機器學習-專案作業三

組別:第五組

組員:李欣哲 M10912065

黃錦泓 M10912075

摘要

本次專案作業共分成兩個小節，第一小節是使用 MDS 將台北、桃園、新竹、台中、雲林、台南、高雄高鐵站彼此之間的距離畫在 2D 平面上。第二小節的資料集則是 Drink Dataset，內容是由各種飲料的特徵所組成，下面會有詳細的介紹。這邊我們以 Drink Dataset 資料集為主進行報告。這裡主要利用 t-SNE 將資料呈現在 2D 平面，並與 PCA 做為比較，進而看出飲料的分類以及飲料之間相似度，可以藉此找出彼此之間的關聯性，可將此資料用於統計飲料的相關度、分析客戶購買習慣、制定市場策略。不僅限於飲料，也可以將其應用在許多商品上，用途廣泛。

關鍵字: Drink Dataset、MDS、PCA、t-SNE

一、緒論

1.1 動機

近年來，人類對於機器學習的機器分析環節越來越重視，不論是在哪個產業中，都能看見它對社會的貢獻。在訓練資料中也許會有許多擁有高維度的資料，需要經過處理才能讓我們更方便使用，且特徵過多可能會導致過擬合、運算速度慢、視覺化不易。因為人類是三維生物，對於高維度的存在與資訊是無法直接理解的，而我們人類又是以視覺為主要感官的生物，所以我們需要透過資料視覺化的方法來幫助我們去解讀這些訓練資料。

因此我們使用了 t-SNE(t-Distributed Stochastic Neighbor Embedding)與 MDS(Multiple Dimensional Scaling)來對 Drink Dataset 做資料降維，使資料讓我們更好理解，同時看出飲料的分類以及飲料之間相似度，可以藉此找出彼此之間的關聯性狀況，幫助我們更進一步的分析、擬定商品的策略。

1.2 目的

我們希望使用 t-SNE 降維方法對 Drink Dataset 資料集裡面的 Class、Drink、Rank、Amount、Count 來進行資料的視覺化，將各類飲料集分集，並以二維視覺方式呈現，讓我們能夠清楚了解其分布與關係。

二、方法

2.1 程式架構概述

(1)導入程式所需函示庫，並將原本 Drink Dataset 資料集，做為 CSV 檔，方便使用。

(2)載入準備好的 CSV 檔。

(3)對資料集進行資料前處裡。

(4)調用 PCA、t-SNE 模組並對資料進行降維。

(5)輸出 PCA、t-SNE 1-of-k 之 2D 分布圖，進行比較。

2.2 執行程式的方法

本次實驗使用 Anaconda 裡的 Jupyter Notebook 系統來進行編輯，環境版本為 Keras2.4.3、TensorFlow2.3.0、Python3.8.8。

(1) 在 Anaconda 上建立環境，設定、安裝上述環境版本。

(2) 在 Anaconda-Environments 上或 Terminal 安裝所需的函式庫。

(3)在 Anaconda-Home 上安裝 Jupyter 並開啟。

(4)開啟 Jupyter 後找尋目標檔案.ipynb 檔，打開即可開始編輯與執行。

三、實驗一(自選資料集 Bank Marketing)

3.1 資料集簡介

本次實驗選用題目提供的 Drink Dataset 資料集，其中記錄了:

(1)Class:
A、B、C、D、E、F、G，共五個 Class。

(2)Drink:
Coke、Pepsi、7Up、Sprite、Latte、Espresso、Cappuccino，共七種飲品。

(3)Rank:
7、6、5、4、3、2、1，共七個 rank。

(4)Amount:
(100, 200)、(200, 10)、(200, 10)、(400, 100)、(800, 10)、(800, 10)、(900, 400)，共七種分布。

(5)Count:
200、100、100、200、100、100、200，七種數量。

Table 1. Drink Dataset

Class	Drink	Rank	Amount($N(\mu,\sigma)$)	Count
A	Coke	7	(100, 200)	200
B	Pepsi	6	(200, 10)	100
C	7Up	5	(200, 10)	100
D	Sprite	4	(400, 100)	200
E	Latte	3	(800, 10)	100
F	Espresso	2	(800, 10)	100
G	Cappuccino	1	(900, 400)	200

