

Machine Learning 2017

Pump It Up: Data Mining the Water Table

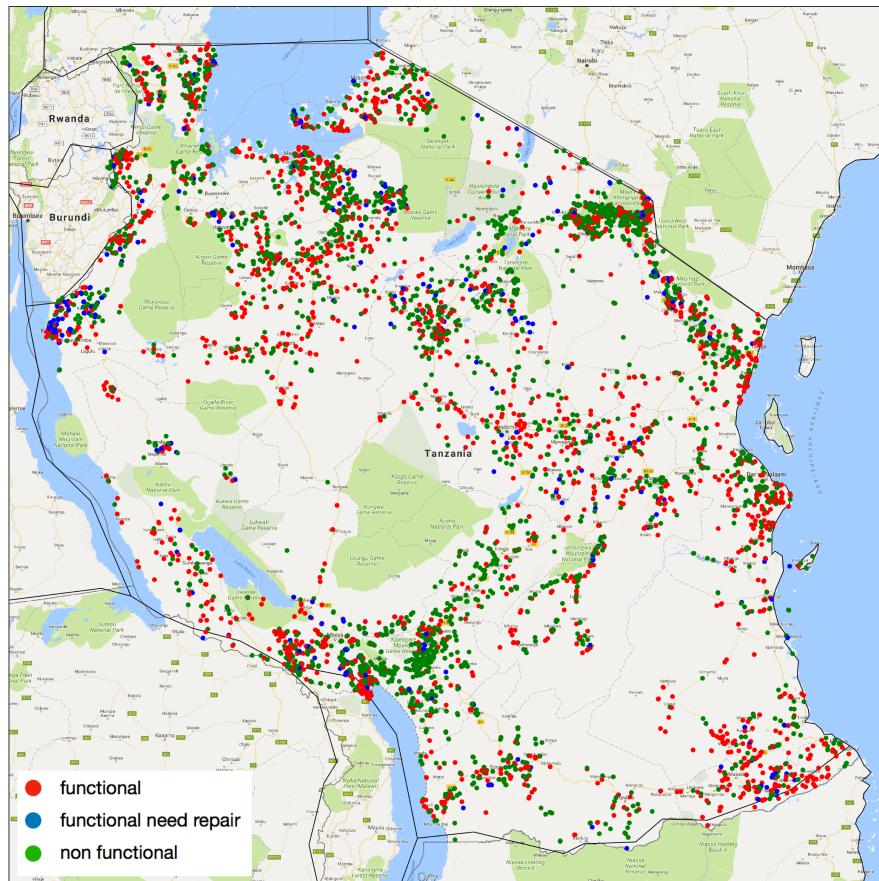
我會 Google 我是資優生

B03502040 劉君猷
B03902042 宋子維
B03902048 林義聖
B03902072 江廷睿

June 29, 2017

1 題目

我們三個題目都有做嘗試，但最後挑選了 Driven Data 的 Pump It Up: Data Mining the Water Table [3] 作為期末報告。

