離職預測

介紹

本組預測題目為離職預測，該資料包含每位員工的在 2014 到 2017 年在公司各項指標的紀錄和每年各季的請假出差次數，預測目標為員工在 2018 年是否會離職。本次研究採用各類 ML 和資料處理方法進行比較，使用 f-score 當作評分標準，選出最優秀的預測解法。

Proposal 修改

原本預定在要使用全連接深度模型進行預測，但在未進行資料處理的狀態直接預測會導致模型無法有效學習，會將所有的答案都填為 0，在此猜測可能原因為訓練資料中未離職的比例就佔了將近 9 成，而本問題又屬於特徵比較多的訓練資料，因此全連接模型無法在僅存的 1 成資料中學習到哪種特徵下會離職，而如果將離職比例進行調整後會導致訓練資料不足 1 千筆，根據以往經驗在訓練項目高達 40 多項而訓練資料不足 1 千筆的狀態無法有效訓練，因此在研究初期就放棄使用全連接深度學習網路。

在 proposal 裡面還有提到我們希望利用該年資訊回答該年離職可能性，不過就訓練結果而言，僅憑單一是完全無法做出任何預測的，因此在之後的實驗裡假設參考的資料要以會有時間變化的資料為訓練資料，所以將畢業學校、性別、曾經工作單位等歷史資料都不列入訓練。

資料前處理

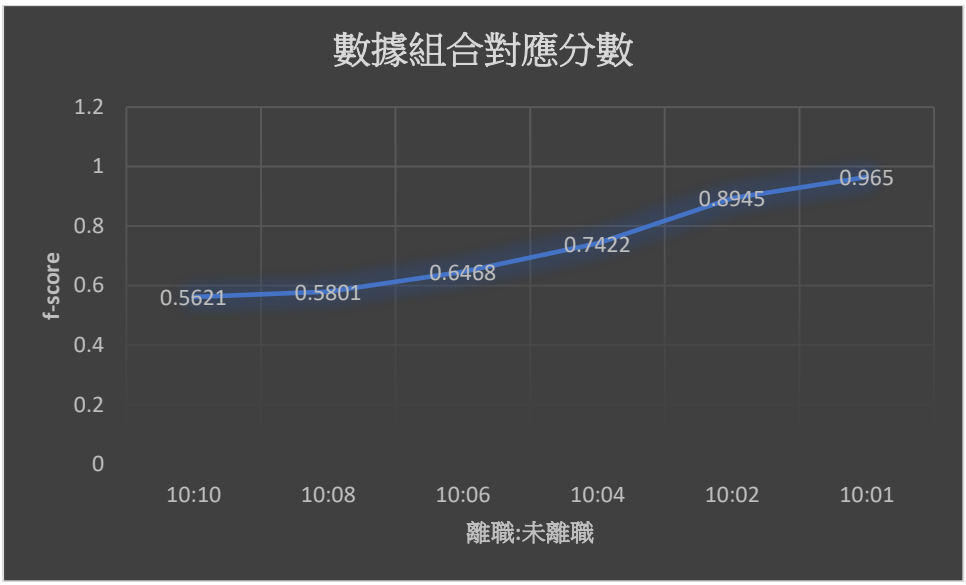
由於該訓練資料在員工離職後仍然會保留員工的位置，因此在訓練資料上有大量的空白資料，如果直接進行訓練會導致訓練結果極差，因此要先將空白的資料清除或必要的補上-1。

因為每個員工離開年份分散在 2014 到 2017 年之間，為了增加訓練的資料所以將所有資料每兩年做切分，如果一個員工在 2014 年持續工作到 2017 年，他他即可被切分出 2014~2015 和 2015~2016 和 2016~2017 共三份資料，之後再特別挑出第二年有離職的資料和兩年都沒離職的資料做組合。

除了訓練資料以外主辦單位還提供員工在每年的 4 個季度出差和休假次數，所以依照年分和員工編號將出差和休假次數資料連接到其他的資料後面。

下表是離職資料比未離職資料比例組合所訓練的結果，使用模型為 XGBClassifier，驗證資料和訓練資料大小為 6:4，可以發現在離職比例越高的情況下訓練結果越好，而上傳上去 aidea 的分數也可以觀測到未離職比越低分數越好的情況，但實際上傳到 aidea 後會發現在未離職比例高的狀況下分數的變異數會比較大平均值大約落在 0.13，比例較低的分數則是平均落在 0.17 但變異

