Homework 4

`TA email: chiehyu.pan@gmail.com`

Problem

巡迴演唱結束回台灣的BlueGreen在觀看Instagram發布的貼文時發現有許多酸民留言說：

> 「唱的很好，下次別唱了。」
>
> 「這個表演看不出有用這麼多經費，經費都拿去吃慶功了是嗎？」

諸如此類的留言層出不窮。BlueGreen懷疑是他們的競爭對手CyanWhite使用網軍來惡意抹黑他們。他心想：

> 「我辛辛苦苦編列預算，立法院表決通過，安排人員出去表演，結果現在我得看人臉色？」
>
> 「那我不就成跪著要飯的了嗎？」
>
> 「不行不行，萬萬不可！」
>
> 「我是想站著，還把錢掙了！」

因此，BlueGreen收集Instagram上真實的帳號以及假帳號的資訊，企圖使用Decision Tree (決策樹) 來辨別真假帳號，藉此打擊網軍。

Dataset

- Instagram fake spammer genuine accounts
- 120 筆資料
- 共有 12 columns，其中 `fake` 是類別 Class，其餘則是特徵 Feature
  - **profile pic：用戶有沒有頭貼**
  - **nums/length username：帳戶名稱中數字所佔比例**
  - **fullname words：用戶名稱單詞數**
  - **nums/length fullname：用戶名稱中數字所佔比例**

- **name==username：帳戶名稱與用戶名稱是否一致**
- **description length：自介長度**
- **external URL：是否有外部網址**
- **private：是否為私人帳號**
- **#posts：有幾則貼文**
- **#followers：粉絲數**
- **#follows：追蹤人數**
- **fake：是否為假帳號**

Assignment Description

從獲取資料到分析資料，本次作業分為五個步驟：

1. 了解資料
2. 前處理
3. 建立模型
4. 優化
5. 解釋

以下分別會在 coding problem 和 report 中說明以上步驟要做的事情。

Coding problem: Learning Decision Tree - 40%

1. **了解資料：**
   - 目的在理解資料的內容、含義
   - 可以使用 `Numpy` 、 `Pandas` 以及繪圖 library `Matplotlib` 和 `seabon` 等來做資料處理與視覺化，方便以圖形得知資料蘊含的內容

> 💡 這個步驟可以知道資料有沒有缺少數值、每個欄位 Column 的屬性是離散或連續、數值或文字、資料有沒有偏頗，哪個部分特別多等等。

2. **前處理：**
    - 目的是將資料處理成適合模型的輸入
    - 可以使用 `Numpy` 、 `Pandas` 以及繪圖 library `Matplotlib` 和 `seabon` 等來做資料處理與視覺化，方便以圖形得知資料處理前後的差異

> 💡 在上個步驟得到資料屬性後，可以根據特性做處理。例如，發現資料有缺失，可以使用插值法補齊或是刪除該筆資料；當模型需要離散的資料，將連續型資料切分轉為離散型資料；文字資訊轉為數字型態；oversampling / undersampling 補齊正反資料等等。

3. **建立模型：**
    - 將資料依據 8:2 分成 train 和 test sets，可以使用 `sklearn` 的 `train_test_split()`
    - 建立 Decision tree 模型
    - 須依照講義的演算法實作（需手刻演算法，不可使用`sklearn` 或其他 library 快速建立）：

function DECISION-TREE-LEARNING($examples, attributes, parent\_examples$) **returns** a tree

if $examples$ is empty **then return** PLURALITY-VALUE($parent\_examples$)
else if all $examples$ have the same classification **then return** the classification
else if $attributes$ is empty **then return** PLURALITY-VALUE($examples$)
else
  $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE($a, examples$)
  $tree \leftarrow$ a new decision tree with root test A
  for each value v_k of A **do**
    $exs \leftarrow \{e : e \in examples$ **and** $e.A = v_k\}$
    $subtree \leftarrow$ DECISION-TREE-LEARNING($exs, attributes - A, examples$)
    add a branch to $tree$ with label $(A = v_k)$ and subtree $subtree$
  return $tree$

    - 使用 train set 去訓練 model
    - 使用 test set 去驗證 model 成效

> 💡 演算法建議定義 Class DecisionTree 裡面包含 fit(), predict() 等 methods 使程式碼簡潔明瞭。

4. **優化：**
   - 使用 Accuracy 來得知 model 的好壞
   - 根據 model 的 Accuracy 來改變參數，重新訓練更好的 model

> 💡 $Accuracy = \frac{正確預測數}{總樣本數}$

> 💡 可以透過改變樹高、剪枝等方式調整決策樹

5. **解釋：**
   - 視覺化最終建立的 tree 並在 report 解釋其意義

> 💡 舉例來說，決策樹的判斷準則，什麼情形可以知道他是 fake or real

Report - 60%

1. **了解資料：**
   - 說明你從中得知資料有什麼屬性
   - 需有程式碼截圖與說明，以及資料視覺化的說明
2. **前處理：**
   - 說明你發現資料有什麼問題或是為了後續處理，因此做什麼前處理
   - 需有程式碼截圖與說明，以及資料視覺化的說明
3. **建立模型：**
   - 說明你是如何建立決策樹
   - 需有程式碼截圖與說明

4. **優化：**
    - 說明你怎麼優化模型
    - 需有程式碼截圖與說明
5. **解釋：**
    - 解釋決策樹的意義
    - 需有程式碼截圖與說明，以及資料視覺化的說明

Notice

- 請使用 python 3 完成作業，版本 $\geq$ 3.8
- 不可以直接使用上述未提及的演算法和 library，除了 `math` 和 `random`。
- 撰寫程式碼，**變數命名必須有意義**、須包含**註解**
- 程式碼檔名取名為 **DT.py**
- 報告命名為 **report.pdf**
- 禁止抄襲
- 繳交格式：請將兩份程式碼和報告壓縮成 `.zip` 檔案並命名為 **`學號.zip`**

```
P12345678.zip
> DT.py
> report.pdf
```