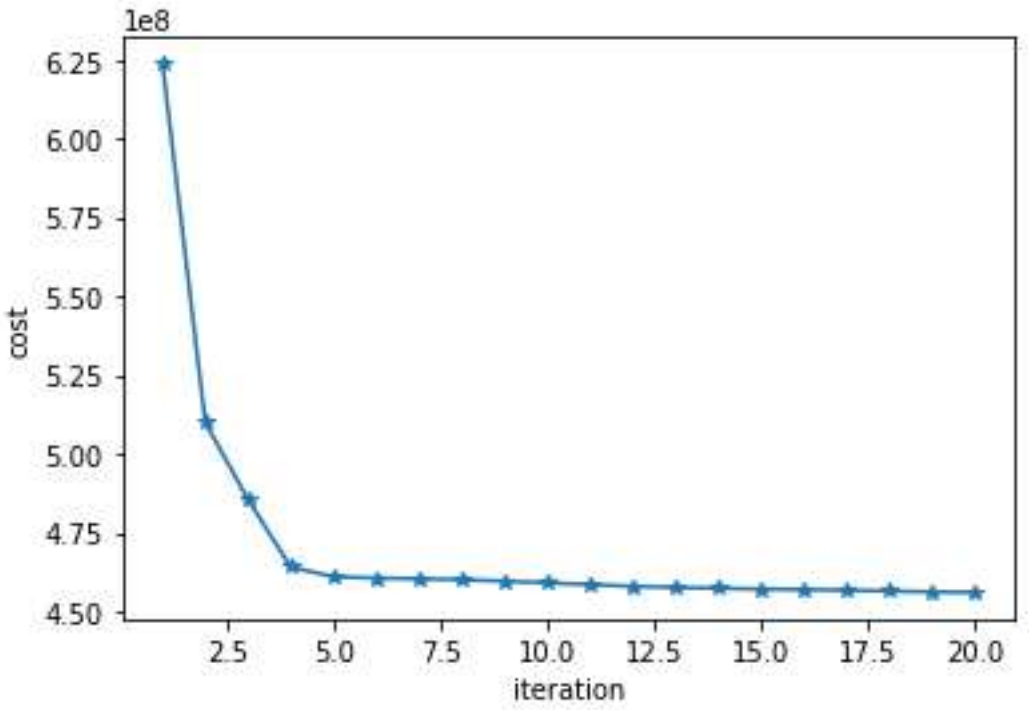
Part A

a. Exploring initialization strategies with Euclidean distance 1. Using the Euclidean distance (refer to Equation

1) as the distance measure, compute the cost function ϕ(i) (refer to Equation 2) for every iteration i. This means that, for your first iteration, you will be computing the cost function using the initial centroids located in one of the two text files. Run the k-means on data.txt using c1.txt and c2.txt. Generate a graph (line plot) where you plot the cost function ϕ(i) as a function of the number of iterations i=1..20 for c1.txt and also for c2.txt.

