**Q1**

a)

|  |  |  |
| --- | --- | --- |
|  | Counters in Main Memory | Frequent items |
| Pass1 | b:6,a:5,d:3,c:2,e:2 | b,a,d |
| Pass2 | ab:4,bd:3,ad:2 | ab,bd |

b)

Confidence = 3/6=1/2

Support = 3/7 s(bd)=3

No

c)

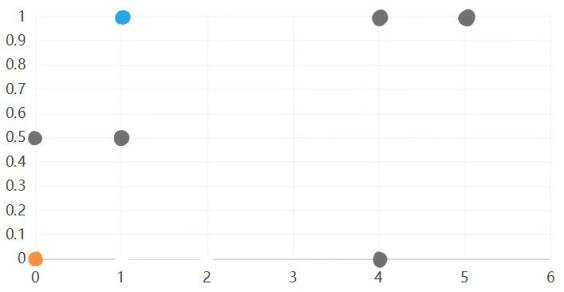
|  |  |  |  |
| --- | --- | --- | --- |
|  | Counters in Main Memory | Counters for the buckets | Frequent items |
| Pass1 | 1:2,2:2,3:2,4:2,7:2,5:1,6:1 | B0:3 ,B1:3 ,B2:1 | 1,2,3,4,7 |
| Pass2 | (1,3):1,(3,4):1,(2,7):2 | B0:3 ,B1:3 ,B2:1 | (2,7) |

**Q2**

1.

P1=(0, 0), P2=(0, 1/2), P3=(1, 1/2), P4=(1, 1), P5=(4, 0), P6=(4, 1),P7=(5, 1)

P1 and P4 are selected as centroids



For each point P2,P3, P5, P6,P7, we determine the closest centroid.

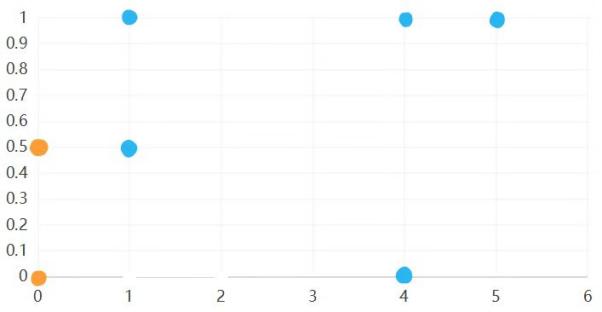
d(P2, P1) = 1/2, d(P2, P4) = √5/2,P2 is assigned to the orange cluster.

d(P3, P1) = √5/2, d(P3, P4) = 1/2,P3 is assigned to the blue cluster.

d(P5, P1) = 4, d(P5, P4) = √10,P5 is assigned to the blue cluster.

d(P6, P1) = √17, d(P5, P4) = 3,P6 is assigned to the blue cluster.

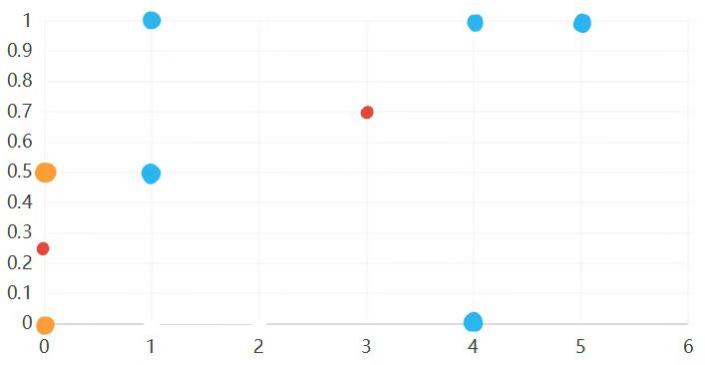
d(P7, P1) = √26, d(P5, P4) = 4,P7 is assigned to the blue cluster.



We recompute the centroids C1, C2 as follows:

C1 = (P1+P2)/2 = (0,1/4)

C2 = (P3 + P4 + P5 +P6 +P7)/5 = (3,7/10)



For each point, we determine the closest centroid:

d(P1, C1) = 1/4, d(P1, C2) = √949/10, P1 stays orange.

d(P2, C1) = 1/4, d(P2, C2) = √226/5, P2 stays orange.

