Machine Learning 2017 Final Project - Pump it Up: Data Mining the Water Table

Team name, members and your work division

NTU_b02705027_隨便B		
B04611038	劉記良	前處理、Model 設計、coding、上台報告
B02705027	陳信豪	畫各種圖、實驗一二三四、coding
R05942056	時丕澔	協助部分實驗三、coding

Preprocessing/Feature Engineering

1. 移除的 feature

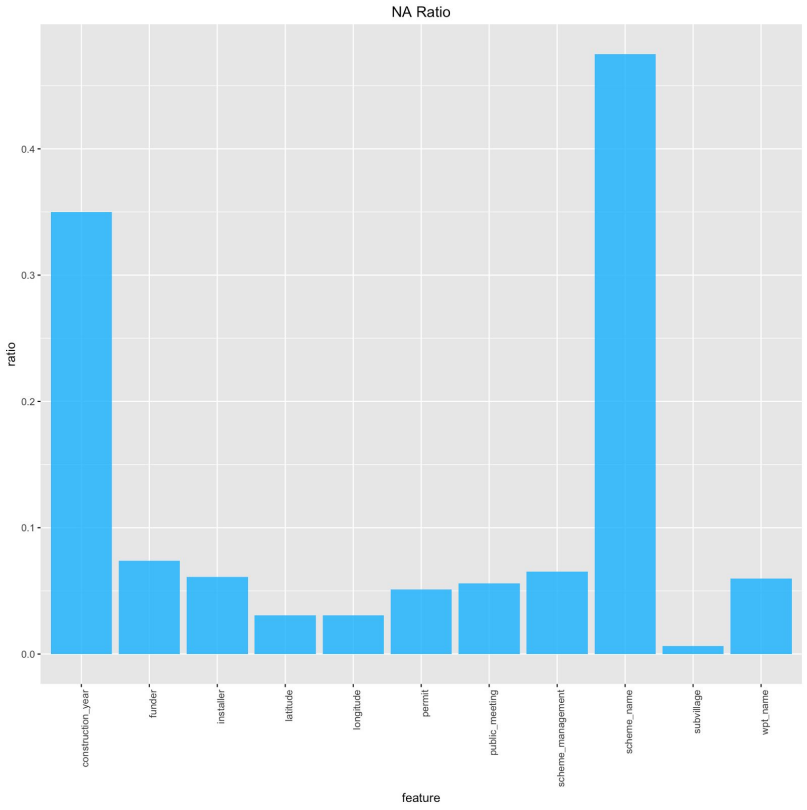
   參考了 driven data 內的討論區，我們得知下面這些 feature 是對結果較無效果的，或是已有重複且分類較細的 feature。

id	id 沒有經過有意的排序，呈現 random 的分部
num_private	不知道是什麼，drivendata 也沒有詳細說明
waterpoint_type_group	與 waterpoint_type 重複
source_type	與 source 重複
payment_type	與 payment 重複
extraction_type_group	與 extraction_type 重複
quantity_group	與 quantity 重複
management_group	與 management 重複
quality_group	與 water_quality 重複
wpt_name	waterpoint_type 比較重要
scheme_name	與scheme_management 重複

source_class	與 source 重複
date_recorded	不重要
recorded_by	資料單一，都是 GeoData Consultants Ltd
subvillage	與經緯度和 region 的概念重複