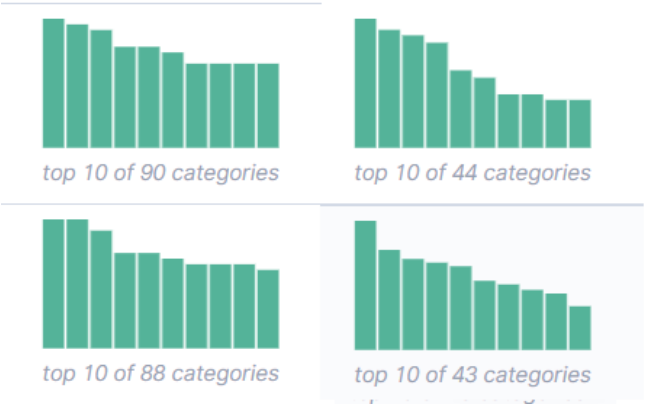
數會比較小，推測是低離職比資訊含量比較高因此比較穩定，而高離職比的訓練要看當時的資料有沒有剛好讓模型學到，而由此上述結果可猜測將離職比提高可以提升模型學習到離職特徵的機會，並且實際 2018 年的離職率可能遠高於之前訓練資料的比例。



資料篩選和清理

在資料篩選上我們採用了 kibina 分析，可以發現有許多資料有級偏差的狀況，像是下圖是工作經歷 3、4、5 的分布狀況。

所以我們挑選分布比較均勻多樣化的特徵出來，如下圖是生產總額和歸屬部門的分布圖。



並且刪除了一些歷史因素特徵，如畢業科系、業學校、性別。利用挑選過後的資料雖然在測試資料上的分數變差，但在 aidea 上可以獲得較小變異差的結果，但實際平均分數卻比未挑選的還低。

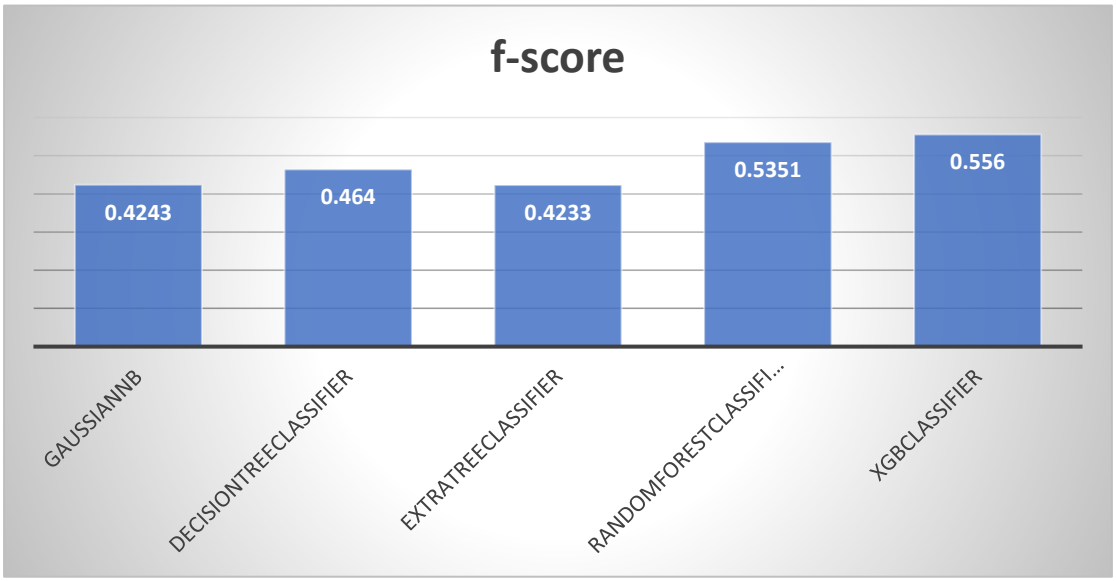
除了原本的訓練資料以外也對每季度的請假和出差資料進行訓練，可以發現如果有用到每季度的資料可在驗證資料裡面得到較高的分數，但卻無法在 aidea 上面拿到更高的分數，因此判斷可能原本的資料就已經有 overfitting 的問題，所以再增加更多特徵無法得到更好的訓練結果。

