員工會不會烙跑?

第九組

心理 黃琮祐
經濟 周柏翰
機械 吳岱桀
統計 謝宜均

目錄

壹-摘要

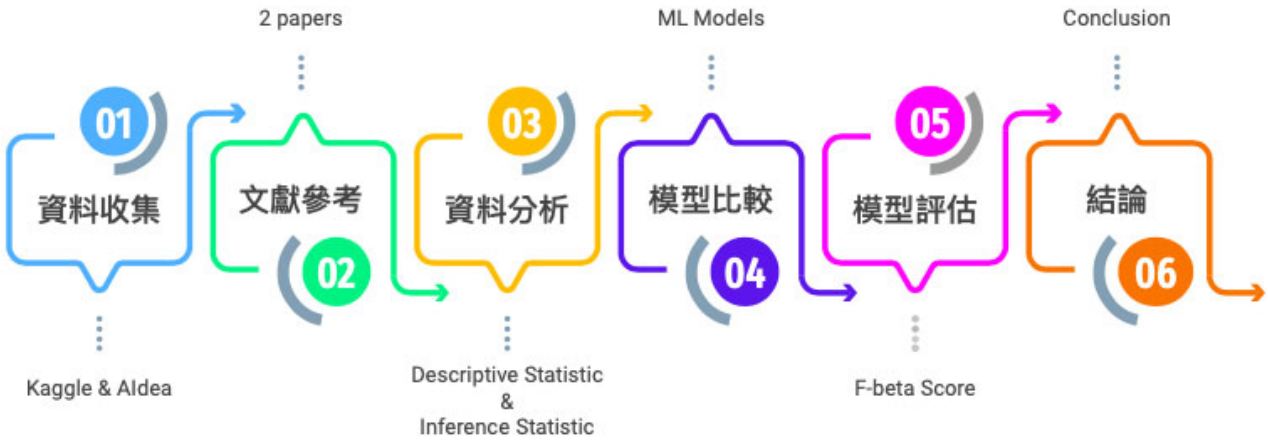
本報告使用了AIdea人工智慧共創平台上的一筆關於員工離職的資料，我們欲使用員工與工作單位的特徵去預測這位員工這一年會不會離職，經過缺失值的處理再做One hot encoding處理類別變數之後，將資料切成2014~2016年為train、2017為test。使用以下幾種方法：logistic Regression, Decision Tree, AdaBoost, KNN, Random Forest, KNN, Naive Bayes, QDA, Neural Network，然後以SMOTE套件解決離職樣本過少資料不平衡的問題，使用train資料訓練出能預測員工這一年是否會離職的模型，最後使用test資料以F beta-score指標評估每一個模型的表現。結果發現各模型普遍表現不佳，F beta-score最多只能達到0.19，經過ensemble機制能有更好的表現但不會有大幅度的優化，或許我們沒辦法以這筆資料準確預測出員工的離職傾向。

貳-選題動機

身為即將畢業的大學生，對於未來總有一定的想像，不論是繼續就讀碩士或是進入職場都是一個全新的挑戰。而對於職場的未知，我們心中多少都會帶有一點的憧憬跟忐忑。擔心自己會不會淪為只能月領22K的草莓族，擔心老闆會不會刁難自己等等，也會擔憂自己的能力能不能符合社會所要，希望自己不要成為被人挑的柿子。

而事實上，員工離職的因素百百種。身為企業決策者，總是希望能廣納賢士壯大公司。若能夠清楚地掌握員工離職的原因，不但能加以改善留住人才，還能避免公司虧損甚至倒閉。而對於準畢業生的我們而言，也想在進入職場前，了解什麼樣的因素會使人想要烙跑，除了能幫助我們找到適合自己的職場元素，也可用做前車之鑑提醒自己，不要落入社會陷阱。

參- 分析流程



肆-問題敘述與目標

彼得·德魯克曾說：「用人不在於如何減少人的短處，而在於如何發揮人的長處。卓有成效的管理者善於用人之長。」

人才是企業最重要的資源之一，找到對的人很重要，但另一件重要的事是如何留住對的人。提早發現員工離職傾向並留任優秀人員，是企業持續成長的重要議題。此議題中蒐集了多個可能會影響員工離職的因素，如年齡層、績效、最高學歷、出差數、請假數...等。建立模型來分析員工未來是否會有離職的風險、預測未來員工是否會離職及早啟動留才管理機制。

目標：預測公司員工會不會辭職

- Step 1. 由10665筆Train資料建立模型(2014~2016年)
- Step 2. 用另外3654筆Test資料預測離職與否(2017年)
- Step 3. 自行計算 F beta score 與準確率
  並與其他隊伍的Leaderboard分數進行比較

評分方式：採用常用於評估二元分類問題模型好壞的F beta-score(β = 1.5)

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

$$F_\beta = \left(1+\beta^2\right)\cdot\frac{precision\cdot\ \text{recall}}{\left(\beta^2\cdot\ \text{precision}\right)+\ \text{recall}}$$

其中，$precision = \frac{true\ positives}{true\ positives + false\ positives}$　$recall = \frac{true\ positives}{true\ positives + false\ negatives}$

伍-資料描述與探索

資料來源：Aldea人工智慧共創平台(train.csv)

資料型態：14319筆資料，47個變項

資料內容：包含員工本身的特徵與工作單位的特徵：年齡、性別、婚姻狀況、學歷、工作資歷、工作地點、工作部門、職等...等46個獨立變數(X)。

預測變數Y: PerStatus（是否離職）。

資料處理流程：

- Step1: 刪除**“最高學歷”**及**“畢業學校類別”**兩變數，因為該兩變數缺失值過多 ，對資訊的提供過少。
- Step2: 剔除同樣的73筆資料，因為該73筆個體在所有變數中皆有缺失值。
- Step3: 做One hot encoding處理類別變數。
- Step4: 針對變數的共線性，刪除one hot encoding中多餘的變數，最後變數增量為85個。
- Step5:最終新的資料切成training data及testing data，分別命名為onehot_del_train.csv & onehot_del_test.csv。

最終資料統整：

Name	Rows	Columns
train.csv(given)	14319	47
clean_train.csv	10665	45
onehot_del_train.csv	10665	85
onehot_del_test.csv	3654	85

資料範例：

	yyyy	PerNo	PerStatus	sex	工作分類	職等	廠區代碼	管理層級	工作資歷1	工作資歷2	...
0	2014	1	0	1.0	1.0	3.0	19.0	4.0	0.0	1.0	...
1	2015	1	0	1.0	1.0	3.0	19.0	6.0	0.0	1.0	...
2	2016	1	0	1.0	1.0	3.0	19.0	6.0	0.0	1.0	...
3	2014	3	0	0.0	1.0	4.0	8.0	1.0	0.0	0.0	...
4	2015	3	0	0.0	1.0	4.0	8.0	1.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
10660	2015	8769	0	1.0	1.0	7.0	8.0	1.0	0.0	0.0	...
10661	2016	8769	1	1.0	1.0	7.0	8.0	1.0	0.0	0.0	...
10662	2014	8774	0	1.0	1.0	7.0	8.0	1.0	0.0	0.0	...
10663	2015	8774	0	1.0	1.0	7.0	8.0	1.0	0.0	0.0	...
10664	2016	8774	0	1.0	1.0	7.0	8.0	1.0	0.0	0.0	...

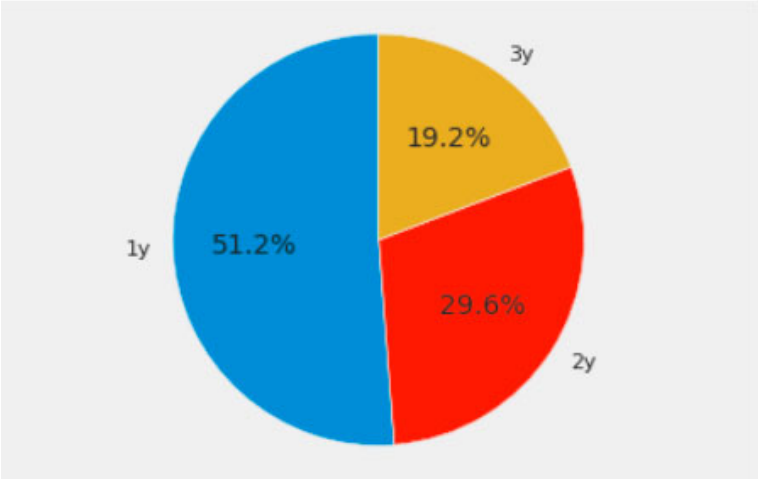
資料樣態：

序	名稱	描述
1	yyyy	西元年分
2	PerNo	員工編號
3	PerStatus	離職與否(1-是 0-否)
4	sex	性別
5	工作分類	分成兩類（1、2）
6	職等	共8類(1-8)
7	廠區代碼	共15類
8	管理層級	共6類(1-6)
9	工作資歷1~5	是否有某種工作資歷(1-是 0-否)
10	專案時數	同名稱
11	專案總數	同名稱
12	當前專案角色	共4類
13	特殊專案占比	同名稱
14	工作地點	共9類
15	訓練時數ABC	同名稱
16	生產總額	同名稱
17	榮譽數	同名稱

