	...	...	...	...	...
850	2022/11/28	2G0	15	1088	4.0	0.0	1669593600
851	2022/11/28	2G0	16	926	4.0	0.0	1669593600
852	2022/11/28	2G0	17	769	4.0	0.0	1669593600
853	2022/11/28	2G0	18	652	4.0	0.0	1669593600
854	2022/11/28	2G0	19	506	4.0	0.0	1669593600

855 rows × 7 columns

產線二之驗證資料集(2023.1月):

	date	oven_id	layer_id	accumulation_hour	oven_num	time_index
0	2022/12/28	2B0	1	906	0.0	1674864000
1	2022/12/28	2B0	2	905	0.0	1674864000
2	2022/12/28	2B0	3	905	0.0	1674864000
3	2022/12/28	2B0	4	9907	0.0	1674864000
4	2022/12/28	2B0	5	744	0.0	1674864000
...	...	...	...	...	...	...
90	2022/12/28	2G0	15	2042	4.0	1674864000
91	2022/12/28	2G0	16	1866	4.0	1674864000
92	2022/12/28	2G0	17	1686	4.0	1674864000
93	2022/12/28	2G0	18	1554	4.0	1674864000
94	2022/12/28	2G0	19	1392	4.0	1674864000

95 rows × 6 columns

產線二之預測資料集(2023.2月):

	date	oven_id	layer_id	accumulation_hour	oven_num	time_index
0	2023/2/6	2G0	1	1940	4.0	1675641600
1	2023/2/6	2G0	2	970	4.0	1675641600
2	2023/2/6	2G0	3	1357	4.0	1675641600
3	2023/2/6	2G0	4	3464	4.0	1675641600
4	2023/2/6	2G0	5	1943	4.0	1675641600
5	2023/2/6	2G0	6	3116	4.0	1675641600
6	2023/2/6	2G0	7	3152	4.0	1675641600
7	2023/2/6	2G0	8	2383	4.0	1675641600
8	2023/2/6	2G0	9	2783	4.0	1675641600
9	2023/2/6	2G0	10	2697	4.0	1675641600
10	2023/2/6	2G0	11	2521	4.0	1675641600
11	2023/2/6	2G0	12	1676	4.0	1675641600
12	2023/2/6	2G0	13	2133	4.0	1675641600
13	2023/2/6	2G0	14	1877	4.0	1675641600
14	2023/2/6	2G0	15	1629	4.0	1675641600
15	2023/2/6	2G0	16	1441	4.0	1675641600
16	2023/2/6	2G0	17	1200	4.0	1675641600
17	2023/2/6	2G0	18	1050	4.0	1675641600
18	2023/2/6	2G0	19	847	4.0	1675641600

二、演算法和模型介紹(介紹方法細節):SARIMA

輸入 : time index (也就是從開始記錄異常燈管的第幾個禮拜)
輸出 : 當個禮拜那個爐發生異常燈管的數量, 若沒有則為0

SARIMA模型的簡單介紹:

1. **季節性(Seasonal)**(參數s):SARIMA模型考慮了數據中的季節性模式, 這是指數據具有重複的周期性變化, 例如每年、每季度、每月等。季節性模式通常是時間序列數據中的重要特徵。

2. **自回歸(Autoregressive)**(參數p):SARIMA中的"AR"部分表示自回歸, 這意味著模型使用過去時間點的觀測值來預測未來的值。AR部分描述了時間序列數據中的自相關性。

3. **整合(Integrated)**(參數d):SARIMA中的"I"部分表示整合, 這表示對時間序列數據進行了差分操作, 以使數據變得穩定。差分操作的次數稱為差分階數, 它可以幫助處理數據中的趨勢。

4. **移動平均(Moving Average)**(參數q):SARIMA中的"MA"部分表示移動平均, 這意味著模型使用過去時間點的誤差來預測未來的值。MA部分描述了時間序列數據中的隨機性。

