國立台北大學資訊工程學系專題報告

如何在網路上減低自己被人肉搜索的風險？

專題組員：王瑞翊、王靜瑩

專題編號：PRJ-NTPUCSIE-104-005
執行期間：104 年 7 月至 105 年 6 月

1. 摘要

在網路上留下自己的個人資訊，有時可能不覺得危險，因為大家都這麼做，但有心人就能把這些零碎的資訊結合在一塊分析，進而找出你的現實身分，也就是所謂的「人肉搜索」，本專題著重於探討哪些私人資訊被放在網路上後，被肉搜成功的機率將大幅提高，進而想出應對的方法。

2. 簡介

現今是社群網站的時代，人們都已沉浸於其中，習慣以它們接收快速的訊息流和作為社交的媒介，亦經常把自己的最新動態放在網路上，跟朋友或是不認識的人分享，日復一日，每個人在網路上已留下了為數可觀的「足跡」。

然而，一些有心人便會想辦法利用這些足跡，加上其他網路資訊像是部落格、論壇，甚至是網路 IP 位址，實際找出你這個人，你的居住地、學校或公司，一些你本來不打算在網路上洩漏的私人資訊，可能都會因為你經常的按讚、發表文章、發表照片、發表影片和到景點打卡[1]，而一步步讓你處在容易被人肉搜索的危險中。

於是我們希望透過這次專題，分析社群網站使用者在網路上所提供的資訊，來看出什麼樣的資訊會使人們容易曝光，並且想出如何防範的作法。我們主要著重在台灣本土第一大的資訊社交平台 PTT 以及全球廣泛使用的社群網站 Facebook，利用使用者在此兩者上所提供的資訊，經過前處理後再撰寫程式來計算兩邊資訊的相似度，再使用資料探勘軟體來對資料作進一步的分析。

3. 專題進行方式

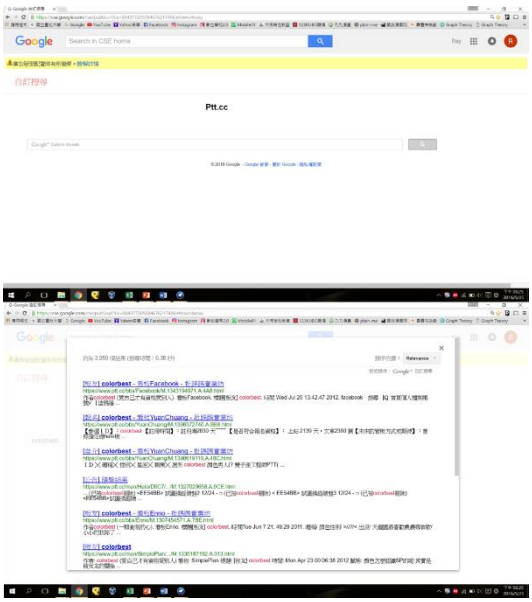
我們的專題分成以下四個步驟：

(1) 資料蒐集
(2) 資料前處理
(3) 利用分析工具分析
(4) 觀察結果並討論

以下針對每個步驟說明實驗方法。

(1) 資料蒐集

利用人工的方式分別蒐集 PTT 以及 Facebook 上的個人資料，PTT 上的資料來源是 Facebook 版裡的自我介紹，利用有留下 Facebook 帳號連結的使用者資料，扣除 Facebook 帳號連結已失效的幾位，總計有 190 筆的使用者資料，並且利用 Google 自訂搜尋（如圖一所示），擷取使用者在 PTT 上的所有發文和推文記錄。



圖一：Google 自訂搜尋

為了蒐集資料的統一性以及後續處理的方便，我們討論並訂出了 11 個我們覺得需要蒐集的個人屬性，這 11 個屬性分別是：姓名、性別、生日、血型、出沒地、學校、公司、職業、興趣、感情狀態和網址，若無該屬性資料的話則該欄填空值。並且為了怕兩人定義的混淆，針對出沒地、學校、興趣和感情狀態這四項更另外定義了記錄方式。

- 出沒地：採用最大範圍的概括城市來代表。例如：若地點為白河，則出沒地則記錄為台南市，三峽區的話則記錄為新北市。
- 學校：針對相同學校採用統一的稱呼。例如：北一女一律記錄為北一女中，建中一律記錄為建國中學。
- 興趣：我們定義了一系列的興趣分類來使記錄不會因為蒐集資料的人員差異而產生分歧（如表一所示）：