序	名稱	描述
18	是否升遷	(1-是 0-否)
19	升遷速度	同名稱
20	近三月請假數AB	同名稱
21	近一年請假數AB	同名稱
22	出差數AB	同名稱
23	出差集中度	同名稱
24	年度績效等級ABC	同名稱
25	年齡層級	同名稱
26	婚姻狀況	共3類(1-3)
27	年資層級ABC	同名稱
28	任職前工作平均年數	同名稱
29	最高學歷	共3類(1-3)
30	畢業學校類別	共3類(1-3)
31	畢業科系類別	共9類(1-9)
32	眷屬量	同名稱
33	通勤成本	同名稱
34	歸屬部門	同名稱

備註：白色底色為類別變數，灰色底色則為連續變數。

離職年數比例：



這張圖顯示出只待1年就離職的佔比很大，因此我們想要將其獨立出來看，是否有哪些因素影響這群人離職。
1Y：317人
2Y：183人
3Y：119人
共：619 人
與離職前一年比較，那個變數有改變的人數(2-3年離職)

序	名稱	次數(全302)
1	近三月請假數A	233
2	生產總額	229
3	年度績效等級B	207
4	訓練時數C	195
5	近一年請假數A	189

序	名稱	次數(全302)
6	專案時數	168
7	專案總數	162
8	特殊專案佔比	150
9	年度績效等級C	131
10	出差集中度	127

從這張表格可以發現，在2-3年離職的人當中，影響前三名的變數分別是近三月請假數A、生產總額以及年度績效等級B，可能代表A事由的請假數量、生產總額以及年度績效等級B有重要的影響。

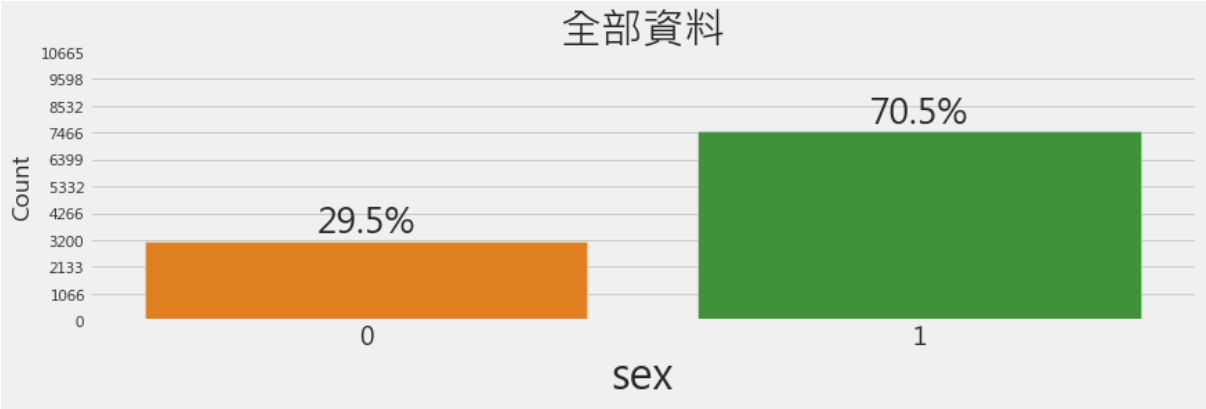
圖表呈現

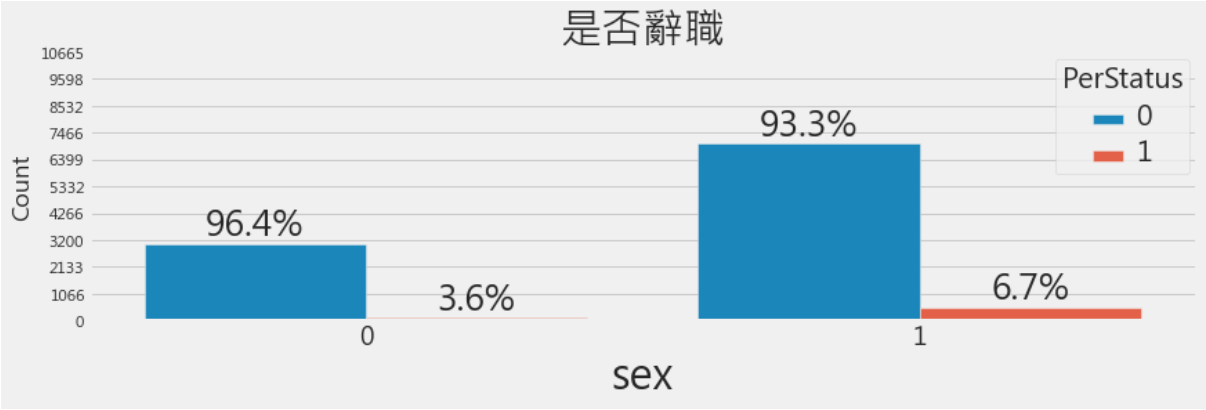
類別變數：

大部分的類別變數，可以發現不同類別當中，數量高的類別其離職數量也較高，這代表著我們很難利用長條圖的方式去看出這個類別變數的重要性，因為很可能是本身高數量所帶來的離職效應而非因為這個類別有多重要。

