# 以 Python 及決策樹套件與最近鄰居法套件開發專案進行分類預測與分類績效評估

B10423004 蔡宗穎，國立雲林科技大學
B10423013 曾鈺雯，國立雲林科技大學
B10423044 王彥淇，國立雲林科技大學

## 摘要

本團隊使用決策樹（Decision tree）及最近鄰居法（K_Nearest_Neighbors）針對 Image Segmentation 資料集進行分類預測及分類績效評估，藉由圖像中的資料，例如：顏色的位置、比例等，去預測為 7 種室外圖像中的一種。並在決策樹中透過 Gini、Entropy 來比較分類結果，在 KNN 中透過 L1、L2 norm 來比較分類結果，並比較哪種較佳，最後依照資料分析的結果，來判斷照片場景是否如預期，

演算法：決策樹、最近鄰居法

# 一、緒論

## 1.1 動機

近年來圖像辨識越來越被看重，不管是臉書、Google 等等，都在這方面投下大量的人力資源進行探討及研發，希望能在電腦視覺技術上有所突破，提供人們更多的幫助，解決社會上遇到的困難，像是停車場常常看到的車牌辨識系統，就是一個透過圖像辨識來幫助人們生活更便利。

圖像辨識在大家認識中是需要使用較困難的技術，我們希望能用最容易理解的兩種演算法，針對所選的資料集進行預估。

## 1.2 目的

利用目前所學的兩種演算法，針對已經從圖像中提取出來的數據，去預測圖片是屬於七種戶外環境中的哪一種。

# 二、方法

## 2.1 程式架構

1. 讀取資料集
2. 對屬性欄位進行標準化的預先處理
3. 使用決策樹套件，並分別用 Gini 和 Entropy 去建立不同模型，並找出最佳的深度
4. 繪製不同深度決策樹訓練及測試正確率，及 Gini 和 Entropy 正確率的比較
5. 繪製正確率最高的決策樹
6. 使用最近鄰居法套件，並分別用 L1 和 L2 距離去做預測，並找出鄰居數量(K)
7. 繪製不同鄰居數量訓練及測試正確率，及 L1 和 L2 距離正確率的比較
8. 分別將決策樹及最近鄰居法預測資料輸出成 Excel 檔

## 2.2 執行程式的方式

透過 Spyder 執行 main.py 檔，需有 segmentation.data 與 segmentation.test 檔，使用 python3.6 版本，注意 matplotlib(畫折線圖套件)須為 2.2.3 版，其他需安裝的套件為 pydotplus(dot 檔轉成 pdf 套件)、sklearn(演算法套件)、graphviz(需設定環境變數，畫 dot 檔圖套件)、xlsxwriter(輸出 excel 套件)、numpy(科學計算套件)。

# 三、實驗

## 3.1 資料集

使用 Image Segmentation 資料集，實例是從 7 個室外圖像的資料庫中隨機抽取，每個實例為 3x3 的區域。有 19 個屬性欄位，均為連續屬性，資料筆總共 2310 筆。類別分佈為磚面、天空、葉子、水泥、窗戶、小路、草。

## 3.2 前置處理

正規化 : 是在資料庫中組織資料的程序。 其中包括建立資料表，以及在這些資料表之間根據規則建立關聯性，這些規則的設計目的是：透過刪除重複性和不一致的相依性，保護資料並讓資料庫更有彈性。

資料集分割 : 因為訓練集過少，所以把訓練集和測試集合併，隨機分割 3 成為測試資料，7 成為訓練資料，所以每次執行結果都是浮動的

## 3.3 實驗設計

決策樹(Decision tree) : 是一種過程直覺單純、執行效率也相當高的監督式機器學習模型，適用於 classification 及 regression 資料類型的預測，與其它的 ML 模型比較起來，執行速度是它的一大優勢。

Gini : 決策樹裡面使用了 Gini 來計算不純度(Impurity)