表一：興趣分類

閱讀寫作	聊天哈拉	電玩對戰	運動釣魚	動畫漫畫
投資理財	兜風閒晃	古董收藏	園藝花卉	逛街購物
音樂欣賞	星象命理	下棋彈琴	美食烹調	攝影繪畫
遊山玩水	發呆睡覺	豢養寵物	電腦網路	登山健行
電視電影	塑身美容	語言學習	唱歌跳舞	吃喝玩樂

- 感情狀態：我們的記錄方法分為以下幾項：單身、有男友、有女友、已婚、已婚有小孩。

以下為我們蒐集 PTT（如圖二所示）和 Facebook 資料（如圖三所示）的格式：

2_top_PTT - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
1:王小明;煞气a明;Min Wang                          //名稱
2:男                                                //性別
3:1970/10/31                                        //生日
4:O                                                 //血型
5:台北市;新北市                                     //出沒地
6:建國中學;台灣大學                                 //學校
7:老師                                              //職業
8:信義國小                                          //公司
9:動畫漫畫;電玩對戰;運動釣魚;閱讀寫作;電視電影     //興趣
10:單身                                             //感情狀態
11:http://minwang1031.pixnet.net/blog               //網址
```

圖二：資料蒐集格式（PTT 資料）

87_top_FB - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
1:王陽明                                            //名稱
2:男                                                //性別
3:1970/10/31                                        //生日
4:O                                                 //血型
5:台北市                                            //出沒地
6:台灣大學                                          //學校
7:                                                  //職業
8:                                                  //公司
9:動畫漫畫;電玩對戰;運動釣魚;電視電影               //興趣
10:                                                 //感情狀態
11:                                                 //網址
```

圖三：資料蒐集格式（Facebook 資料）

(2) 資料前處理

為了之後分析的需要，我們使用程式來計算不同資料中同屬性

的相似度(similarity)。針對不同屬性的特點，我們為不同的屬性定義了不同的相似度計算方式（空值不計）：

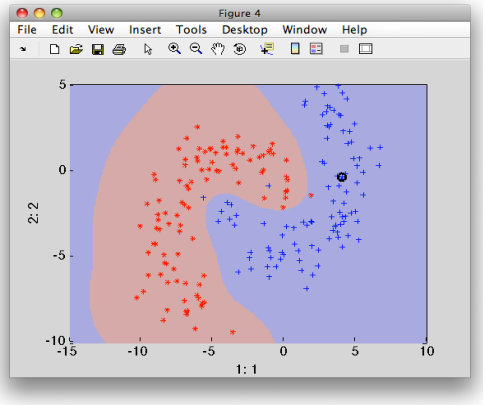
- LCS（Longest Common Subsequence)：用於名稱，e.g. Ruby Wang VS Ruby0701，最長子字串為 Ruby，所以相似度為 $\frac{4}{12} = \frac{1}{3}$ 。
- Binary（1 或 0)：用於血型、公司、職業、感情狀態和網址，以感情狀態為例：有男友和有女友，儘管「有」和「友」兩字相同，但在意義上卻是截然不同，所以相似度為 0。
- Jaccard：主要概念為 $\frac{\text{交集}}{\text{聯集}}$ ，用於出沒地、學校和興趣，已出沒地為例：台北市;彰化市 VS 台北市;新北市，則相似度為 $\frac{1}{3}$ 。
- 自我定義：用於生日，e.g. 月和日若有對上，相似度為 0.8；年、月和日若都有對上，相似度為 1。其他則為 0。

