MDA\_HW3\_KMeans

107062321 王劭元

在這次的作業中我主要共用到了6個mapper，2個reducer

**mapperE\_dist**：在這個mapper中，我是將每個點去與centroid算出他們之間的Euclidean distance後，以我希望的形式包好後回傳

**mapperE\_min**：在這個mapper中，傳進來的資料會是data中的點，後面接著十個centroid，我用迴圈掃過10個centroid後，會得到是哪個centroid距離最短，並且距離為多少，最後要回傳時，我除了回傳是哪個centroid之外，我還會回傳這個centroid的index以利於後面可以sortByKey，讓最後csv檔的輸出可以順利一點。並且我在回傳時，我在最後的KV pair就會直接回傳這個點所產生的cost為多少，這樣最後我對data所有點的這個值用api sum()起來，就可以得到這次的總cost為多少。除此之外，由於發現data中有重複出現的點會影響到cost，所以我還會去算若同樣的點後面還接了幾個，最後的總cost會去乘上點的數量。

**mapperM\_dist**：在這個mapper中，我是將每個點去與centroid算出他們之間的Manhattan distance後，以我希望的形式包好後回傳

**mapperM\_min**：在這個mapper中，與mapperE\_min實作的方式幾乎一樣，我只是為了要回傳不同的cost。所以這邊只有在回傳的cost算法不一樣，其他都跟mapperE\_min一樣。

**mapperNew\_cen**：在這個mapper中，傳進來的資料會是其中一個centroid接上，現在屬於這個centroid，或者說新的這個cluster的點。所以要計算新的cluster的中心時，就把後面用迴圈全部掃過，並且把各dimension的值加起來平均後，即可得到新的centroid。

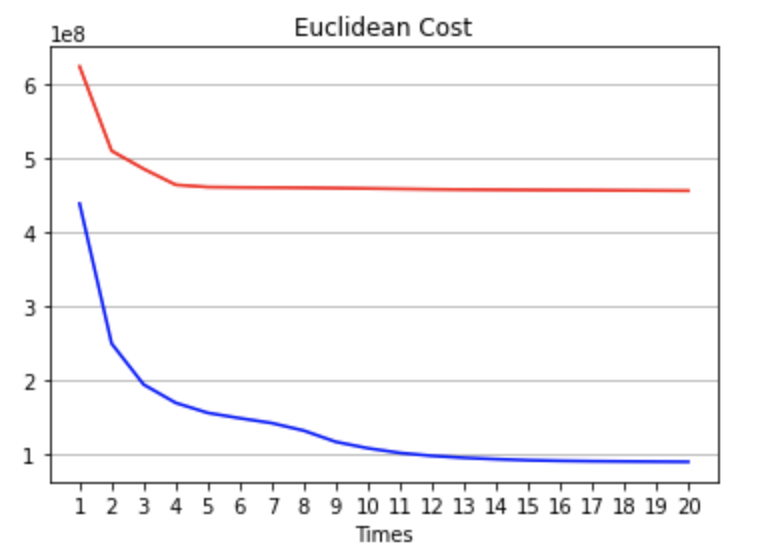
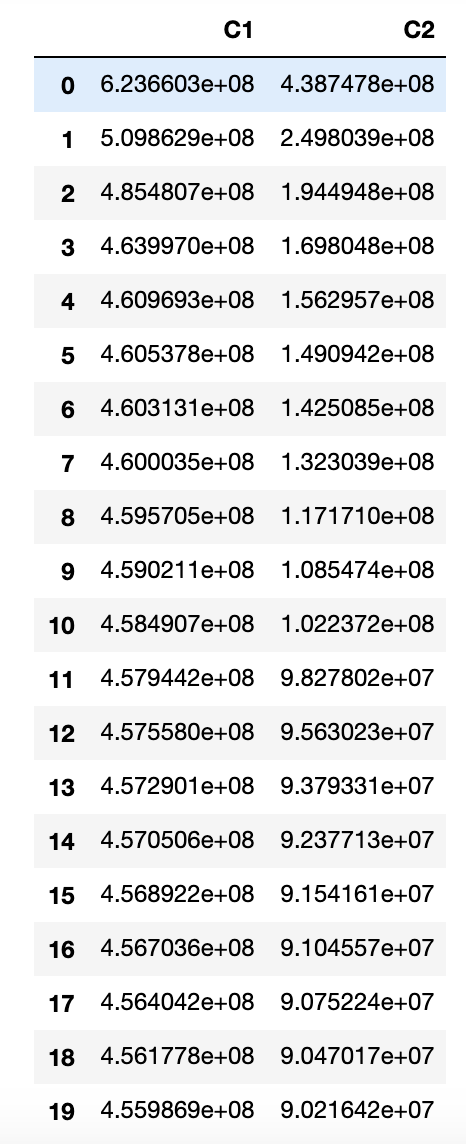
**mapperinit**：在這個mapper中，我為了讓每次iteration完，紀錄centroid的rdd回復成我一開始的型態所用的。會回傳 (0, centroid)。

