**FDA HW3-2 Report**

**學號：F64051164**

**系級：109資訊系**

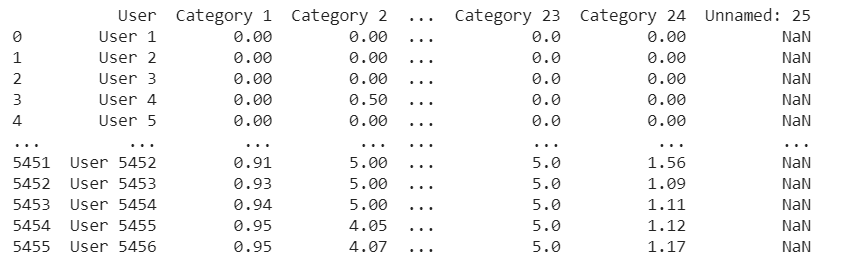
**姓名：張嘉榮**

1. **簡介**

這次我選擇的資料集是Kaggle上的Travel Review Ratings，藉由這個資料集去挖掘其中包含的隱藏訊息。資料集全部有25個欄位，每個欄位是每個User對某個旅遊項目的評分(例如教堂、花園、購物中心等等)，數量級在[0,5]這個區間，評分可為小數。

1. **資料預處理**
2. **讀取csv檔案並存成DataFrame格式。**

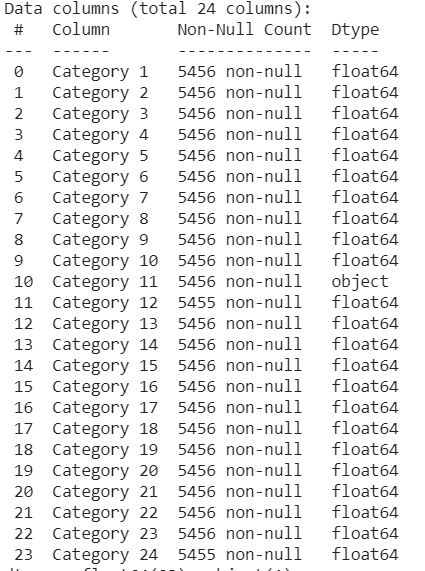
training\_set = pd.read\_csv('google\_review\_ratings.csv')



讀取資料集之後，我發現User和Unnamed這2個欄位是多餘的，因為User的ID對整體趨勢沒有幫助，而Unnamed也一樣。

1. **去除User和Unnamed欄位。**

training\_set.drop(columns = ['User','Unnamed: 25'] ,inplace = True)