2. 缺值處理

資料中有缺值的欄位



*值為缺值佔總資料的比例

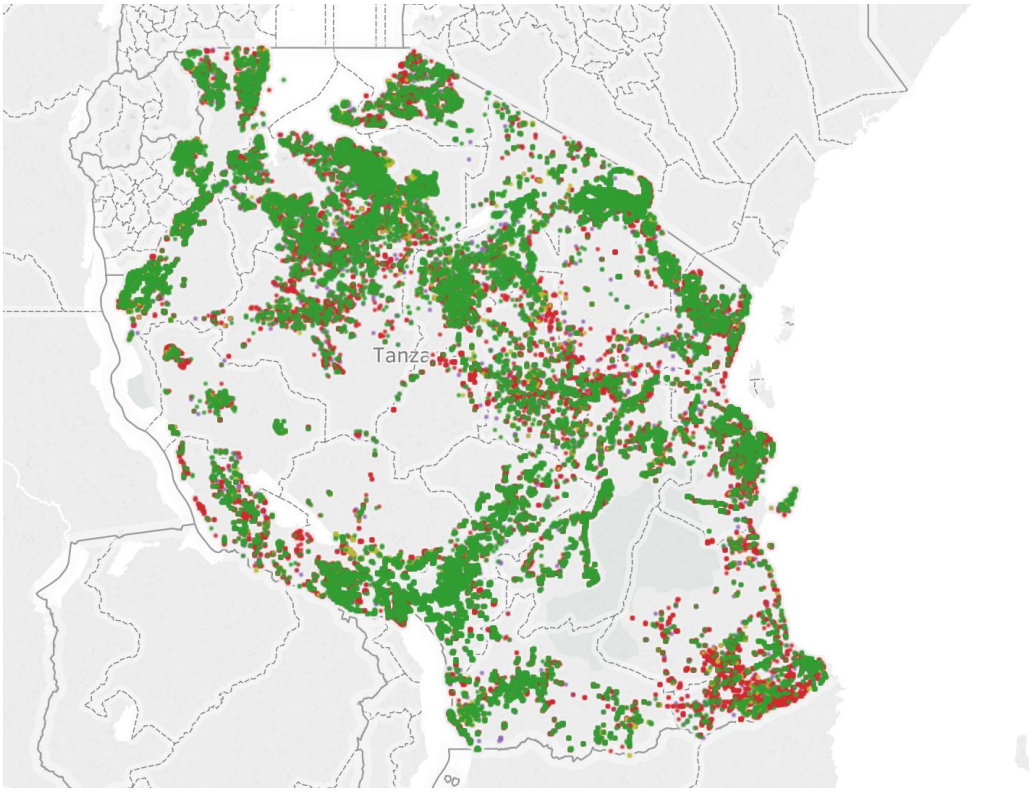
我們可以看到其中有不少我們採用的 feature，因此要做處理

longitude	值為 0 的是缺值，報告後面會說明處理方式
latitude	值為 > -0.1 的是缺值，報告後面會說明處理方式
funder	值為 '0' 的是缺值，以 'Other' 取代
installer	值為 '0' 或是 '-' 的是缺值，以 'Other' 取代
constuction_year	值為 0 的是缺值，以 median (2000) 取代

3. Label Encoding:
   利用 sklearn.preprocessing 的 LabelEncoder 將每一個 string 的 label 轉換成數字。

4. 經緯度預測 (keypoint)

   在討論區中我們發現了這張圖



(註：綠點為 functional, 紅點為 non functional, 橘點為 functional needs repair)

可以發現大部分的紅綠橘點是成區塊集中出現的，也就是說其實地點和結果是有成高度正相關的，所以如何填補經緯度這最具代表性的地理 feature 的 missing data 會是一個非常關鍵的地方。

而我們發現其他資料中，像是 region code 等是有機會把這些 missing data 給 predict 出來的。所以我們就用其他的地理性質 feature，架了 XGB model 把經緯度的缺值給預測出來，經過實作發現有 predict 出來的效果會比直接用 median 取代缺值好。之後亦嘗試使用 DNN 來預測，效果也不錯。再經過觀察後發現其實經緯度基本上就只是 region code 對到兩個數字而已，所以本來就不怎麼需要擔心 overfitting 的問題。

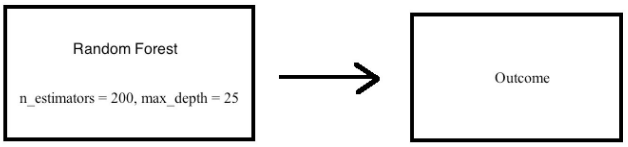
若是有時間，未來還有機會可以運用爬蟲與資料處理，直接的將這些 data 與 Google Map 給連接起來，近一步取得更準確的資料，不單單只是去用其他資料 predict，而是利用現有的資料庫與資源去將 missing data 更加有效率的填補。

4. One-Hot

我們嘗試將大部分的非數值 feature 轉換成 One-Hot encoding，這樣做的缺點是資料量會有點大，training 的 data 可以達到 3.3G 左右，testing 的 data 也有 800 多 MB。但優點是資料會變的比較簡單，而我們期待訓練出來的 model 會有較佳的 performance。(但根據實驗四的結果，發現其實並非如此）

