Introduction to Artificial Intelligence

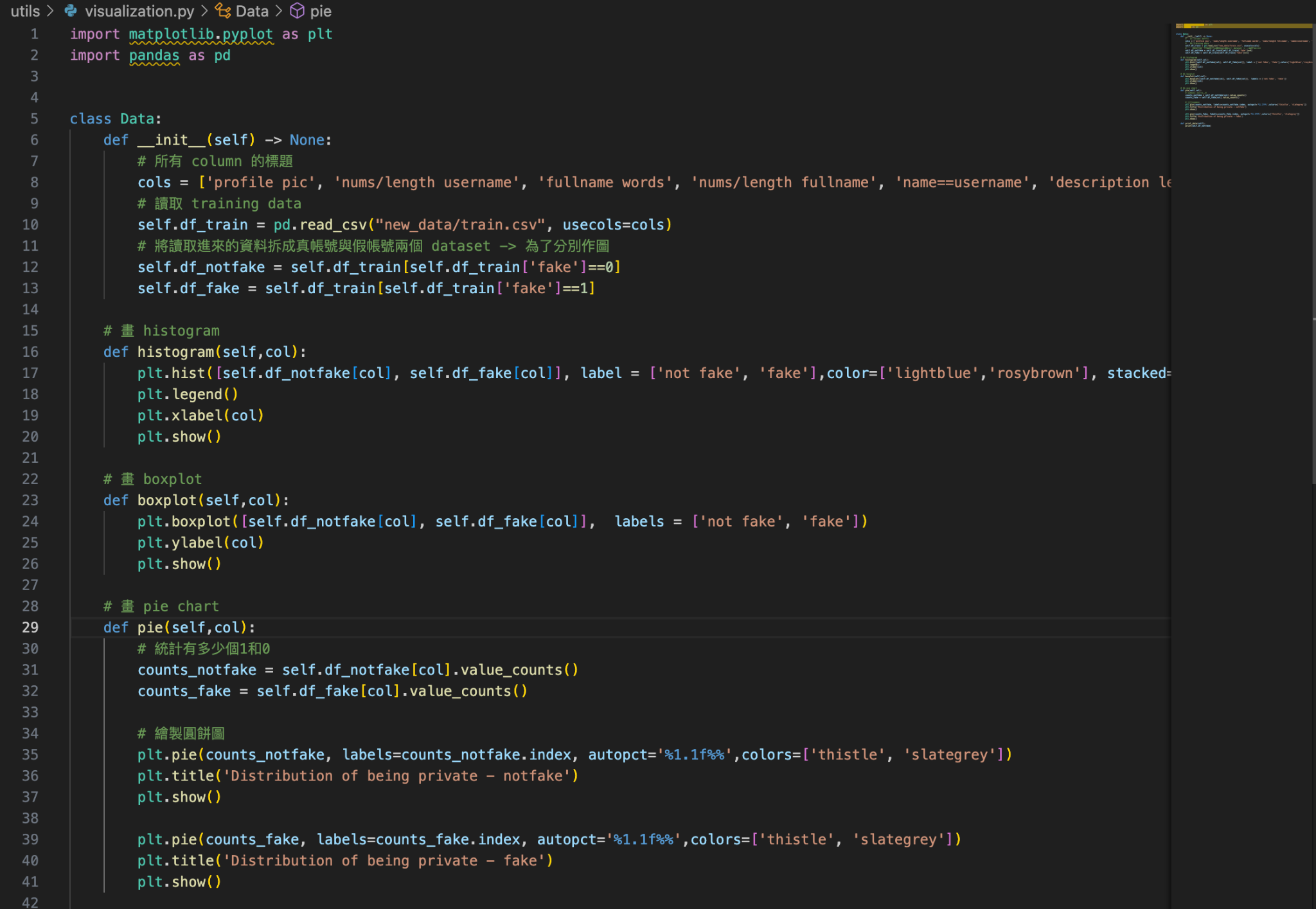
Homework 4 : Learning Decision Tree

E14096407 范芳瑜

1. 了解資料：

我使用 pandas 和 matplotlib 來讀取 dataset 並作圖。在簡單觀察資料過後，發現各種欄位有離散也有連續的，因此我用 pie chart 來表現離散資料，用 histogram 與 box plot 來表現連續的資料，並以呈現出真假帳號間的差距為主要目的分析各個欄位資訊。

1.1 程式碼截圖：

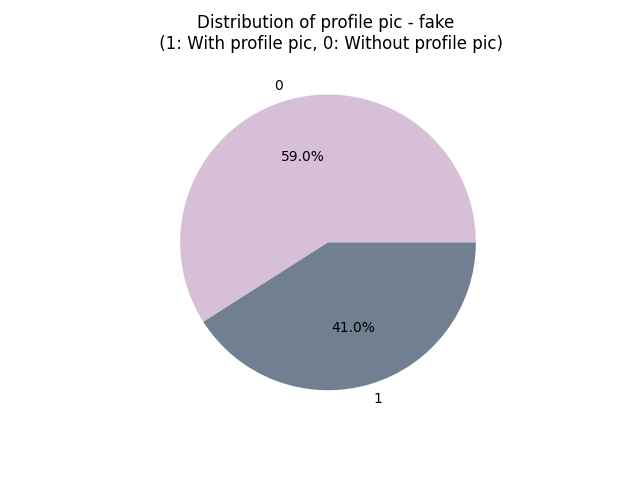
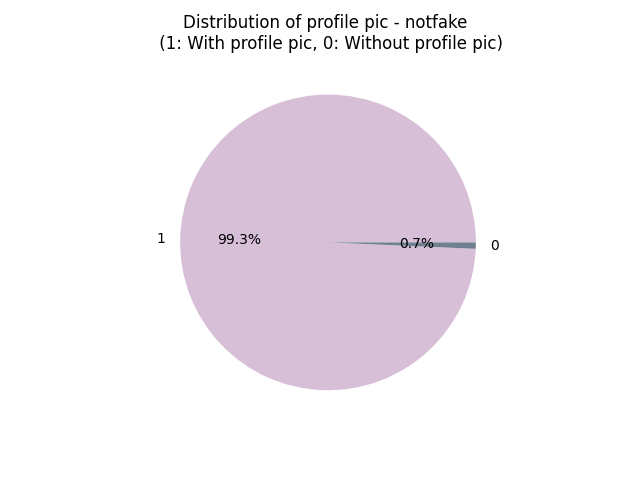


內容主要在讀取資料並生成視覺化圖表。

1.2 針對每個 feature 分析：

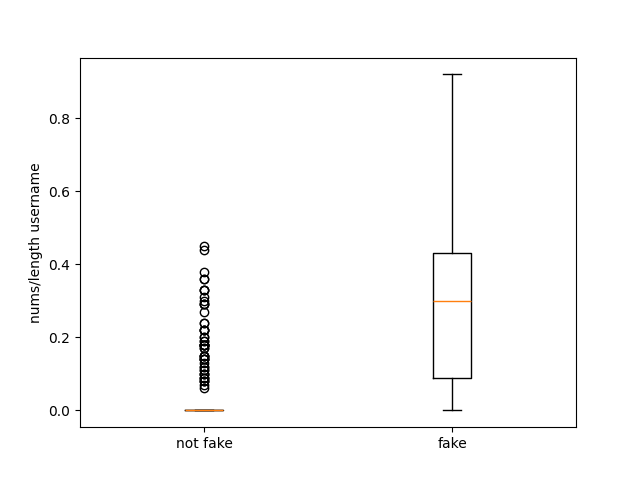
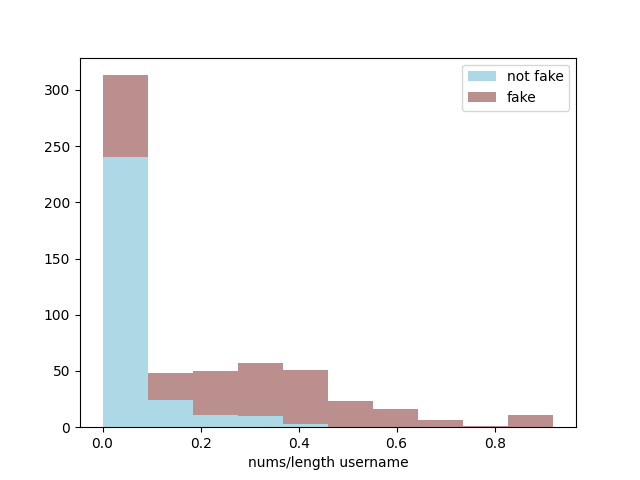
**profile picture**