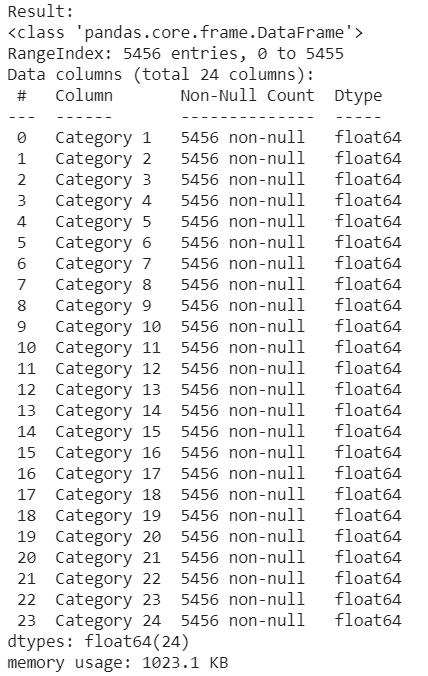


去除多餘資料集之後，我使用Dataframe的info函數確認資料集內容，我發現Category 11仍然有缺失值或錯誤的值。

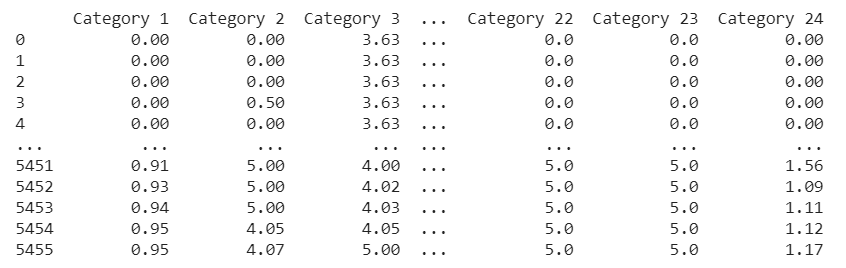
1. **處理Category 11中的NAN、NULL、錯誤數值，並用平均數填補。**

training\_set['Category 11'] = pd.to\_numeric(training\_set['Category 11'], errors = 'coerce')

training\_set = training\_set.fillna(training\_set.mean())

