CS6220 Data Mining Techniques

Assignment 3

Yaxin Huang

HW2 Question 1

(I) The complete decision tree trained by using the training data is the one below. The bold nodes are the nodes selected to form the decision tree; The blued information gains are the larger information gains compared to the ones got from split using other features:

Top [6L 4H], Entropy = 0.9710
If split by Education, IG = 0.1246
(1) High School [4L 1H], Entropy = 0.7219
If split by Career, IG = 0.2628
(1.1) Management [2L 1H], Entropy = 0.9183
(1.2) Service [2L 0H], Entropy = 0
If split by Experience, IG = 0.3219
(1.3) Less than 3 [1L 0H], Entropy = 0
(1.4) 3 to 10 [2L 0H], Entropy = 0
(1.5) More than 10 [1L 1H], Entropy = 1
If split by Career, IG = 1
(1.5.1) Management [0L 1H], Entropy = 0
(1.5.2) Service [1L 0H], Entropy = 0
(2) College 2L 3H, Entropy = 0.9710
If split by Career, IG = 0.4200
(2.1) Management [0L 2H], Entropy = 0
(2.2) Service [2L 1H], Entropy = 0.9183
If split by Experience, IG = 0.9183
(2.2.1) Less than 3 [1L 0H], Entropy = 0
(2.2.2) 3 to 10 [0L 1H], Entropy = 0
(2.2.3) More than 10 [1L 0H], Entropy = 0
If split by Experience, IG = 0.1710
(2.3) Less than 3 [1L 1H], Entropy = 1
(2.4) 3 to 10 [0L 1H], Entropy = 0
(2.5) More than 10 [1L 1H], Entropy = 1

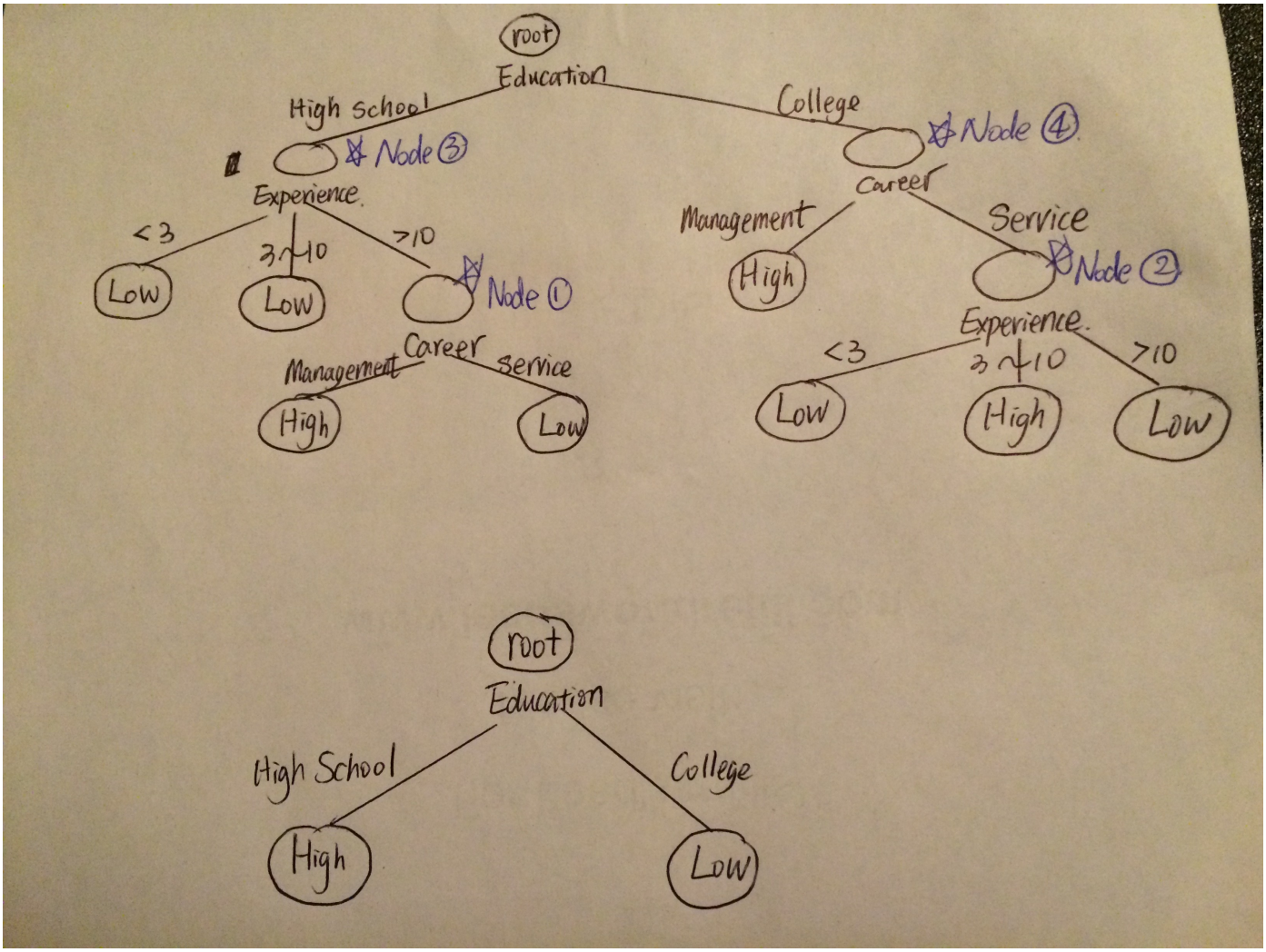
If split by Career, IG = 0.1246
(3) Management [2L 3H], Entropy = 0.9710
(4) Service [4L 1H], Entropy = 0.7219