根據上面四個參數, 在交叉驗證下R-square表現最好的參數當作我的最佳預測模型。

二、演算法和模型介紹(介紹方法細節):SARIMA

如圖所示, 我根據每組不同的參數去訓練不同的訓練集, 並算出在不同驗證集上的R square後做平均, 並挑選R square最好的參數。

圖片來源:https://segmentfault.com/a/1190000043266169

二、演算法和模型介紹(介紹方法細節)：Random Forest / Decision Tree Classifier

簡介：

Train Data 的異常數量為常態分布，以2、3 為大宗。因此在資料數量足夠，並且假設當月各層級都有燈管損壞的情況下，以一個月為時間區間進行分類判斷，最後進行加總即可預測出當月各爐可能的損壞數量。總得來說，**此方法不看各別層級的正確性，只看整個爐加總的正確性**

建模方式：

考量到 G 爐的資料數量不足，不利於分類判斷，因此將同一產線每個爐的資料整併後再建模。一條產線一個模型，共兩個模型。

二、演算法和模型介紹(介紹方法細節)：Random Forest / Decision Tree Classifier

輸入欄位：9 欄，含爐、層編號、累計使用時數、水冷板進水流量、溫度等設定參數

具體欄位名稱：oven_id、layer_id、accumulation_hour、water_volume_lower、water_volume_upper、in_temp、out_temp、a_temp、b_temp

Train Data：該產線除了最後一個月之外的所有資料（2022/8、2023/1~）

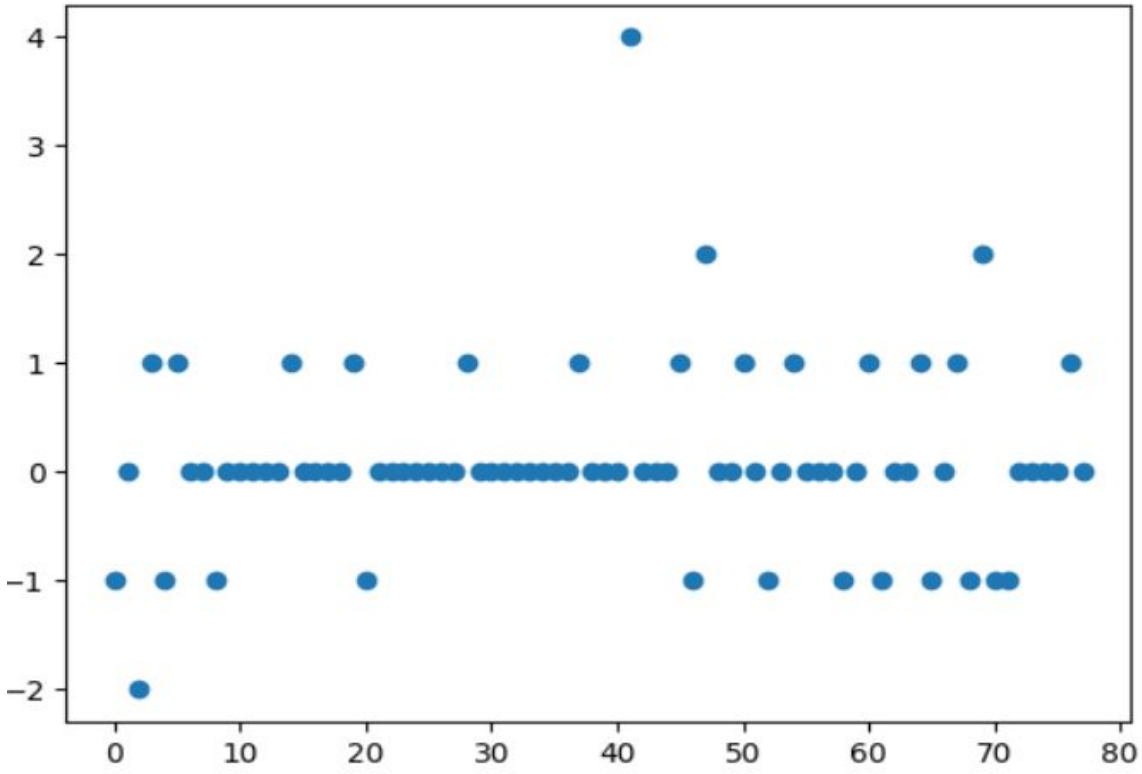
Valid Data：該產線最後一個月的資料。一律將時數加至當月月底（2022/8/31、2023/1/31），確保跟 Test Data 性質一致