Model Description

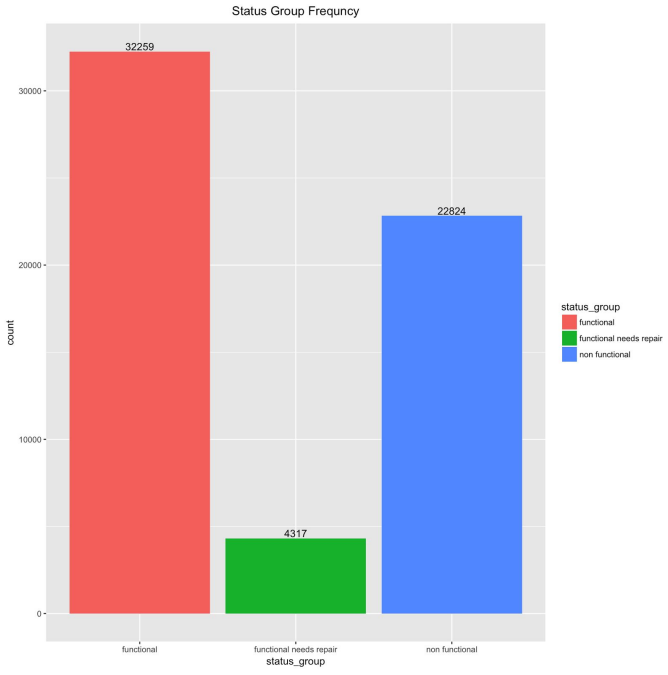
Model 1 Random Forest



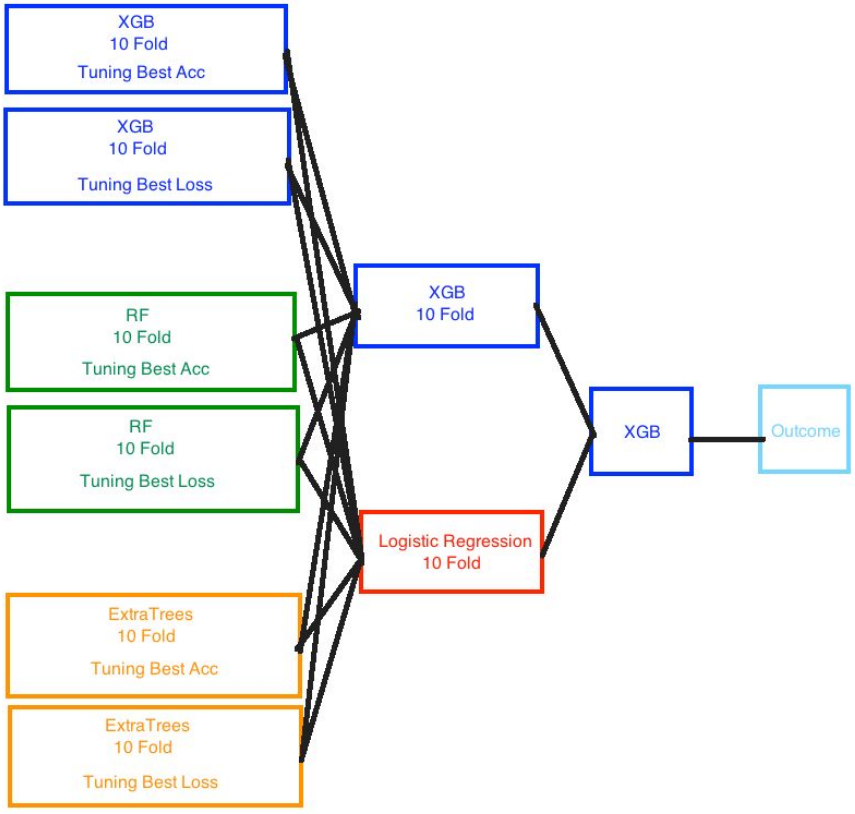
第一個 model 中我們採用較為單純的架構，只有用 skleran 中的 random forest。參數設定為 n_estimators = 200, max_depth = 25。

優點：運算量小、效果優良，結果利用 voting 所得，所以曲線平滑，與先前觀察經緯度的圖符合。

缺點: 不能很好的解決樣本數不平衡的問題，也剛正好是這個問題中最大的問題，我們可以看到在這次的 training data set 中，functional, non functional, functional need repair 的資料是極度不平衡的。



Model 2 Multiple Model and stacking



我們的 Model 2 主要可以分成三個 Stage1、Stage2、Blending 三個階段。

Stage 1

我們這邊個採用了 XGBoost、Random Forest、ExtraTrees 三個模型，另外又再往下分 Tuning 結果 validation accurary 最高者與 validation loss 最低者，所以總共會有六個 model。

參數為使用 Bayesian optimization 所得，設定如下:

XGB loss 最低:

```
XGBClassifier(max_depth=14,
              learning_rate=0.0588,
              n_estimators=250,
              objective='multi:softprob',
              nthread=8,
              gamma=0.6890,
              min_child_weight=7.6550,
              subsample=0.8,
              colsample_bytree=0.8)
```

XGB accuracy 最高:

```
XGBClassifier(max_depth=15,
              learning_rate=0.03599,
              n_estimators=385,
              objective='multi:softprob',
              nthread=8,
              gamma=0.6836,
              min_child_weight= 4.3704,
              subsample=0.8,
              colsample_bytree=0.8)
```

RandomForest loss 最低:

```
RandomForestClassifier(n_estimators=384,
                       min_samples_leaf=2,
                       max_features=0.5060,
                        max_depth=26)
```

RandomForest accuracy 最高:

```
RandomForestClassifier(n_estimators=346,
                       min_samples_leaf=5,
                       max_features=0.5112,
                       max_depth=25)
```

ExtraTreesClassifier loss 最低:

```
ExtraTreesClassifier(n_estimators=341,
                       min_samples_split=5,
                       max_features=0.7769,
                       max_depth=25)
```

ExtraTreesClassifier accuracy 最高:

```
ExtraTreesClassifier(n_estimators=387,
                       min_samples_split=3,
                       max_features=0.6636,
                       max_depth=25)
```

Stage 2:
我們在 Stage 2 採用了 XGBoost 和 Logistic Regression 兩種 model，並將 Stage 1 中六個 model 的結果 Stack 起來。

參數設定如下：

```
XGBClassifier(max_depth=7,
        learning_rate= 0.02358,
        n_estimators=189,
        gamma=0.07479,
        min_child_weight=3.0666,
        subsample=0.49698,
        colsample_bytree=0.9517,
        reg_alpha=0.2065,
        objective='multi:softmax')

LogisticRegression(C=1200,max_iter=800)
```

Blending:
最後，我們再次使用 XGBoost 將 Stage 2 中兩個 model 的結果 Stack 起來，得到最後的結果。

參數設定如下：

```
XGBClassifier(max_depth=7,
                learning_rate= 0.02358,
                n_estimators=189,
                gamma=0.07479,
                min_child_weight=3.0666,
                subsample=0.4970,
                colsample_bytree=0.9517,
                reg_alpha=0.2065,
                objective='multi:softmax')
```

優點: 準確度高

缺點: 跑的時間久、model 數量多且 model 檔案大、如何reproduce是個問題

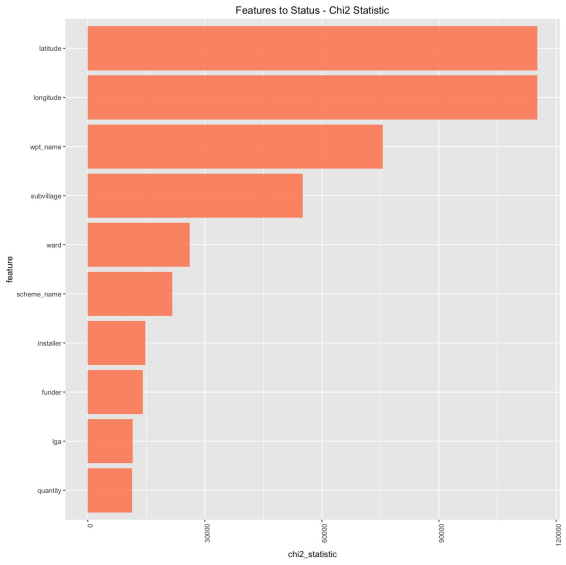
Experiments and Discussion

實驗與討論一：以不同方式探討 Feature 重要性（此部分以 R 語言實作）

(1) CHI2 Test

藉由 CHI Square Test 比較各個 feature 與 label status 的相關性。

經由 CHI Square Test 可以得到檢定值 Statics，Statics 如果愈高就代表與 label status 的相關性愈高。以下是 CHI2 Test 經排序後找出來的前十名 feature。