If split by Experience, IG = 0.0200
(5) Less than 3 [2L 1H], Entropy = 0.9183
(6) 3 to 10 [2L 1H], Entropy = 0.9183
(7) More than 10 [2L 2H], Entropy = 1

(II) Post-pruning:
We only use the decision tree structure we got from the last part. And here we have three validation items:

Instance	Education	Career	Experience	Salary
1	High School	Management	>10	High
2	College	Management	< 3	Low
3	College	Service	3~10	Low

The tree before pruning and after pruning is shown in the picture below:



And here are the steps showing how to get the final tree:
Here the prunable nodes are denoted as node 1, 2, 3, 4 as the blue annotations in the picture says, and the “Improve” is calculated by subtracting keep errors from prune errors

Step 1, consider the node 1 & node 2:

Node	Keep Errors	Prune Errors	Label after Pruning	Improve
1	0	0	High	0
2	1	0	Low	1

So now prune the branches under node 2 and get the node 2 replaced by leaf node with “Low” label.

Step 2, consider the node 1 and node 4:

Node	Keep Errors	Prune Errors	Label after Pruning	Improve
1	0	0	High	0
4	1	0	Low	1

So now prune the branches under node 4 and get the node 4 replaced by a leaf node with label “low”.

Step 3, consider the node 1:

Node	Keep Errors	Prune Errors	Label after Pruning	Improve
1	0	0	High	0

Although pruning does not make improvement but it won't incur any errors regarding the validation set. For the sake of simplicity, we prune the node 1 here and result in node 1 replaced by a leaf node labeled “High”.

Step 4, consider the node 3:

Node	Keep Errors	Prune Errors	Label after Pruning	Improve
3	0	0	High	0

For the same reason as the last step, we prune the node 3 and it results in node 3 replaced by a leaf node labeled “High”.

Above are all the steps needed and the final tree is displayed in the picture as well.

Question 1

(I) Build the Naive Bayes classifier for the given training data:
Possible Education (i.e. Education Level): {High School, College, Graduate}
Possible Career: {Management, Service, Retail}
Possible Experience (i.e. Years of Experience): {Less than 3, 3 to 10, More than 10}
Possible Salary: {Low, High}

We need to apply Laplace Smoothing on Education and Career.
P(Salary = Low) = 6/10 = 3/5
P(Salary = High) = 4/10 = 2/5