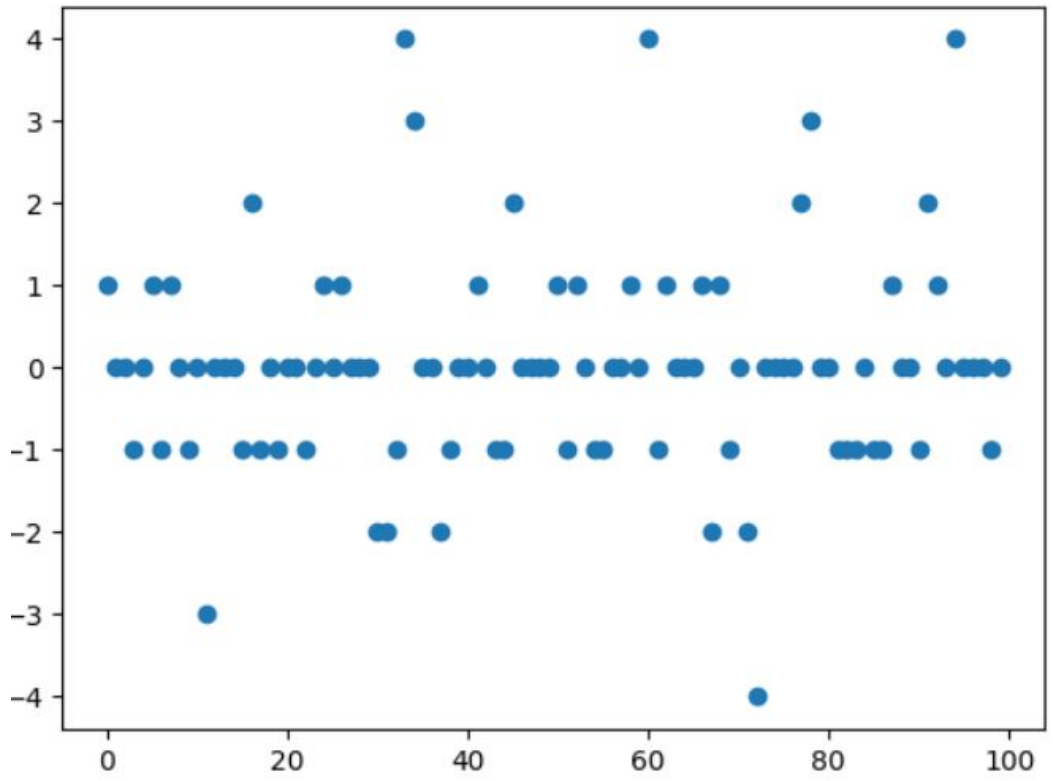
Test Data：當月月底各爐層各層別的狀態（2022/9/28、2023/2/28）。透過各爐層累積使用時數（accumulation_hour.csv）進行加總，獲得指定時間的累計時數以及進水流量、溫度等資訊。