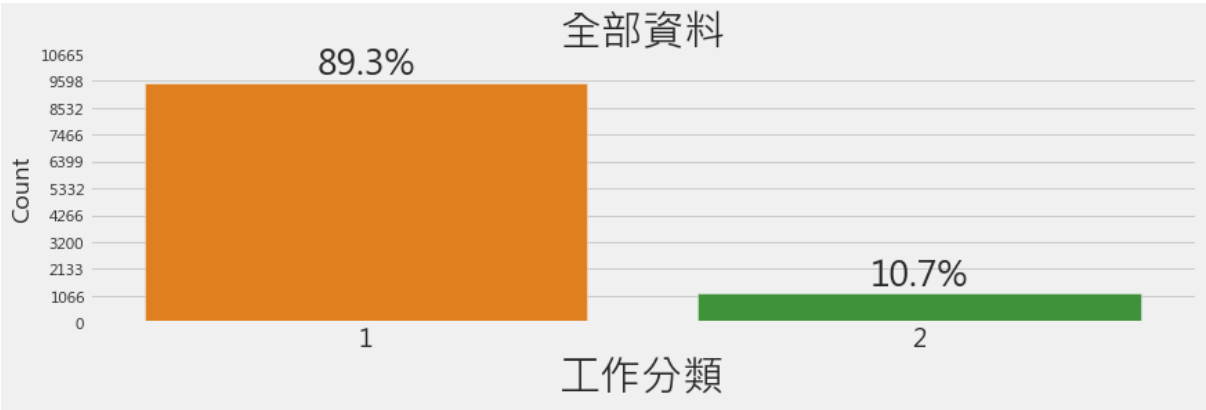
下面，我們將透過一些圖表來呈現這樣的結果作為幾個例子，但由於變相的數

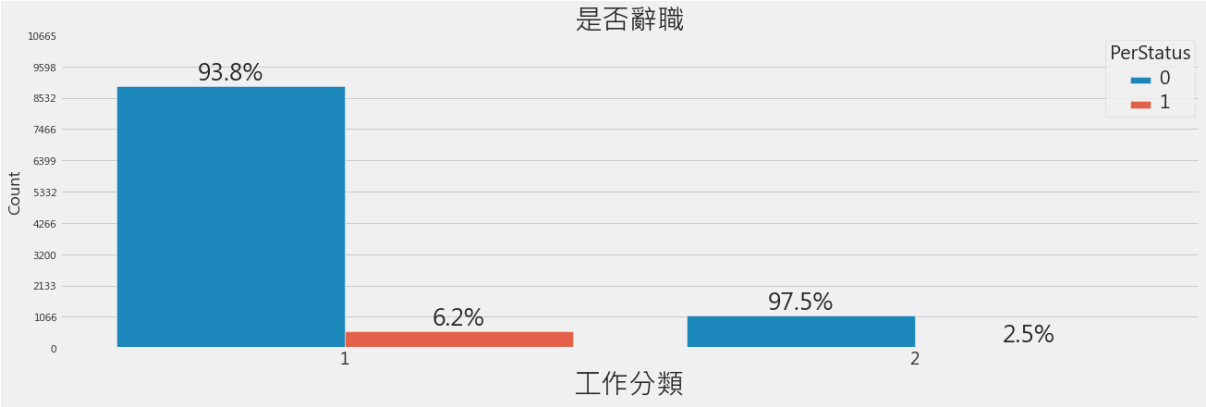
量太多，因此我們僅列舉部分變數來做表述。



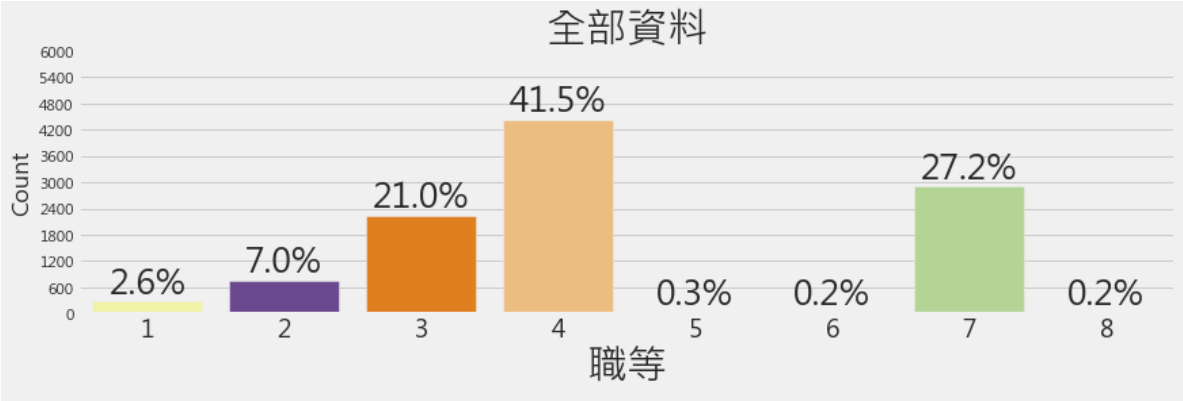


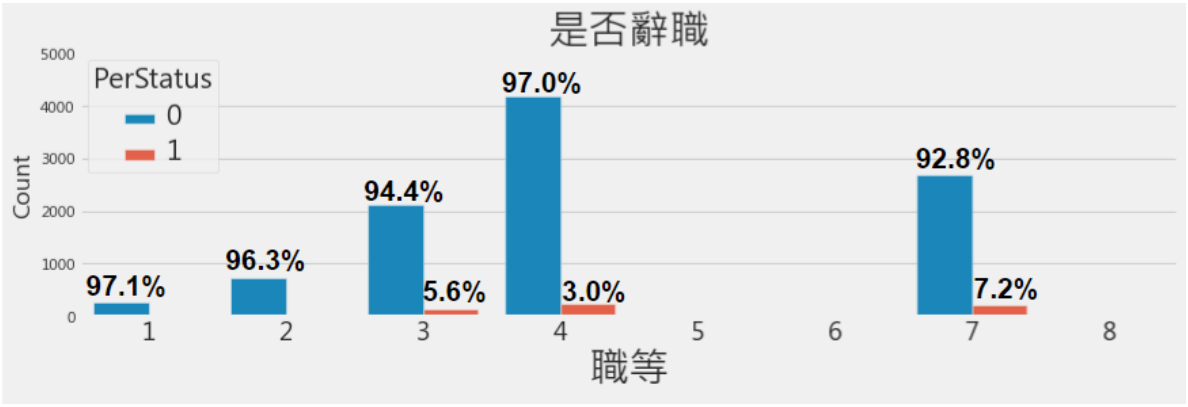
以性別為例：資料中男性數量本來就比女性數量大，因此即便有離職的是以男性居多，我們也不能斷定性別是一個重要的影響變數。





以工作分類為例：可以發現離職的人幾乎都是類別1工作分類類別1的比例高出類別2許多，因此離職資料中也是以類別1較多。





以職等為例：職等的情況與性別、工作分類類似，不能作為顯著的指標。

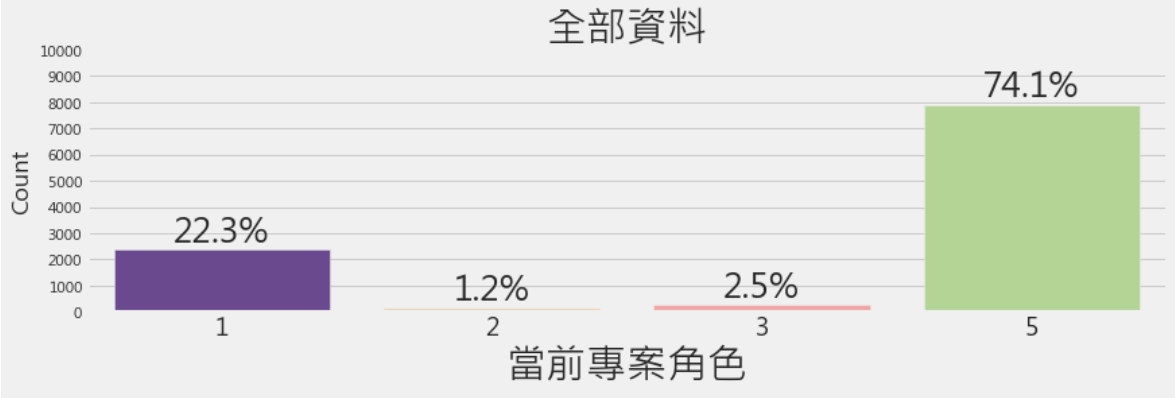
其餘類別變數：

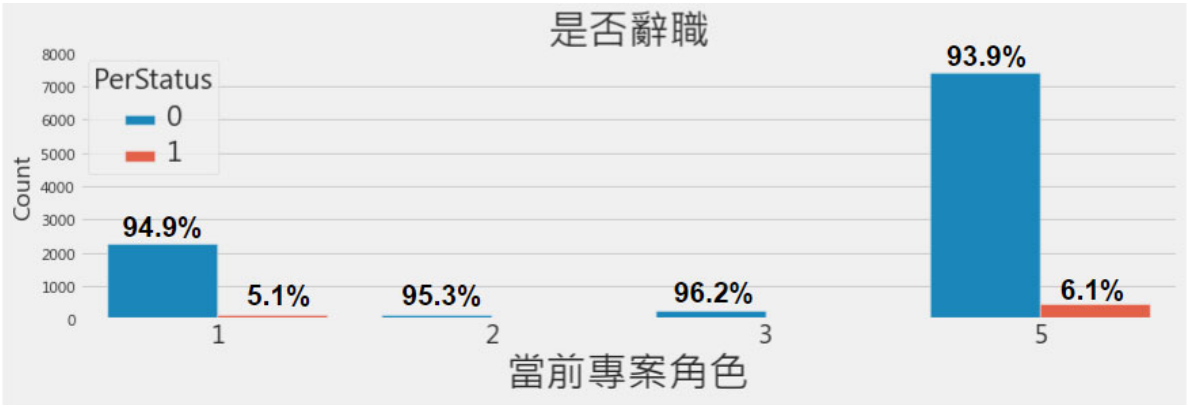
其於類別變數如職等、廠區代碼、管理層級、工作資歷(1、2、3、4、5)、是否升遷、畢業狀況、婚姻狀況、畢業科系、工作地點等類別都有與上述變數類似的情況。

我們或許得用其他的方式去檢索重要的變數。因為光使用長條圖的方式難以看出重大的差異。

"唯一比較特別的變數是當前專案角色"

可以發現離職的資料一大部分都是專案角色5，或許代表當前專案角色為5的人有較高離職傾向。





連續變數：

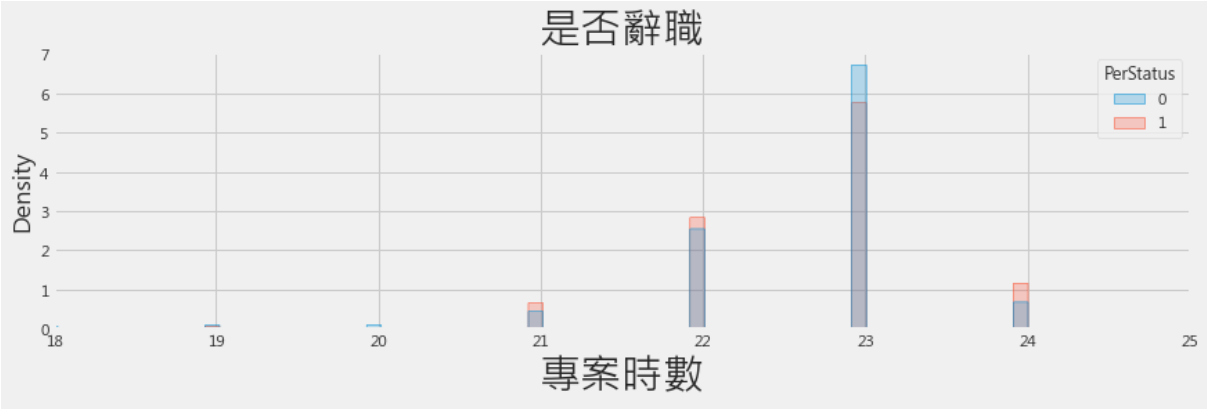
連續變數上，我們所畫的圖雖然大部分也以長條圖為主，這是因為若以折線圖來畫，會呈現五指山的情形，而相較於類別變數，我們也將有離職跟沒有離職按照不同變數，將其密度圖畫出來。