前處理後的範例（如圖四所示）：

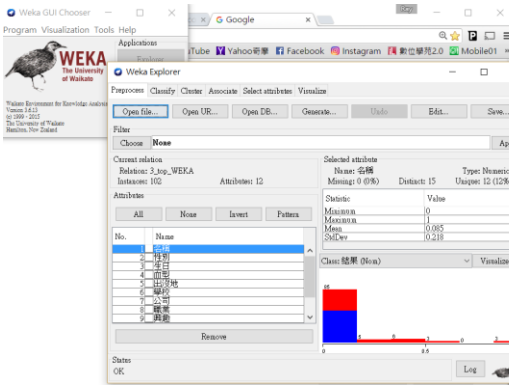
圖四：用程式將資料轉換成分析工具所適用之格式

(3) 利用分析工具分析

我們總共使用兩種分析工具，分別為 libSVM[2]（如圖五所示）和 Weka 資料探勘軟體（如圖六所示）。**libSVM（A Library for Support Vector Machines）**為一個易於使用和快速有效的 SVM 模式識別與回歸的軟體，藉由將資料點投射到高維空間，並且在不同類別中找出辨識分割線的一種機器學習方法。**Weka** 以 java 為基礎的資料探勘(Data mining)與機器學習（Machine learning）軟體，整合了大量的資料探勘演算法。



圖五：libSVM 範例



圖六：Weka 範例

此專題我們使用它們兩者中的三種演算法來分析：

- SVM（Support Vector Machines）：藉由將資料點投射到高維空間，並且在不同類別中找出辨識分割線的一種機器學習方法。
- ADTree（Alternating Decision Tree)[3]:Decision Tree 方法的一種，藉由整合弱的 classifier，給予強的 classifier 較高的權重，以達到更好的 training 結果。
- BFTree(Best-first decision tree)[4]：在所有可以分開的節點中選取讓分開(split)後的亂度降低最多的節點。

實驗以 N-fold cross-validation 進行，N設為 10，意思即是每次取 10%的資料，將它們的結果（同一人或非同一人）隱藏起來，然後取剩餘的 90%資料來做 training，學習這 90%的資料有什麼特性，之後再用此特性來檢測一開始取的 10%資料，看這個 rule 的準確率有多少。

(4) 觀察結果與討論

分析出幾次結果後，我們依照結果數值的高低和分布情形，思考造成它們結果如下的原因，並且調整資料的分布和屬性的參照，一一記錄下來，甚至由數據結果來評估使用者們的心理。

4. 主要成果與評估

在進入成果的講述之前，我們先說明要如何證明高、中和低資料量真的符合高、中和低，我們撰寫程式來跑我們所抓到的 190 位使用者有提供的 PTT 資料和 FB 資料。以確認各個使用者屬性的填寫率是否符合我們的預期，經過我們的驗證，發現的確符合我們的歸類（如表二所示）。高資料量的使用者 PTT 資料和 FB 資料填寫率最高（59.45%），中資料量的使用者填寫率次之（50%），而低資料量的使用者填寫率最低（44.12%）。

表二：資料提供率

高資料量	59.45%
中資料量	50%
低資料量	44.12%

不過即使高資料量的使用者資訊提供率最高，但若是 PTT 和 Facebook 的資料對應率不高，也是沒有意義的。因此接下來，我們也撰寫程式來確認高、中和低資料使用者的資訊對應率。驗證之後發現，發現高、中和低資料的對應率大致上也是呈現高、中和低的趨勢（如表三所示）。

表三：資料對應率

Label 1	高資料	中資料	低資料
名稱	0.33	0.22	0.06
性別	0.98	0.92	0.88
生日	0.22	0.06	0.04
血型	0.06	0.03	0
出沒地	0.49	0.4	0.22
學校	0.37	0.09	0.08
公司	0	0	0
職業	0.02	0	0
興趣	0.84	0.76	0.69
感情狀態	0.06	0.08	0.04
網址	0.02	0	0

確認資料的蒐集與歸類沒問題之後。即進入我們的成果討論。在本次專題中，我們使用 LibSVM、ADTree 和 BFTree 來對全部資料共 190 筆做訓練(training)。而因為總共是 190 位使用者的 PTT 和 Facebook 資料，所以對應的情況總共有 190*190 = 36100 筆。在這之中，對應到是同一個使用者的是 190 筆，除外的皆是不屬於同一個使用者的資料。因此我們使用程式，在餘下的三萬多筆資料中用隨機挑出 190 筆來和 190 筆一起做訓練，總共產生出 5 組隨機數據（如表四所示）。觀察數值之後，我們發現 ADTree 的平均結果最佳，標準差也最小，因此為了確保實驗的準確度以及可信度，我們決定之後的實驗都以 ADTree 為主。