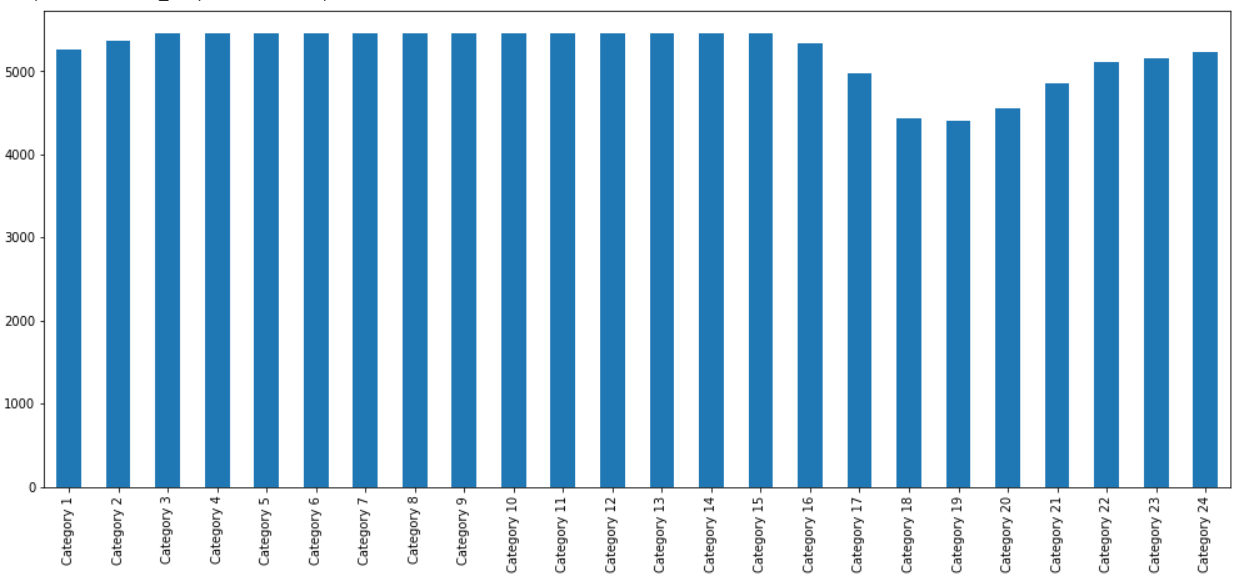


**最後的Dataframe:**



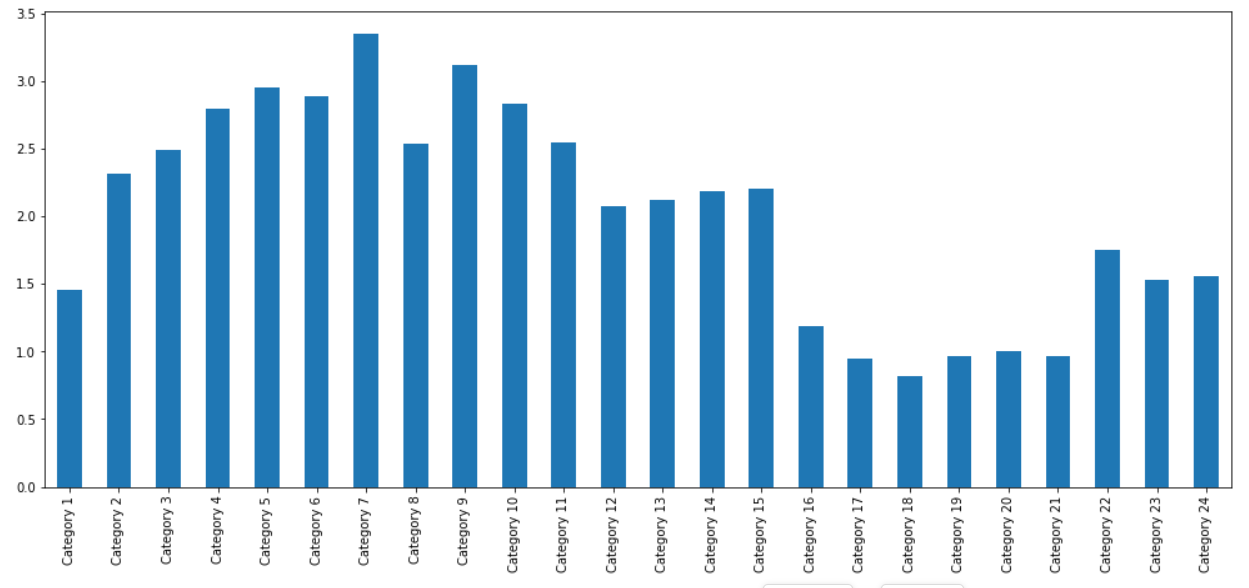
1. **資料集整體趨勢分析**

經過預處理之後，接下來便是資料集的趨勢分析，首先從所有Category的遊客評分數量下手：



除了幾個Category的數量略少之外(最低約4500個rating)，其它的Category的rating數量都很接近，因此可說是一個平衡的資料集。

確定資料的平衡以後，再來確認每個Category的平均評分：

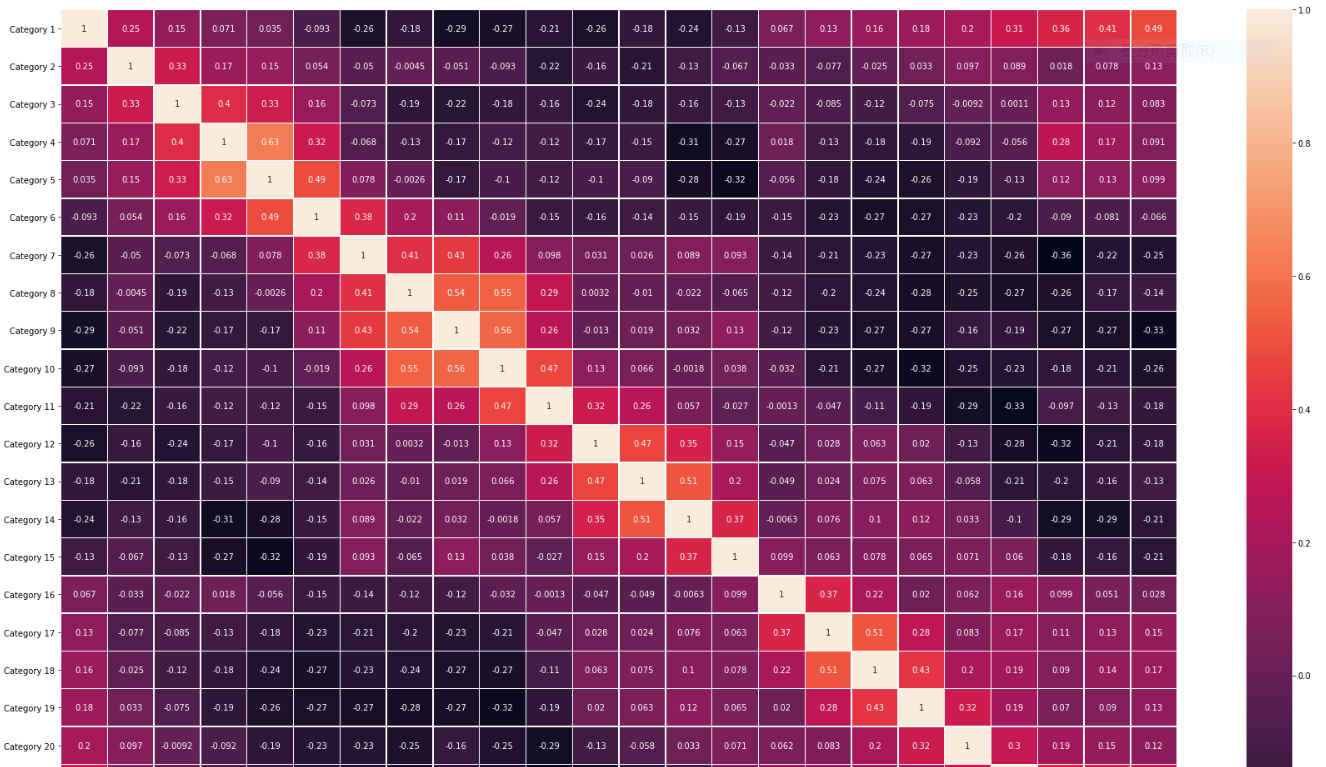


由上圖可知Category 7(大型購物商場)的平均評分最高，Category 18(健身房)的平均評分最低。