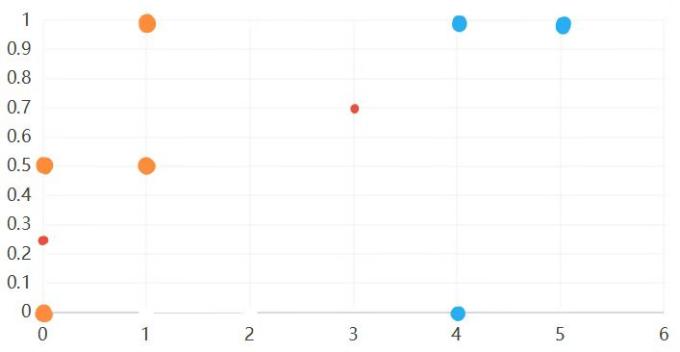
d(P3, C1) = √17/4, d(P3, C2) = √101/5, P3 gets orange.

d(P4, C1) = √2, d(P4, C2) = √409/10, P4 gets orange.

d(P5, C1) = √257/4, d(P5, C2) = √149/10, P5 stays blue.

d(P6, C1) = √265/4, d(P6, C2) = √109/10, P6 stays blue.

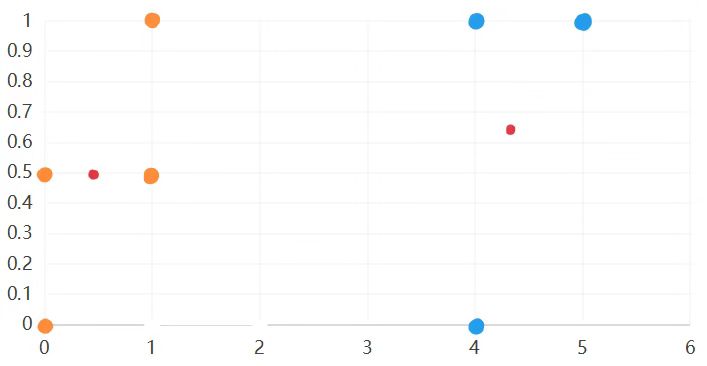
d(P7, C1) = √409/4, d(P7, C2) = √409/10, P7 stays blue.



We recompute the centroids C1, C2 as follows:

C1 = (P1 + P2 + P3 + P4 )/4 = (1/2,1/2)

C2 = (P5 +P6 +P7)/3 = (13/3,2/3)



d(P1, C1) = √2/2, d(P1, C2) = √173/3, P1 stays orange.

d(P2, C1) = 1/2, d(P2, C2) = √677/6, P2 stays orange.

d(P3, C1) = 1/2, d(P3, C2) = √401/6, P3 stays orange.

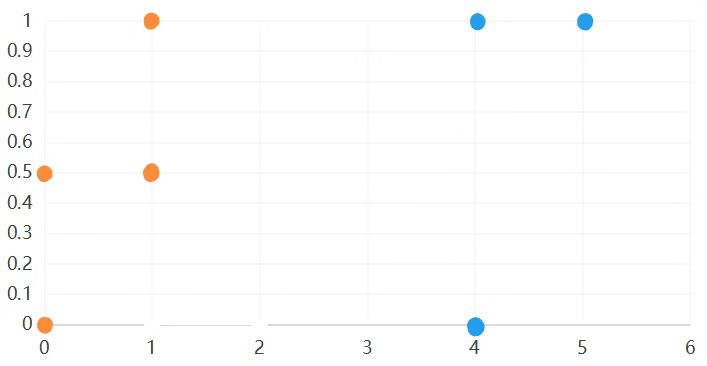
d(P4, C1) = √2/2, d(P4, C2) = √101/3, P4 stays orange.

d(P5, C1) = √50/2, d(P5, C2) = √5/3, P5 stays blue.

d(P6, C1) = √50/2, d(P6, C2) = √2/3, P6 stays blue.

d(P7, C1) = √82/2, d(P7, C2) = 2√2/3, P7 stays blue.

As the clustering did not change, the algorithm terminates with the following final clustering:



2.

I assume there are 6 points in the one-dimension euclidean space,P1=0,P2=1,P3=3.2,P4=3.4,P5=3.4,P6=5.6. Suppose that P1 = 0, P2 = 1 and P6 = 5.6 are chosen as initial centroids for the K-means algorithm, K=3.

8498d50b0891989ba478f8ff924b07a

d(P1,P3)=3.2,d(P2,P3)=2.2,d(P6,P3)=2.4,P3 is assigned to the purple cluster.

d(P1,P4)=3.4,d(P2,P4)=2.4,d(P6,P4)=2.2,P4 is assigned to the blue cluster.

d(P1,P5)=3.4,d(P2,P5)=2.4,d(P6,P5)=2.2,P5 is assigned to the blue cluster.