二、演算法和模型介紹(介紹方法細節)：Random Forest / Decision Tree Classifier

Random Forest Valid Data 預測結果差距散佈圖

X 軸：資料編號；Y 軸：預測結果與正確答案之間的差距

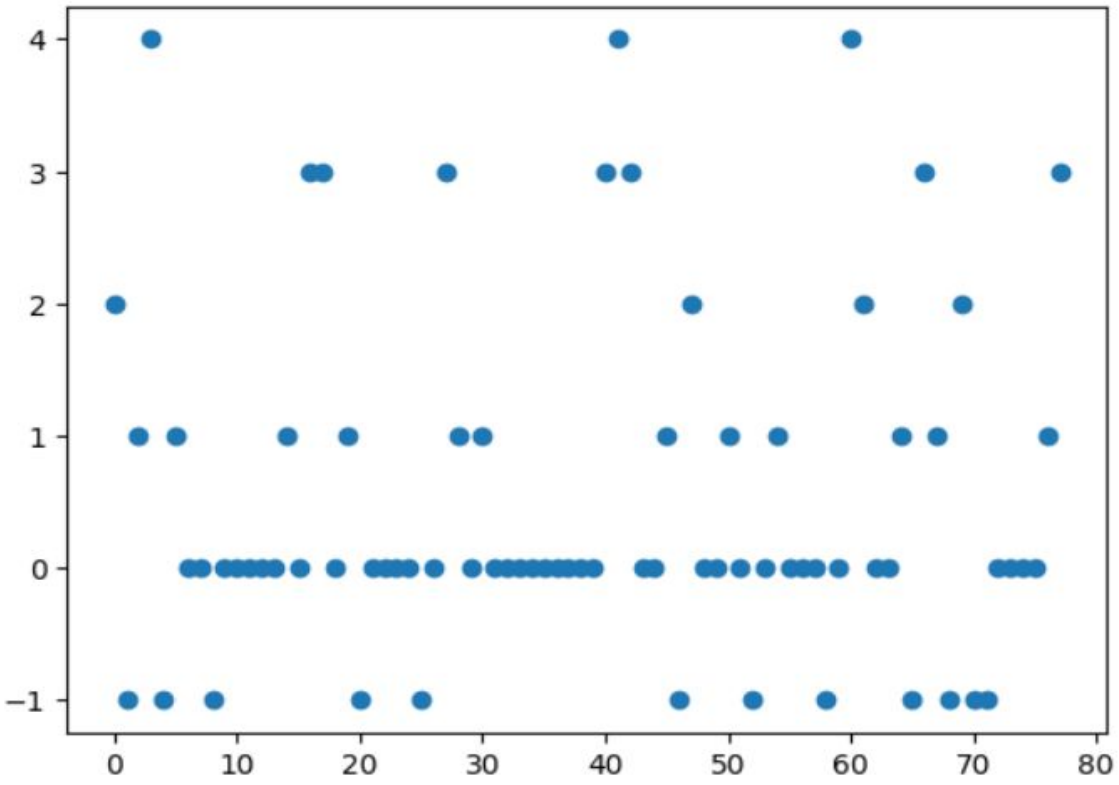


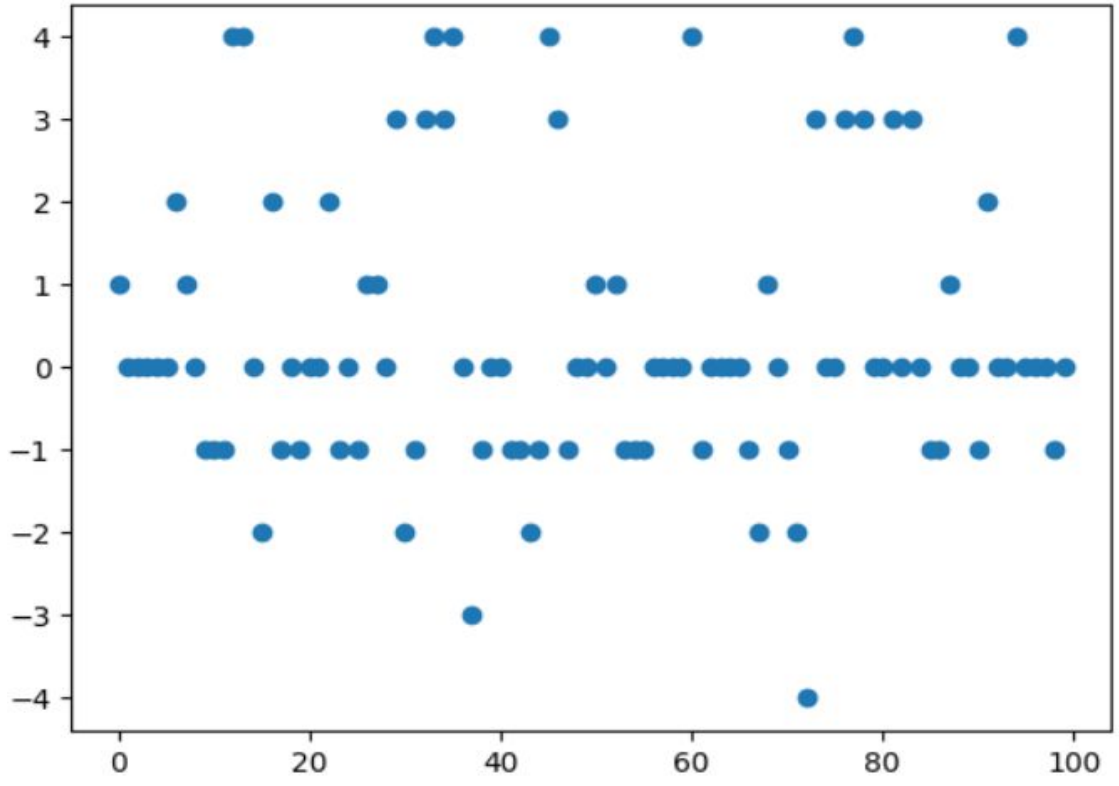


二、演算法和模型介紹(介紹方法細節)：Random Forest / Decision Tree Classifier

Decision Tree Valid Data 預測結果差距散佈圖

X 軸：資料編號；Y 軸：預測結果與正確答案之間的差距





三、預測結果:產線一

各模型以2022年8月為驗證資料來預測各爐的壞掉支數之結果
將所有模型之驗證結果與8月真實資料算出Absoulte Error
Absolute Error最小者，將被標記為藍色，
意味者要以該模型預測9月的結果為最後正確答案

	Random Forest	Decision Tree	SARIMA	Linear Regression	Answer
1B0	62	63	31.325	86	64
1C0	47	74	69.577	63	41
1D0	50	50	74.640	48	47
1E0	42	49	48.091	29	42
1G0	0	0	未訓練	0	0

	Random Forest	Decision Tree	SARIMA	Linear Regression	**最終提交之答案**
1B0	39	49	-6.349	86	49
1C0	39	60	68.200	62	39
1D0	43	58	77.220	48	48
1E0	40	58	38.454	29	40
1G0	36	40	未訓練	0	0

三、預測結果:產線二

各模型以2022年8月為驗證資料來預測各爐的壞掉支數之結果
將所有模型之驗證結果與8月真實資料算出Absoulte Error
Absolute Error最小者，將被標記為藍色，
意味者要以該模型預測9月的結果為最後正確答案

	Random Forest	Decision Tree	SARIMA	Linear Regression	Answer
2B0	84	66	92.216	77	80
2C0	65	75	72.955	69	73
2D0	62	65	63.231	60	67
2E0	59	64	55.866	36	62
2G0	0	0	未訓練	0	0

	Random Forest	Decision Tree	SARIMA	Linear Regression	**最終提交之答案**
2B0	55	39	91.033	66	66
2C0	56	52	69.982	65	70 (四捨五入)
2D0	56	51	59.588	61	51
2E0	49	48	56.410	37	48
2G0	57	39	未訓練	0	0