由Category的平均評分可以知道User對不同的設施有著不同的偏好，那兩個Category的相關程度也非常重要，如果有正相關，我就能確保以下的思維是正確的：

User 1對Category A的評分很高，若Category A和Category B的相關程度高，User對Category B的評分也會很高，那有著相同偏好的User 2和User 1是不是就能分成同一個cluster，而cluster的label本身就相當於這些Category的交集、聯集、差集呢？

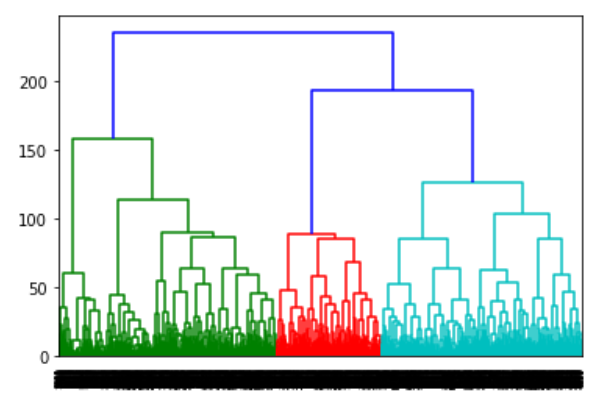
相關矩陣熱感圖(圖片有點大，**完整圖片請參考程式碼**)：



從這個圖可以知道有許多Category有一定程度的相關，例如Category 4和Category 5的相關係數超過0.6，且相鄰2個Category的相關程度很高，因此我開始實踐我的想法。