7b3dc4fe02092b1168c16476040c415

C1=P1=0

C2 = (P2+P3)/2 = 2.1

C3 = (P4 + P5 +P6)/3 =4.13

abcbf3a0f569bda11c2ac964055e88d

d(P1, C1) = 0, d(P1, C2) = 2.1, d(P1, C3) = 4.13,P1 stays orange.

d(P2, C1) = 1, d(P2, C2) = 1.1, d(P2, C3) =3.13,P2 gets orange.

d(P3, C1) = 3.2, d(P3, C2) = 1.1, d(P3, C3) =0.93,P3 gets blue.

d(P4, C1) = 3.2, d(P4, C2) = 1.1, d(P4, C3) =0.93,P4 gets blue.

d(P5, C1) = √257/4, d(P5, C2) = √149/10, d(P5, C3) =,P5 stays blue.

d(P6, C1) = √265/4, d(P6, C2) = √109/10, d(P6, C3) =,P6 stays blue.

543d402b6e442047ab42993101b153d

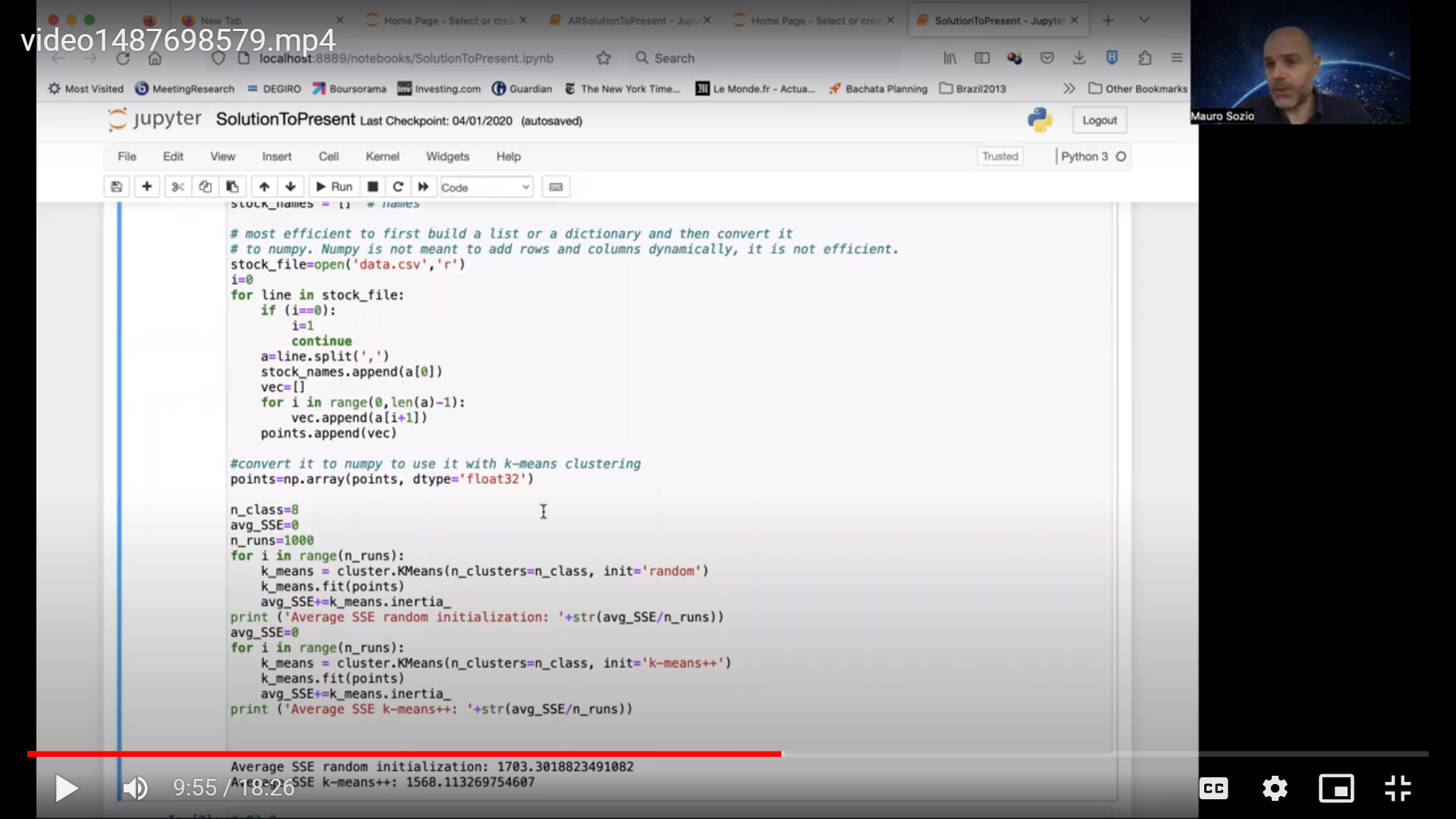
C1= (P1+P2)/2=0.5

C2 = null

C3 = (P3+P4 + P5 +P6)/4 =3.9

So it produces an empty cluster.