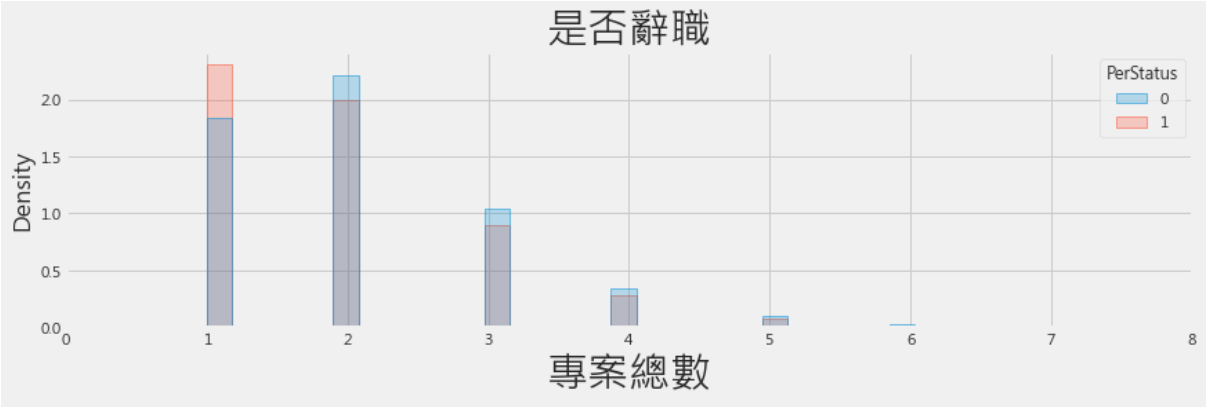
在專案時數上，不論有沒有離職似乎都落在20~25左右

在專案總數上，離職的人似乎少於沒離職的人

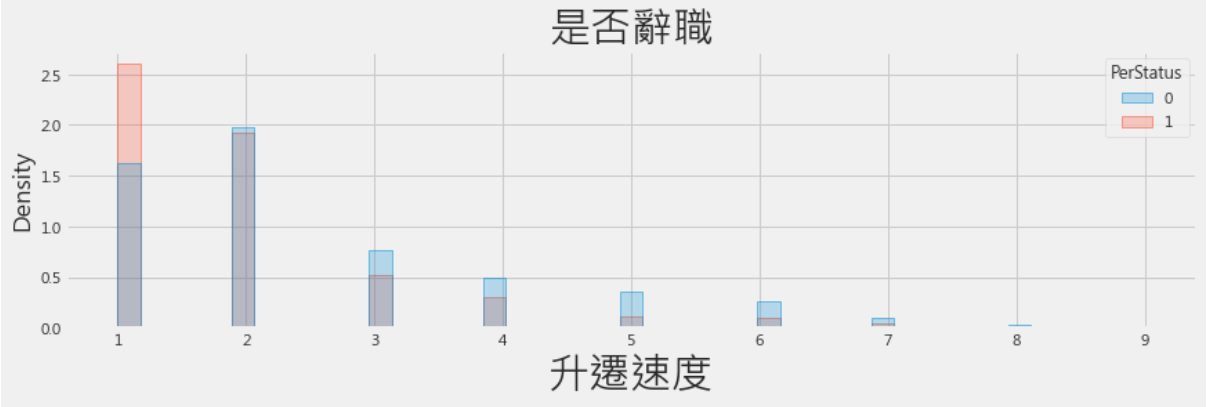
我們猜想離職的人可能在每一個專案花的時間較多，可能都是較為麻煩的專案

(此圖的看法是如果離職的人比例較高上面多出來的就會是紅色)