1. **Clustering實驗結果與討論**

首先，我先用dendrogram先初將這些User分群：

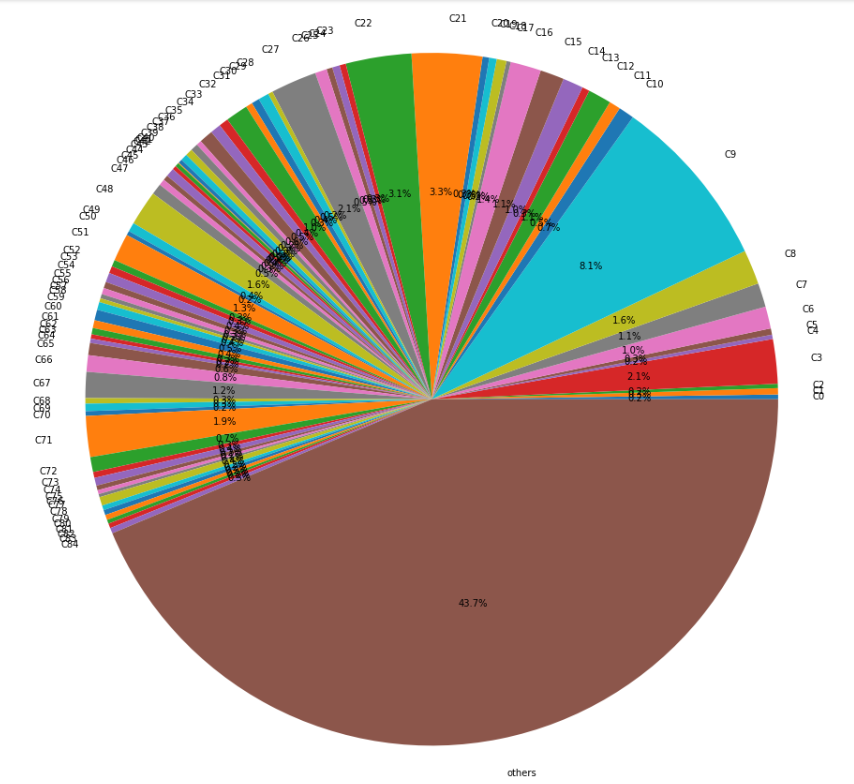


根據上面的bottom-up的dendrogram可以看到全部的User大致上可以分成3個cluster，雖然可以再用dendrogram分的更細緻，但是dendrogram難以區分noise，而且我認為不一定每個User都可以完完全全地分成某一類，也就是outlier，強硬地把這些outlier分成同一大群並不是好的策略，所以應該將outlier區分開來，因此我採用DBSCAN。