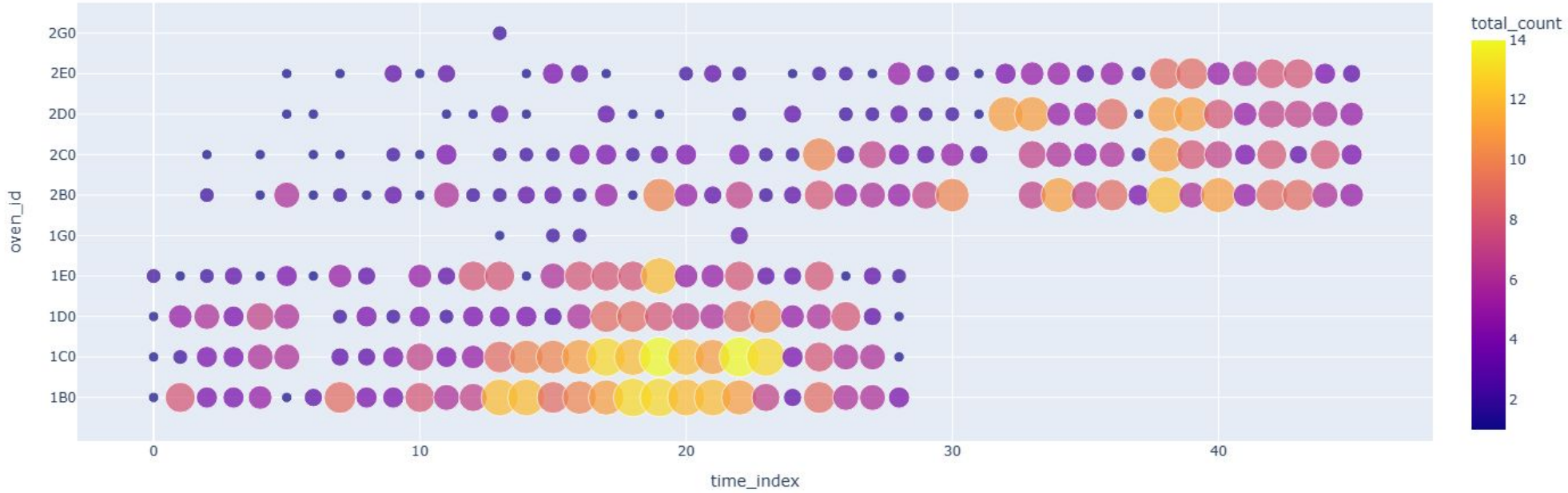
三、預測結果:最後提交之預測結果

鍋爐	預測答案
1B0	**49**
1C0	**39**
1D0	**48**
1E0	**40**
1G0	**0**
2B0	**66**
2C0	**70**
2D0	**51**
2E0	**48**
2G0	**0**