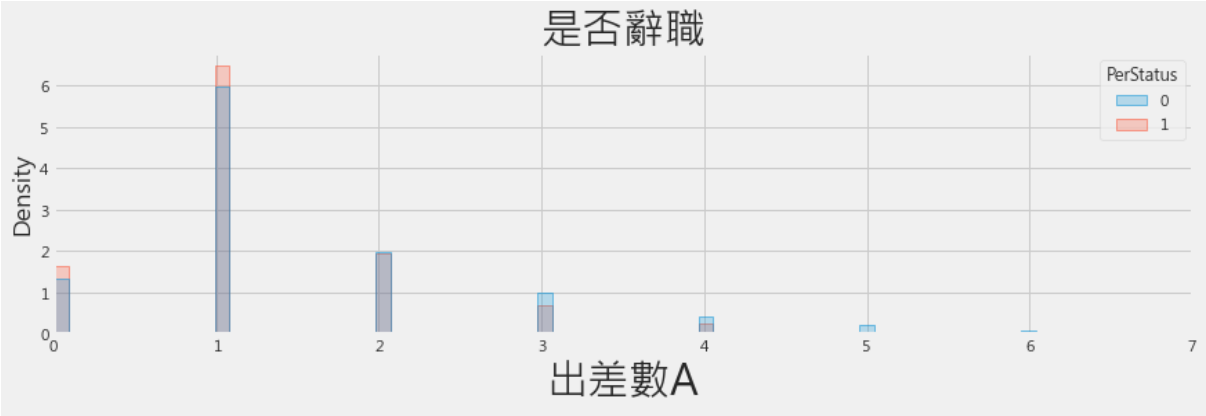


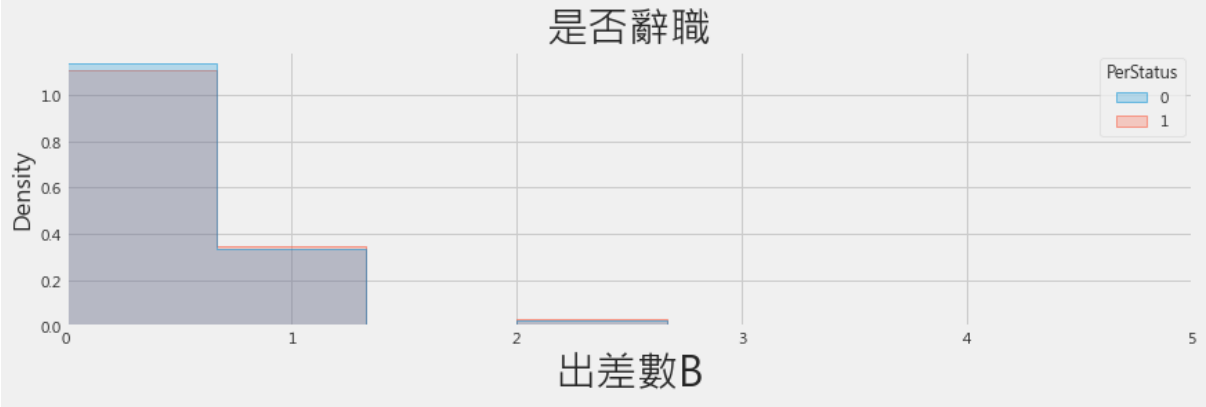


在升遷速度上，離職的人似乎慢於沒離職的人。



在出差數A上，不論有無離職，看不出明顯的差異，而出差數B也是相同的狀況。
而在出差集中度上，也跟出差數AB一樣，並沒有看出明顯的差異

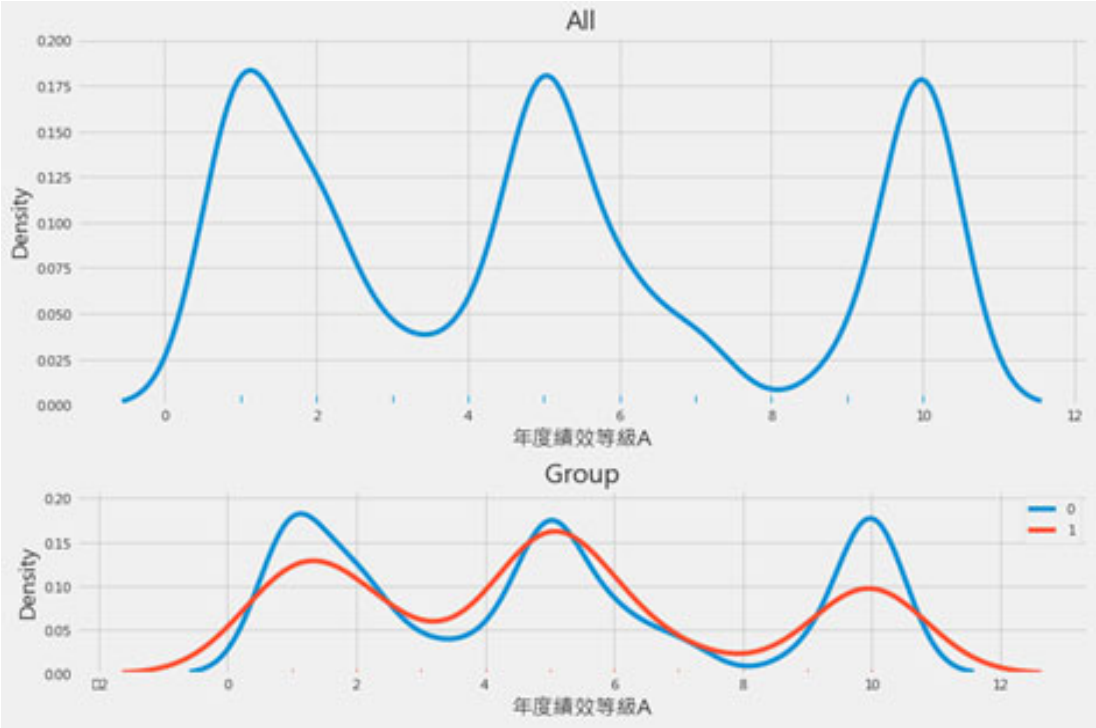


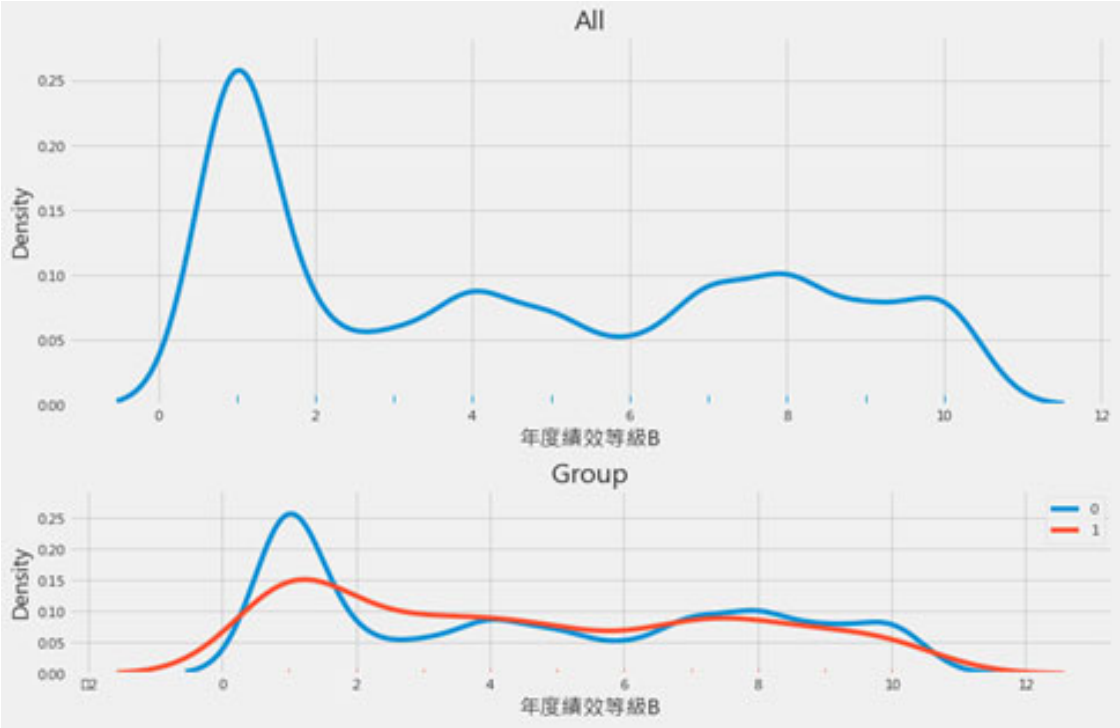


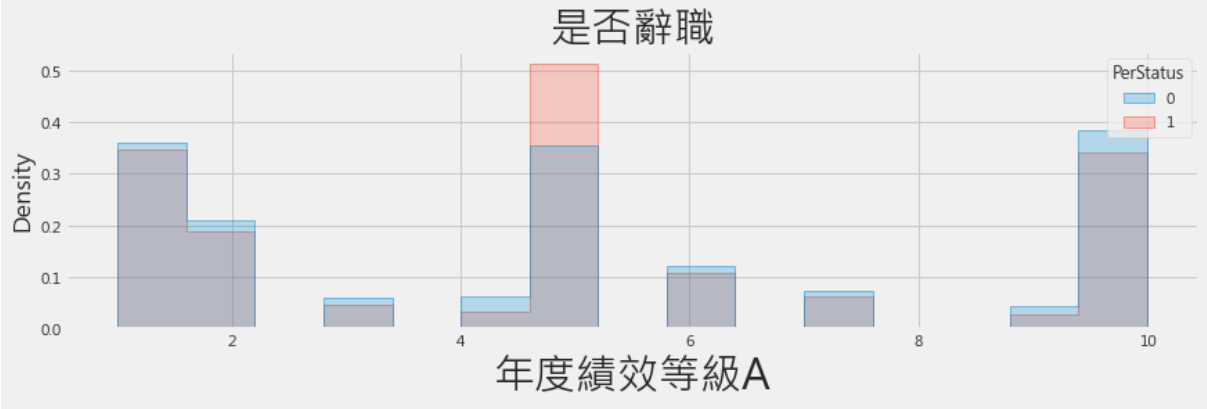
在年度績效等級A中，可以發現離職的人在等級5明顯多過沒離職的人