P(Education=High School | Salary=Low) = (4+1) / (6+3) = 5/9
P(Education=College | Salary=Low) = (2+1) / (6+3) = 3/9 = 1/3
P(Education=Graduate | Salary=Low) = 1 / (6+3) = 1/9
P(Career=Management | Salary=Low) = (2+1) / (6+3) = 3/9 = 1/3
P(Career=Service | Salary=Low) = (4+1) / (6+3) = 5/9
P(Career=Retail | Salary=Low) = 1 / (6+3) = 1/9
P(Experience=Less than 3 | Salary=Low) = 2/6 = 1/3
P(Experience=3 to 10 | Salary=Low) = 2/6 = 1/3
P(Experience=More than 10 | Salary=Low) = 2/6 = 1/3

P(Education=High School | Salary=High) = (1+1) / (4+3) = 2/7
P(Education=College | Salary=High) = (3+1) / (4+3) = 4/7
P(Education=Graduate | Salary=High) = 1 / (4+3) = 1/7
P(Career=Management | Salary=High) = (3+1) / (4+3) = 4/7
P(Career=Service | Salary=High) = (1+1) / (4+3) = 2/7
P(Career=Retail | Salary=High) = 1 / (4+3) = 1/7
P(Experience=Less than 3 | Salary=High) = 1/4
P(Experience=3 to 10 | Salary=High) = 1/4
P(Experience=More than 10 | Salary=High) = 2/4 = ½

(II) Use the model to classify the new instances:

Instance	Education Level	Career	Year of Experience
1	High School	Service	Less than 3
2	College	Retail	Less than 3
3	Graduate	Service	3 to 10

Instance 1:
P(Salary=Low | Instance 1) * P(Instance 1)
= P(Education=High School, Career=Service, Experience=Less than 3 | Salary=Low) * P(Salary=Low)
= P(Education=High School | Salary=Low)
* P(Career=Service | Salary=Low) * P(Experience=Less than 3 | Salary=Low) * P(Salary=Low)
= (5/9) * (5/9) * (1/3) * (3/5)
= 5/81

P(Salary=High | Instance 1) * P(Instance 1)
=P(Education=High School, Career=Service, Experience=Less than 3 | Salary=High) * P(Salary=High)

= P(Education=High School | Salary=High) * P(Career=Service | Salary=High)
* P(Experience=Less than 3 | Salary=High) * P(Salary=High)
= (2/7) * (2/7) * (¼) * (2/5)
= 2/245

Since P(Salary=Low | Instance 1) * P(Instance 1) = 5/81
> P(Salary=High | Instance 1) * P(Instance 1) = 2/245,
therefore P(Salary=Low | Instance 1) > P(Salary=High | Instance 1) and instance 1 should be labeled salary = Low.

Instance 2:
P(Salary=Low | Instance 2) * P(Instance 2)
= P(Education=College, Career=Retail, Experience=Less than 3 | Salary=Low) * P(Salary=Low)
= P(Education=College | Salary=Low) * P(Career=Retail | Salary=Low)
* P(Experience=Less than 3 | Salary=Low) * P(Salary=Low)
= (1/3) * (1/9) * (1/3) * (3/5)
= 1/135

P(Salary=High | Instance 2) * P(Instance 2)
= P(Education=College, Career=Retail, Experience=Less than 3 | Salary=High) * P(Salary=High)
= P(Education=College | Salary=High) * P(Career=Retail | Salary=High)
* P(Experience=Less than 3 | Salary=High) * P(Salary=High)
= (4/7) * (1/7) * (¼) * (2/5)
= 2/245

Since P(Salary=Low | Instance 2) * P(Instance 2) = 1/135
< P(Salary=High | Instance 2) * P(Instance 2) = 2/245,
therefore P(Salary=Low | Instance 2) < P(Salary=High | Instance 2), and instance 2 should be labeled as salary = High.

Instance 3:
P(Salary=Low | Instance 3) * P(Instance 3)
= P(Education=Graduate, Career=Service, Experience=3 to 10 | Salary=Low) * P(Salary=Low)
= P(Education=Graduate | Salary=Low) * P(Career=Service | Salary=Low)
* P(Experience=3 to 10 | Salary=Low) * P(Salary=Low)
= (1/9) * (5/9) * (1/3) * (3/5)
= 1/81

P(Salary= High | Instance 3) * P(Instance 3)
= P(Education=Graduate, Career=Service, Experience=3 to 10 | Salary=High) * P(Salary=High)
= P(Education=Graduate | Salary=High) * P(Career=Service | Salary=High)
* P(Experience=3 to 10 | Salary=High) * P(Salary=High)
= (1/7) * (2/7) * (¼) * (2/5)
= 1/245
Since P(Salary=Low | Instance 3) * P(Instance 3) = 1/81
> P(Salary= High | Instance 3) * P(Instance 3) = 1/245,
therefore P(Salary=Low | Instance 3) > P(Salary= High | Instance 3), and instance 3 should be labeled as salary = Low.