表四：全部資料 training 結果

	LibSVM	ADTree	BFTree
Train1	68.95%	75.53%	73.42%
Train2	72.89%	74.21%	71.32%
Train3	71.32%	76.84%	75.53%
Train4	72.63%	77.11%	75.79%
Train5	71.05%	75.26%	75%
平均	71.37%	**75.79%**	74.21%
標準差	1.28%	**0.97%**	1.51%

而 ADTree 的準確率也會因高、中和低資料量的分別，準確率大致上也是呈現高到低的情況（如表五所示）。

表五：ADTree 的 training 結果

	高資料	中資料	低資料
Train1	86.27%	68.18%	68.63%
Train2	81.37%	69.32%	59.80%
Train3	79.41%	70.45%	60.78%
Train4	78.43%	69.32%	69.61%
Train5	81.37%	71.59%	70.59%
平均	81.37%	69.77%	65.88%

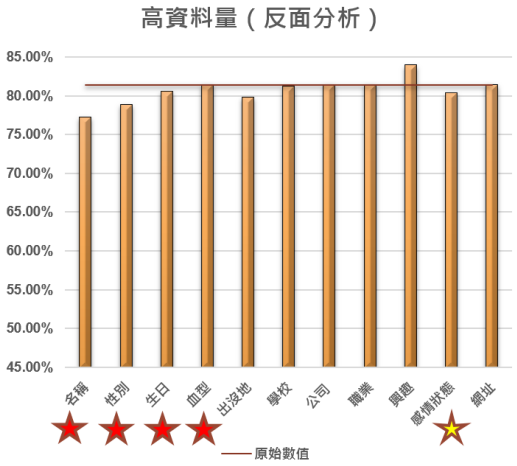
接下來，我們開始分析，除了資料對應率的高低會影響準確率外，某些屬性(名稱、性別、生日和出沒地等等)是否也會對於準確率產生重要的影響呢？因此我們對於高、中、低和全資料量做了正面分析和反面分析（如表六所示）。

表六：分析法

正面分析	反面分析
一次只看一個屬性的相似度，看只有各個屬性單一情況下的準確率誰為佳。	以全屬性所計算之準確率為基準，每次拿掉一個屬性的相似度資訊，看哪個屬性被拿掉後準確率會下降，表示其為重要屬性。

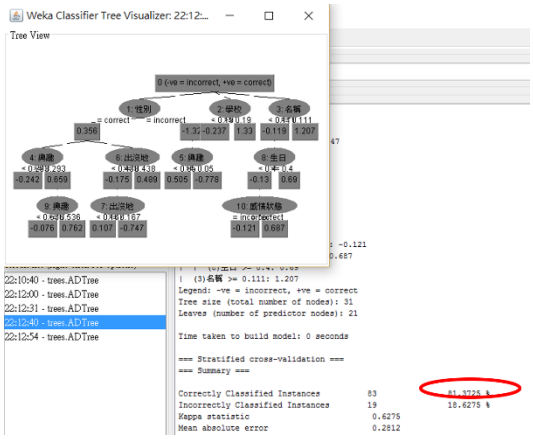
在正面分析下，發現光單看性別、學校、名稱、出沒地、生日、興趣和血型的準確率都超過 50%(如圖七所示); 而在反面分析下，發現拿掉名稱、性別、生日、出沒地、學校或感情狀態時，都會使得準確率下降（如圖八所示）。在我們歸納兩個分析結果後，可以得知 ：名稱、性別、生日、出沒地和學校為重要的屬性。