而我們也嘗試將出差時數或是請假時數這類資料進行標準化，不過不管是在測試或是 aidea 都到正確率下降的結果。

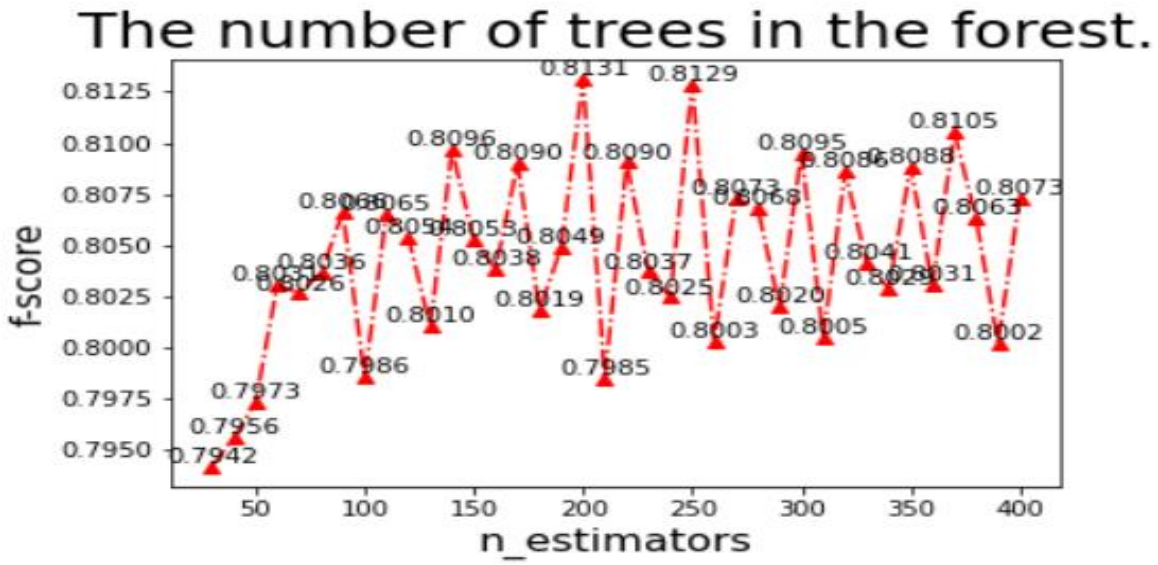
模型選用和調整

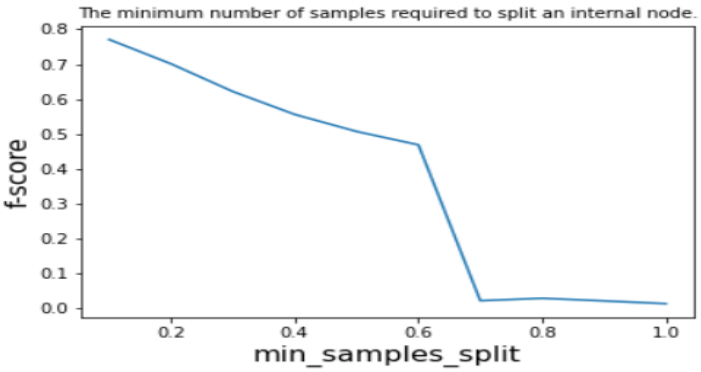
在模型選用上我們測試了幾個模型，在離職資料和未離職資料以 10:3 並且同樣訓練狀況下列出以下列表。

