# Mining Big Data assignment 3

Made by

Yu Zhang a1795409

## Exercise 1

1. When support threshold is 5, the frequent items are the number from 1 to 20, which is {1, 2, 3, 4, …, 20}. These numbers appear at least 5 times in the backets set.
2. The confidence of {5,7} ->2 = support({5,7}&{2})/support({5,7}) = 1/2.
   Which basket 35 [1,5,7,35] and basket 70 [1,2,5,7,10,14,35,70] include 5 and 7 and 2 only be included in basket 70
   The confidence of {2,3,4} ->5 = support({2,3,4}&{5})/support({2,3,4}) = 1/8.
   Which 2,3,4 is included in basket 12,24,36,48,60,72,84,96, and 5 only be included in basket 60.
   The result generated by python:

```
[1, 2, 3, 4, 6, 12] include 2 and 3 and 4
[1, 2, 3, 4, 6, 8, 12, 24] include 2 and 3 and 4
[1, 5, 7, 35] include 5 and 7
[1, 2, 3, 4, 6, 9, 12, 18, 36] include 2 and 3 and 4
[1, 2, 3, 4, 6, 8, 12, 16, 24, 48] include 2 and 3 and 4
[1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60] include 2 and 3 and 4
[1, 2, 5, 7, 10, 14, 35, 70] include 5 and 7
[1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72] include 2 and 3 and 4
[1, 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 84] include 2 and 3 and 4
[1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 96] include 2 and 3 and 4
1 : 100 2 : 50 3 : 33 4 : 25 5 : 20 6 : 16 7 : 14 8 : 12 9 : 11 10 : 10 11
 : 9 12 : 8 13 : 7 14 : 7 15 : 6 16 : 6 17 : 5 18 : 5 19 : 5 20 : 5 21 : 4
 22 : 4 23 : 4 24 : 4 25 : 4 26 : 3 27 : 3 28 : 3 29 : 3 30 : 3 31 : 3 32
 : 3 33 : 3 34 : 2 35 : 2 36 : 2 37 : 2 38 : 2 39 : 2 40 : 2 41 : 2 42 : 2
 43 : 2 44 : 2 45 : 2 46 : 2 47 : 2 48 : 2 49 : 2 50 : 2 51 : 1 52 : 1 53
 : 1 54 : 1 55 : 1 56 : 1 57 : 1 58 : 1 59 : 1 60 : 1 61 : 1 62 : 1 63 : 1
 64 : 1 65 : 1 66 : 1 67 : 1 68 : 1 69 : 1 70 : 1 71 : 1 72 : 1 73 : 1 74
 : 1 75 : 1 76 : 1 77 : 1 78 : 1 79 : 1 80 : 1 81 : 1 82 : 1 83 : 1 84 : 1
 85 : 1 86 : 1 87 : 1 88 : 1 89 : 1 90 : 1 91 : 1 92 : 1 93 : 1 94 : 1 95
 : 1 96 : 1 97 : 1 98 : 1 99 : 1 100 : 1
```

   Code in appendix: Assignment_3_Ex1

## Exercise 2

The top 10-page ranks are:

597621,0.0006157351781325349

41909,0.0006140081588843619

163075,0.0006025873974702713

537039,0.0005991415885020174

384666,0.0005245270644172195

504140,0.0005100101621964171

486980,0.0004832264701234674 6

605856,0.0004785703285939145 5

32163,0.0004749846637488116

558791,0.0004727257416922059 3

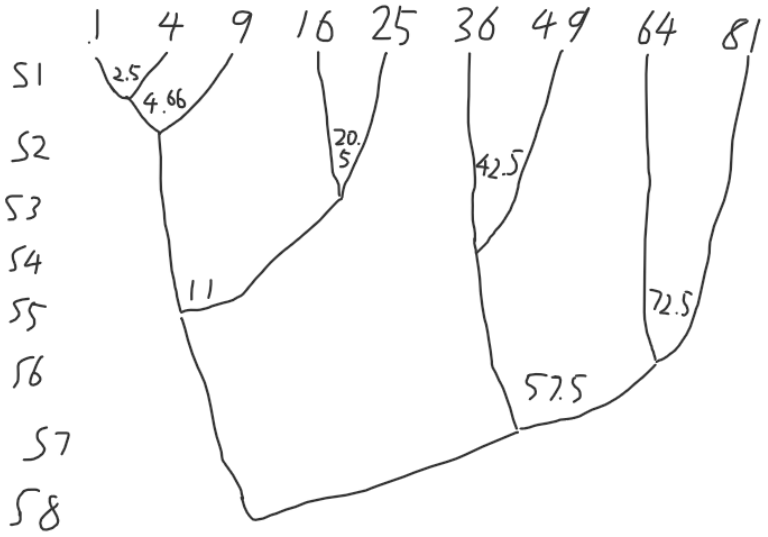
Method:

1. The code use pandas to read the file, generate the initial dataframe with from and to colums.
2. Use pandas function to calculate the number of outbound links of each page, set each rank as 1, calculate each page's contribution to its destinations.
3. Build the matrix to show the relation between pages' link and contribution flow. Since the data is large and there are many empty slots. Use CSR sparse matrix, which the format is (data(contribution), (col(from),row(to))).
4. Build an vector to initial the rank as the 1/number of pages
5. Put them into the formula, iterate multiple time then out put the result

Code in appendix: Assignment_3_Ex2

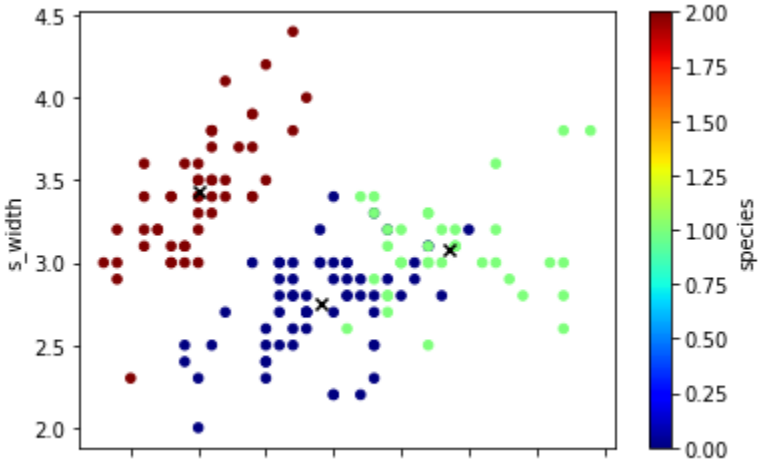
## Exercise 3

1. Final cluster is [[[[[[1,4],9], [16,25]],[[36,49]],[64,81]]]]




Each step compare each element's distance. Combine the least distance pair as the cluster. And merge the pair as one and use the mean of elements as its value. Loop until reach the root cluster

2. (1)




The point has been clustered by color, the centroid is the “x” on the plot.

(2) By Elbow method. Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k.

K = 1, WSS = 291.610254

| species | s_length | s_width | p_length | p_width | 0 |
|---|---|---|---|---|---|
| 0 | 876.5 | 458.6 | 563.7 | 179.9 | 291.610254 |

K = 2, WSS = 128.33665

| species | s_length | s_width | p_length | p_width | 0 | 1 |
|---|---|---|---|---|---|---|
| 0 | 611.2 | 280.0 | 481.0 | 164.5 | 97.335994 | 385.861967 |
| 1 | 265.3 | 178.6 | 82.7 | 15.4 | 210.494699 | 31.000671 |

K = 3, WSS = 94.224869

| species | s_length | s_width | p_length | p_width | 0 | 1 | 2 |
|---|---|---|---|---|---|---|---|
| 0 | 267.3 | 120.0 | 222.9 | 80.1 | 28.541989 | 195.734249 | 72.752166 |
| 1 | 250.3 | 171.4 | 73.1 | 12.3 | 250.745805 | 24.085262 | 169.432605 |
| 2 | 358.9 | 167.2 | 267.7 | 87.5 | 113.197966 | 207.144267 | 44.597618 |