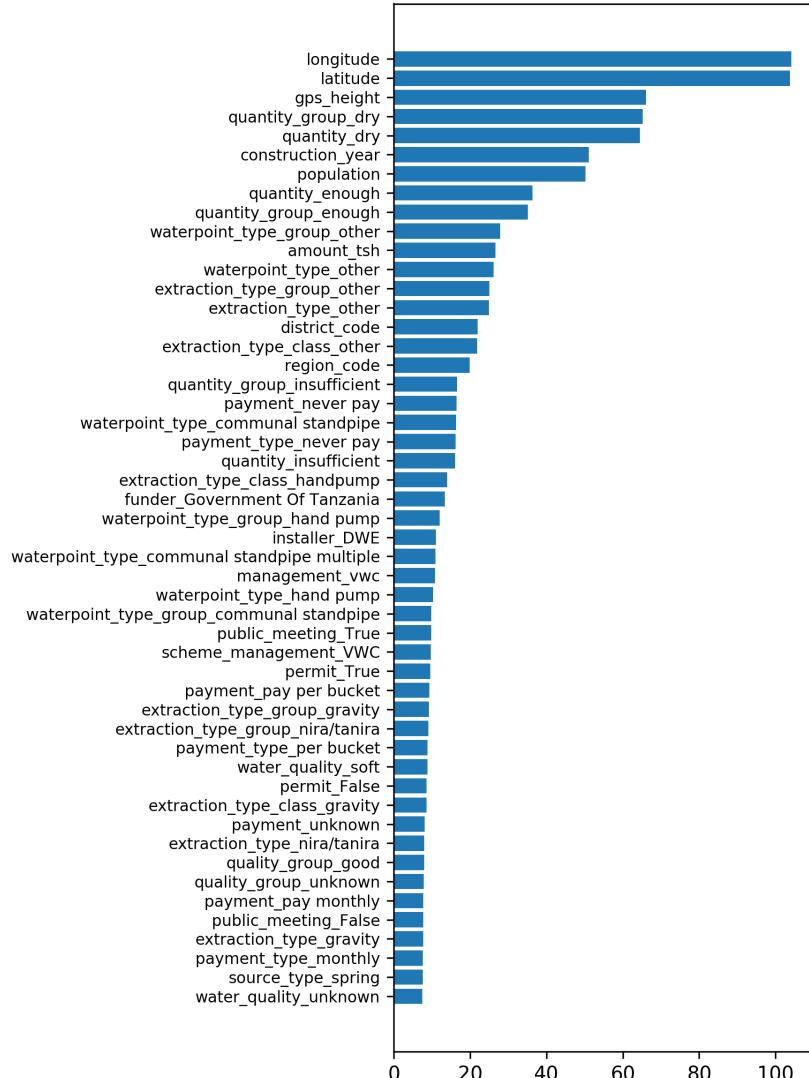


(a) 水泵的空間分佈

此題的目標是要透過 Taarifa 提供坦尚尼亞水利局的數據集，來預測抽水機是否為“待修”、“完好”、或是“故障”三種狀態之一。所以一開始，我們即在地圖上將這三種狀態視覺化 (Figure 1a)，並藉此觀察其分佈。

2 前處理與特徵工程

在觀察完視覺化的圖片後，我們即開始研究每個特徵所代表的意思與意義，加上參考 Driven Data 論壇內的一些討論。我們做了以下的處理。



(a) 特徵重要性

2.1 特徵工程

在研究完各個特徵所代表的意思後，先把一些內容重複的項目跟有太多缺失值的特徵先行刪除，例如“extraction_type”和“extraction_type_group”在記錄的內容相同，所以我們將後者移除；而“num_private”太多缺失值，並將它設定為控制值做實驗發現對精準度沒有太大影響，所以也將之移除。接著我們使用 feature importance 來找出最重要的幾個特徵 (我們在這邊只提供了最高頻的前 50 名)，再把次序比較下面較不重要的特徵刪除。

這裡還要特別說明的是，由於數據集中有連續數字和非數字的類別型特徵，於是我們使用的 pandas 的特殊函式 “`get_dummies()`” 來將所有屬於類別型的特徵做 one hot encoding 的操作，轉換成可以訓練的格式。

2.2 前處理

根據前面的 feature importance 圖可知 “`gps_height`” 和 “`constuction_year`” 是屬於很重要的特徵的效果。然而他們的缺失值不算少，所以這部分我們參考了討論區裡面有一位網友 `matthew_brown_iowa` 提供的方法 [2]，將這兩種特徵的缺失值用中位數來取代。

然後有個地方必須特別指出，經過我們的觀察，由於這塊區域是在坦尚尼亞，所以經緯度應該會在一個固定的範圍內，而根據訓練資料中顯示，在經度中會出現 $-2e^{-8}$ 的值；而在緯度中會出現 0。但這其實是用來代表缺失值而不是指經緯度分別是 $-2e^{-8}$ 和 0。

3 架構與做法

我們使用了三種架構，分別是 1) 極致梯度提升; 2) 隨機森林; 3) 深度神經網路。參考一些 kaggle 上的比賽，發現前幾名的方法都是用“極致梯度提升”，所以它自然就成為了我們的第一種方法嘗試，經過調整其學習度和循環的參數，我們成功使用它通過了簡單基礎線和強力基礎線。接著為了要分析不同架構下的效果，我們使用了一樣是基於決策樹的“隨機森林”，並且也嘗試了“深度神經網路”。