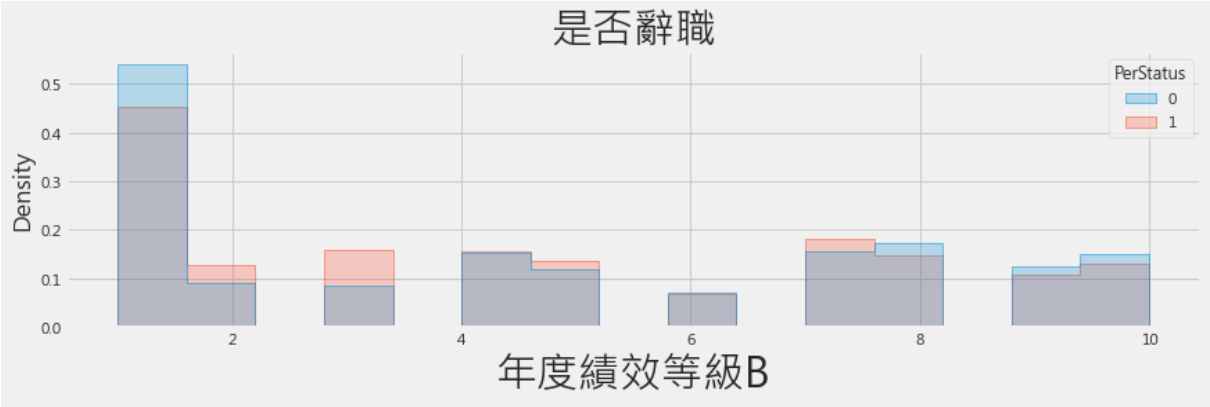
在年度績效等級B中，離職的人明顯在等級1少過沒離職的人

折線圖中，因為此連續型變數五指山的情形沒那麼嚴重，也特別畫出，其中0跟1分別為沒有離職與有離職的密度圖

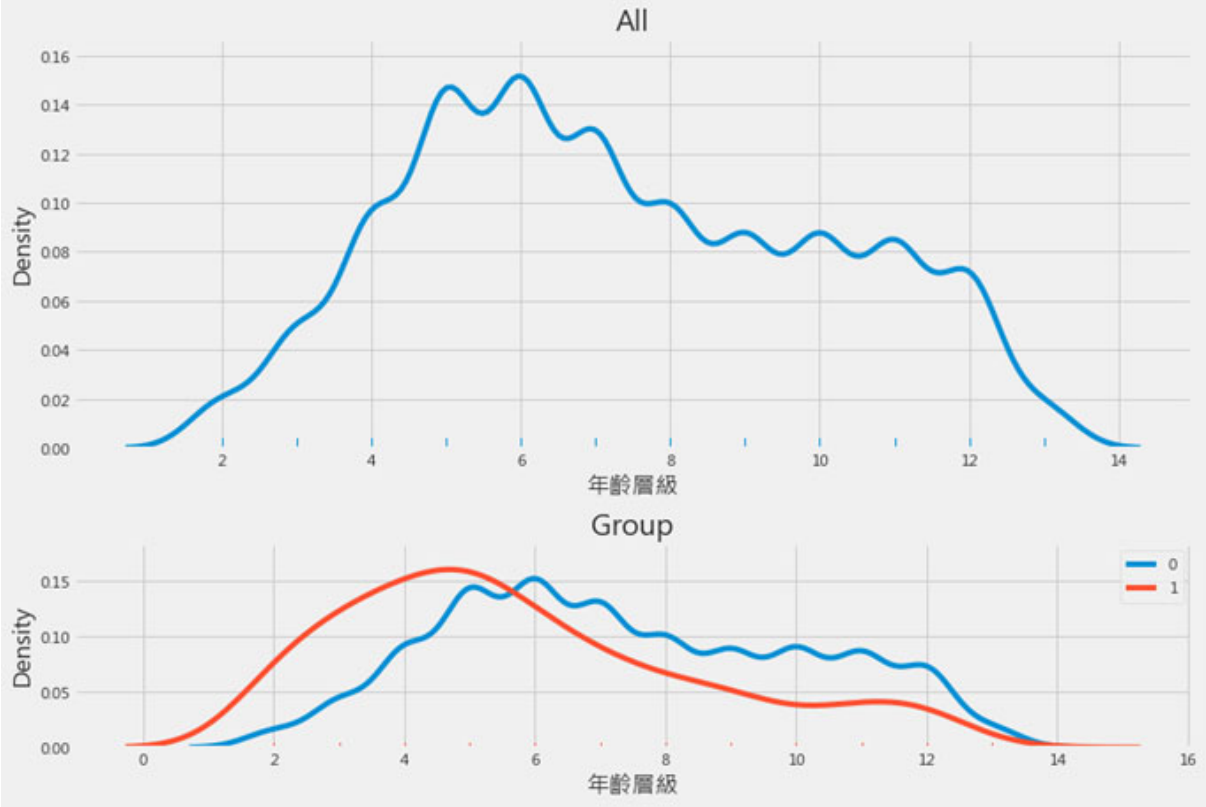


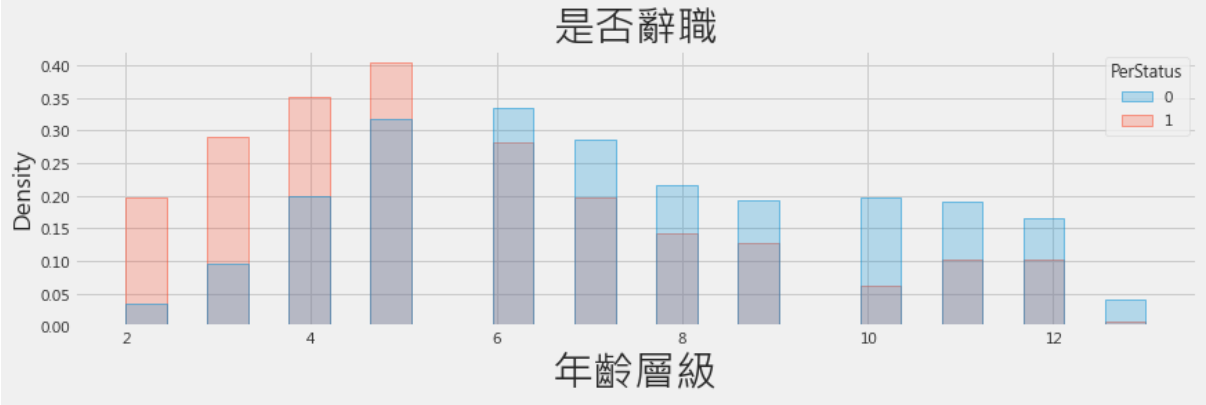




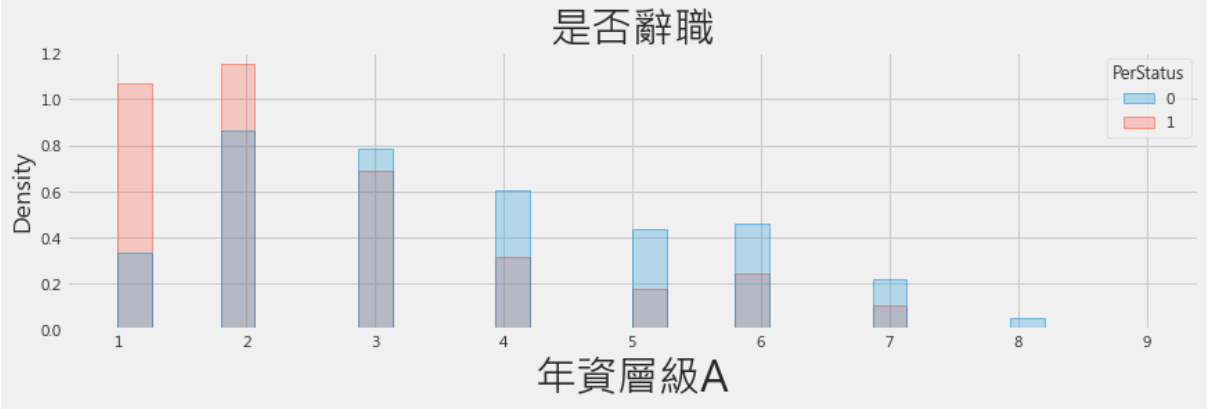


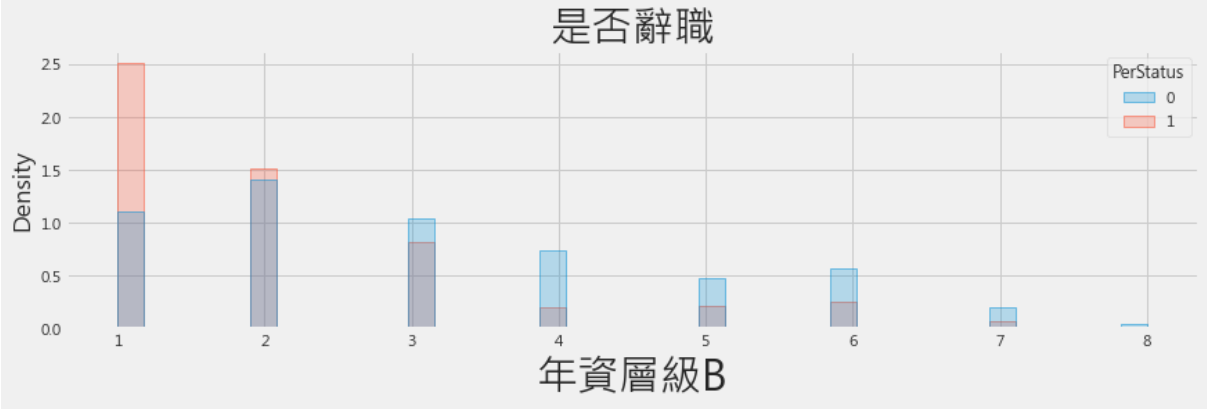
在年齡層級中，可以看出離職的人平均年齡低於沒離職的人，折線圖部分中的0與1分別代表沒有離職與有離職的密度圖，因為沒有五指山的情形故特別畫出。