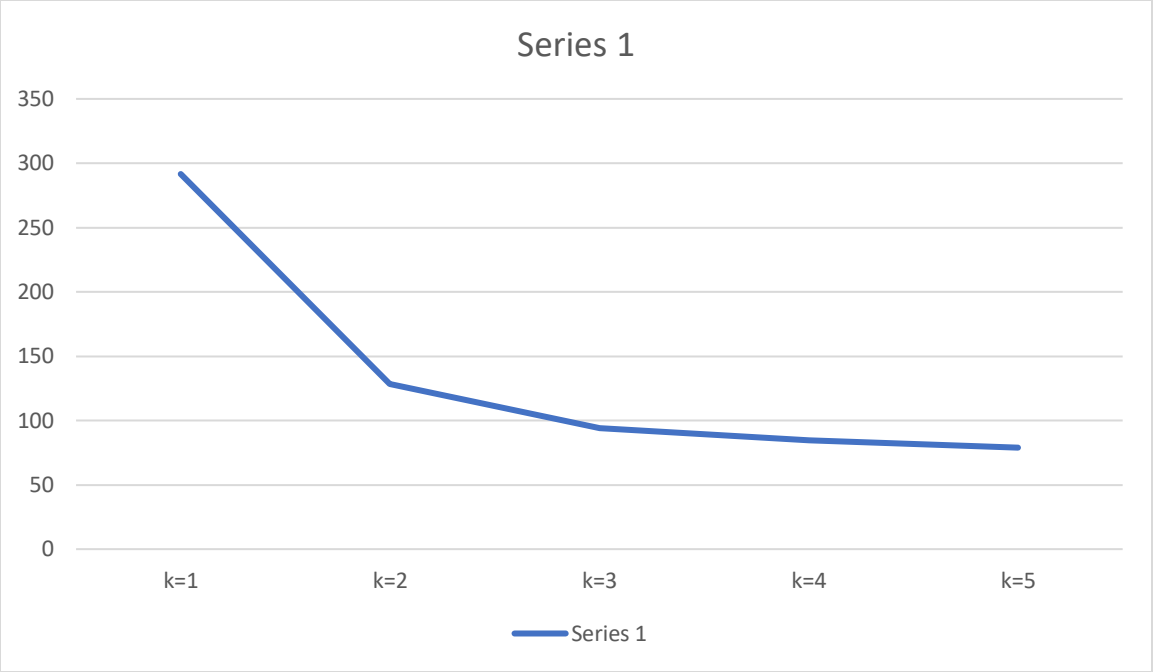
K = 4, WSS = 84.556055

```
df.groupby(['species']).sum()
```

| length | s_width | p_length | p_width | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|
| 167.4 | 79.0 | 119.6 | 37.0 | 16.390776 | 87.700241 | 41.668476 | 82.739745 |
| 250.3 | 171.4 | 73.1 | 12.3 | 146.417645 | 24.085262 | 203.536988 | 269.181526 |
| 295.8 | 136.3 | 232.7 | 81.3 | 66.867895 | 191.247023 | 28.221078 | 70.420562 |
| 163.0 | 71.9 | 138.3 | 49.3 | 63.890335 | 123.931252 | 34.503803 | 15.858939 |

K =5, WSS = 78.897402

| idth | p_length | p_width | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|
| 06.8 | 47.3 | 7.6 | 12.171261 | 27.643310 | 96.758152 | 180.477727 | 136.198241 |
| 64.6 | 25.8 | 4.7 | 14.393338 | 6.255348 | 49.659493 | 88.703799 | 67.338747 |
| 79.0 | 119.6 | 37.0 | 88.427949 | 88.409214 | 16.390776 | 82.739745 | 41.668476 |
| 71.9 | 138.3 | 49.3 | 126.156912 | 120.417684 | 63.890335 | 15.858939 | 34.503803 |
| 36.3 | 232.7 | 81.3 | 194.694584 | 186.856590 | 66.867895 | 70.420562 | 28.221078 |




As the chart shows, the change rate of WSS dimmish rapidly when k = 3 to k = 4. So it should pick k = 3 for the k-means.

Method:

1. Input the Iris data features as pandas dataframe, draw the plot with firs two dimensions.
2. Set each feature's maximal and minimal numbers, build a random coordinate generator function, use to set the random point in reasonable range.
3. Set k, generate k number of random point on the plot as centroid, save the random point's coordinate as a list. Add a species column to the dataframe, initial as -1.
4. Loop start. Calculate the distance to every centroid, generate the column of distance of every centroid. Select the minimal one as the species name.
5. Calculate the mean coordinate by group of species, renew the centroid coordinate by each mean. If one species does not have any plot belongs to (means this point is ), handle an exception, generate another set of random point. Repeat the loop, until the species column doesn't not change compared to previous iteration.
6. Loop end, the final centroid coordinate is the k-means coordinate, draw the plot, set data point to each species' color. Calculate the WSS under the K value.

Code in appendix: Assignment_3_Ex3-4D