由圓餅圖可以看出真實帳號有非常高的機率有profile picture，而假帳號則有40%沒有設置profile picture，因此遠高於真實資料的0.7％，因此 profile picture==0 的情況下帳號為假帳號的可能性可能較高。



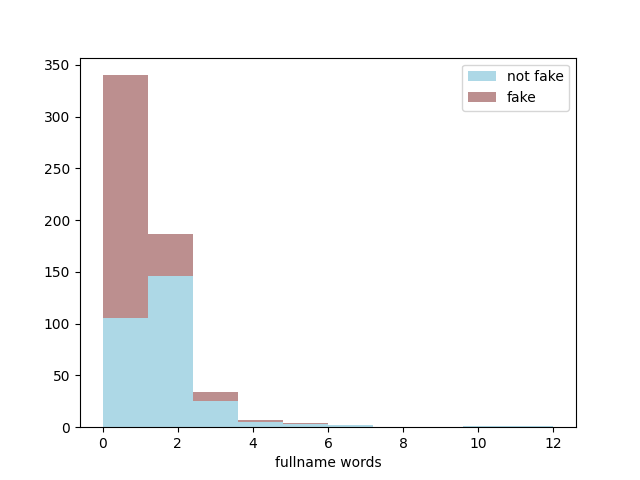
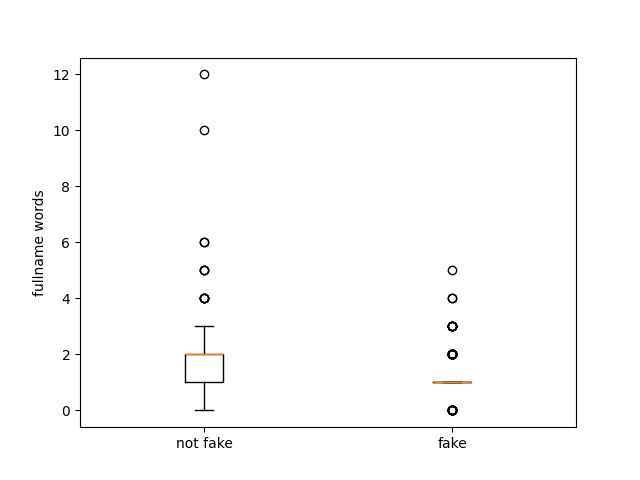
**num/length username**

此 feature 的值介於 0 到 1 之間（為連續資料），可以看見真實帳號的情況下，數字站帳號名稱的比例幾乎都在 0.5 以下，而在 histogram 中可以看出假帳號除了在0.1附近特別高以外，其他比例都有分佈，因此，參考 boxplot ，在做資料離散化時會以區分出 0.5 以上或以下為基礎，即可區分出假帳號與真帳號的差別。



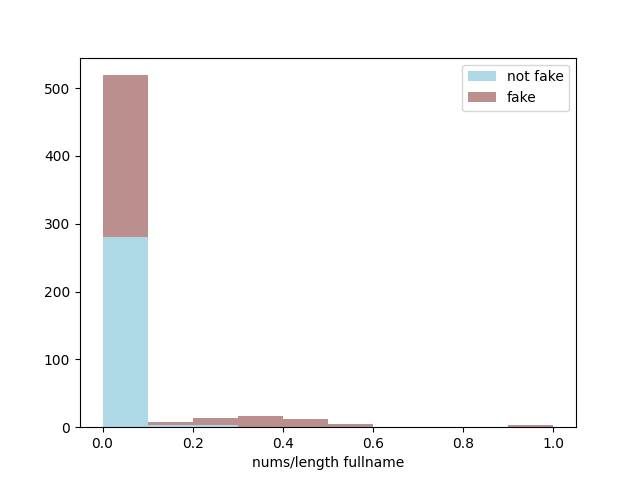
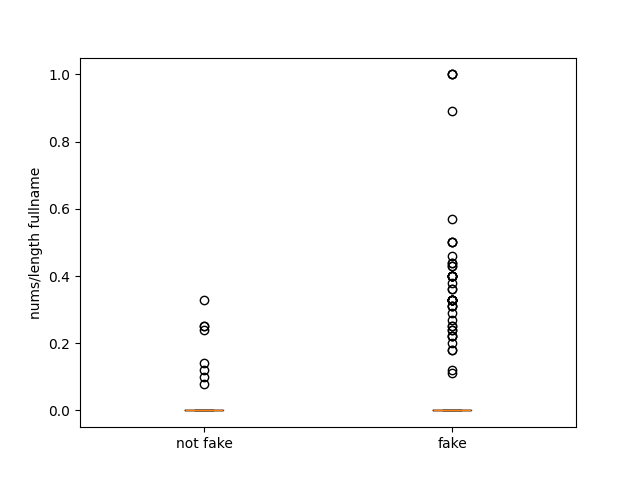
**fullname word**

從 boxplot 可以發現真假用戶兩者的名稱字數分佈差異不大，唯有假用戶在字數長度為 1 的情況下數量遠高於真用戶，因此推斷此 feature 並非能良好區分出真假帳號的重要 feature。



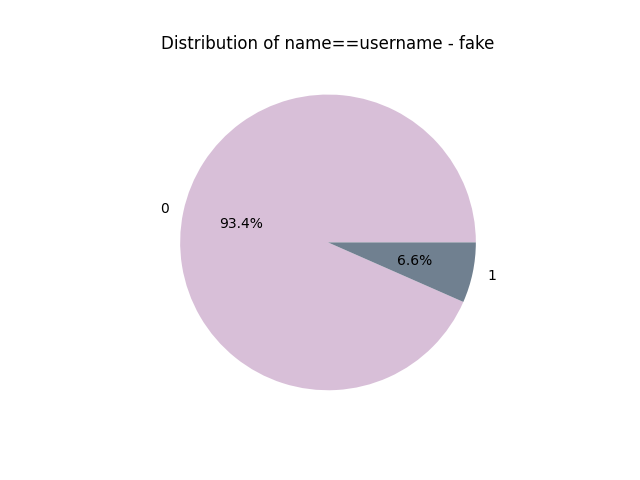
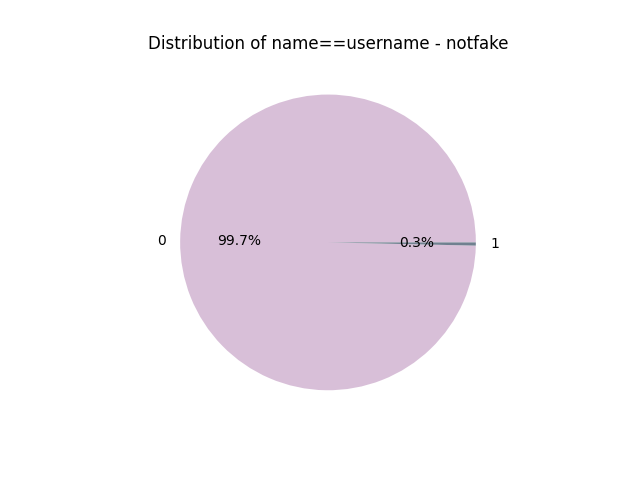
**nums/length fullname**