**Reducer**：兩個reducer的功用都蠻像的，都是為了讓我可以將資料串接在一起。一次是在每次iteration一開始，我的centroid和data這兩個rdd的kv pair都長成 (0, centroid) (0, data)，的形式，所以我先把他們join起來後，算完每一個點和centroid的距離後，為了要去計算min distance，要把同一個點對不同的centroid的這些pair串起來，故我這裡用了一次reduceByKey(lambda a, b:a+b)。

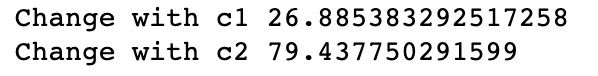
第二次是在把每個centroid的與某個data point的cost算完後，為了要算出總cost要把所有屬於這個cluster得data point串在centroid的後面，於是我在這邊用了第二次的reduceByKey(lambda a, b:a+b)。

**(a)**

**(1)** A plot of cost vs. iteration for 2 initialization strategies



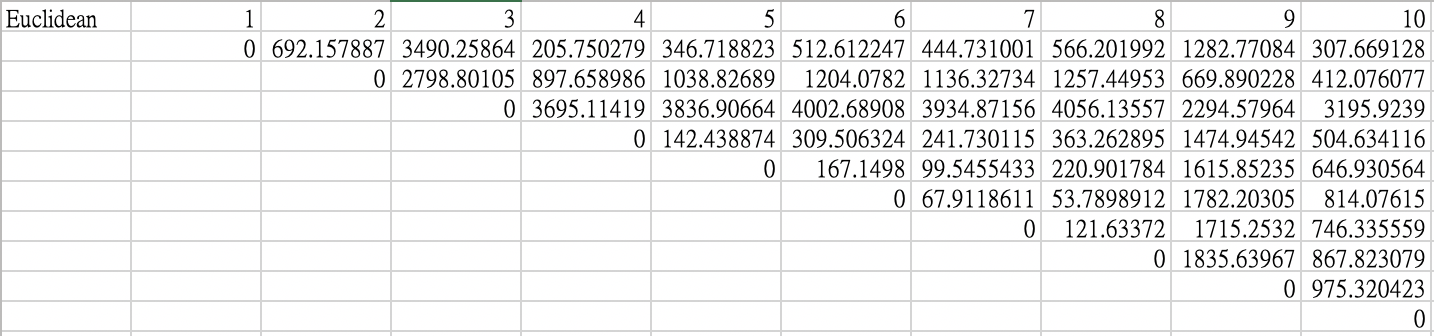
(b) Percentage improvement values and your explanation



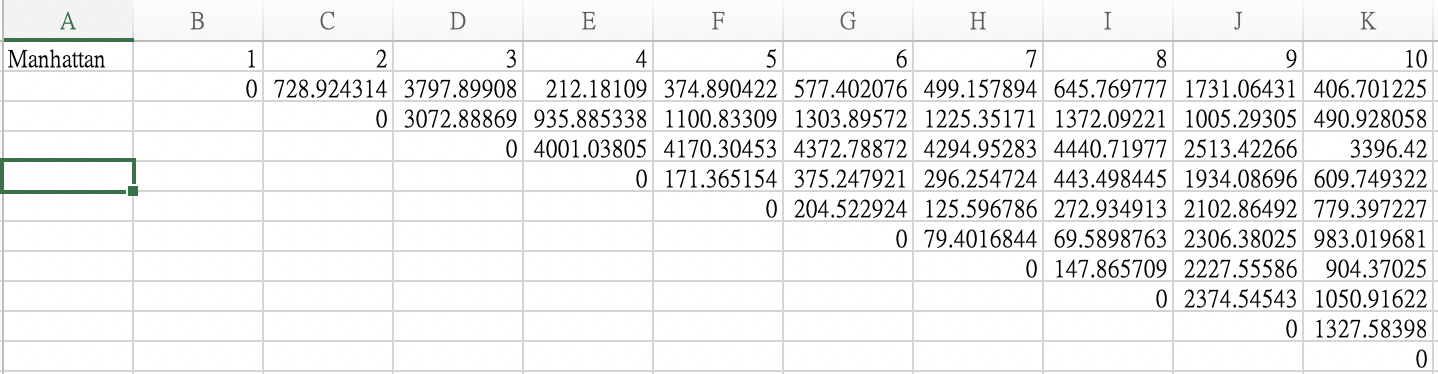
由於Euclidean計算cost的方式為距離的平方，所以對於cluster距離遠近、離群值較為敏感，故在這種情況，使用c2事先刻意挑好讓最遠的點可以自己形成一種cluster的方式，最後得到結果以及下降的幅度也較好。