3.1 極致梯度提升

極致梯度提升是一個基於決策樹的訓練方式，藉由逐次加入一顆新的樹來調整前一次的結果以求更能適應訓練資料。我們使用“交叉驗證”的方式，想藉此觀察不同“提升”回數的效果與差異，於是產生了分析圖 3。從中可以發現，在訓練資料上的精準度只會越來越高，所以我們才事先切出驗證資料來做觀測，而驗證資料在大概接近 200 次時已經沒有太大的進步。

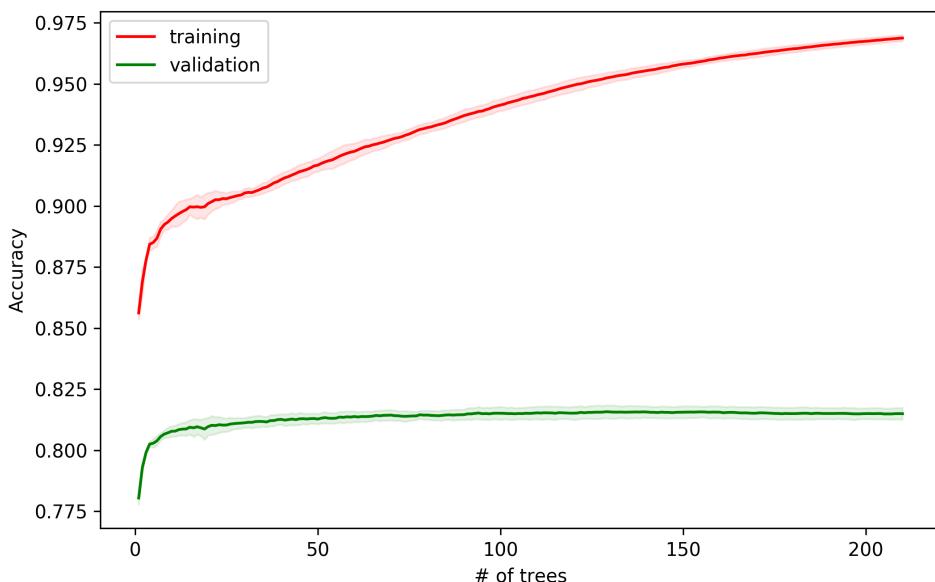


Figure 3: 極致梯度提升之訓練過程

然而，由於“梯度提升”的特性，此種模型比較不會產生“過度適應”的現象，所以最終在上傳網站做評分時，我們並沒有如做分析圖時那樣，先用“交叉驗證”的方式找出最佳的提升次數，接著再使用挑出來的最佳參數重新訓練一次產生答案，而是直接固定提升次數（我們總共提升了 210 次），這樣可以大量減低訓練時所花的時間。在此參數及仔細調整學習率和樹深度下，單一模型效果可以達到 0.81 左右，而最後調出來的參數如下。

- booster: gbtree
- objective: multi:softmax
- eta: 0.025
- max_depth: 23,
- colsample_bytree: 0.4
- eval_metric: merror,
- num_class: 4

3.2 隨機森林

我們測試的第二個模型是隨機森林。隨機森林整合多棵決策樹的變異量數，理論上愈多棵數效果會愈好，但樹的數量會直接影響到訓練所需的時間。因此以下為樹的數量與準確度作圖，以觀察足夠的樹的數量。從圖中 4 可以看到，在多於 25 棵樹後，準確度就沒有明顯的上升了。所以我們相信，500 棵樹已經足夠了。然而即使是使用了 500 棵樹，最終的準確度仍然只有大約 0.80 °(除樹的數量以外，其餘參數使用 sklearn 預設參數)