圖七：高資料量正面分析

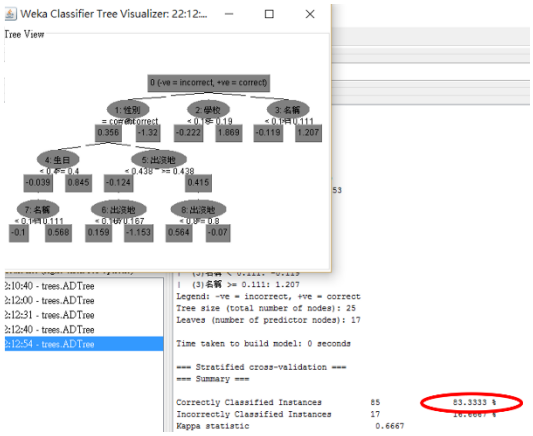


圖八：高資料量反面分析

為了證明所分析出的重要屬性是正確的所做的實驗。我們使用 weka 以及高資料量的使用者資料來做實驗。被歸類為高資料量的使用者共有 51 名，也就是 label 1(PTT 和 Facebook 兩邊為同一使用者)的資料有 51 筆，配上另外 random 出 51 筆沒對上的資料。上圖是原始數值，也就是以 11 個屬性下去訓練的結果，準確率為 81%（如圖九所示）；下圖則是用我們在上一段所提到的正面分析和反面分析所得到的結果來做訓練，也就是光用名稱、性別、生日、出沒地和學校來做訓練。我們可以發現，儘管只剩 6 個屬性，但準確率卻提高為 83%（如圖十所示），故可以驗證我們實驗結果出來的重要屬性為正確。



圖九：weka training 高資料量全屬性



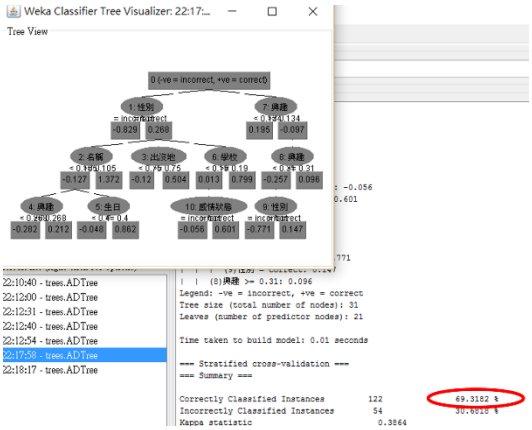
圖十：weka training 高資料量重要屬性

對於中資料量，在正面分析（如圖十一所示）和反面分析（如圖十二所示）後，得知名稱、性別、生日、血型、出沒地和學校為中資料量的重要屬性。

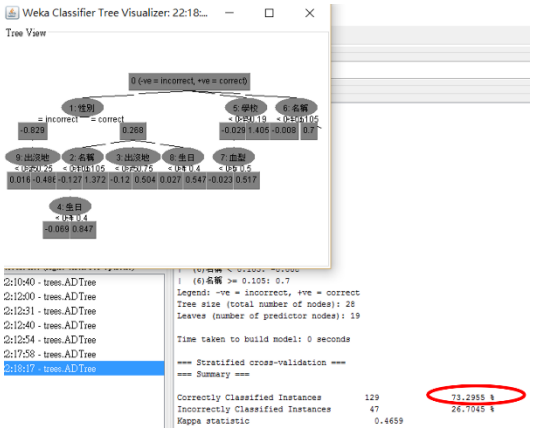
圖十一：中資料量正面分析

圖十二：中資料量反面分析

在 weka 的 training 下，中資料量以全屬性來 training 的準確率為 69%（如圖十三所示），而只用重要屬性 training 的準確率為 73%（如圖十四所示）。