此 feature 與 nums/length username 較為類似，同樣在做資料離散化時會以 0.5 以上或以下為基礎，以區分出假用戶與真用戶。

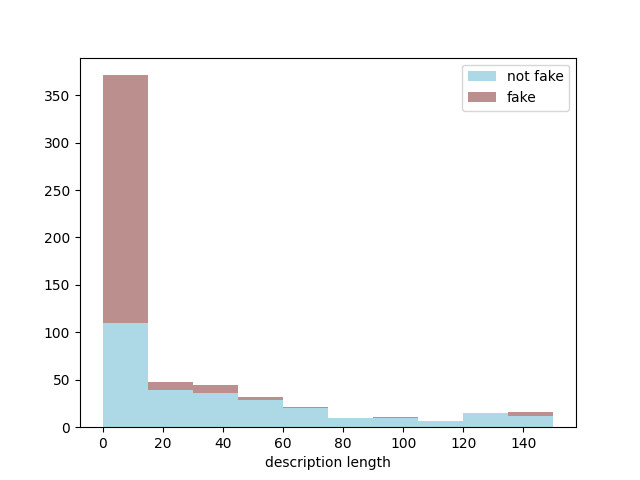
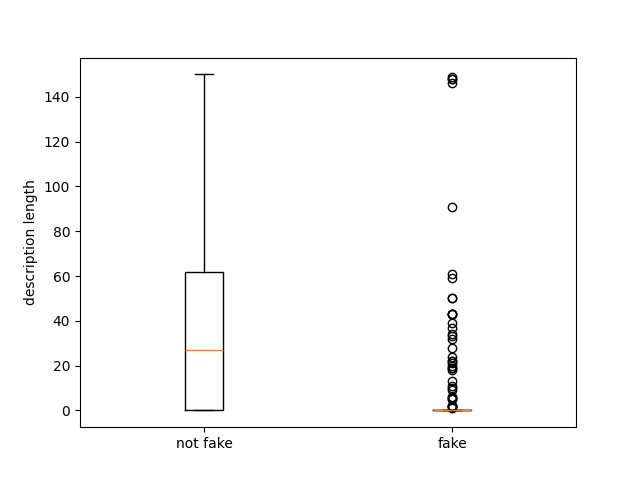


**name==username**

假用戶的用戶名稱等同於帳號名稱的狀況略高於真用戶，但兩種用戶發生這種狀況都極低，因此不能確定此 feature 的重要程度。

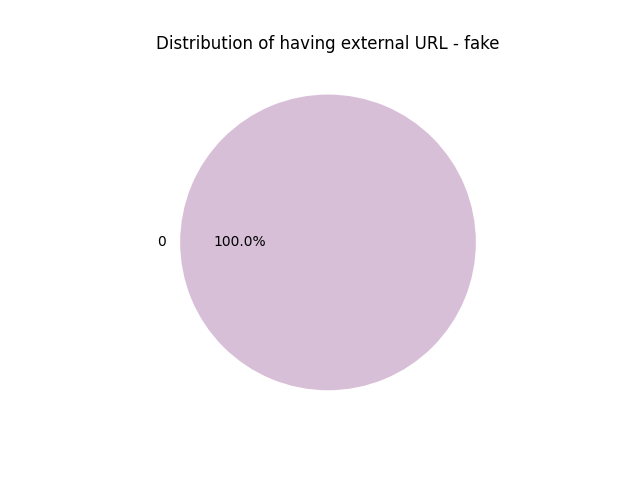
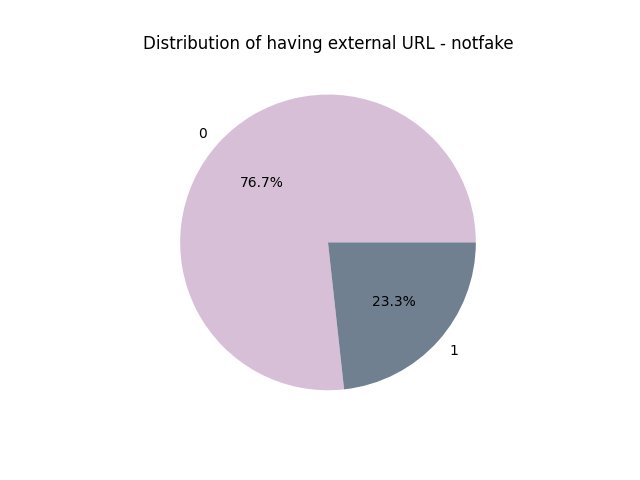


**description length**

假帳號的 description length 整體偏極低，而真帳號雖然也偏低，但沒有假帳號這麼集中。在資料離散化時會以長度 20 為單位切割。

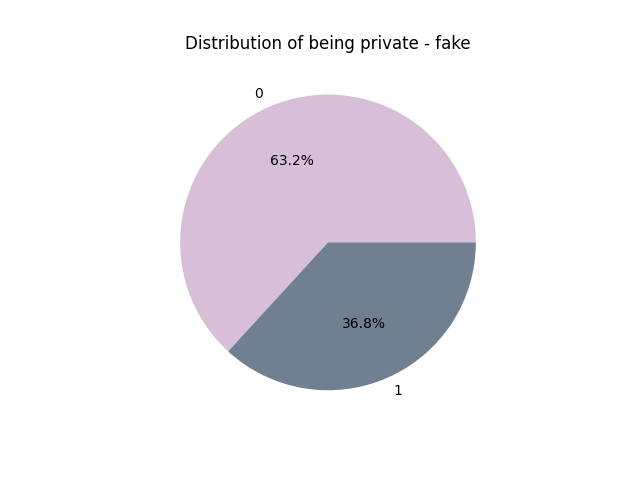
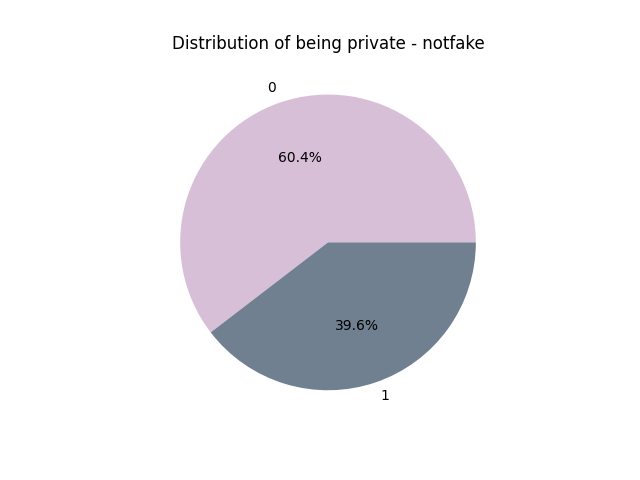
**external URL**

training data 中的假帳號完全沒有 external URL，而真帳號中有 23.3％ 有包含 external URL ，因此也許可以粗略推斷有 external URL 的帳號較為可能是真帳號。