圖一、Drink Dataset 資料集

3.2 前置處理

3.2.1 原始資料生成並轉 CSV 檔

因原始的資料集是一張表格，其中有記錄各種飲料的分布，所以按照其自然分布生成對應的個數筆資料並將其轉換為 CSV 檔，使程式更方便讀取。

	A	B	C	D	E
1	class	drink	rank	avg	amount
2	A	Coke	7	100	288.233
3	A	Coke	7	100	513.6981
4	A	Coke	7	100	279.2964
5	A	Coke	7	100	-16.7317
6	A	Coke	7	100	503.4451
7	A	Coke	7	100	-72.7063
8	A	Coke	7	100	194.1238
9	A	Coke	7	100	602.2391
10	A	Coke	7	100	133.1269
11	A	Coke	7	100	220.7089
12	A	Coke	7	100	355.5374

圖二、生成資料集 CSV 檔

3.2.2 Label Encoding

由於我們的資料集 class 欄位顯示為 A、B、C、D、E、F、的非數字型態，並無法進行運算，所以我們要將其轉換為數值，這邊我們將其轉換為數字的格式，例如:A 轉換為 0、B 轉換為 1、C 轉換為 2 等等，轉換成這樣的形式即可進行訓練的運算。

```
labelencoder = LabelEncoder()
data['class'] = labelencoder.fit_transform(data['class'])
```

圖三、Label Encoding 程式

	class	drink	rank	amount1	count
0	0	Coke	7	100	200
1	1	Pepsi	6	200	100
2	2	7UP	5	200	100
3	3	Sprite	4	400	200
4	4	Latte	3	800	100
5	5	Espresso	2	800	100
6	6	Cappuccino	1	900	200

圖四、對 class 欄位 Label Encoding 後結果。

3.2.3 丟棄 drink 欄位

將訓練資料中的 drink 欄位可能對於我們的可視化無關，且其欄位為英文單字，程式無法進行運算，所以我們將其忽略。

```
data = data.drop('drink', axis=1)
```

圖五、drop drink 欄位程式碼

3.3 實驗設計

3.3.1 讀取 CSV 檔、進行預處理

讀取事先準備好的 CSV 檔，並進行前面提到的預處理。

```
data = pd.read_csv('C:/Users/eric/Desktop/ML/3HW/Drink_Dataset.csv')
data.head(10)
```

	class	drink	rank	amount1	count
0	A	Coke	7	100	200
1	B	Pepsi	6	200	100
2	C	7UP	5	200	100
3	D	Sprite	4	400	200
4	E	Latte	3	800	100
5	F	Espresso	2	800	100
6	G	Cappuccino	1	900	200

圖六、讀取 CSV 檔

```
labelencoder = LabelEncoder()
data['class'] = labelencoder.fit_transform(data['class'])
```

```
data.head(10)
```

	class	drink	rank	amount1	count
0	0	Coke	7	100	200
1	1	Pepsi	6	200	100
2	2	7UP	5	200	100
3	3	Sprite	4	400	200
4	4	Latte	3	800	100
5	5	Espresso	2	800	100
6	6	Cappuccino	1	900	200

圖七、labelencoder 後結果

3.3.2 讀取 PCA、t-SNE 模組進行運算

從 sklearn 函式庫 import PCA、t-SNE 兩個模組，設定個別參數後進行運算。

```
from sklearn.decomposition import PCA # Principal Component Analysis module
from sklearn.manifold import TSNE # TSNE module
```

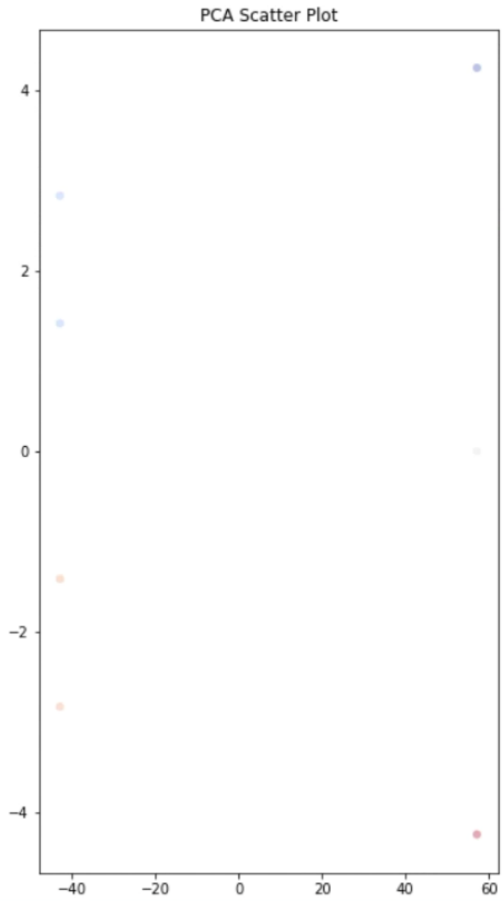
```
# Turn dataframe into arrays
X = data.values

# Invoke the PCA method. Since this is a binary classification problem
# Let's call n_components = 2
pca = PCA(n_components=2)
pca_2d = pca.fit_transform(X)

# Invoke the TSNE method
tsne = TSNE(n_components=2, verbose=1, perplexity=40, n_iter=2000)
tsne_results = tsne.fit_transform(X)
```

```
[t-SNE] Computing 6 nearest neighbors...
[t-SNE] Indexed 7 samples in 0.000s...
[t-SNE] Computed neighbors for 7 samples in 0.001s...
[t-SNE] Computed conditional probabilities for sample 7 / 7
[t-SNE] Mean sigma: 1125899906842624.000000
[t-SNE] KL divergence after 250 iterations with early exaggeration: 40.707672
[t-SNE] KL divergence after 2000 iterations: 0.158585
```