The final result is:

Instance	Education Level	Career	Year of Experience	Salary
1	High School	Service	Less than 3	Low
2	College	Retail	Less than 3	High
3	Graduate	Service	3 to 10	Low

Question 2

Two clusters: C_1 = {(1, 1), (2, 2), (3, 3)} , C_2 = {(5, 2), (6, 2), (7, 2), (8, 2), (9, 2)}

(a) The mean vectors m_1 and m_2:
m_1 = 1/3 [(1,1) + (2,2) + (3,3)] = (2, 2)
m_2 = 1/5[(5,2) + (6,2) + (7,2) + (8,2) + (9,2)] = (7, 2)

(b) The total mean vector m:
m = 1/8 [(2,2) * 3 + (7,2) * 5] = (41/8, 8/5) = (5.125, 1.6)

(c) The scatter matrices S_1 and S_2:

$$S_1=\sum_{x\in C_1}(x-m_1)\cdot(x-m_1)^T=\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}$$

$$S_2=\sum_{x\in C_2}(x-m_2)\cdot(x-m_2)^T=\begin{bmatrix}10 & 0\\ 0 & 0\end{bmatrix}$$

(d) The within-cluster scatter matrix S_W:

$$S_W=\sum_{i=1}^{K}S_i=\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}+\begin{bmatrix}10 & 0\\ 0 & 0\end{bmatrix}=\begin{bmatrix}12 & 2\\ 2 & 2\end{bmatrix}$$

(e) The between-cluster scatter matrix S_B:

$$S_B=\sum_{i=1}^{2}N_i(m_i-m)(m_i-m)^T=\begin{bmatrix}46.875 & 7.5\\ 7.5 & 1.28\end{bmatrix}$$

(f) The scatter criterion $tr(S_B)$ /$tr(S_W)$:

$$\frac{tr(S_B)}{tr(S_W)}=\frac{46.875+1.28}{12+2}\approx 3.44$$

Question 3

In this question I used Rstudio to plot graphs.
We have 2-dimensional data listed as below:

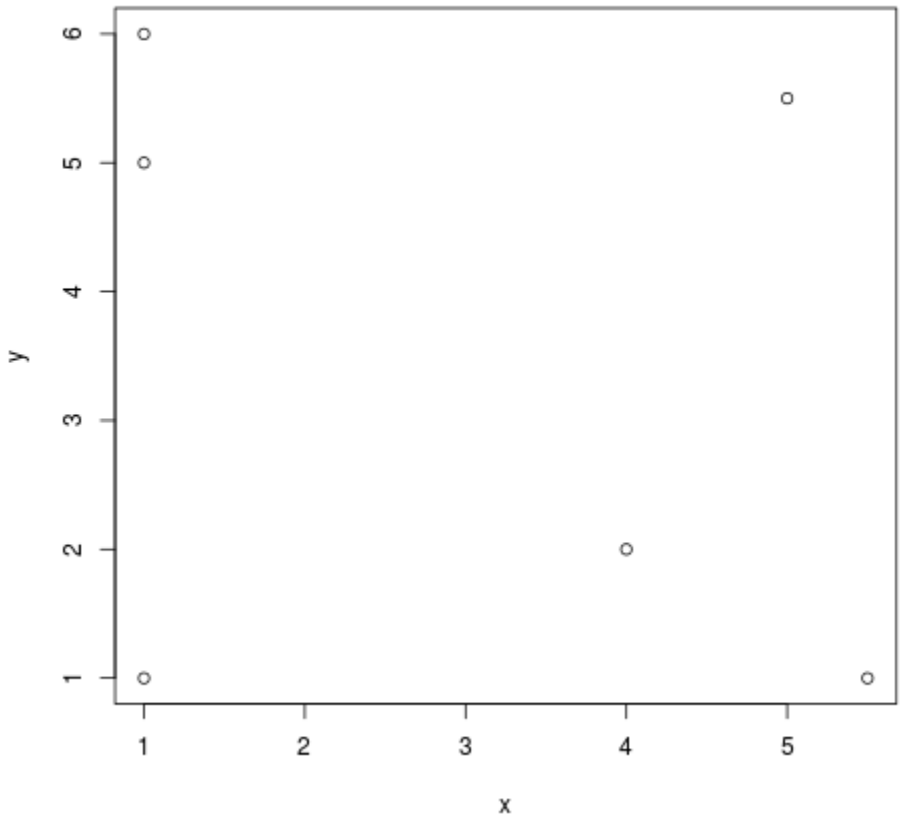
Index	Data Vector
1	(1, 6)
2	(1, 5)
3	(1, 1)
4	(5, 5.5)
5	(4, 2)
6	(5.5, 1)