**private**

從圓餅圖可以看出真假帳號在是否為私人帳號的分佈幾乎相同，因此推斷此 feature 並不能良好區分出真假用戶。



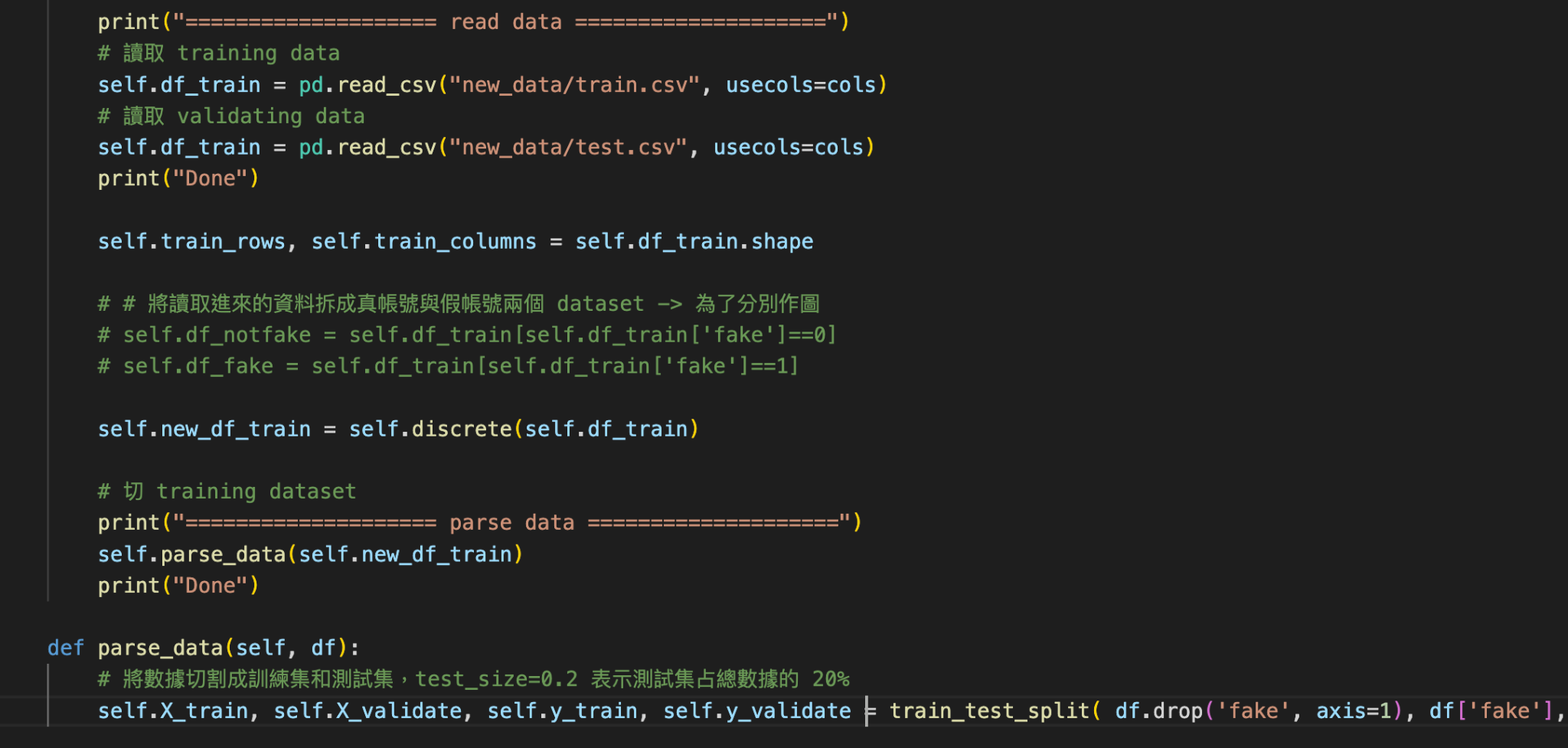
**#posts #followerrs #follows**

這三個 features 除了假帳號有較高比例特別少之外，其他分佈並沒有特別顯現出真假帳號的差別。

2. 前處理

2.1 程式碼說明

在 class Data 中，首先使用 pandas 讀取資料、將資料離散化，再以sklearn 切割成 training data與testing data 兩部份，其中 training data 會用來建構 decision tree，而 testing data 用以確認模型是否 overfitting。



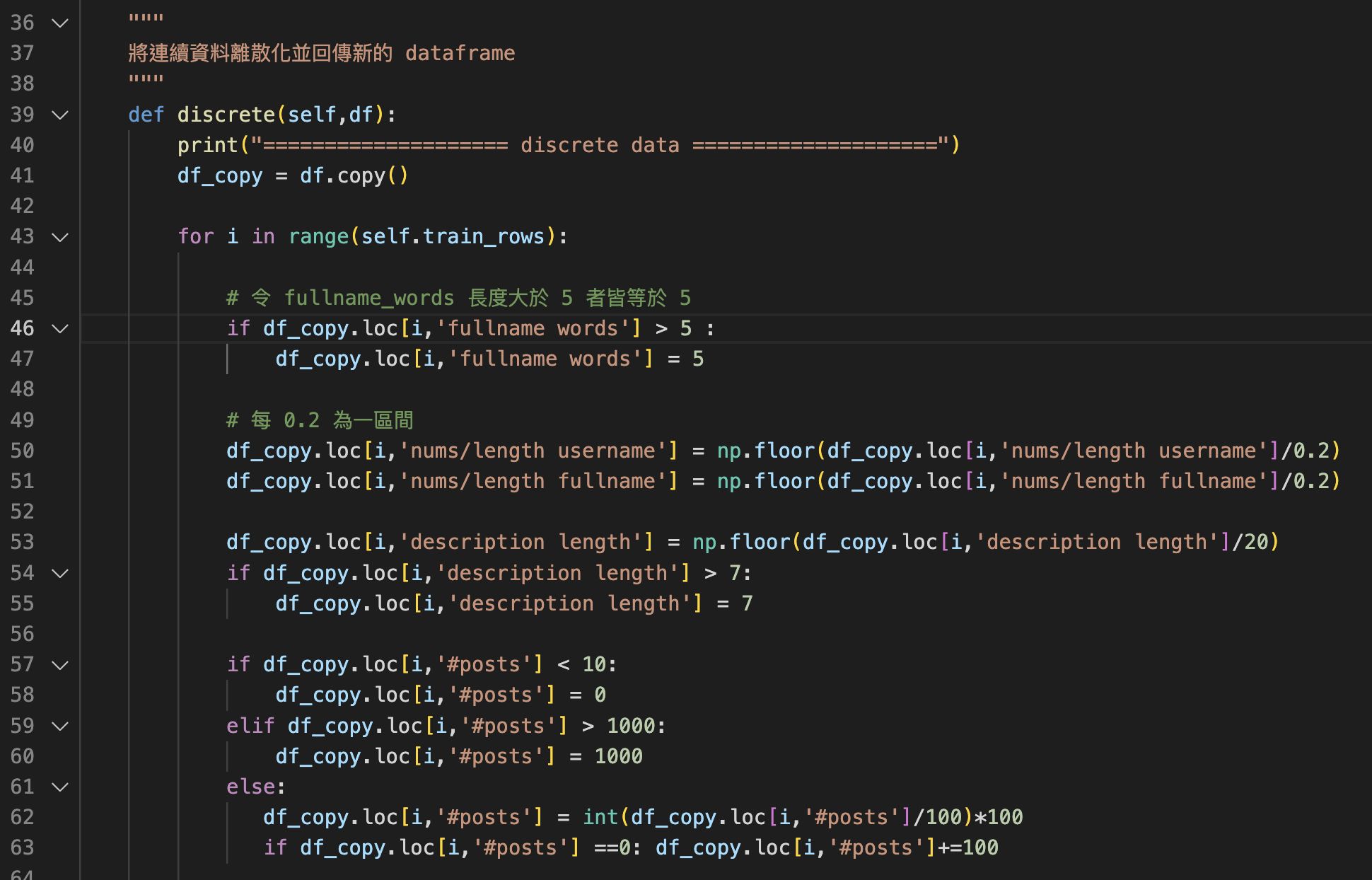
將連續資料離散，目前的做法是依據先前資料分析的結果，將資料以能夠區分出真假帳號為目標切成區段。以下根據程式說明每個連續feature的切割方式：

fullname word：若 full name 長度大於 5 ，皆視為 5 （因為長度大於 5 的個例不多），其他則為原來的數值（1, 2, 3, 4）

num/length username & fullname：因其為範圍 0~1 之間的連續小數，我將他用以 0.2 為單位切割成五等分。

#posts：post 的分佈差距較大，因此將大於 1000 的都視為1000，小於 10的都視為 0，剩下的每 100 為一區間。

#followers ＆ #follows：這兩者的數量範圍都很大，因此用比較大的範圍區分出極小與極大的區間，分別為 <100, 100~500, 500~1000, >1000 四種。