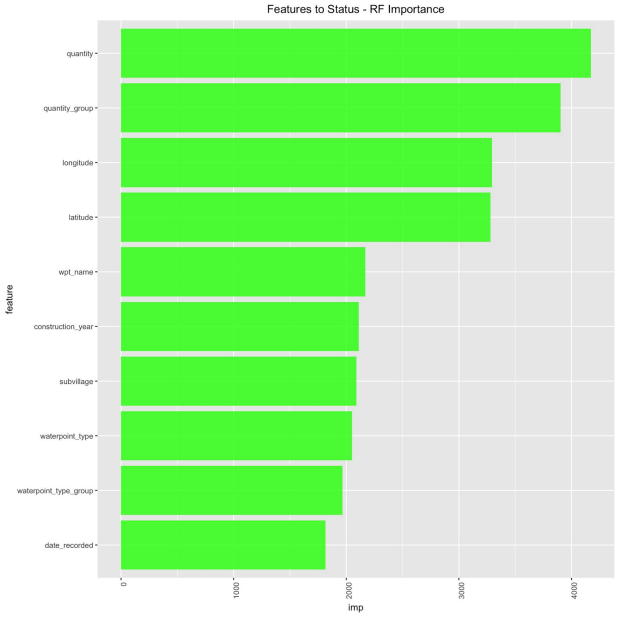


(2) RandomForest Importance

藉由 RandomForest Importance 探索各個 feature 對 model 的重要性。

經由 RandomForest Importance 可以得到檢定值 MeanDecreaseGini，MeanDecreaseGini 如果愈高就代表該 feature 對 RandomForest model 愈為重要。

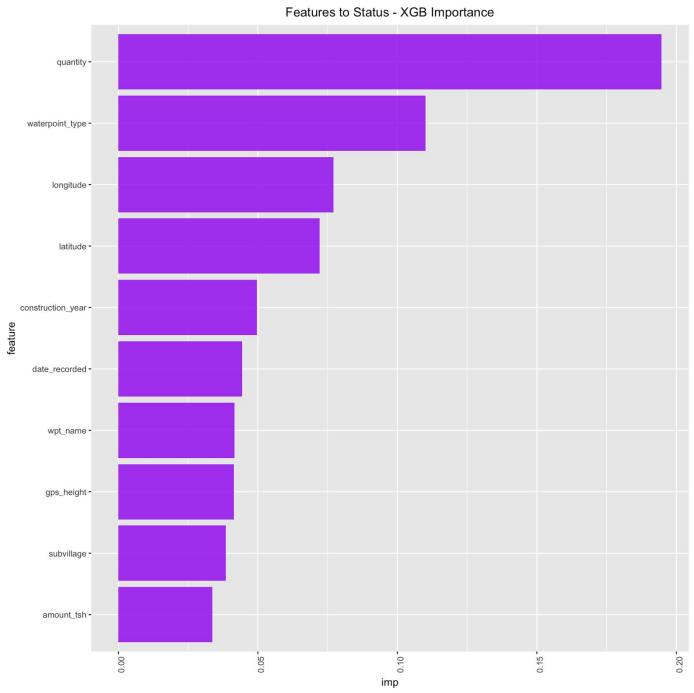
以下是 RandomForest Importance 經排序後找出來的前十名 feature。



(3) XGB Importance

藉由 XGB Importance 探索各個 feature 對 model 的重要性。

經由 XGBImportance 可以得到檢定值 Gain、Cover 和 Frequency，Gain 如果愈高就代表該 feature 對 XGB model 愈為重要。以下是 RandomForest Importance 經排序後找出來的前十名 feature。



結論：

我們從 CHI2 Test、RandomForest Importance、 XGB Importance 三個面向去探討了 feature 的重要性。並發現地理區位類型的 feature 非常重要，如 longitude、latitude、subvillage 都有在榜上。尤其是經緯度在三個榜上都是前五名。這或許可以解釋成因為只要有一個地方的水源有問題或是枯竭的話，就會連帶影響到整片地區的水井，因此地理環境就是影響結果的最重要的因素。

另外，時間也是影響結果的一大因素， construction_year 在 randomForest 和 XGB 都有上榜。其他重要的 feature 還有如 quantity、wpt_name 和 waterpoint_type 等。

實驗與討論二：10-fold Model 比較

在此我們探討， XGBoost、RandomForest 和 ExtraTrees 這三者的 model performance。方法是運用 10-fold cross validation，算出各個 model 十個 cv fold 的平均 accuracy 和平均 log loss。

10-fold：

將原本的 training dataset 亂數均分成 10 個小的 dataset，每次的 cv fold 會取其中一個小的 dataset 作為 validation/test dataset，其餘的作為真正的 training dataset，在Model 2中，我們最後會將這些結果疊起來作為下一階段的輸入。
三個 model 都會跑十次 fold，彼此之間對應的 train-test dataset 都相同。

feature採用：amount_tsh、gps_height、longitude、latitude、basin、region、region_code、district_code、lga、population、public_meeting、scheme_management、permit、construction_year、extraction_type、management、payment、water_quality、quantity、source、waterpoint_type