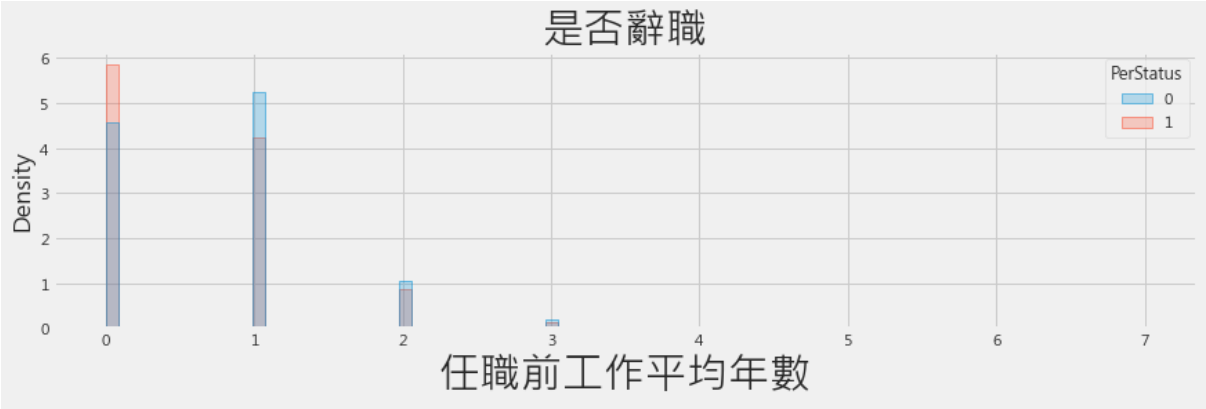


在年資層級AB中，沒離職的人似乎都高於離職的人

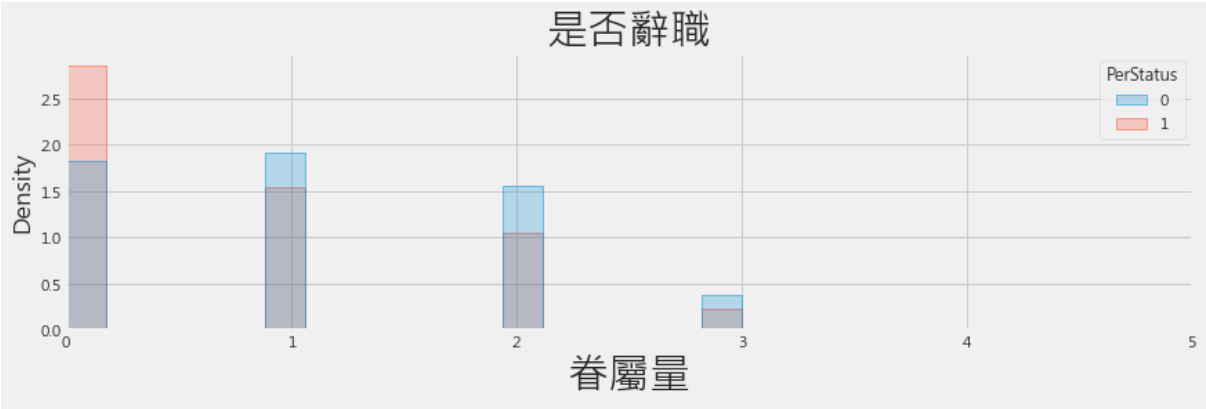




在任職前工作平均年數中，離職的人明顯少於沒離職的人

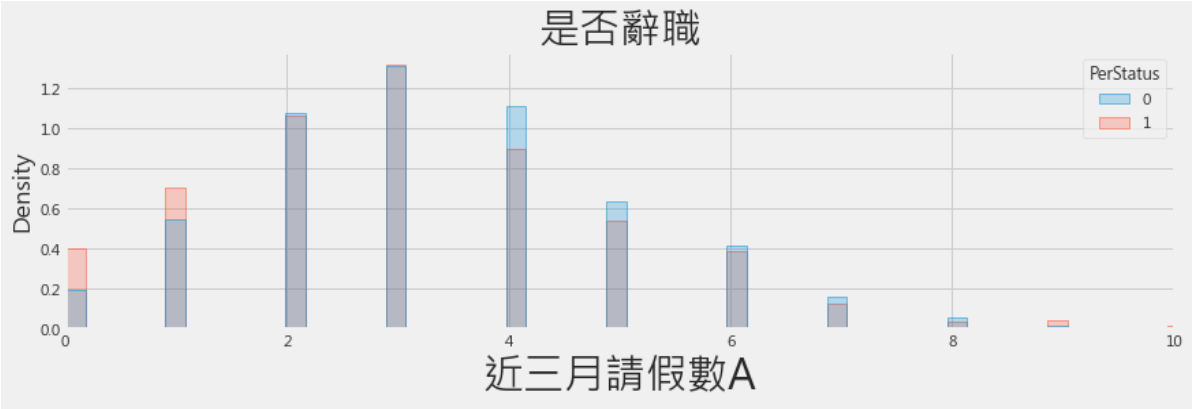


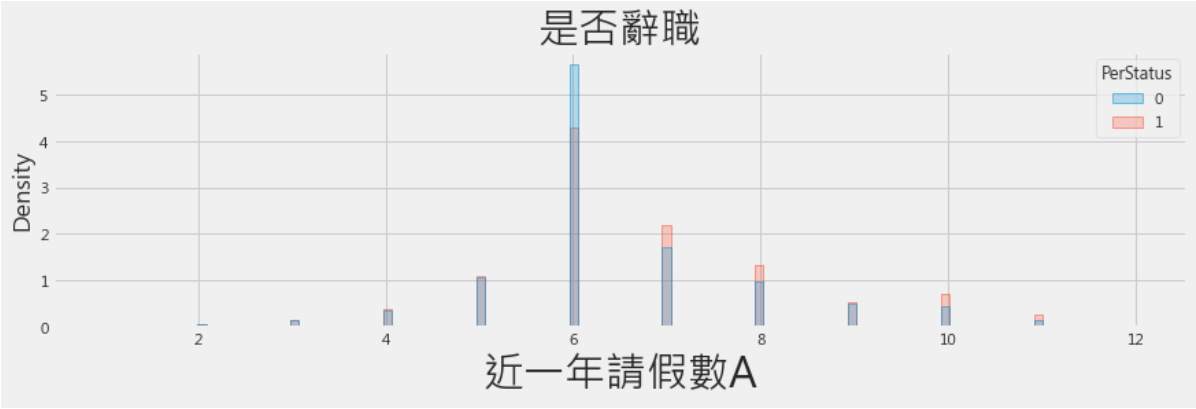
在眷屬量中，離職的人明顯少於沒離職的人

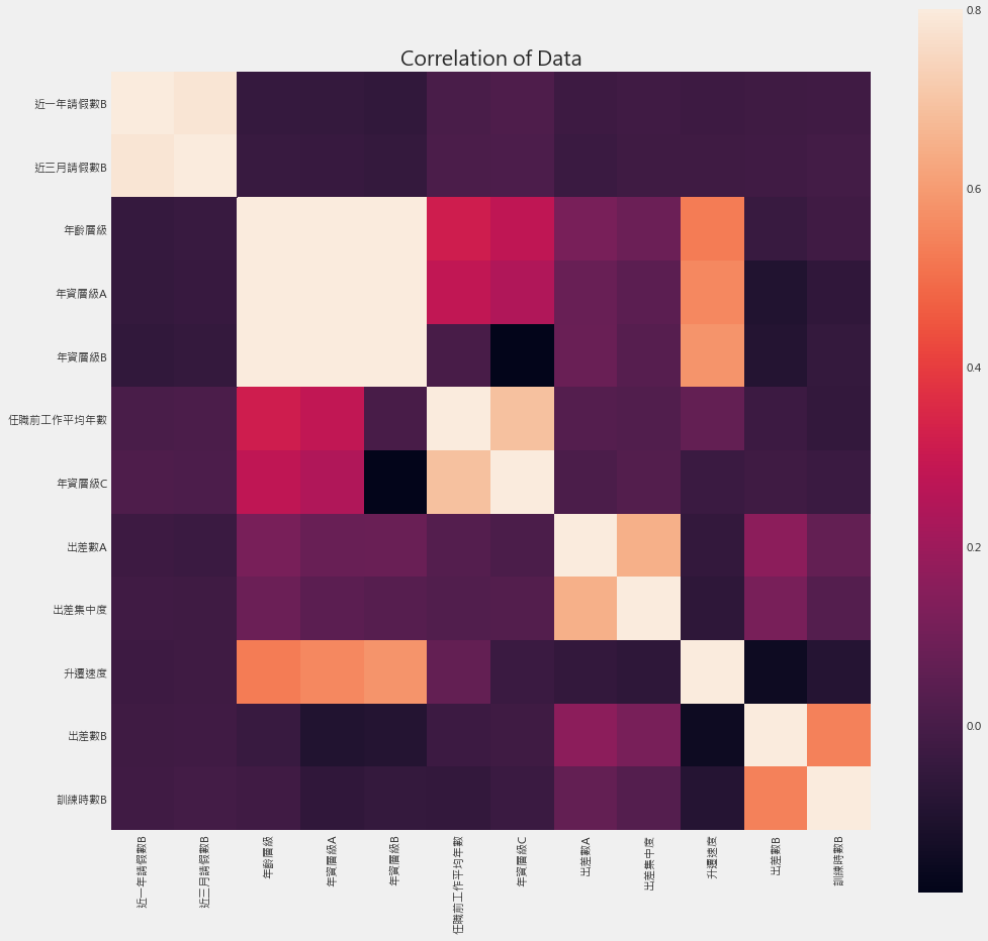


在近三月或近一年請假數A中，兩組差異並不明顯

而近三月或近一年請假數B中也是相同的情況


是否辭職
PerStatus
0
1
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
2
4
6
8
10
近三月請假數A


是否辭職
PerStatus
0
1
Density
0
1
2
3
4
5
2
4
6
8
10
12
近一年請假數A


Correlation of Data
近一年請假數B
近三月請假數B
年齡層級
年資層級A
年資層級B
任職前工作平均年數
年資層級C
出差數A
出差集中度
升遷速度
出差數B
訓練時數B
0.8
0.6
0.4
0.2
0.0

我們將各個次序型變項剔除掉一些不重要的變項，畫出HEAT MAP發現：

spearman相關係數＞.6的有:

- 近一年請假數B v.s. 近三個月請假數B
- 年齡層級 v.s. 年資層級 A,B
- 任職前工作平均年數 v.s. 年資層級 C
- 出差數A v.s. 出差集中度

spearman相關係數大約＝.6的有:

- 升遷速度 v.s. 年齡層級 ,年資層級 A,B
- 出差數B v.s.訓練時數B

陸-具體預測方法

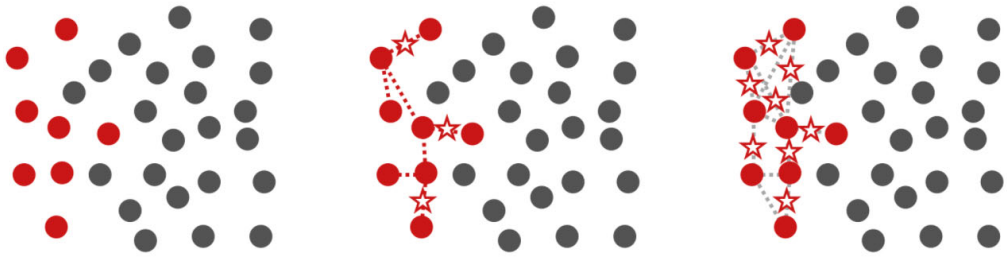
在做完基本的資料處理後，這部分主要進行配適模型，資料預測的步驟。首先，我們將整理過後的資料集，切成Training & Testing Data，Training Data為2014-2016的資料，而Testing Data則是為2017年的資料。我們最終要做的便是透過訓練2014-2016年的資料，進而去預測該這體在2017年是否會離職。我們主要會使用以下幾種機器學習方法去訓練模型，包括：logistic Regression, Decision Tree, AdaBoost, KNN, Random Forest, KNN, Naive Bayes, QDA, Neural Network。

在訓練的過程中，我們發現了嚴重Imbalanced Data的問題，於是使用了SMOTE (Synthetic Minority Oversampling Technique)”樣本合成方法進行處理，SMOTE主要概念為透過一些演算法針對極少值的那部分產生相似的樣本。

1. 找出與陽性個體 $\mathbf{x}_i$ 的最近的 k 個陽性鄰點 (k-nearest neighbors)
2. 在 k 個鄰點中隨機選擇一個，稱作 $\mathbf{x}_j$，我們會利用該鄰點用來生成新樣本
3. 計算 $\mathbf{x}_i$ 與 $\mathbf{x}_j$ 的差異 $\Delta = \mathbf{x}_j - \mathbf{x}_i$
4. 產生一個 0 – 1 之間的隨機亂數 η
5. 生成新的樣本點 $\mathbf{x}_i^{(new)} = \mathbf{x}_i + \eta\Delta$

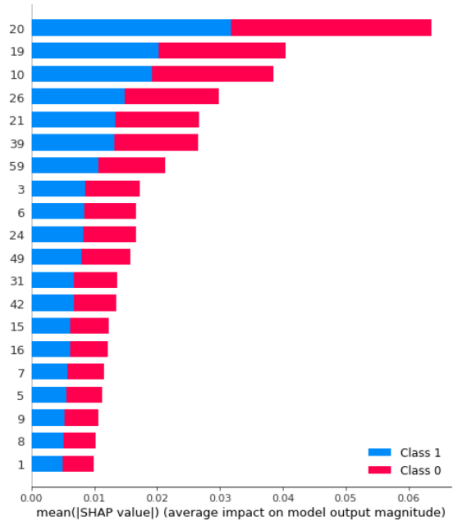
如下圖所示，最左邊為原始資料；中間圖深色的個體是被用來找 k-nearest neighbors 的陽性個體，此處假設我們選擇 k＝3，則對於這兩個點 SMOTE 演算法會先

辨認出最近的 3 個鄰近點，接下來會隨機挑選其中 1 個鄰點用來產生新樣本，最後會在被挑到的個體與對應鄰點的連線上隨機產生一個新的個體，並當作這個個體是陽性的；當我們選擇了很多不同的個體去找出 k-nearest neighbors 以及合成新樣本後，最後會的結果會像右圖所示。



SMOTE 在分類表現上通常可以給我們不錯的分類結果，但最大的缺點在於可能會讓算法對於「特徵重要性」(Variable Importance) 失真，因為樣本是透過合成而來，而非實際收集資料。因此，如果用了 SMOTE 這樣的方法，分類模型的解釋性通常會大幅降低。

在變數的選擇上會依據模型的類別有所不同，有些放入全部變項，有些則是只放入先前在資料探索中發現可能有用的變項。大部分的模型都並沒有特別去挑選變數，選擇放入所有變項進行模型的配飾。然而在每個機器模型調整參數的方法上，我們使用了Optuna這個套件幫助調整出最佳化的參數，進行模型的優化。再者，我們 Random Forest 從85個變數中找出了前20個最重要的變數，結果如下圖呈現：