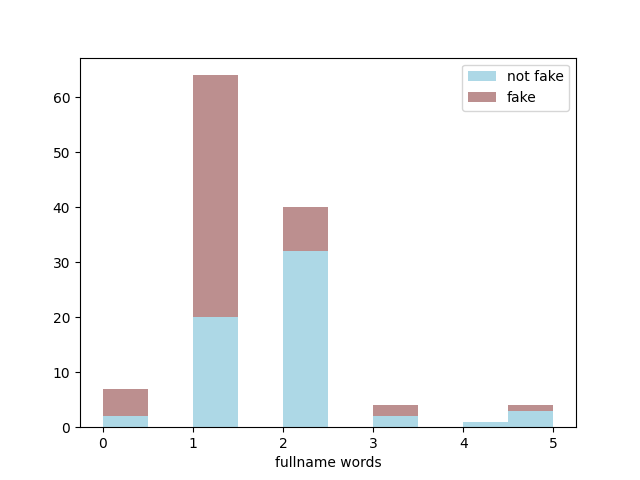




2.2 資料視覺化

處理後的資料將極值集中成一類，視覺化會更突顯真假帳號之間到差異，如下圖，左方為連續資料的 histogram，右方為經過離散化的 histogram，可以發現左圖的資料較集中，無法看出更詳細的資料分佈，而在資料離散化後，可以看到資料集中區域更詳細資訊（x 軸數值較小，間距也較小）。

一張含有 文字, 螢幕擷取畫面, 圖表, Rectangle 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 圖表, Rectangle 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 圖表, Rectangle 的圖片

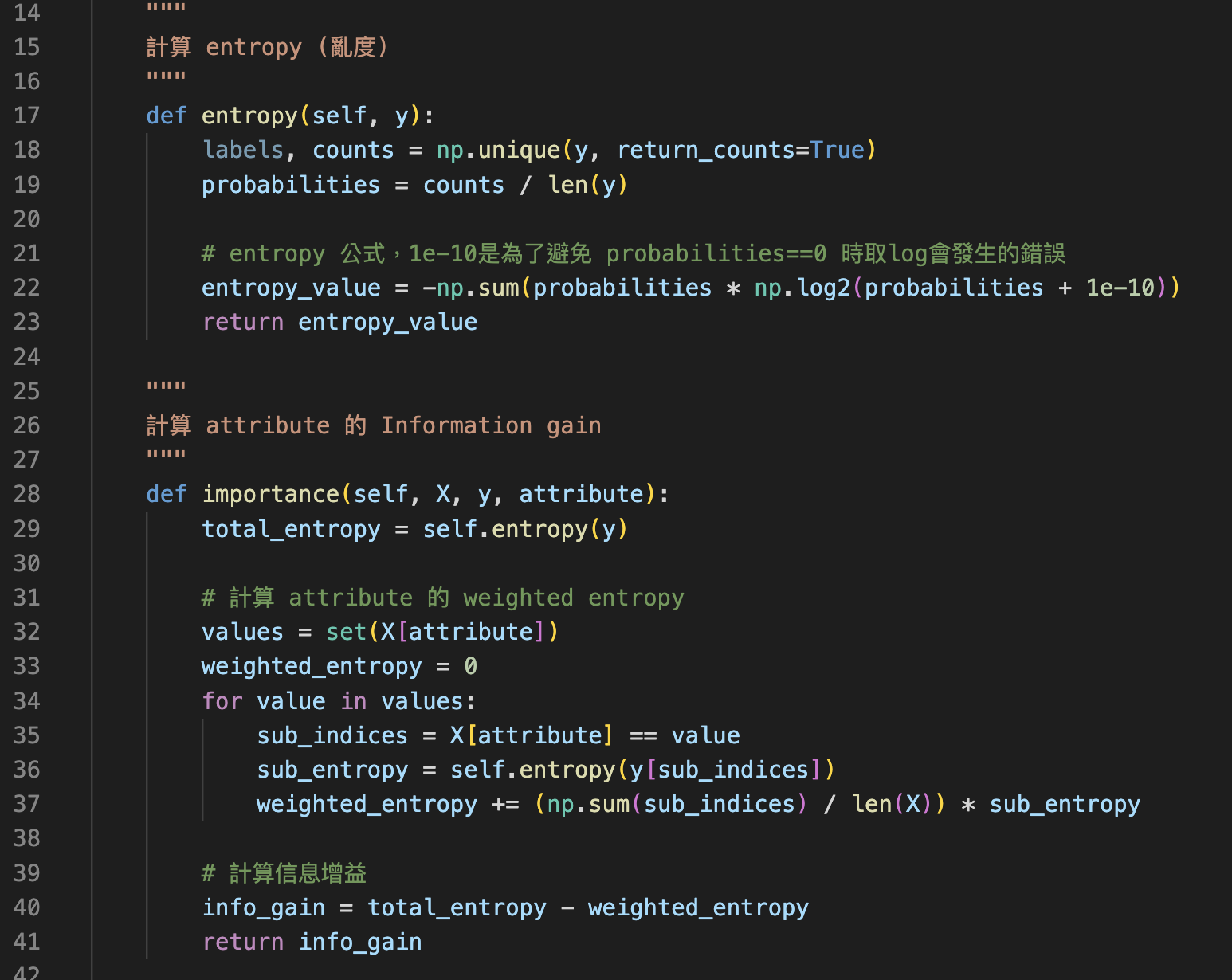
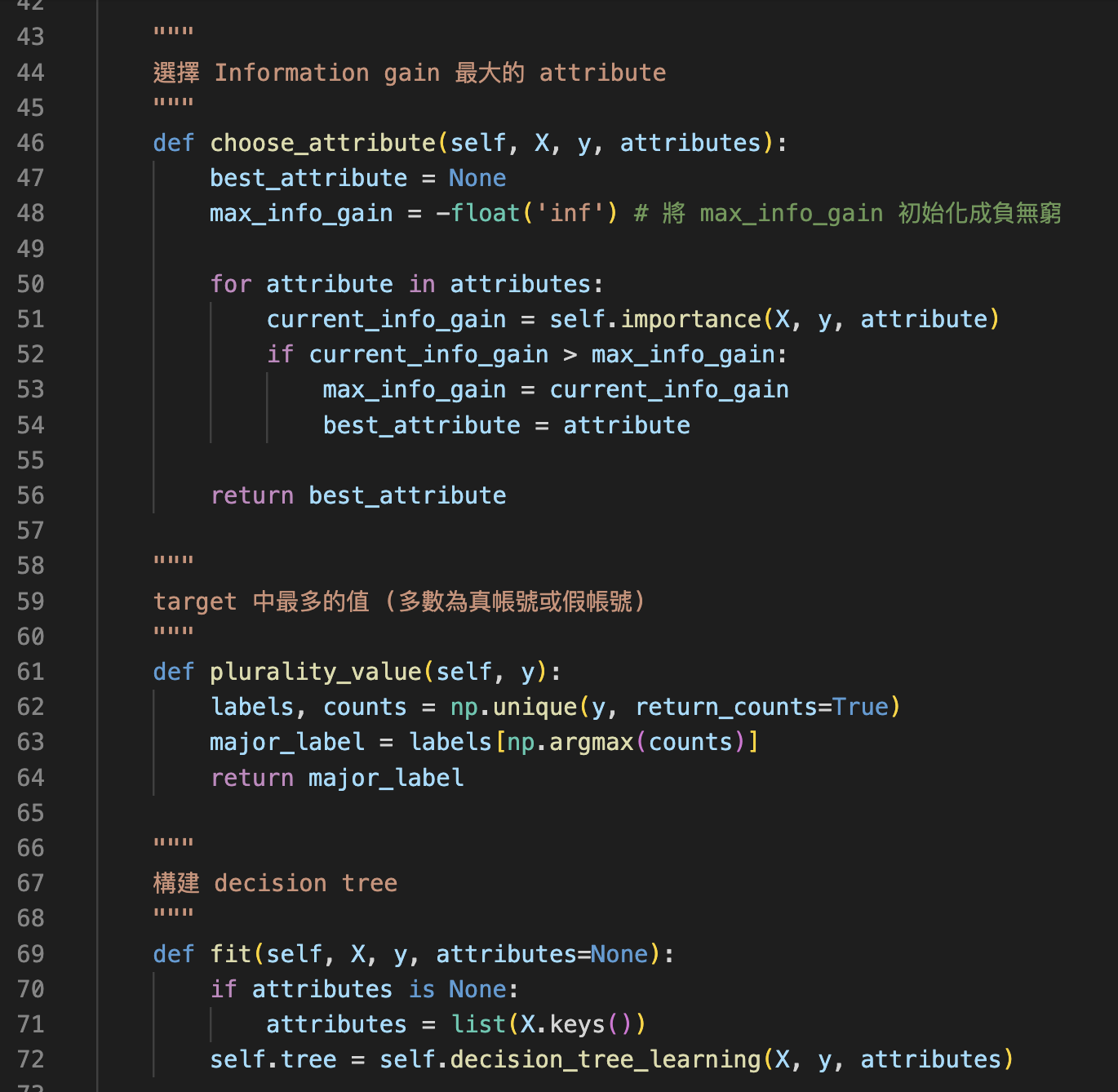
自動產生的描述