**Q3**



1. While using init=’random’ and the default values for the other parameters, sum of squared errors (SSE) is 1691.3028573043973
2. init='k-means++', sum of squared errors (SSE) is 1591.2393590861036.

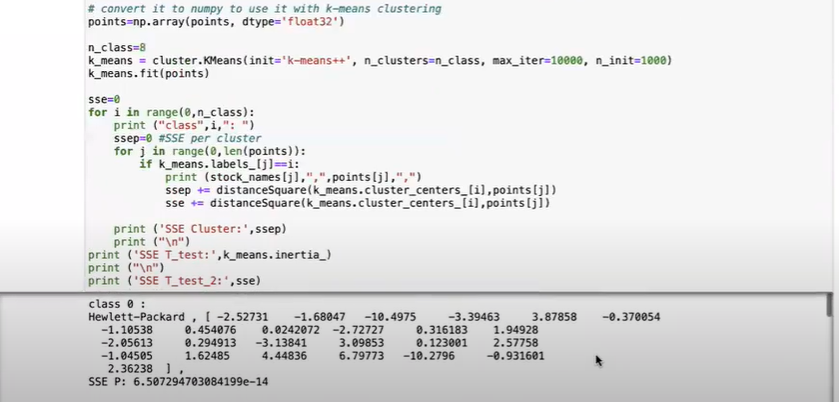
n\_init=100, sum of squared errors (SSE) is 1603.9558306515232.

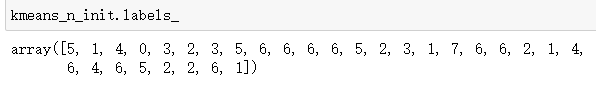
I think these two parameters impact the results the most.

‘k-means++’ selects initial cluster centroids using sampling based on an empirical probability distribution of the points’ contribution to the overall inertia. So it initializes the centroids better than before. It will not necessarily improve the results.

‘n\_init ’ means number of times the k-means algorithm is run with different centroid seeds. The final results is the best output of n\_init consecutive runs in terms of inertia. Increasing the value of the ‘n\_init’ should always improve the results.

3)





Lable 2 cluster of Food stocks : Kraft,Procter & Gamble,AT&T,McDonalds,Coca-Cola

Lable 3 cluster of Manufacturing stocks: DuPont,Caterpillar,Alcoa

Lable 5 cluster of Bank stocks: American Express,Bank of America,Walt Disney,JPMorgan Chase

Lable 6 cluster of Technology stocks:Verizon,Microsoft,IBM,The Home Depot,Wal-Mart,General Electric,United Technologies,Travelers,3M

**Q4**

1)

(Margin=1, BI-RADS=4)->(Severity=0) the support is 0.299688, the confidence is 0.911392

(Margin=1, Severity=0)->(BI-RADS=4) the support is 0.299688, the confidence is 0.911392

(Margin=1, Severity=0, Density=3)->(BI-RADS=4) the support is 0.238293, the confidence is 0.927126

2)

('Margin=1', 'Shape=2')->('Severity=0')

('Shape=1', 'Margin=1', 'Density=3')->('Severity=0')

The margin, shape and density of the lesion can help us determining whether a lesion is benign or malign. Circumscribed margin and oval shape of the lesion support it is benign, circumscribed margin, round shape, and low density support it is benign.

3)

(Margin=1, BI-RADS=4)->(Severity=0) the support is 0.299688, the confidence is 0.911392

(Margin=1, BI-RADS=4, Density=3) ->(Severity=0) the support is 0.238293, the confidence is 0.905138

High BI-RADS leads to benign result instead, with high support and confidence.

4)

(Age=35) -> (Severity=0) the support is 0.012486992715920915, the confidence is 0.923076923076923

We should ignore it as its support value is very small, so we don’t have enough data to support this rule.

5)

('Age>=50','BI-RADS=5','Density=3')->('Severity=1') the support is 0.24476987447698745, the confidence is 0.9140625000000001.

('Age>=40', 'Shape=4', 'BI-RADS=5') ->('Severity=1') the support is 0.23744769874476987, the confidence is 0.908