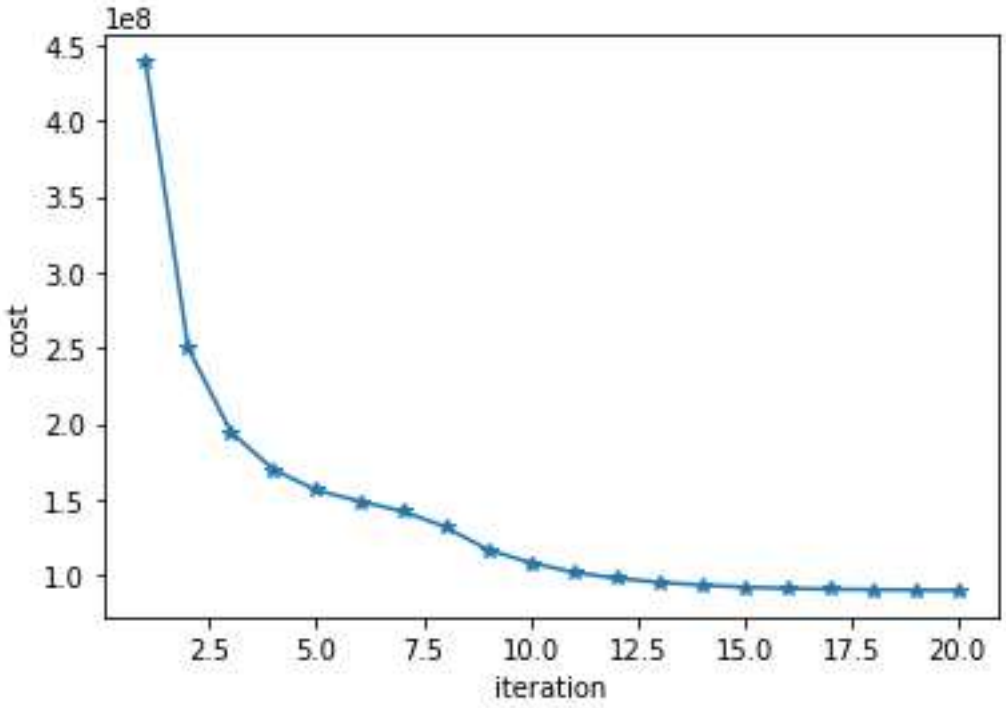
Iterations & graph plot using c1.txt as centroids and calculating Euclidean distance

iteration 1 : 623660345.3064235
iteration 2 : 509862908.29754597
iteration 3 : 485480681.87200826
iteration 4 : 463997011.6850107
iteration 5 : 460969266.572994
iteration 6 : 460537847.98277014
iteration 7 : 460313099.65354246
iteration 8 : 460003523.88940686
iteration 9 : 459570539.3177353
iteration 10 : 459021103.3422901
iteration 11 : 458490656.1919807
iteration 12 : 457944232.5879742
iteration 13 : 457558005.1986796
iteration 14 : 457290136.3523032
iteration 15 : 457050555.0595639
iteration 16 : 456892235.61535746
iteration 17 : 456703630.7370357
iteration 18 : 456404203.0189769
iteration 19 : 456177800.54199505
iteration 20 : 455986871.02734846



Iterations & graph plot using c2.txt as centroids and calculating Euclidean distance

iteration 1 : 438747790.027918
iteration 2 : 249803933.62600294
iteration 3 : 194494814.40631396
iteration 4 : 169804841.45154333
iteration 5 : 156295748.80627596
iteration 6 : 149094208.10896605
iteration 7 : 142508531.61961588
iteration 8 : 132303869.40653005
iteration 9 : 117170969.8371908
iteration 10 : 108547377.17857017
iteration 11 : 102237203.31799614
iteration 12 : 98278015.74975717
iteration 13 : 95630226.12177445
iteration 14 : 93793314.05119292
iteration 15 : 92377131.96821107
iteration 16 : 91541606.25423913
iteration 17 : 91045573.83042422
iteration 18 : 90752240.10140836
iteration 19 : 90470170.18122767
iteration 20 : 90216416.17563146



2) Is random initialization of k-means using c1.txt better than initialization using c2.txt in terms of cost ϕ(i)? Explain your reasoning.

We know that: "Sum of squares of distances of every data point from its corresponding cluster centroid should be as minimum as possible". This sum of squares is nothing but cost value.

Centroid initialization for k-means using c2.txt is better than c1.txt in terms of cost using Euclidean distance measure because as seen in the above two graphs the cost values for every iteration is high when c1.txt is considered as centroids. And also, the c1.txt centroids are the first 10 data points of the dataset, but c2.txt have centroids far away compared to c1.txt. The clusters stabilize just after 4 iterations using c1 centroids, but the clusters stabilize after 10 iterations using c2 centroids with low cost value compared to c1. In my understanding may be the impact of outliers is less with c2 centroids as they are far away from each other.