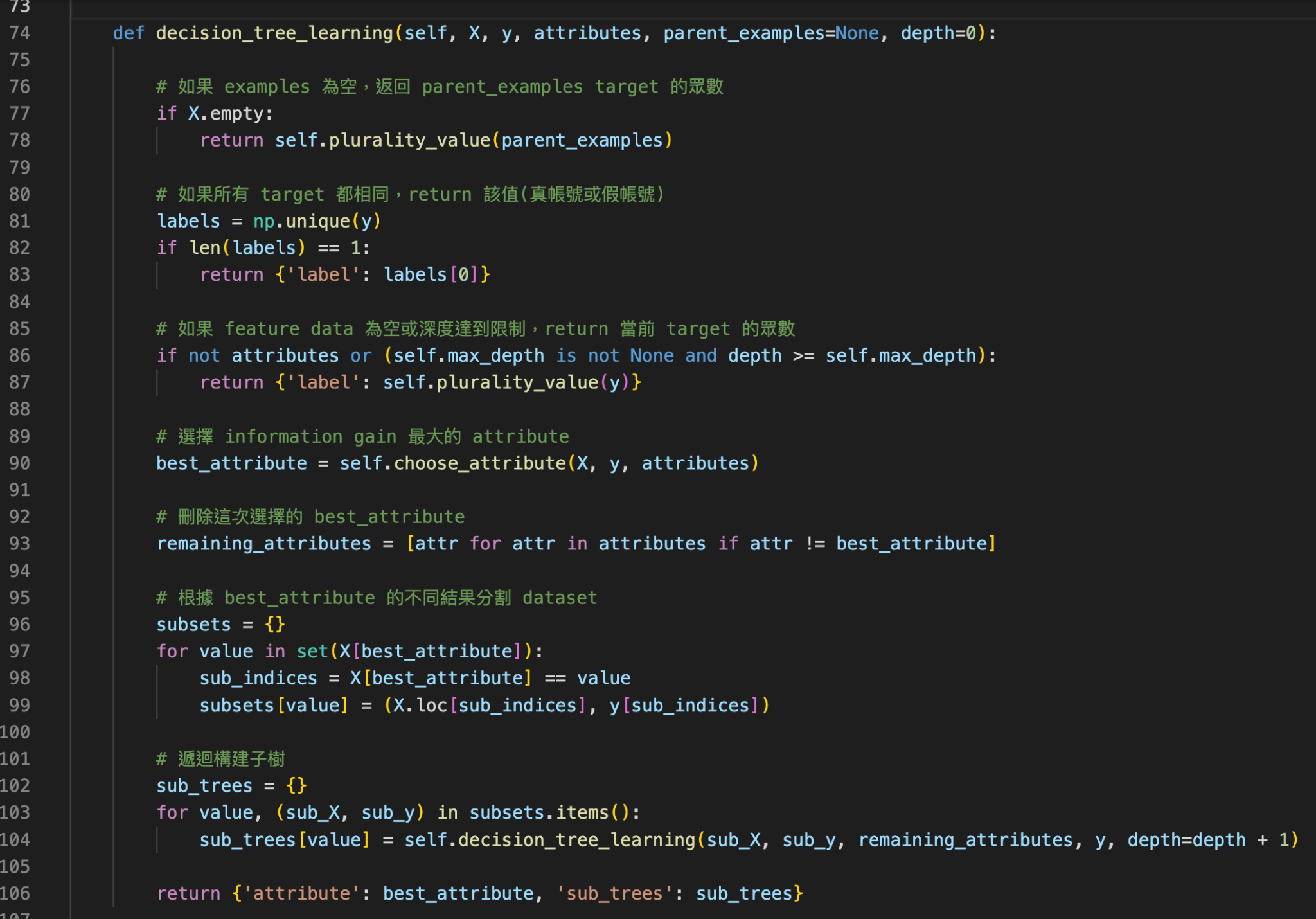
3. 建立模型

class decision tree 的演算法皆參考講義，僅使用 pandas 與 numpy 兩個 library。（圖中的 graphviz 是用來視覺化 decision tree並產生圖片，並未在演算法中使用到）

以下特別說明幾個重要 function 的用途：

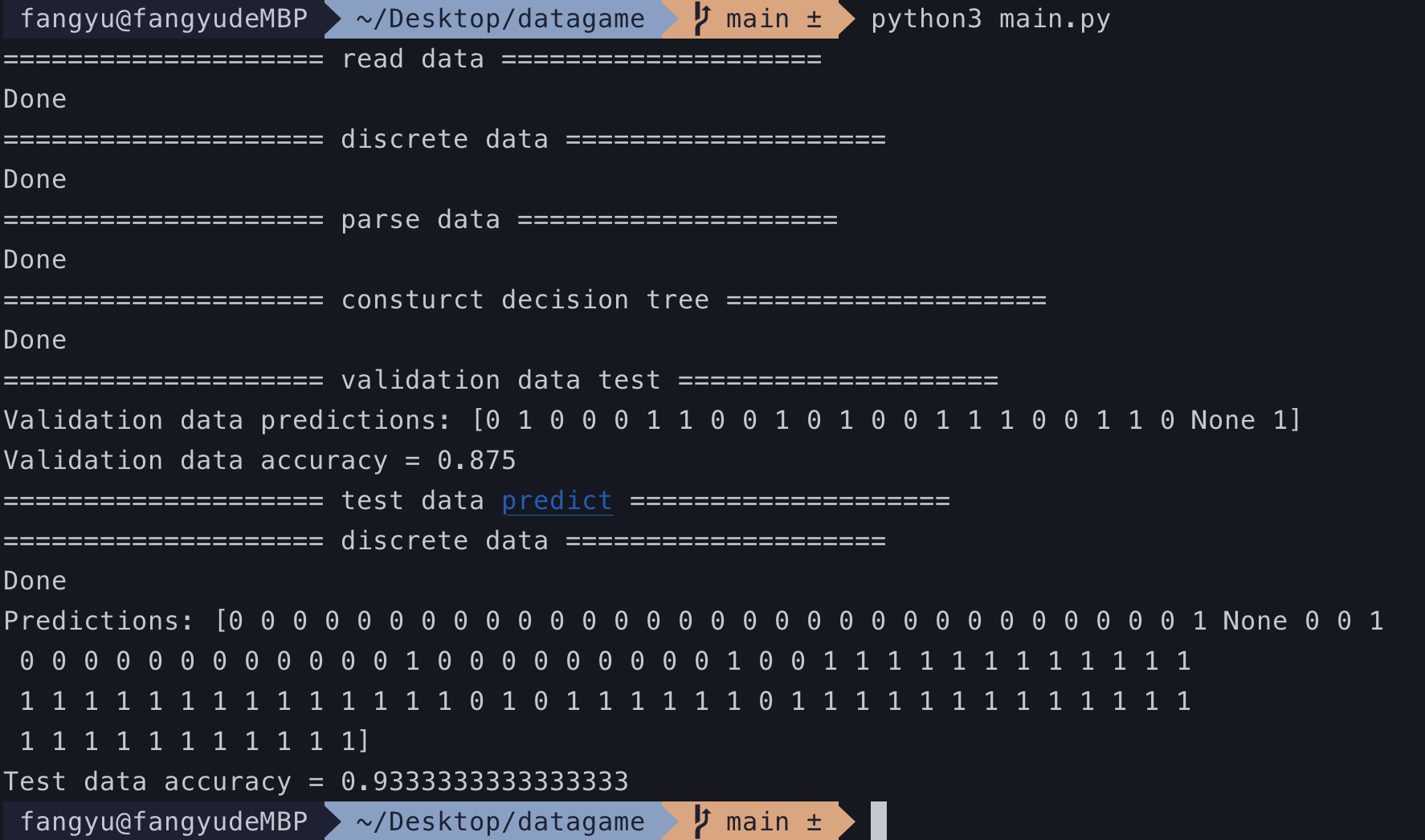
1. fit ： 要開始構建 decision tree 時呼叫的 function，需要 input training data。
2. decision\_tree\_learning：主要構建 decision tree 的 function，由 fit 呼叫，在其中會不斷選擇重要的 feature 作為節點，並在該節點下遞迴生成 subtree，當 training data 全部分類完或 attributes 用完時會結束。
3. plurality\_value：取眾數
4. importance：計算某個 attribute 的 information gain ，用來挑選 attribute，公式為 total entropy – weighted entropy。
5. entropy：計算 entropy，公式與與講義上相同。
6. predict：input 要預測的data，用上述幾個 function train 出來的 tree 預測結果。