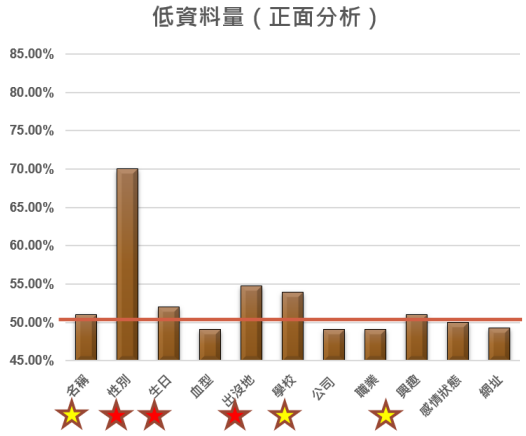


圖十三：weka training 中資料量全屬性

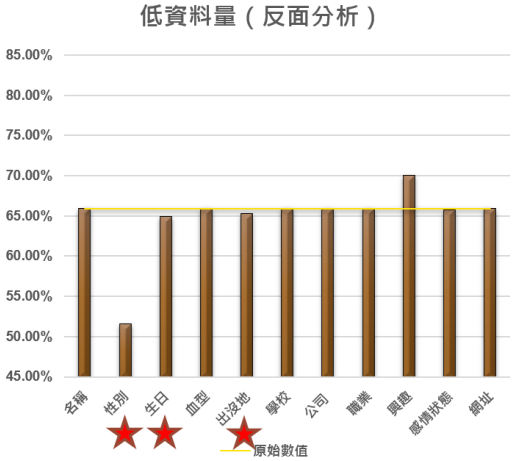


圖十四：weka training 中資料量重要屬性

對於低資料量，在正面分析（如圖十五所示）和反面分析（如圖十六所示）後，得知性別、生日、出沒地為低資料量的重要屬性。

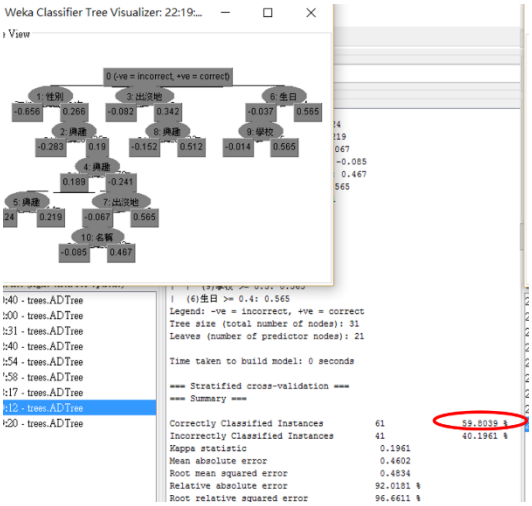


圖十五：低資料量正面分析

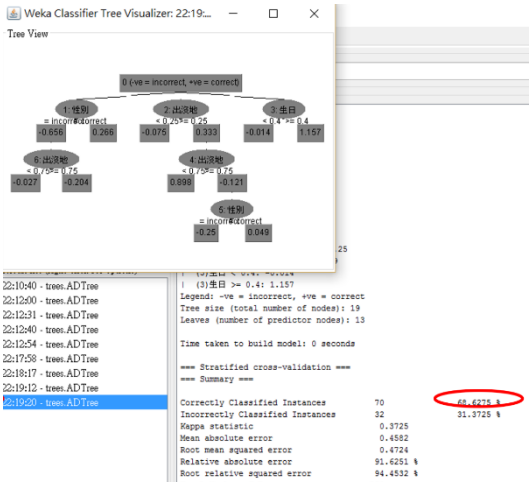


圖十六：低資料量反面分析

在 weka 的 training 下，低資料量以全屬性來 training 的準確率為 60%（如圖十七所示），而只用重要屬性 training 的準確率為 68%（如圖十八所示）。