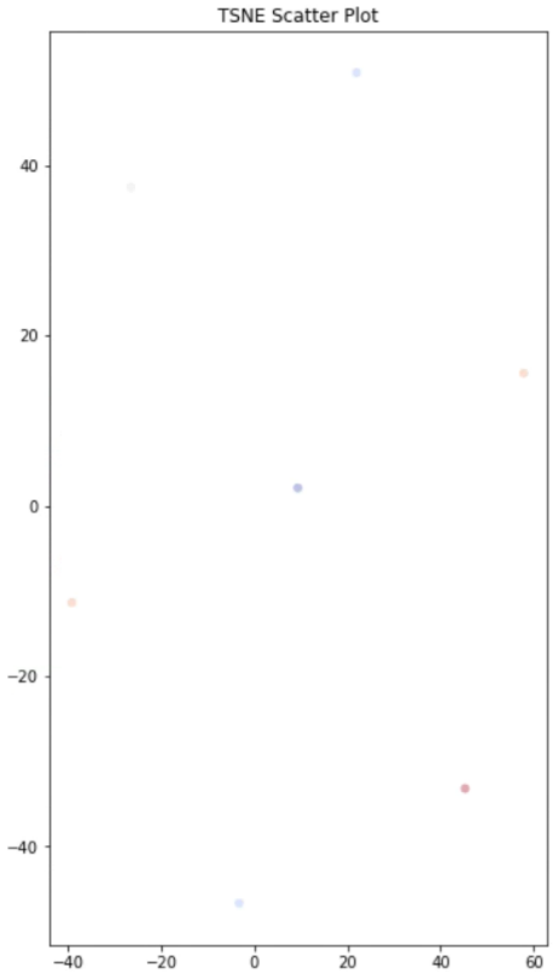
圖八、讀取 PCA、t-SNE 模組運算程式碼

3.3.3 結果

結果分為 PCA 與 t-SNE，分別將兩張分布圖畫出來，分布如下。



圖九、PCA 視覺化二維分布圖



圖十、t-SNE 視覺化二維分布圖

四、實驗二(高鐵站視覺化)

4.1 資料集簡介(自製)

實驗二的資料集為自製資料集，蒐集台北、桃園、新竹、台中、雲林、台南、高雄高鐵站的經緯度，在利用經緯度計算彼此之間距離，下圖為計算完成之資料。

	Taipei	Taoyuan	Hsinchu	Taichung	Yunlin	Tainan	Kaohsiung
0	0.00	30.66	54.97	138.30	183.75	267.16	289.90
1	30.66	0.00	28.82	117.08	163.63	250.66	274.62
2	54.97	28.82	0.00	88.50	135.15	223.01	247.36
3	138.30	117.08	88.50	0.00	46.70	136.24	161.57
4	183.75	163.63	135.15	46.70	0.00	90.91	116.92
5	267.16	250.66	223.01	136.24	90.91	0.00	26.56
6	289.90	274.62	247.36	161.57	116.92	26.56	0.00