變數重要性：

1.	年資層級A	11. 廠區代碼_19.0
2.	年齡層級	12. 職等_3.0
3.	近一年請假數A	13. 廠區代碼_10.0
4.	歸屬部門	14. 出差集中度
5.	年資層級B	15. 年度績效等級A
6.	廠區代碼_6.0	16. 榮譽數
7.	工作資歷5_0.0	17. 訓練時數C
8.	訓練時數A	18. 近三月請假數A
9.	生產總額	19. 升遷速度
10.	眷屬量	20. 專案總數

備註：圖中的x軸為變數的重要性程度，y軸代表變數原始的數字代碼。舉例來說，最重要的變數為最上面那個原始代碼20，則他所對應的便是右側排名第一的“年資層級A”，第二重要的是代碼19的變數，對應到右側則是“年齡層級”，而第20重要的為代碼1，代表“專案總數”... 以此類推。

(1) Logistic Regression

然而針對 Logistic Regression 這個方法，我們使用了兩種方式去進行模型的配飾，分別是在使用變項(X variables)上有所不同(全部變數以及使用上方Random Forest挑過的變數)，我們想看看兩者間是否有線著差異， 結果如下表格：

方法	Logistic Regression	Logistic Regression
使用變數	Random Forest 挑選出的重要變數	全部變數
使用資料	全部筆數	全部筆數
F_beta score	0.0579	0.1023
判別機率	0.3	0.3
Confusion Matrix	預測 實際 0 1 0 3431 49 1 166 8	預測 實際 0 1 0 3410 70 1 159 15

小結：邏輯斯迴歸的結果很差，雖然大部分沒有離職(0)的人有被成功判別，但我們重視的是離職的人(1)的判別，這部分可能因為變數的比例太過懸殊(資料不平衡情況嚴重)，導致模型傾向於將所有結果都判別成0。

(2) Other Machine Learning Models

最後，在處理完Imbalanced Data的問題後，我們分別配飾了 Decision Tree, AdaBoost, KNN, Random Forest, KNN, Naive Bayes, QDA, Neural Network 這9個機器學習模型。我們將資料以年份分成四組個別訓練模型，想看不同年份之間的模型配飾結果是否有明顯差異，以及哪個組別會有比較好的表現，模型配飾結果(以F Beta Score為評估好壞標準)如下：

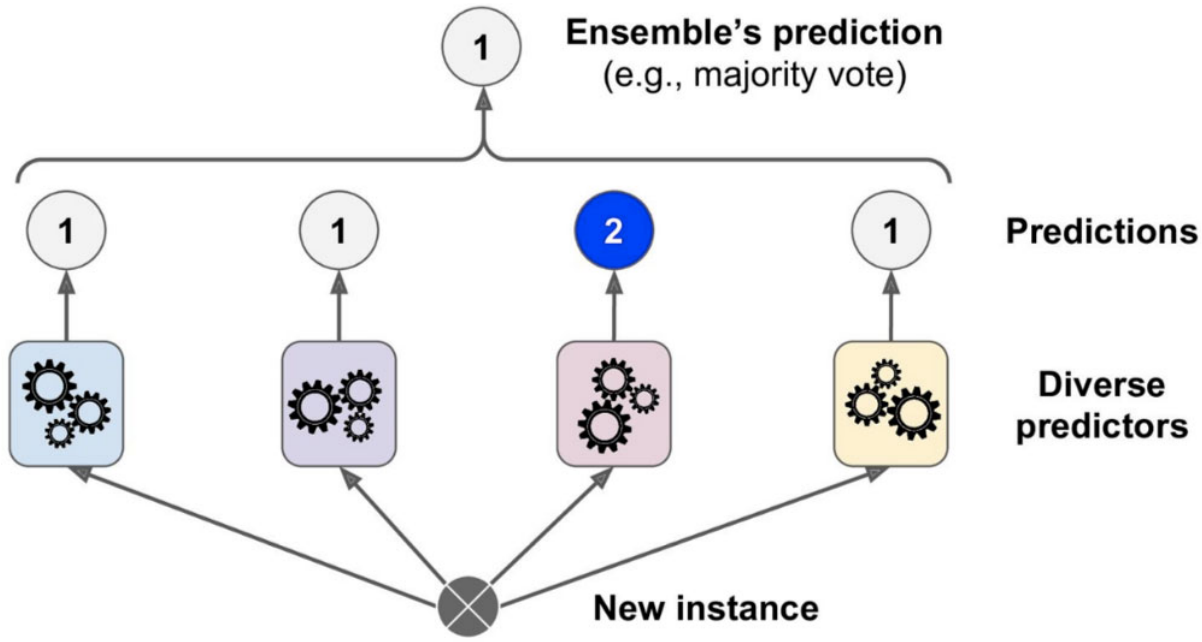
ML Model	all	2016	2015	2014
KNN	0.14	0.10	0.13	0.12
Naive Bayes	0.19	0.19	0.17	0.19

Decision Tree	0.08	0.12	0.09	0.08
Random Forest	0.12	0.14	0.09	0.10
AdaBoost	0.08	0.08	0.05	0.01
QDA	0.15	0.15	0.15	0.15
Neural Net	0.16	0.14	0.15	0.12

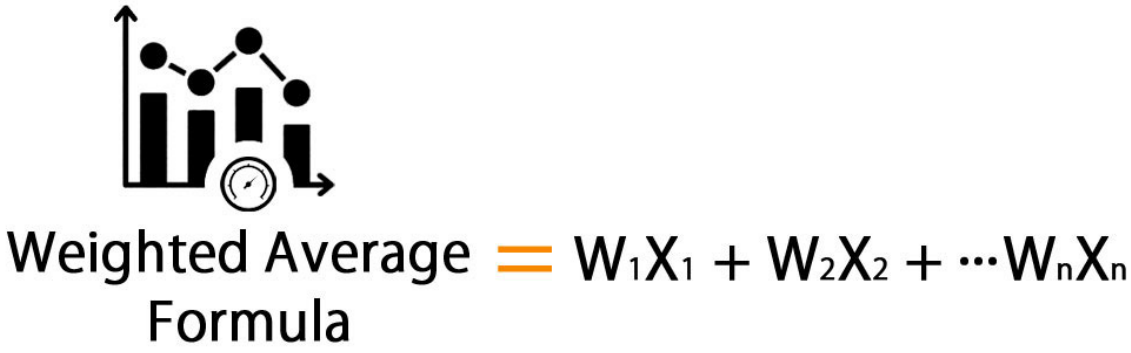
根據上面結果，我們可以發現all (全部年份)以及2016這兩組的表現差不多，相對比較好，而其中發現使用 Naive Bayes 模型的結果則在每個組別中表現皆為最佳。

柒-實驗評估分析

單純從實驗結果來看，我們的表現並不是很好。由於每種模型都已使用自動參數調整的套件進行hyperparameter search去尋找能讓模型表現最好的參數組合，所以可以先排除模型並未找到最佳解的可能。在後續我們有嘗試XGBoost、LightGBM、CatBoost以及簡單的神經網路模型，但是表現並沒有顯著的進步，有的結果可能還會更差。



為了解決單一模型表現不佳的問題，我們採用ensemble的方法，也就我們挑選在訓練後表現相對較好的幾個模型讓他們進行「投票」評分，也就是額外訓練一組合成器使其能學習要分配給各模型多少權重，如下圖所示X_i代表不同的模型預測結果而W_i代表模型分配的權重，最終的預測結果將採用不同模型預測結果的加權分數。



使用這樣的機制，能使預測結果穩定得到F beta-score = 0.21的分數，整體結果會比較穩健一點，但缺點是會導致模型解釋性較差。

捌-結論及未來展望

透過這次的分析報告，我們可以發現某些變數的卻可能是影響是否離職的原因，如年齡、升遷速度等，從圖表分析中可以初步提供離職員工可能具備的屬性，從中了解到整體趨勢。然而，即便有初步的分析提供我們在後續建模時哪些變數可能對結果影響較大，從實驗結果來看仍然無法使模型有顯著進步，了解其他參賽者多數也表現不好，我們分析到此筆資料可能無法那麼容易單純使用挑選重要變數去訓練模型的方法就能得到良好結果。

礙於時間關係，本次報告還有許多實驗與模型架構改善的方法未完成，例如使用Graph Neural Network進行建模以及依照依照在職年數不同分別建立模型去進行預測等，實屬可惜。不過整體上，透過這樣的分析流程能讓我們對此議題有更進一步的了解，算是有不少收穫。

玖-文獻參考與附錄

1.Foley, A. (2019). Using Machine Learning to Predict Employee Resignation in the Swedish Armed Forces (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-265013

2.Dai, Weihuang & Zhu, Zijiang. (2021). Employee Resignation Prediction Model Based on Machine Learning. 10.1007/978-3-030-53980-1_55.

拾-小組成員貢獻

	貢獻
吳岱桀	探索式資料分析、圖表繪製、整體模型架構設計、特徵工程
謝宜均	資料前處理、機器學習模型配飾及評估結果、口頭報告、分析流程及具體預測方法說撰寫
黃琮祐	資料前處理(R)、變項篩選(LASSO)、口頭報告、資料描述與探索、基本統計分析圖表詮釋、摘要撰寫
周柏翰	分類模型配飾、口頭報告、資料描述與探索、基本統計分析圖表詮釋、動機描述