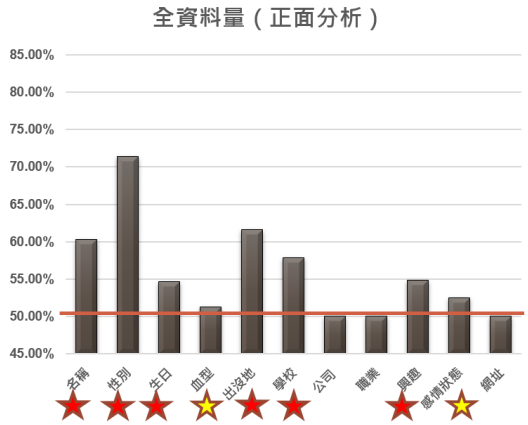


圖十七：weka training 低資料量全屬性

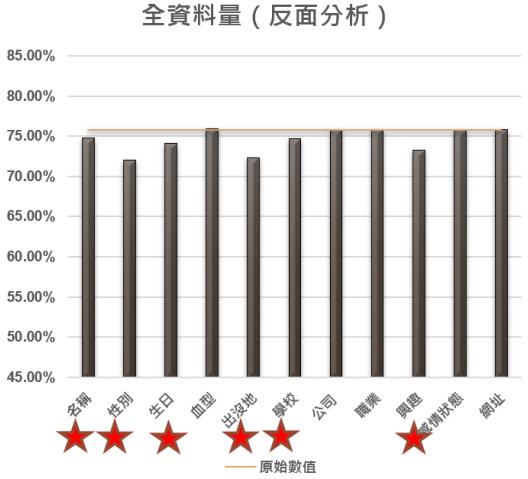


圖十八：weka training 低資料量重要屬性

對於全資料量，在正面分析（如圖十九所示）和反面分析（如圖二十所示）後，得知名稱、性別、生日、出沒地、學校和興趣為全資料量的重要屬性。

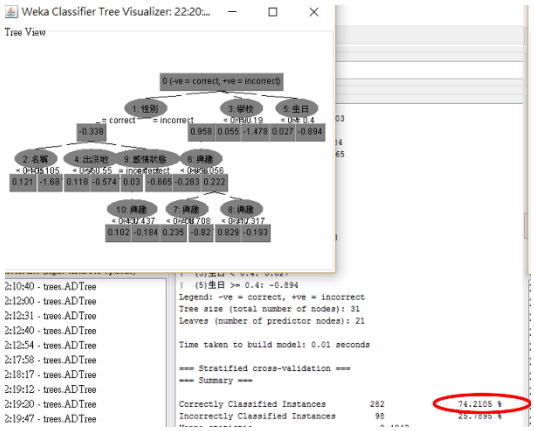


圖十九：全資料量正面分析

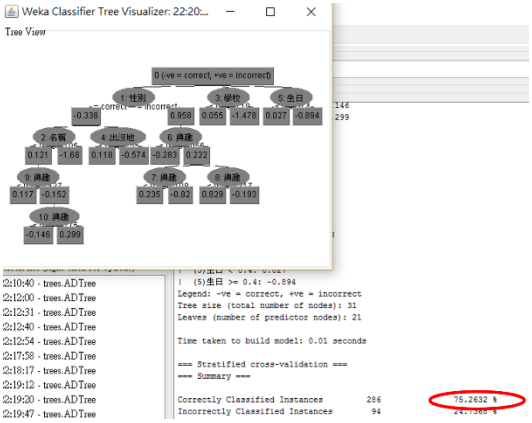


圖二十：全資料量反面分析

在 weka 的 training 下，全資料量以全屬性來 training 的準確率為 74%（如圖二十一所示），而只用重要屬性 training 的準確率為 75%（如圖二十二所示）。



圖二十一：weka training 全資料量全屬性



圖二十二：weka training 全資料量重要屬性

最後我們統整高、中、低和全資料量的結果，發現皆為重要屬性為**性別**、**生日**和**出沒地**此三項。

5. 結語與展望

實驗一開始我們拿高、中和低資料量做原始數值分析的時候，發現高資料量的準確率恆高於中低族群，經由這點我們可以發現 ： 高資料量族群不但提供資料多，且資料對應率也高，表示他們通常是喜歡在網路上展現自己的族群，不在乎個人隱私；而中資料量的使用者雖然給的資料比起低資料量族群多，但是能夠對應上的資料反而不見得多，例如從幾次訓練的數值觀察，可以發現低資料量的準確率反而比中資料量的來得高。表示中資料量的族群比較懂得保護自己的隱私，雖然填不少資料，但在 PTT 和 Facebook 兩邊給的資料卻有一定的不一致，從而降低了兩邊的相似度 ； 而低資料族群雖然覺得提供資料少就能降低被人肉搜尋的風險，但在實驗中卻顯示，某些屬性卻常在兩個平台上都提供，導致資料對應上的機率反而有時比起中資料量來得高。而在此實

驗中所找出的三個重要屬性：性別、生日和出沒地，它們剛好是現代人們覺得很習以為常公開的資訊，例如在網路上提供自己的性別和生日以便和他人交際，和時常透過「打卡」來讓自己的出沒地曝光。但在實驗結果顯示，若同時有這三個屬性的存在，被肉搜出來的機率將大幅提高。

因此我們可以得到以下的結論：避免被人肉搜尋的方法，就是在不同的網路平台上填寫不同的資料，避免兩邊的資料對應上；或是只填不重要的屬性，並且避免同時填性別、生日和出沒地。

對於這個發現，將來希望可以發展出更進一步的應用。或許對於資訊安全或者是網路犯罪的防範等等也能有所幫助。

6. 銘謝

特別感謝指導教授的指導以及提供資料給我們使用的網友們。

7. 參考文獻

[1] S Liu, S Wang, F Zhu, J Zhang, and R Krishnan. "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling." Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014.

[2] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology (TIST) 2.3 (2011): 27.

[3] Freund, Yoav, and Llew Mason. "The alternating decision tree learning algorithm." icml. Vol. 99. 1999.

[4] Shi, Haijian. "Best-first decision tree learning. " Diss. The University of Waikato, 2007.