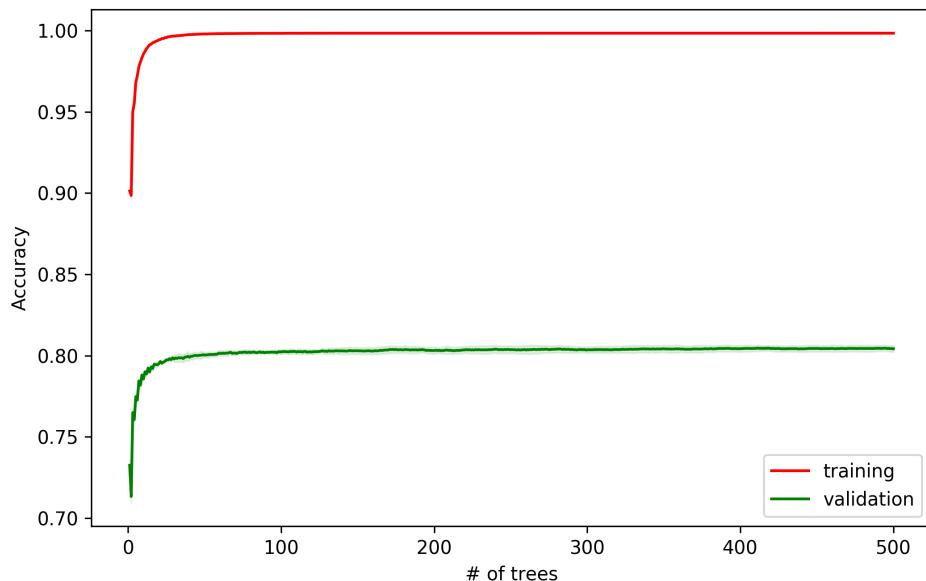


Figure 4: 隨機森林大小與精確度之關係

3.3 深度神經網路

使用架構

- Dense 1024
- BatchNormalization
- Leaky Relu
- Dropout 0.1
- Dense 512
- BatchNormalization
- Leaky Relu
- Dropout 0.1
- Dense 256
- BatchNormalization
- Leaky Relu
- Dropout 0.1
- Dense 3
- Softmax

並使用梯度下降法 Adam 以及 Categorical Crossentropy 作為損失函數。

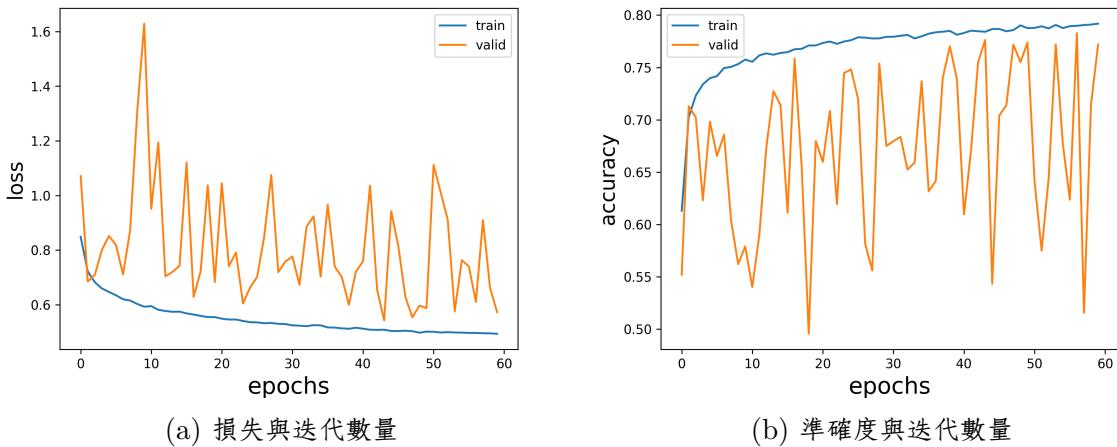


Figure 5: 深度學習網路的訓練過程

從訓練的過程來看，可以看到即使在訓練資料集的部份，神經網路的表現也沒有很好，始終無法達到 0.8 的準確度。同時在驗證資料集上，此深度學習網路的表現無論是損失函數或是準確度都波動得十分劇烈，且最好的表現也沒有很好。