(c)

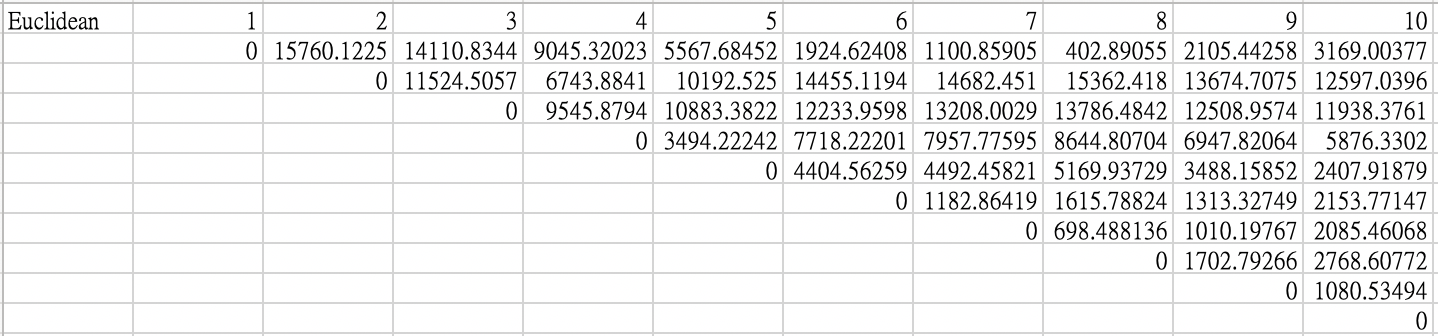
(1) Euclidean distance for all pairs of centroids, with c1 started



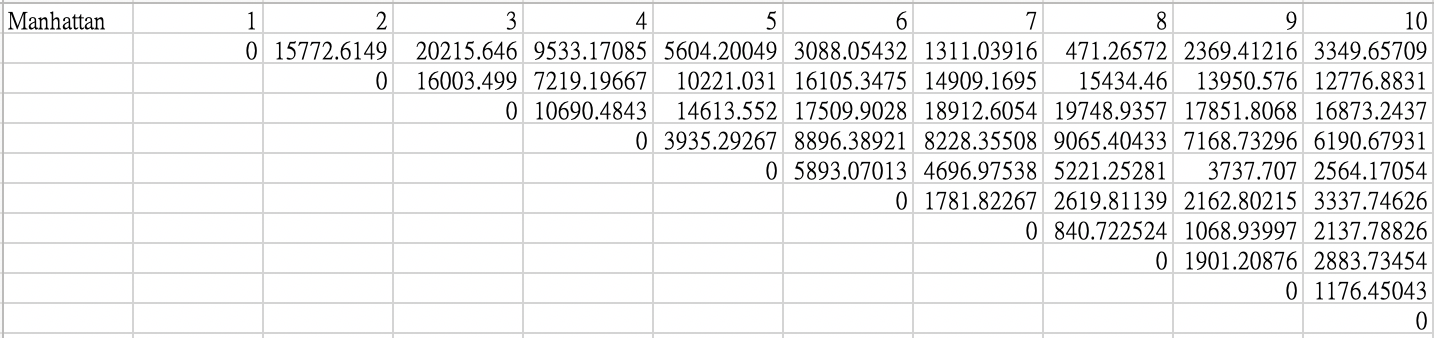
(2) Manhattan distance for all pairs of centroids, with c1 started