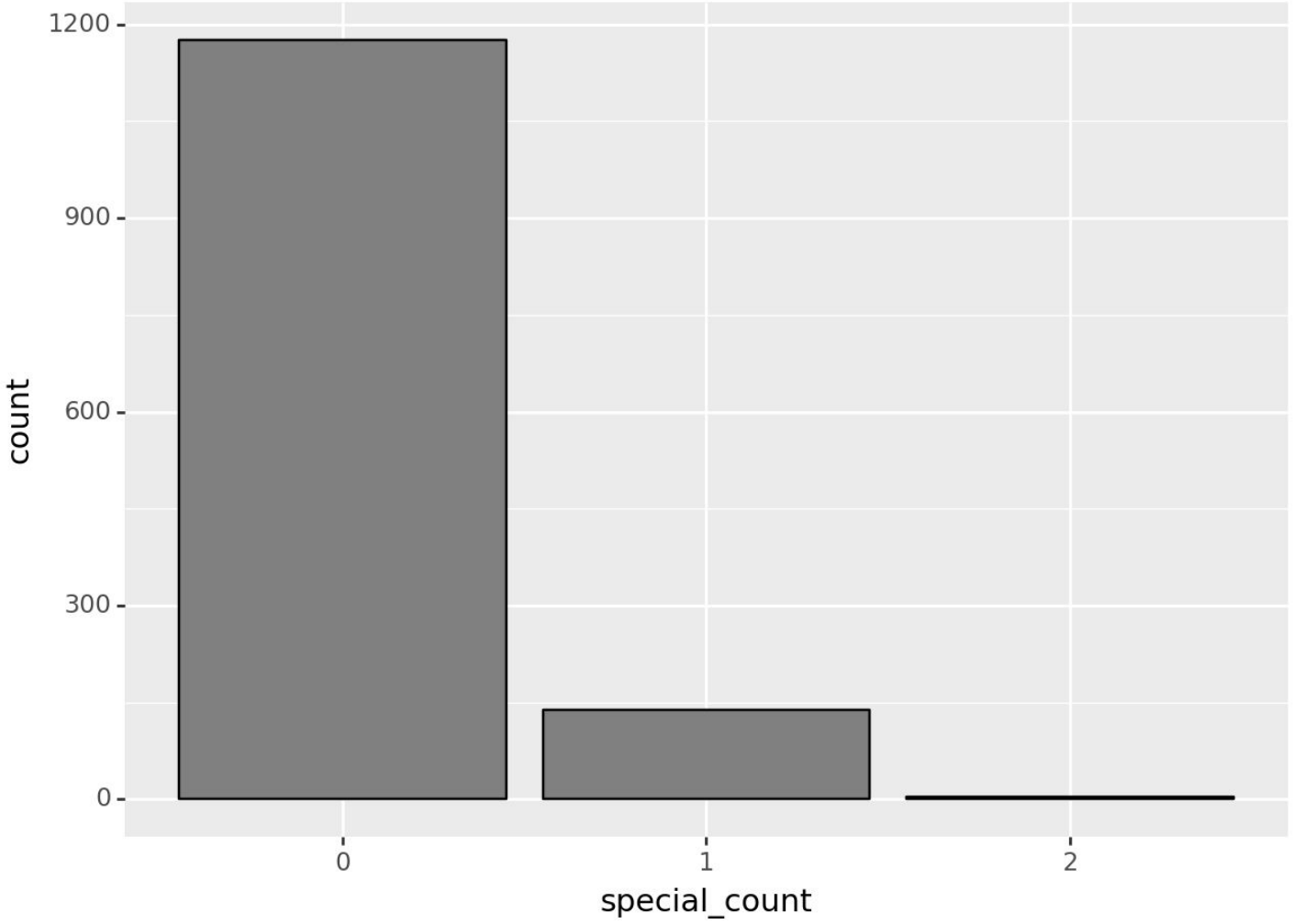
四、補充說明:資料視覺化 & 整體分析

time_index: 根據anomaly_train當中有紀錄的時間為基礎, 以一週為單位劃分出46個時間區段
下圖呈現各個鍋爐在不同時間區段中的異常燈管數量(圓圈的大小愈大且顏色愈亮代表異常數量愈多)
發現:

1. 產線一與產線二皆發生在某段時間中異常燈管數量有先上升後下降的現象, 並在接近最終預測時間段的時候回復到普通水準
2. 相較於產線二, 產線一的異常燈管數量的上下限差距較大
3. 同產線的不同鍋爐之間雖然距具有類似的異常變化趨勢, 但具體趨勢仍有差異
4. 1G0與2G0兩個鍋爐的異常數量十分稀少(作為Linear Regression模型的bias調整的基礎)



四、補充說明:資料視覺化 & 整體分析



處理:**將各個爐層的第1、2、60、61、62、63、121、122號燈管設定為「特殊燈管(special)」**, 他們位於爐層的邊角位置, 具有與其他位置不同的功率設定

左圖呈現爐層發生燈管異常時, 在這些燈管之中有多少是特殊燈管

根據結果, 大部分時候是非特殊燈管發生異常, 但仍有一小部分的異常事件包含了1根特殊燈管的異常狀況, 極少數為2根

後續:**將「單次異常狀況中, 特殊燈管的異常數量」作為變項加入到訓練過程中, 以得到更精準的預測結果**

四、補充說明:資料視覺化 & 整體分析

下圖呈現各個爐層發生單次燈管異常事件時, 異常燈管的總數量的分布

發現:**異常數量接近常態分布, 且集中於2根燈管、3根燈管為大宗**