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

Entropy : 決策樹裡面使用了 Entropy 來計算不純度(Impurity)

$$Entropy(t) = -\sum_{j} p(j \mid t) \log_2 p(j \mid t)$$

最近鄰居法(k_nearest_neighbors) : 在 k-NN 分類中，輸出是一個分類族群。一個物件的分類是由其鄰居的「多數表決」確定的，k 個最近鄰居（k 為正整數，通常較小）中最常見的分類決定了賦予該物件的類別。若 k = 1，則該物件的類別直接由最近的一個節點賦予。

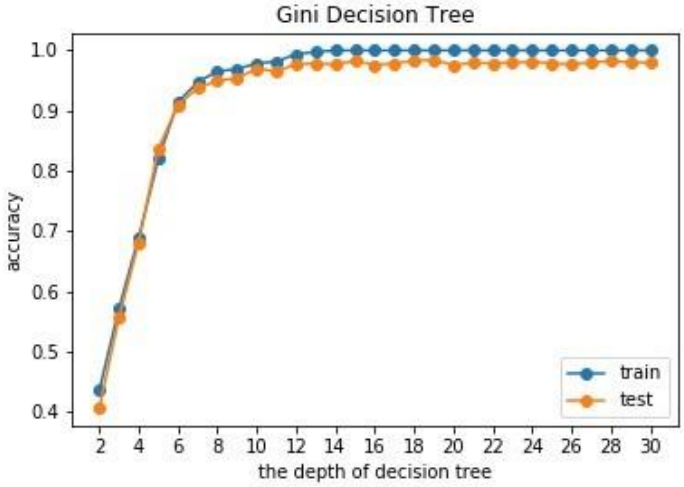
歐式距離(Euclidean Distance) : 在幾何空間中我們在高中就有學過如何計算兩個點之間的距離，以二維平面來說，我們在高中時所用的方法為，計算兩個點之間 x 和 y 各自的差距，並使用畢氏定理來計算兩點間的距離，三維空間可以用一樣的方法來計算。

$$dist(p,q) = (\sum_{k=1}^{n} | p_k - q_k |^r)^{\frac{1}{r}}$$

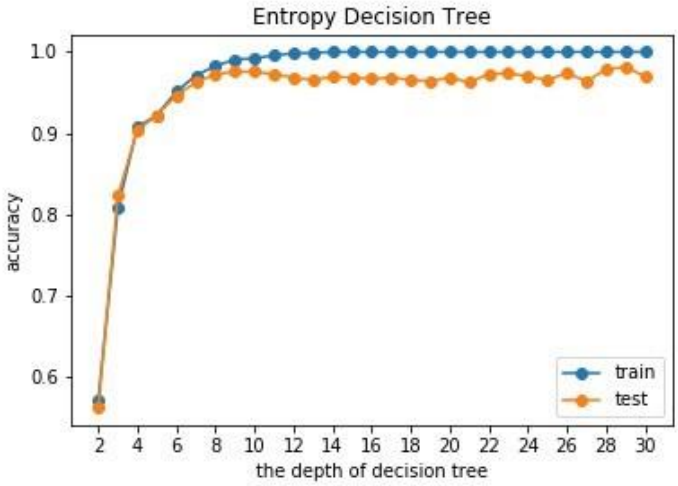
### 3.4 實驗結果

決策樹(Decision tree)