(3) Euclidean distance for all pairs of centroids, with c2 started

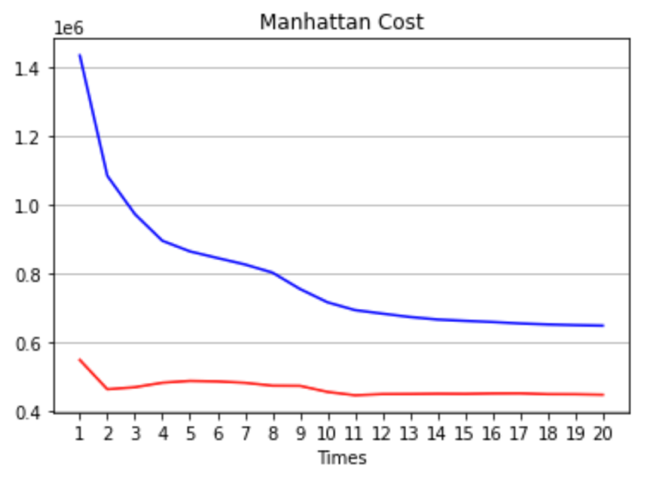
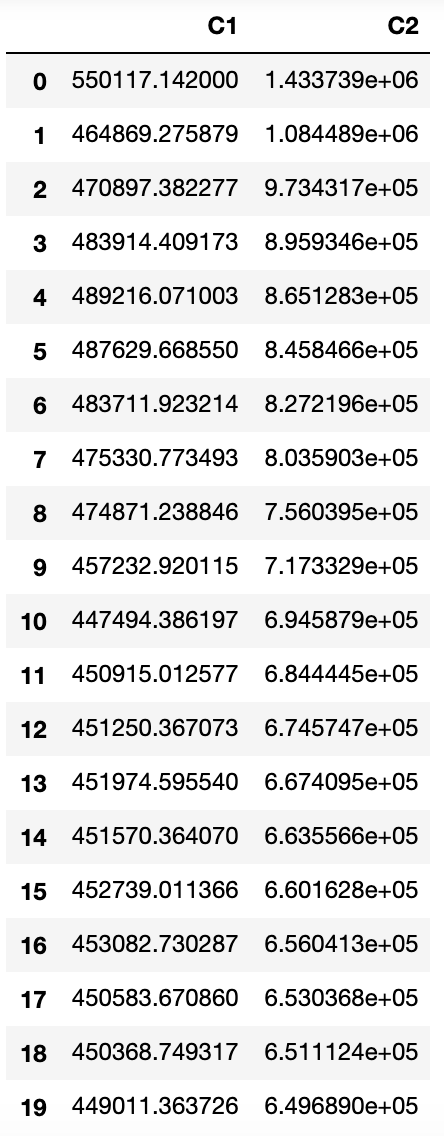


(4) Manhattan distance for all pairs of centroids, with c2 started

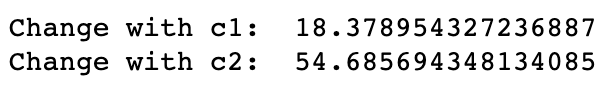


**(b)**

**(1)** A plot of cost vs. iteration for 2 initialization strategies



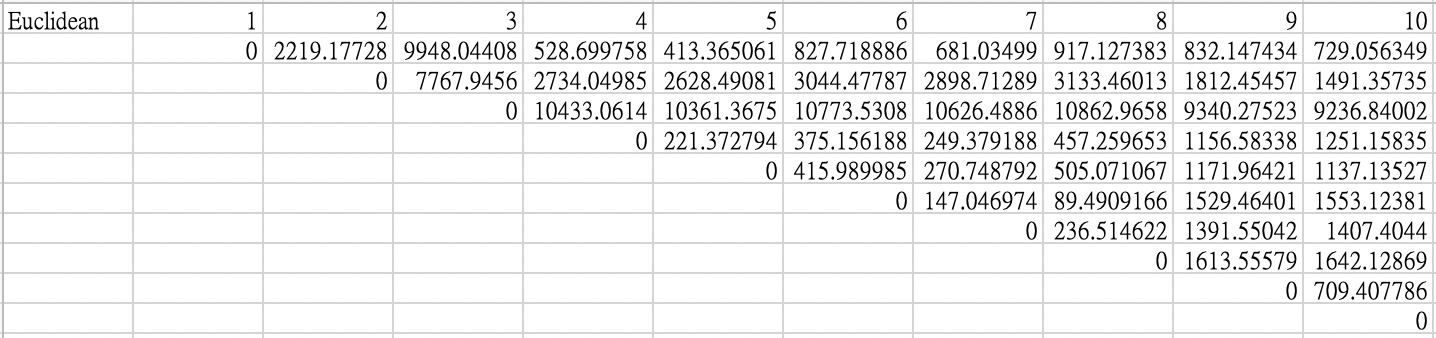
(b) Percentage improvement values and your explanation