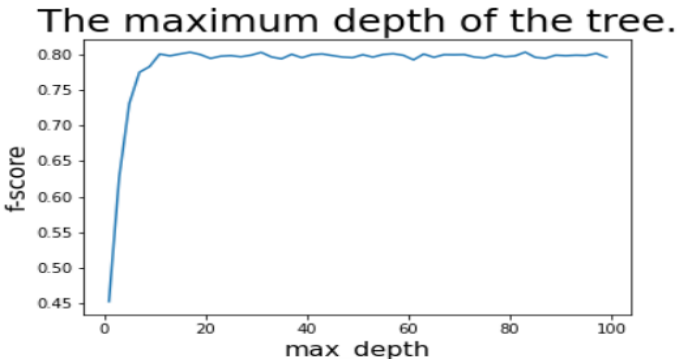
而當離職資料和未離職資料以 1:1 組合時的訓練結果如下圖。



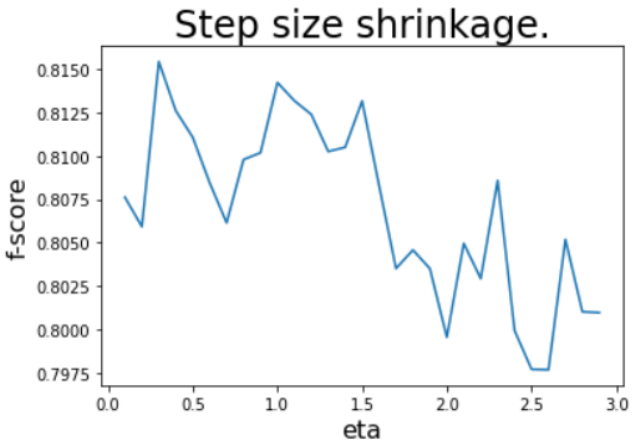
可以發現 GAUSSIANNB 模型比較不容易發生 overfitting 的狀況，但訓練的結果還是略低於其他模型，而 DECISIONTREE 和 EXTRATREE 都有比較好的訓練結果，但還是稍微比 RANDOMFOREST 和 XGBOOST 還低一些，尤其是在訓練資料比較亂的狀況下，所以之後的實驗將以 RANDOMFOREST 和 XGBOOST 為主要模型。