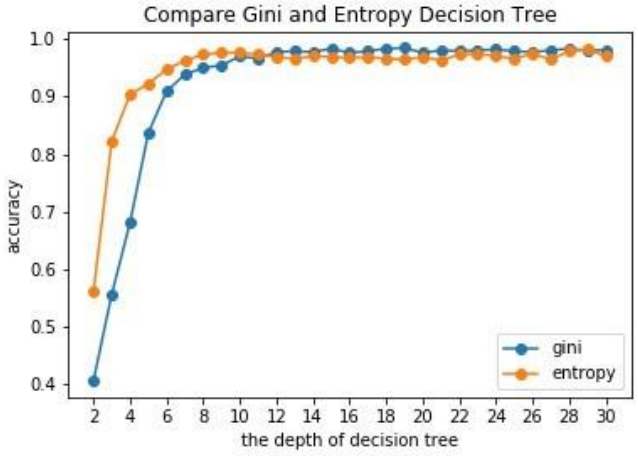
- 圖表 1.使用 Gini 算法，比較訓練及測試正確率




- 圖表 2.使用 Entropy 算法，比較訓練及測試正確率

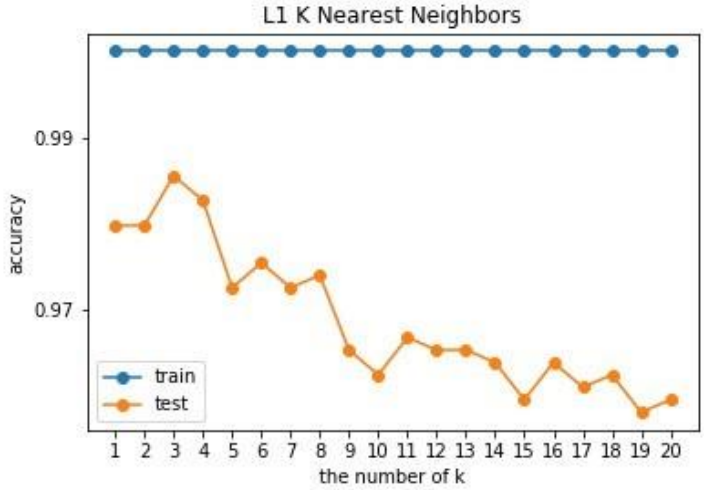



- 圖表 3.比較使用 Gini 與 Entropy 算法的正確率

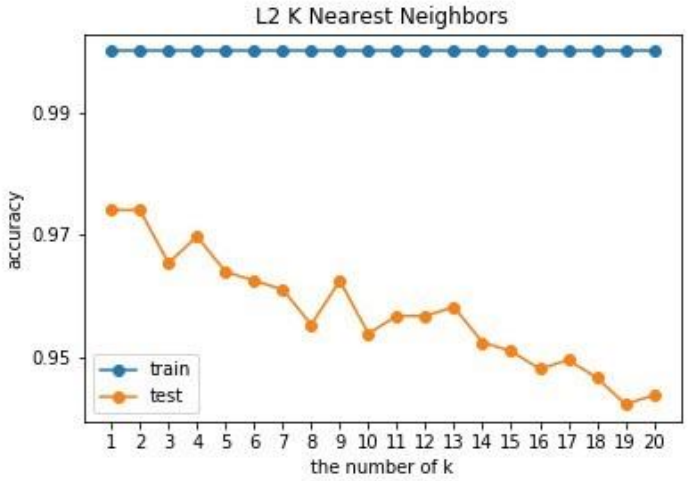


最近鄰居法(k_nearest_neighbors)