**Code:**

**model = DBSCAN(eps = 0.5, min\_samples = 10)**

一開始我隨便設定DBSCAN的超參數，結果如下(圖片可參考Code生成的原圖)：

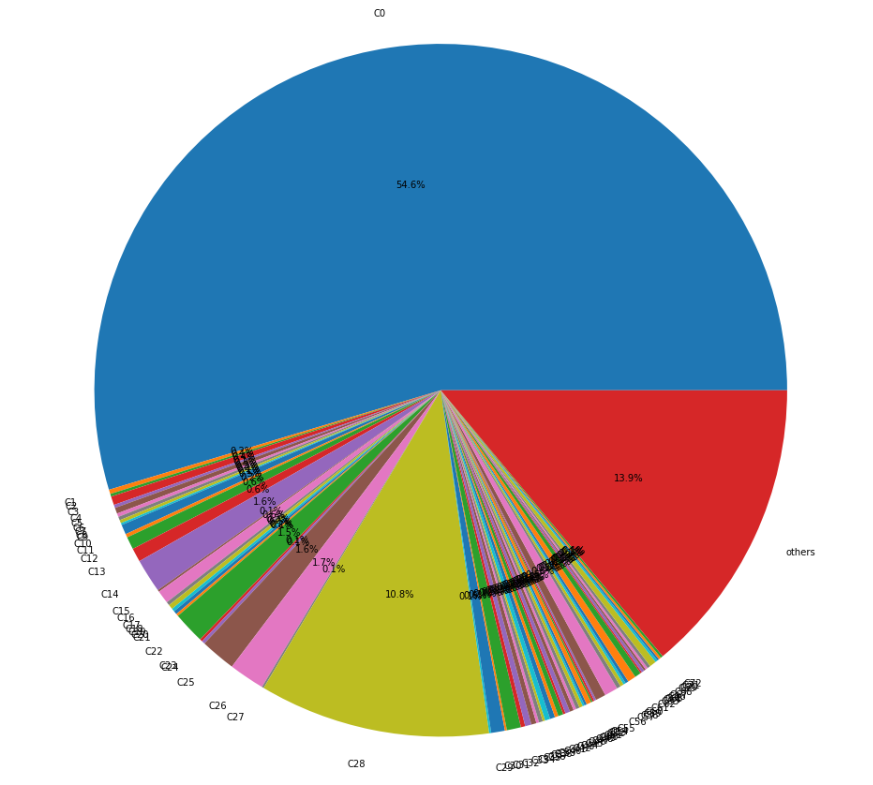


由上圖可知模型生成的outlier的數量太多了，約43.7%，也就是說有43.7%的人被這個模型認為是怪咖，無法被分成一群，而且除了outliers之外的每一個類別所占百分比很小，但這樣明顯是不合理的，因此需要進一步調整模型。

DBSCAN的本質就是從數據的密度來分群，所以設定eps(兩個數據點的最大距離)和min\_samples(成為一個cluster所需之最小數據點數量)就是關鍵。從一開始的超參數設定來看，除了outliers之外的每一個類別所占百分比很小應該是eps太小造成的，因此eps要適中，太大則會造成只有1~2個cluster，沒有分群意義，然後min\_samples設為5，也就是最少5人一個cluster，因此我採用以下程式碼：

**model = DBSCAN(eps = 0.65, min\_samples = 5)**

結果如下：

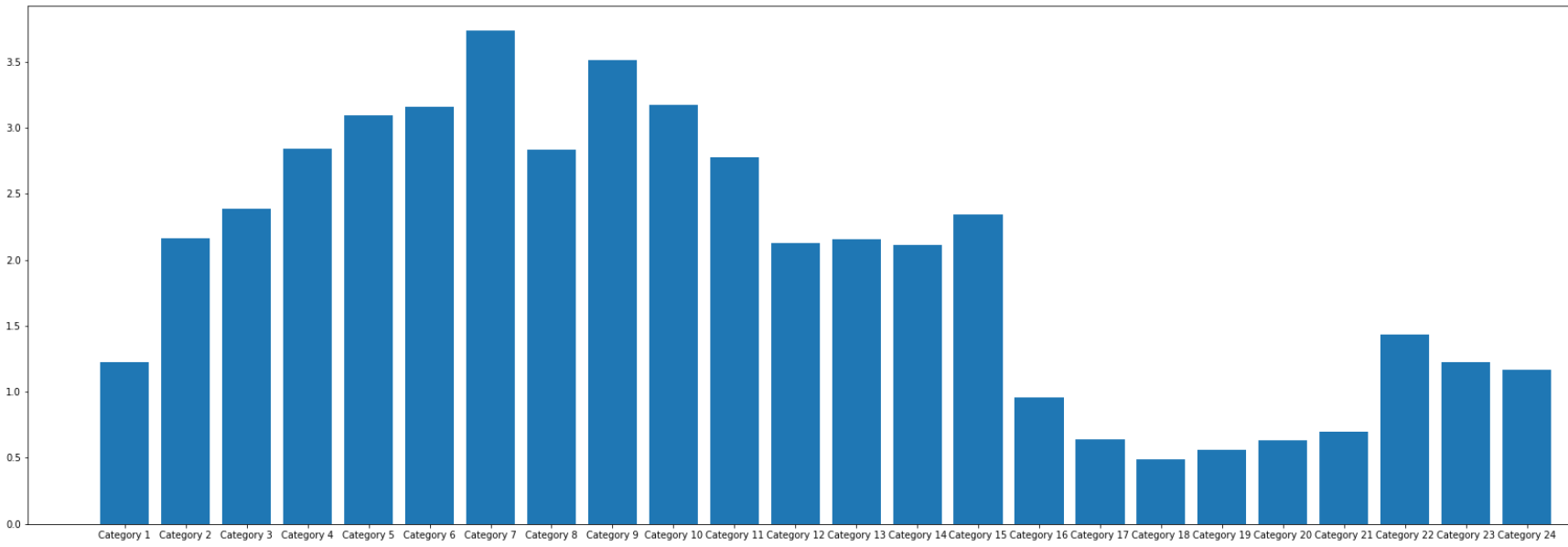


Kaggle上也有kernel是用dendrogram將資料分成2群，但是DBSCAN比較不會被noise影響，而只用dendrogram難以區分outlier，而DBSCAN不僅能夠區分出outlier，它也比較符合將人分群的情境，因為有些人的喜好本來就比較特殊，硬是把這些人分成一群，說他們是同一類本身就不合理。