結果：

	XGBoost	RandomForest	ExtraTrees
cross validation log loss average	0.45899113	0.47378025	0.53027612
cross validation accuracy average	0.8181144	0.81358593	0.80622888
parameter setting	max_depth = 15 learning_rate = 0.03 n_estimators = 400 objective ='multi:softprob' nthread = 8 gamma = 0.5 min_child_weight = 5 subsample = 0.8 colsample_bytree = 0.8	n_estimators = 500 min_samples_leaf = 5 max_features = 0.5 max_depth = 25	n_estimators = 400 min_samples_split = 3 max_features = 0.6 max_depth = 25

*所有 model 的參數均有經過 tuning 或調整來嘗試達到最好的 performance
*log loss：logistic loss 或稱 cross-entropy loss.
*accuracy：(1/N) * ∑ (y_true == y_pred)

Model Performance：XGBoost > RandomForest > ExtraTrees
XGBoost 不愧是眾多人認同的王者，在我們比較的三個 model 之間也有最好的表現，其次 Random Forest 的表現只稍為落後 XGB，而 ExtraTrees 的表現就比較一般了。

實驗與討論三：有無將經緯度納入考量之比較

根據討論區上的說法，經緯度似乎是一個對結果影響很大的因素，因此我決定要對經緯度做一需處理，並假設了三種情形進行實驗。

情形一：不使用經緯度作為 feature
情形二：經緯度缺值以經緯度的 median 取代
情形三： XGB 訓練區域類型的 feature 來預測經緯度，藉以填補缺值
情形四：用 DNN 訓練區域類型的 feature 來預測經緯度，藉以填補缺值

情形三中使用的區域類型 feature 為 region、region_code、district_code、lga、gps_height、ward 這幾項，並透過 XGBRegressor 來進行 training 和預測。

而情形四中使用的區域類型 feature 和情形三一樣，只是換用 DNN model 來進行 training 和預測。

三種情形假設完後我們一樣透過 10-fold cross validation 來進行實驗，看平均的 accuracy 來決定 model performance。

feature採用：與實驗二相同，只是在 longitude、latitude 上有不同處理

結果：

	XGBoost	RandomForest	ExtraTrees
情形一	0.81249144	0.80831659	0.8008753
情形二	0.81704796	0.81346802	0.80629616
情形三	0.8181144	0.81358593	0.80622888
情形四	0.81878764	0.81434353	0.8061783

*欄位中的值為 10-fold cross validation accuracy average

從實驗結果可以看出經緯度確實影響了準確度，把經緯度拿掉會降低 model performance。

另外，比較情形二至四， 以 XGBoost 或 DNN 預測經緯度缺值會稍微比用 median 填補缺值好一些，雖然不甚明顯，但我們之後仍會使用預測的資料來做處理。

實驗與討論四：有無 One-Hot Encoding 之比較

我們運用了 sklearn 的 OneHotEncoder 來將非數值的參數做 One-Hot Encoding。

接著，我們與實驗二相同以 10-fold cross validation 來進行實驗，看平均的 accuracy 來決定 model performance。

feature採用：與實驗二相同，但拿掉了 funder、installer 和 ward，經緯度的處理是將缺值以預測值取代。

由於只是實驗的部分，我們在這裡拿掉 funder、installer 和 ward 這三個參數。原因是這三個 feature 分類的較細，One-Hot 後會導致資料量過大。不拿掉這三個參數，One-Hot feature 總數為 4758， 拿掉後 feature 總數為 248，而為了避免 One-Hot 後的 model 跑太久，我們決定拿掉。

結果：

	Feature 數目	XGBoost	RandomForest	ExtraTrees
One-Hot	248	0.81373701	0.8104882	0.80334997
No One-Hot	21	0.81579106	0.81082487	0.80483142

*欄位中的值為 10-fold cross validation accuracy average

我們原本認為透過 One-Hot 編碼可以提高我們的準確值，但根據實驗結果來看並非如此，無論是 XGBoost、RandomForest 和 ExtraTrees，三者的 10-fold cross validation accuracy average 皆比原本沒有 One-Hot 來得的低。

另外， One-Hot 後的 feature 數量比原本多上許多，會導致 model 訓練得比原本更慢。以 XGBoost 速度而言，原本平均 150 秒一個 cv fold，One-Hot 後變成要 400 秒一個 cv fold。