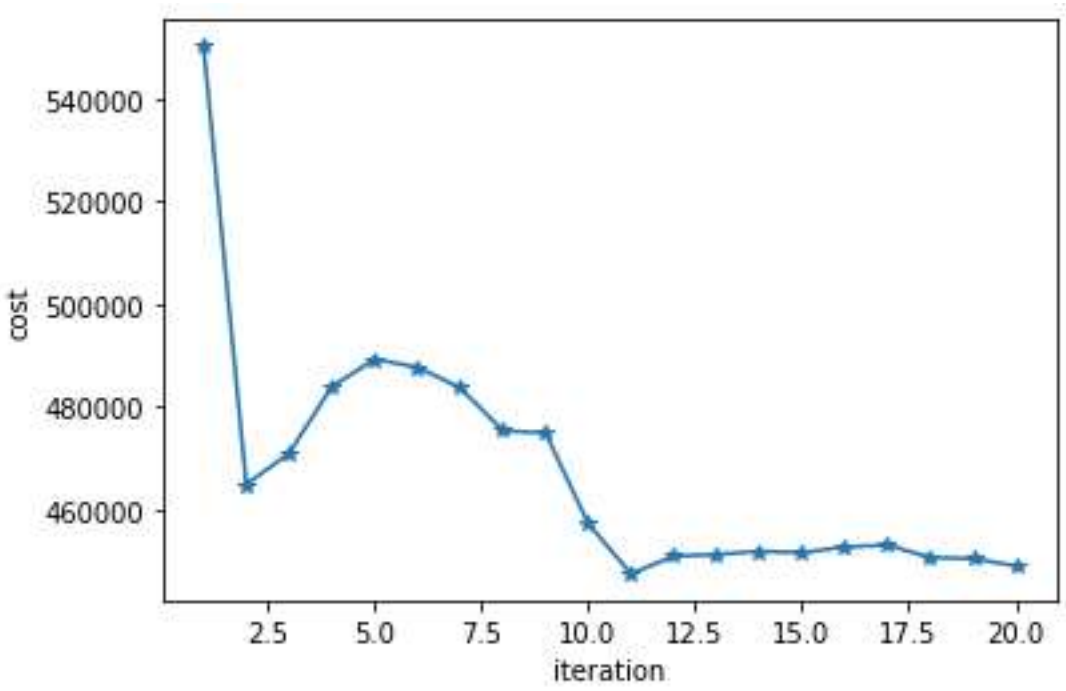
b. Exploring initialization strategies with Manhattan distance

1. Using the Manhattan distance metric (refer to Equation 3) as the distance measure, compute the cost function ψ(i) (refer to Equation 4) for every iteration i. This means that, for your first iteration, you'll be computing the cost function using the initial centroids located in one of the two text files. Run the k-means on data.txt using c1.txt and c2.txt. Generate a graph where you plot the cost function ψ(i) as a function of the number of iterations i=1..20 for c1.txt and also for c2.txt.

Iterations & graph plot using c1.txt as centroids and calculating Manhattan distance

