資料科學導論 Report—HW2

104502518 資工 4A 劉冠聲

程式說明：

輸入檔名和 K 值後進行 KNN 運算，最後將結果顯示於標準輸出中。

一開始想到最直觀的計算策略是將所有 training data point 的距離存起來再進行排序，時間複雜度 O(n*logn)，空間複雜度(n)，由於不想要空間複雜度這麼高，於是想了第二策略，也是目前的實作方式。

第二策略的目的為縮減空間複雜度，在對每一個 training data point 計算完距離後，確認是否比原本 k 個最近的某一點更近，以便更新列表，所需空間即為 k，空間複雜度 O(k)，若列表一有更新即須重新對整個列表做排序，時間複雜度最糟為 O(n*k*logk)。

第三策略為第二策略的延伸，改善時間複雜度，不過並未實作，將列表更新重新排序的方式改為實作類似氣泡排序的方式，由最新加入的值往前依序比較找到適合的位置，由此可將第二策略的時間複雜度優化為 O(n*k)

由於不確定執行程式的電腦是否安裝我使用的畫圖套件(matplotlib)，將程式碼分為 code.py 和 auto_pic.py 兩者，執行 code.py 需使用者手動輸入檔名和 K 值進行 KNN 計算，執行 auto_pic.py 則為自動將 K 從 1 至 20 的結果畫圖顯示。

結果分析

對 train error 來說，k 為 1 時必定是 0%，因為就是取用自己的種類，k 為 2 至 20 時，數值多有振盪，每次執行都不一樣，沒有實驗出明顯趨勢。

對 test error 來說，k 為 1 至 20 時，數值也多有振盪，每次結果都不同，但大多會大於 train error。