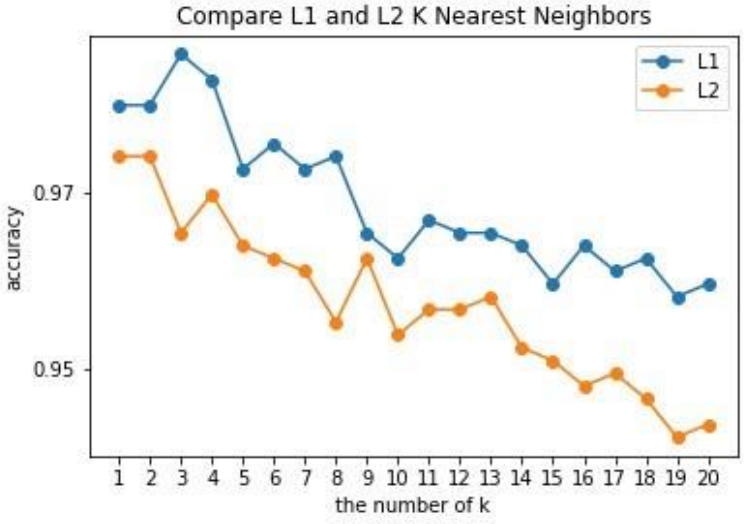
- 圖表 4.使用 L1 距離，比較訓練及測試正確率




- 圖表 5.使用 L2 距離，比較訓練及測試正確率



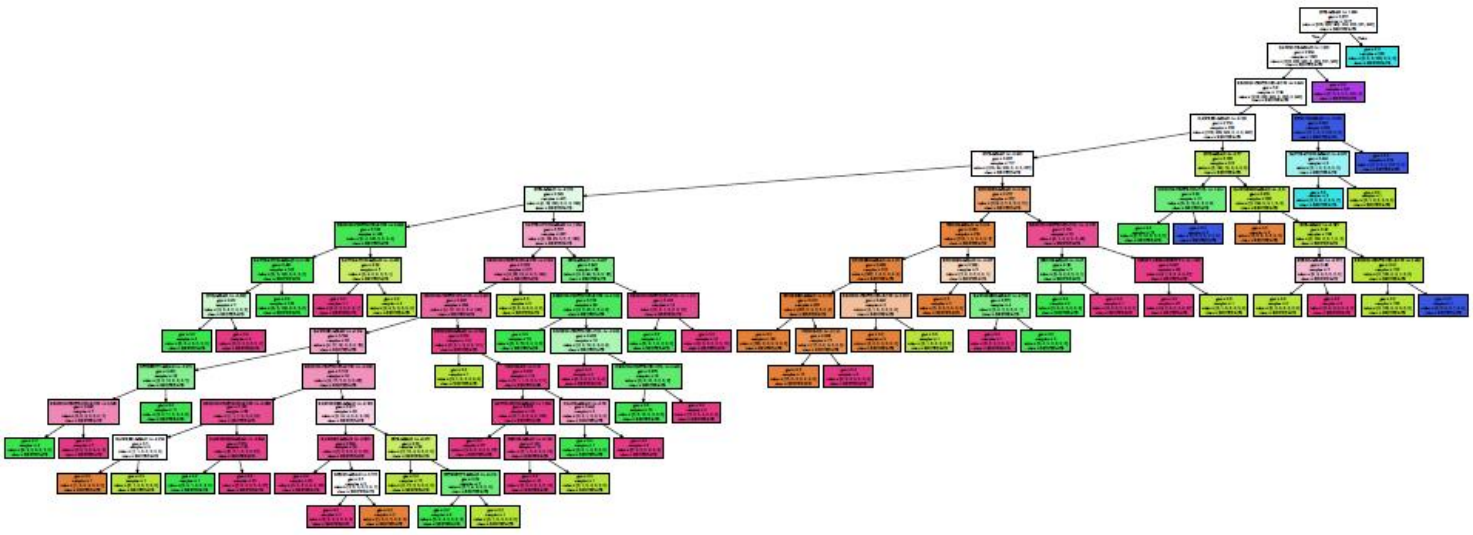

- 圖表 6.比較使用 L1 與 L2 距離的正確率



- 圖表 7. 預測資料部分內容

| | A | B | C |
|---|---|---|---|
| 1 | class | | predict |
| 2 | PATH | → | PATH |
| 3 | CEMENT | → | CEMENT |
| 4 | WINDOW | → | WINDOW |
| 5 | CEMENT | → | CEMENT |
| 6 | FOLIAGE | → | FOLIAGE |
| 7 | FOLIAGE | → | FOLIAGE |
| 8 | WINDOW | → | WINDOW |
| 9 | PATH | → | PATH |
| 10 | BRICKFA | → | BRICKFA |
| 11 | BRICKFA | → | BRICKFA |
| 12 | BRICKFA | → | BRICKFA |
| 13 | WINDOW | → | WINDOW |
| 14 | PATH | → | PATH |
| 15 | BRICKFA | → | BRICKFA |
| 16 | SKY | → | SKY |
| 17 | GRASS | → | GRASS |
| 18 | CEMENT | → | CEMENT |
| 19 | CEMENT | → | CEMENT |
| 20 | GRASS | → | GRASS |
| 21 | PATH | → | PATH |

- 圖表 8. 決策樹圖型



# 四、結論

由於每次切割訓練與測試資料集為隨機切割，所以每次的正確率會有些微差距，正確率平均為 0.95 以上。決策樹最佳深度會隨著切分資料而改變，但最佳鄰居個數皆為 1 個。

在我們的測試下得到 L1 norm 的正確率會大於 L2 norm 的正確率，且 Entropy 的正確率會大於 Gini 的正確率。