iteration 1 : 550117.1420000045
iteration 2 : 464829.26840394654

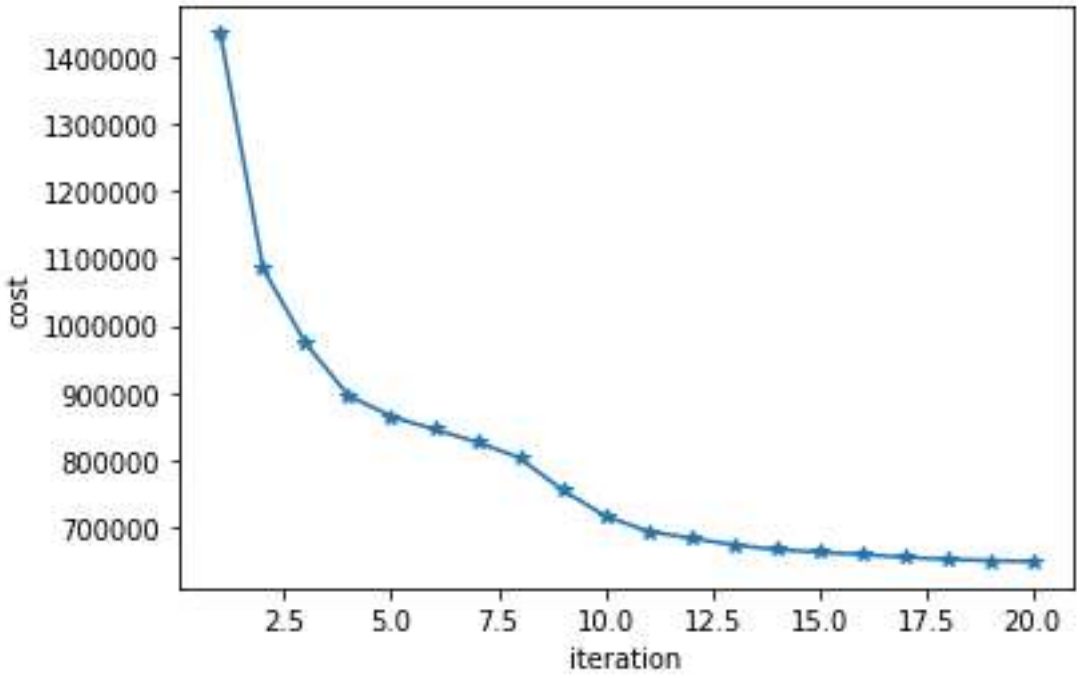
iteration 3 : 470934.15384668263
iteration 4 : 483874.81628509297
iteration 5 : 489234.2347883483
iteration 6 : 487664.6926267901
iteration 7 : 483718.66592851654
iteration 8 : 475337.94763305597
iteration 9 : 474871.9665496577
iteration 10 : 457244.78974174923
iteration 11 : 447493.195604051
iteration 12 : 450891.8358047716
iteration 13 : 451232.5774756949
iteration 14 : 451860.12588546367
iteration 15 : 451567.2235891512
iteration 16 : 452710.0520999444
iteration 17 : 453078.22696184984
iteration 18 : 450646.13556209754
iteration 19 : 450419.97011343326
iteration 20 : 449009.59037188475



Iterations & graph plot using c2.txt as centroids and calculating Manhattan distance

iteration 1 : 1433739.3099999938
iteration 2 : 1084488.7769648738
iteration 3 : 973431.7146620394
iteration 4 : 895934.5925630673
iteration 5 : 865128.3352940796

iteration 6 : 845846.6470313473
iteration 7 : 827219.5827561237
iteration 8 : 803590.3456011107
iteration 9 : 756039.5172761244
iteration 10 : 717332.9025432297
iteration 11 : 694587.9252526843
iteration 12 : 684444.5019967926
iteration 13 : 674574.7475478566
iteration 14 : 667409.469916026
iteration 15 : 663556.6278214998
iteration 16 : 660162.777228758
iteration 17 : 656041.3222947085
iteration 18 : 653036.7540731638
iteration 19 : 651112.4262522653
iteration 20 : 649689.0131843556



2. Is random initialization of k-means using c1.txt better than initialization using c2.txt in terms of cost ψ(i)? Explain your reasoning.

Unlike in the case of Euclidean distance measure, in Manhattan c1.txt centroids are better than c2.txt.

Thought the graph plot looks good for c2 centroids with elbow curve, and there is gradual decrease in the cost, using c1 centroids the cost value is low. With c1 centroids the cost value drops after 2nd iteration and then increases up to 6th iterations and then drops with stabilized cost value after 11th iteration. With c2 centroids cost value stabilizes after 11th iteration but the cost value is high.

In the case of Manhattan distance measure, c1.txt or random initialization is better than c2.txt or far away centroids initialization.