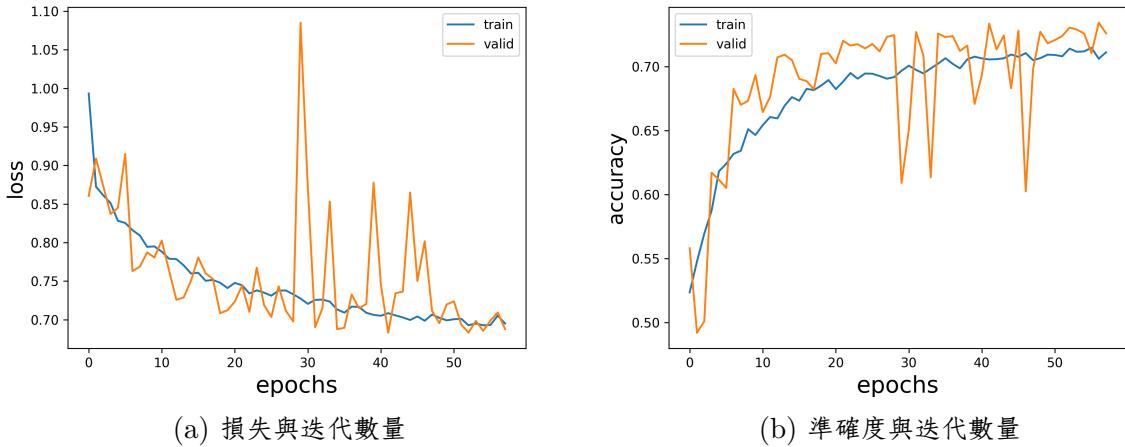


Figure 6: 深度學習網路的訓練過程 (使用 SeLU)

此外，我們也試了最近發表的激發函數 SeLU 來代替原先的 LeakyReLU。從圖 6b 中可以看到，相較於使用 LeakyReLU，測試集上的損失和準確度都較為平緩。然而，此神經網路在訓練資料集上依然沒有太好的表現。不過有趣的是，大約從第 10 次迭代後，此神經網路在驗證資料集的準確度都高於訓練資料集的準確度，但最終的準確度甚至不如使用 LeakyReLU 的神經網路。

4 討論

各別模型的分析如前面介紹所實驗過的模型時所述，在這邊將做整合的綜合分析。根據前段敘述，在單一模型的比較下，是以“極致梯度提升”有著最佳的效果。

Table 1: 單一模型精確度比較

	Accuracy
xgboost	0.81
random forest	0.80
DNN	0.73

由於在單一模型上升有限，於是我們使用了“集成算法”(ensemble)，希望能讓精確度再次上升。這部分由於其他模型怎麼調參皆未達到與極致梯度提升單一模型的效果，所以並沒有考慮把他們加入一起做集成。而是使用不同數量的極致梯度提升做集成同時調整深度與學習率做測試。

在這裡我們使用了 DMLC 提供的套件 [1]，在撰寫程式上來的方便許多，相關極致梯度提升的介紹也可以在參考資料中獲得。根據裡面文件說明，他本身會有隨機種子即使讓每次參數一樣，結果都會不一樣，所以我們特別將種子做設定，讓一樣參數的結果會一致好做比較。所以不同極致梯度提升的生成就是使用不同的種子但其他參數固定 (學習率、深度... 等)，測試了幾種挑幾個表列如下。

Table 2: 參數與分數之關係

	learning rate	depth	random seed range	# of round	Public Score
1	0.022	23	60 - 75	210	0.8271
2	0.025	23	60 - 68	220	0.8271
3	0.025	23	60 - 65	210	0.8274
4	0.025	23	60 - 72	220	0.8275
5	0.025	23	60 - 72	210	0.8275

5 分工分配

我們在最原始的分工分配是三個題目都做嘗試，看哪一個有過簡單基礎線與強力基礎線，分配是：

- Sberbank Russian Housing Market => 君獻、廷睿
- DengAI: Predicting Disease Spread => 義聖
- Pump It Up: Data Mining the Water Table => 子維

然後根據之後繼續衝高記分板的狀況跟潛在可以提升的程度，才統一集中火力一起專注在同一題分析特徵和找各種方法加強。在最後決定以 Pump It Up: Data Mining the Water Table 這題來做報告，所以以下分工分配是基於這個題目提供。

Table 3: Pump It Up 分工表

	程式	分析	報告
君獻		V	V
子維	V	V	
義聖	V	V	
廷睿		V	V

(在這邊“分析”包含了事前特徵工程時的調查與嘗試，以及在做報告時的作圖與解釋。)

References

- [1] *DMLC XGBoost*. URL: <http://xgboost.readthedocs.io/en/latest/>.
- [2] *matthew brown iowa*. URL: <https://community.drivendata.org/t/share-your-approach/65/26>.
- [3] *Pump It Up*. URL: <https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/>.