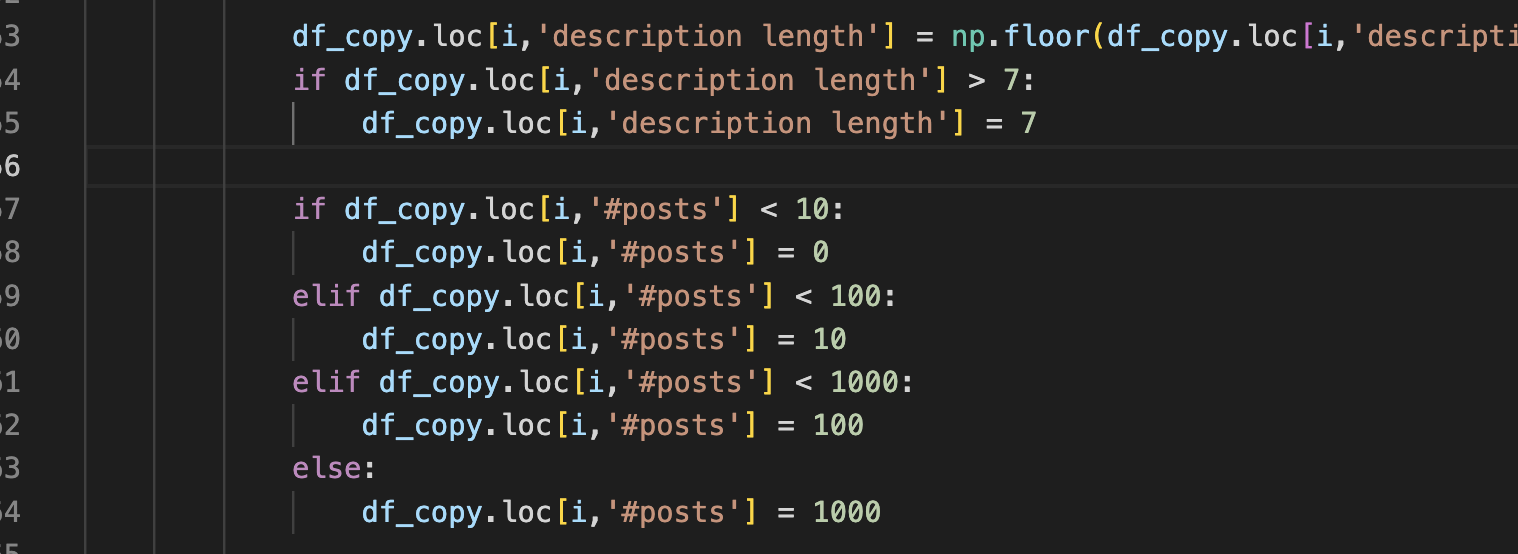


4. 優化：

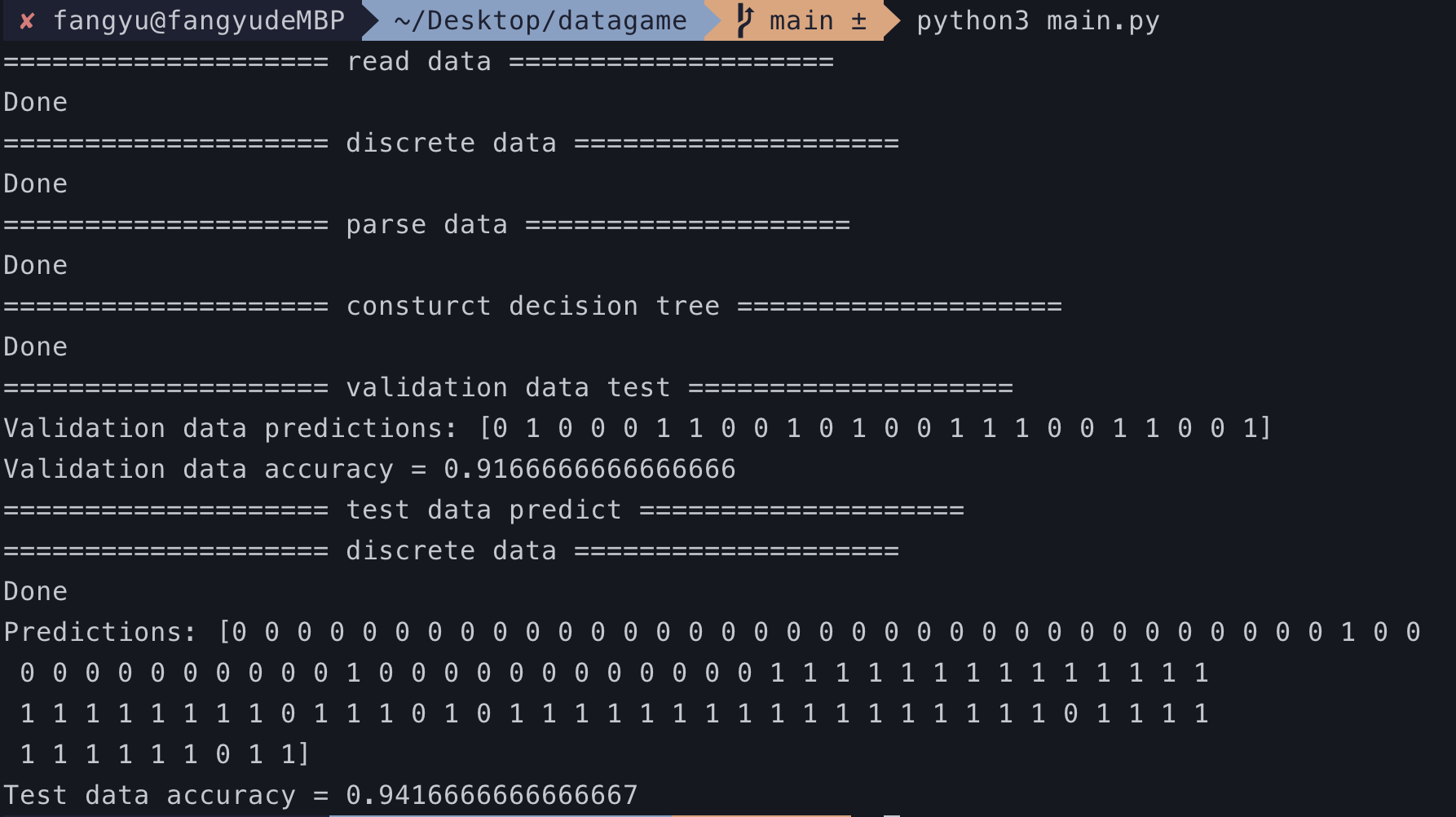
優化前 Accuracy = 0.933



因為觀察到 validating data 的 accuracy 比較低，推測可能分支太複雜，因此在資料前處理時離散化分出來的區段減少，如圖：

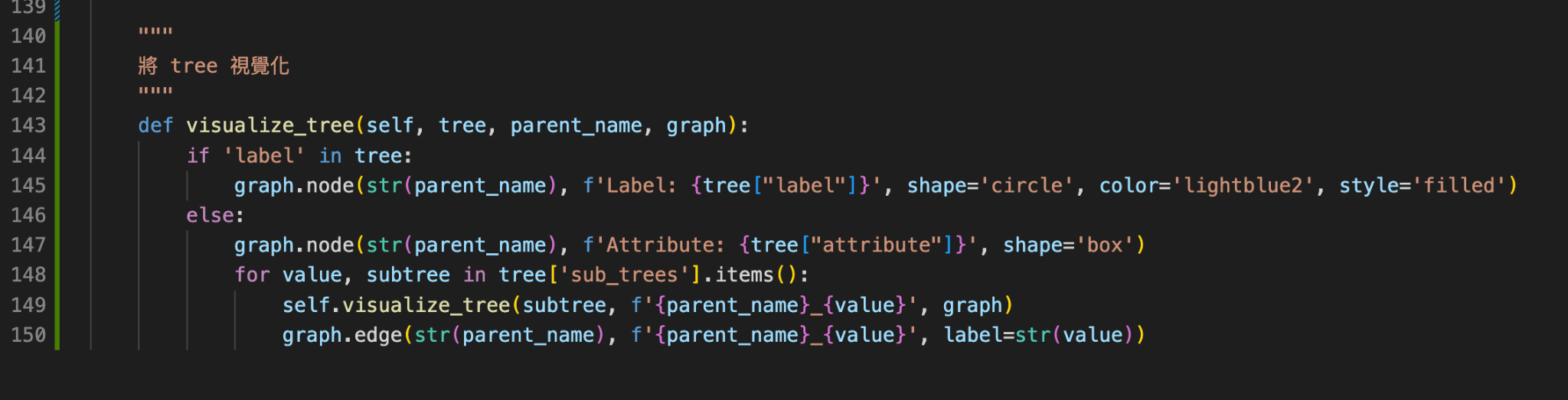


得到優化後 Accuracy = 0.942

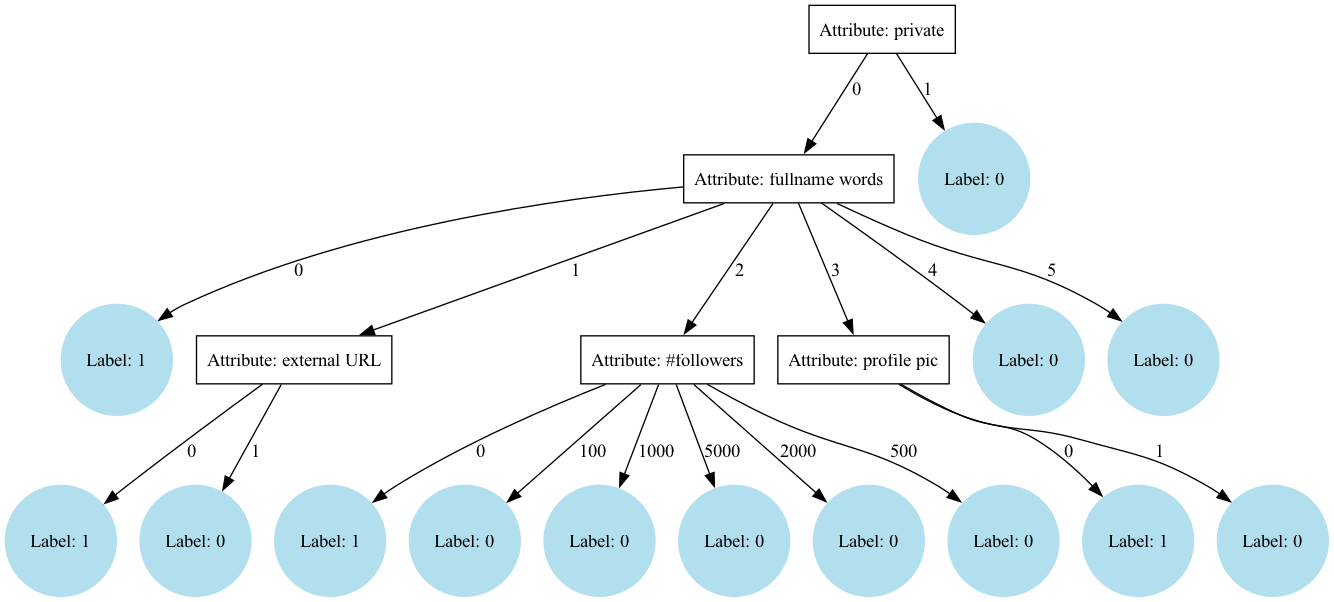


5. 解釋：

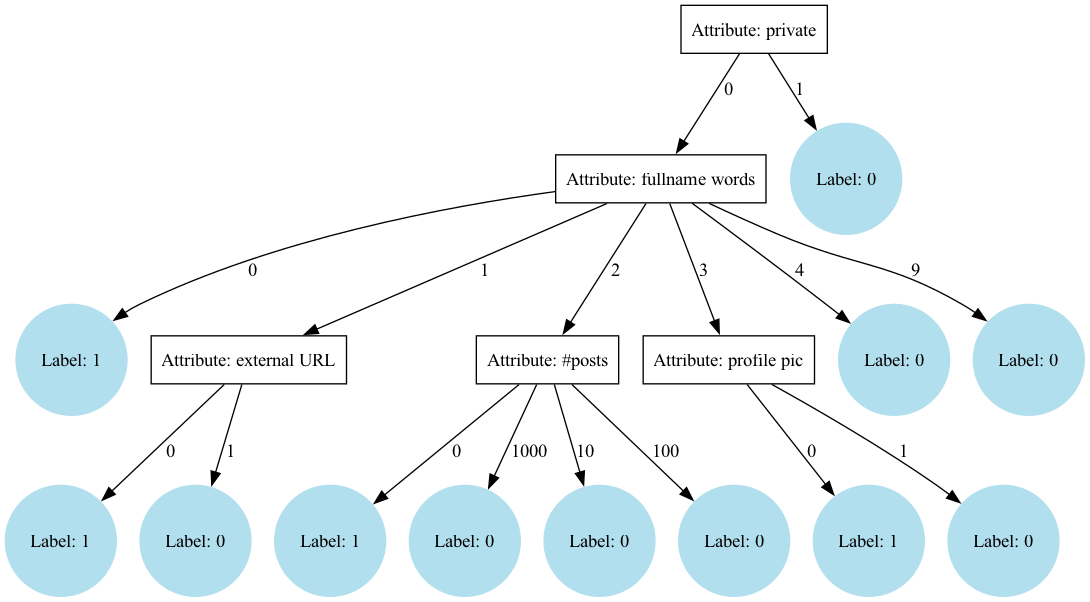
此為將 decision tree 視覺化的 function



優化前的 tree：可見分支較複雜



優化後：分支比優化前少



從圖中可以看見被挑選出成為重要 feature 的有 private、fullname width、external URL、#posts 與 profile pic，許多 features 在資料分析時並不能看出真假帳號的差異，被認為不是重要的feature，因此判斷資料在離散化後的重要性與離散前有很大的差異。

在 root 的 private 最初在分析時認為其無法良好區分真假用戶，結果卻在private == 1時直接判斷其為真實帳號，並有不錯的準確率，這點尚難以解釋。第二層的fullname words 當初在分析時也看不出差異，但在離散化後數值最小的一群被判斷為假帳號，最大的一群被判斷成真帳號。數值在中間的的則由 external URL、#posts 與 profile pic 做第三層分類。#posts 在結果可以看見，只要是 posts 數量大於 10 者皆被視為真實帳號，與是否有 profile pic 和 external URL 這兩點一同說明，資料越缺乏資料的帳號越容易被判斷為假帳號。