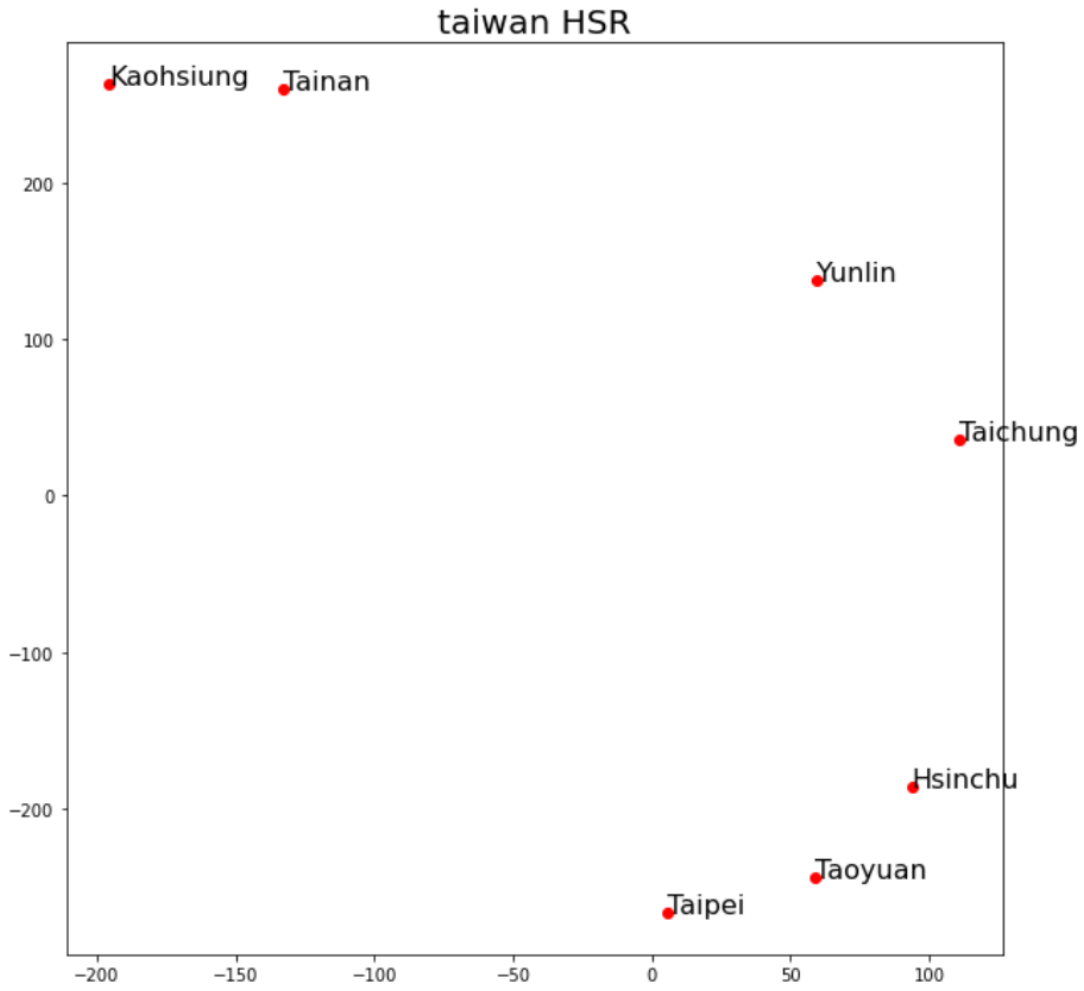
圖十一、高鐵站間相差之距離

4.2 前置處理

實驗二前置處理與實驗一相同。不同的是還要自行利用各高鐵站經緯度來計算距離。

4.3 實驗設計

實驗設計與實驗一略同，不同的是使用 MDS 來進行降維。以下為實驗二 MDS 降維後繪出之視覺化平面分布圖。(從下到上為台灣之北到南，台北、桃園、新竹、台中、雲林、台南、高雄高鐵站分布)



圖十二、MDS 降維後繪出的視覺化 2D 分布圖

五、結論

透過本次實驗使用了 MDS、PCA、t-SNE 三種降維方法來進行資料的視覺化，照理來說 t-SNE 是非線性的降維方法，經過測試分類效果也比其他先前的降維法更有效。本次實驗使用的 t-SNE 分群表現尚可，但也須注意每次使用的分布都會不一樣。使用 PCA 發現相較於 t-SNE 不一樣的是將屬性的相似度也考慮進去，讓我們更能看清楚資料間的關聯性。這次運算的資料量並沒有很大，將來會嘗試使用在更大的數據集上，並觀察、比較其分布。

六、參考資料

[1] 資料降維與視覺化 t-SNE 理論與應用，Mr. Opengate，民國 110 年 5 月 25 日取自: https://mropengate.blogspot.com/2019/06/t-sne.html

[2] 機器學習-降維演算法(MDS 演算法)，itread01，民國 110 年 5 月 27 日取自: https://www.itread01.com/content/1550606051.html

[3] 世上最生動的 PCA 直觀理解並應用主成分分析，LeeMeng，民國 110 年 5 月 28 日取自: https://leemeng.tw/essence-of-principal-component-analysis.html