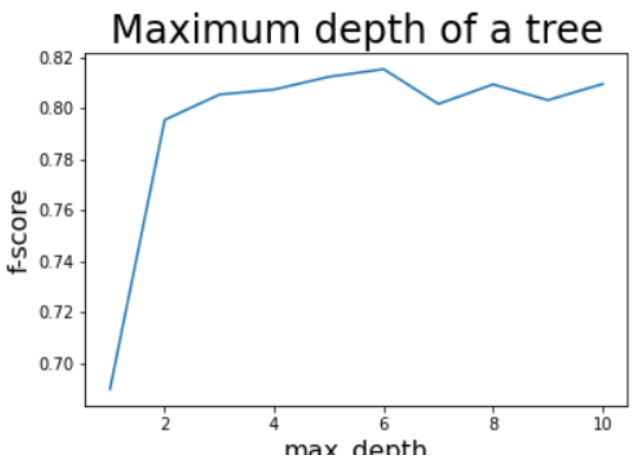


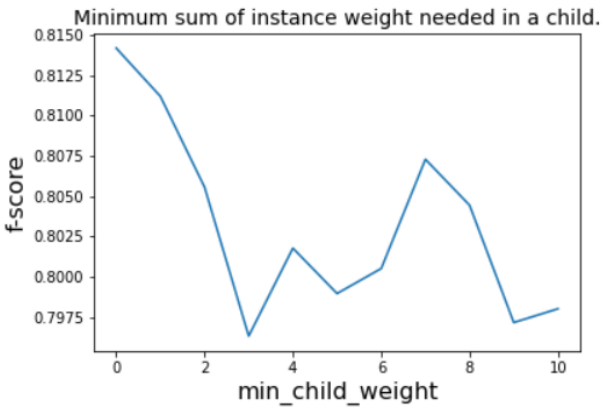


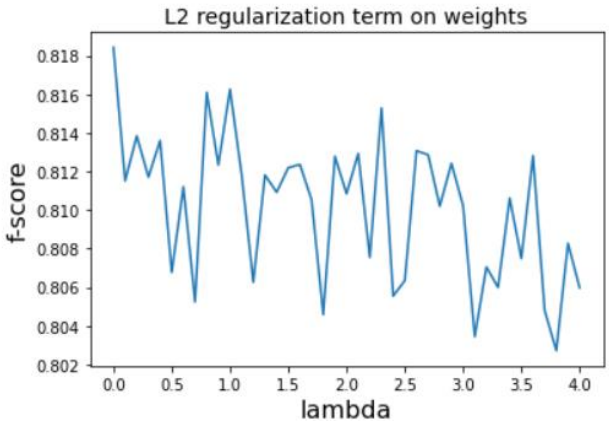


以上三張圖是針對 RANDOMFOREST 的幾項比較有影響的參數做調整，將子樹數量和樹深度調小可以發現稍微降低 overfitting 的問題，不過調整細微參數對於整體的正確率上升幫助並不大。









以上四張圖是針對 XGBCLASSIFIER 做參數主要參數的調整，可以發現跟 RANDOMFOREST 一樣對於細微的參數調整雖然無法對驗證資料的 f-score 有明顯幫助，但在同樣的 f-score 下面在 aidea 上面會有比較穩定的表現。

其他問題

在整個訓練過程中會發現明顯的 overfitting 問題，常常驗證資料的 f-score 可以高達 0.8 但實際測試結果卻不到 0.08，甚至會發生調整完參數後驗證資料 f-score 上升但在 aidea 分數下降的狀況。而我們在這之中發現兩個關鍵因素會導致我們在 aidea 分數有劇烈波動，第一個是隨機數的種子，第二個是測試資料跟訓練資料的比例分配。

隨機數的種子除了影響模型和測試資料跟訓練資料比例以外，對我們實驗最大的影響就是我們的未離職資料是從所有資料裡面抓出不到 1 成的資料，所以隨機抓取到的資料對我們在訓練上會有極大的影響，尤其是當離職資料和未離職資料以 1:1 組成時，同樣的模型在不同隨機數的種子，上傳到 aidea 的結果可以相差 0.1 以上，所以我們在測試完模型後還會額外做一個隨機數的挑選，藉由選取不同隨機數找到最好的訓練結果在以此上傳。除此之外為了減少每次訓練結果被隨機數的種子的影響，我們在同樣模型會訓練 100 次，最後再將訓練結果計算平均輸出。

我們在訓練資料和測試資料的比例上是抓 7:3，在一開始因為怕訓練資料過少所以抓到 8:2 甚至 9:1，但實驗的結果發現會有 overfitting 的問題，並且因為整體資料過少所以如果抓到 8:2 以上有時對模型的小改變無法反映到測試結果上面，但卻會對上傳的結果有影響，而在測試中還有觀察到一個特別的現象，在多種模型下都可以觀測到當訓練集比例較低時反而會有更好的效果。

分析結果

總結上述各種資料清理方法和組合，加上模型的調整進行訓練並且上傳。目前最高在 aidea 上面有 0.1957 的成績，不過該項結果是利用 1:1 離職和未離職資料組合訓練而得出來的結果，根據前面的研究這樣的數據在 public

Leaderboard 就算有很好的表現也有可能是 overfitting，上傳至 private Leaderboard 有可能會分數變很低。而其餘比較好的分數還有由 XGBCLASSIFIER 在離職比和未離職比 10:3 訓練出來的 0.1838 和用 RANDOMFOREST 在同樣訓練資料下的 0.1790，雖然這兩個都比最好的 0.1957 還低一些，但為了確保最後不會 overfitting 還是以 XGBCLASSIFIER 為主要 private Leaderboard 計算結果。

結論

這個題目因為訓練資料分布偏差太大，如果一開始並沒有做任何資料處理或清洗就直接訓練的話是無法得到任何分數的，而在經過一基本的清洗大概就可以在 aidea 上面拿到 0.1 以上的成績，之後的訓練往往會出現在是否會 overfitting 上面，常常會發生在本地成績不如以往上傳上去卻異常出色，或是本地測試成績極好但上傳上去分數卻極低，雖然可以靠選擇 GAUSSIANNB 這類比較簡單的模型來避免，但分數要在上去勢必要用比較複雜的模型，我們在資料挑選和模型參數的調整上並沒有發現有太多的突破點，因此後續主要研究方向在進行資料組合和不同模型的配對，來尋找一個穩定度高且分數不錯的組合。