這次的分類結果如下：

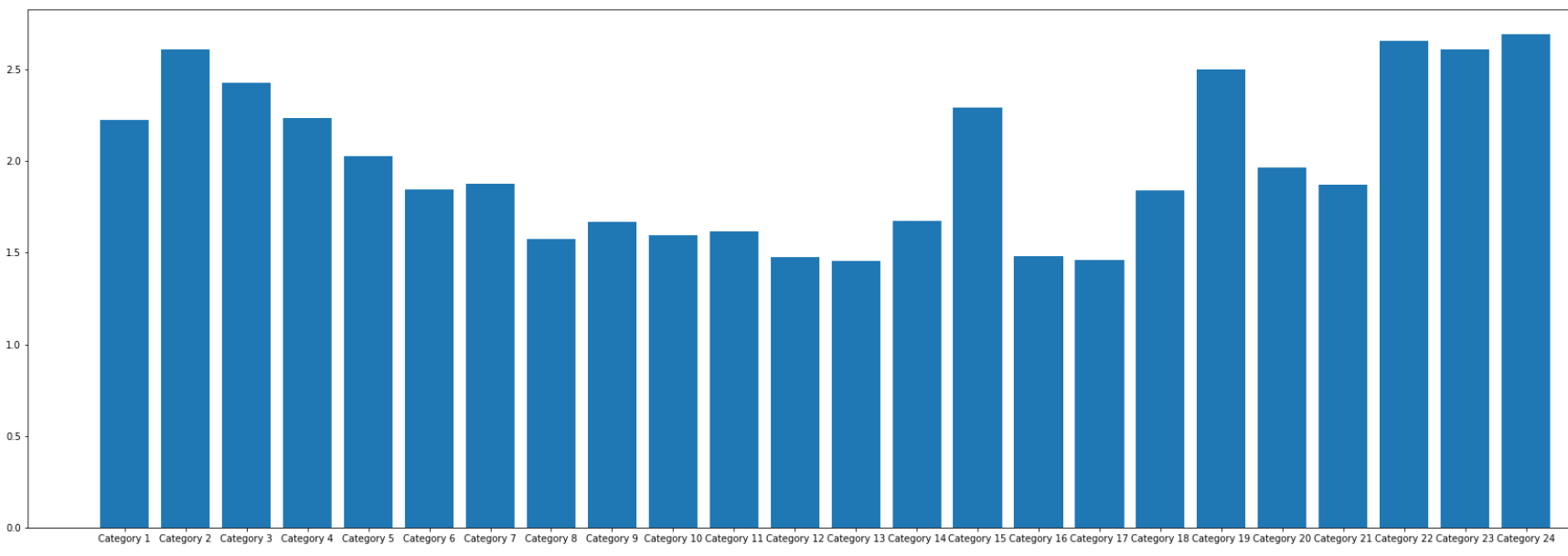
Outliers只剩下13.9%，最大群佔54.6%，第二大群佔10.8%，其它群有些有1.6%、1.7%的百分比。

**最大的cluster(54.6%)中rating的平均值(請參考原圖)：**



由上圖可知Category 7(malls)的分數最高，分在這個cluster的User比較偏愛購物商場，且以Category 7為中心，其他的Category大致上與Category 7越接近rating越高，形成一個山丘形的分布。

**第二大的cluster(10.8%)中rating的平均值(請參考原圖)：**



這個cluster的平均值較低(最大的平均值約2.8左右)，不同於最大的cluster，這個cluster呈現中間低，兩側高的分布。