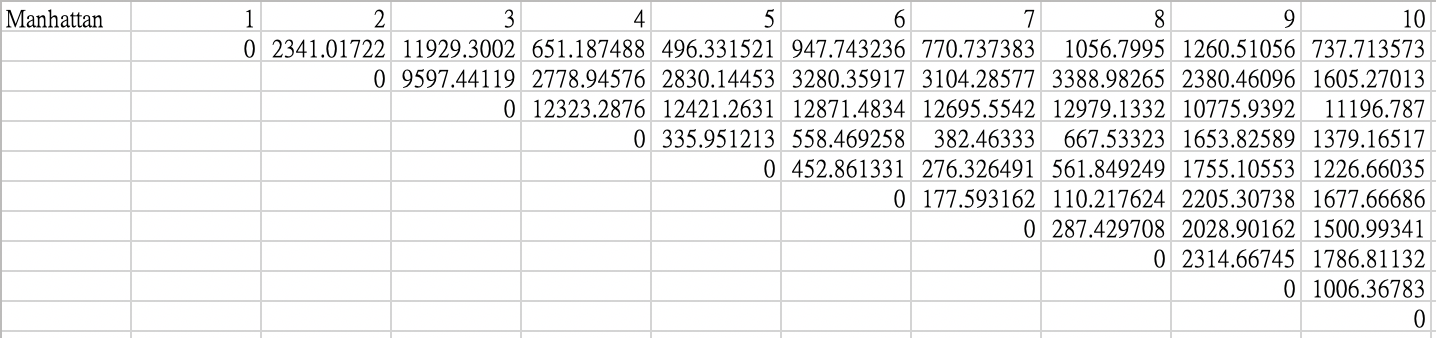
由於Manhattan在計算cost的方式相對來說比較不敏感，Manhattan更希望你能將集中很多點的大群即使距離很近，也可以分成不同cluster，故在這種情況，一開始還特地將距離拉遠來看，最後達到的效果沒有比較好。

(c)

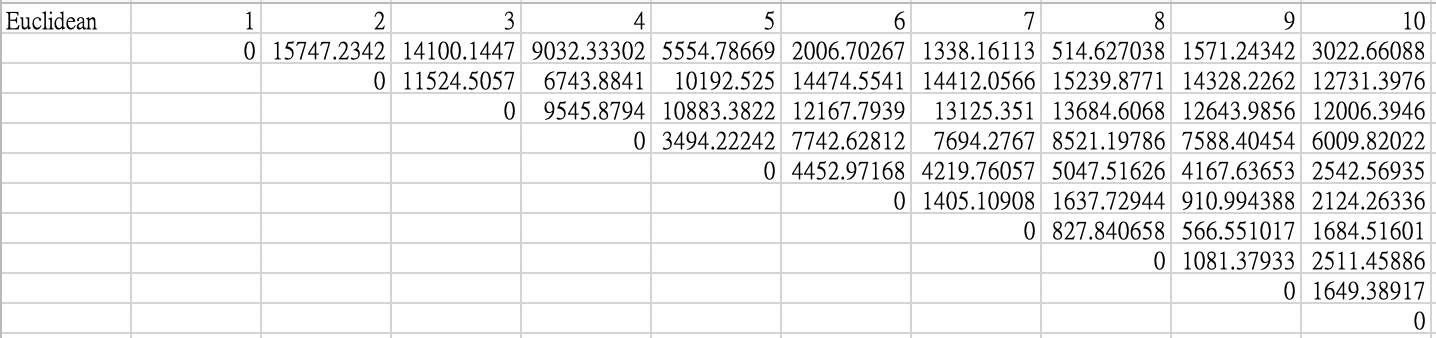
(1) Euclidean distance for all pairs of centroids, with c1 started



(2) Manhattan distance for all pairs of centroids, with c1 started



(3) Euclidean distance for all pairs of centroids, with c2 started



(4) Manhattan distance for all pairs of centroids, with c2 started