First we create a data frame representing the dataset:

```
> data <- data.frame(x=c(1,1,1,5,4,5),y=c(6,5,1,5.5,2,1))
```

And then we plot the data:

```
> plot(data)
```



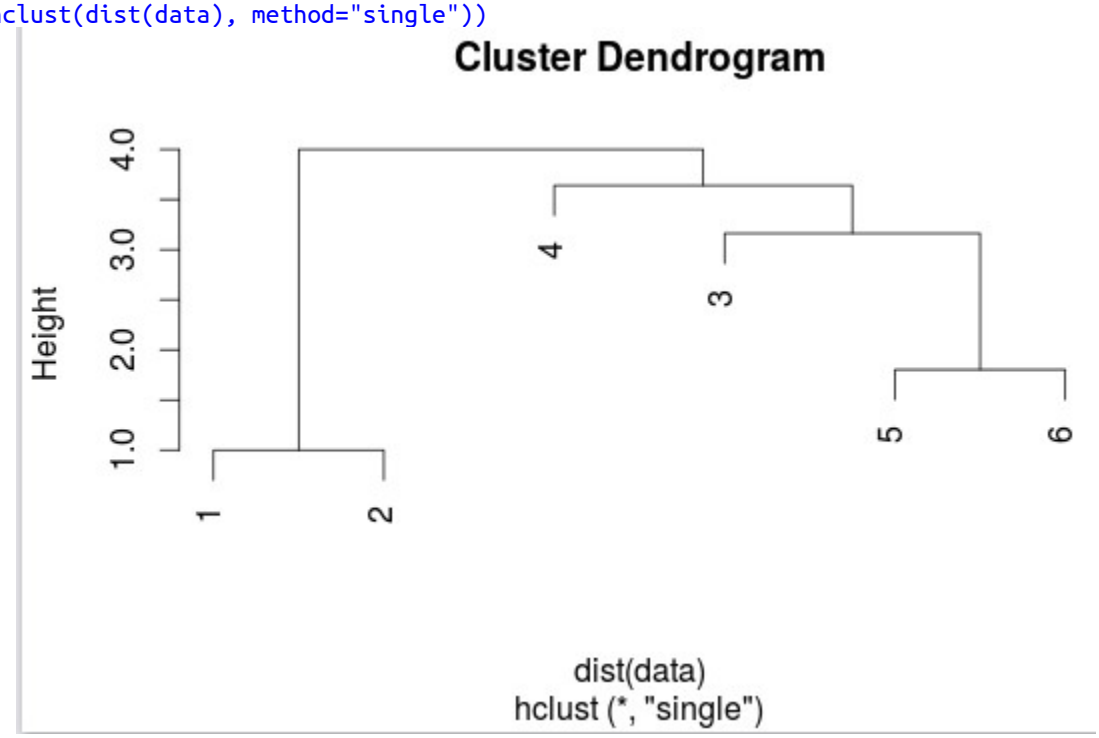
The distance matrix between the datapoints:

```
> dist(data)
          1        2        3        4        5
2 1.000000
3 5.000000 4.000000
4 4.031129 4.031129 6.020797
5 5.000000 4.242641 3.162278 3.640055
6 6.726812 6.020797 4.500000 4.527693 1.802776
```

And then we use different methods to compute the distance between clusters, where “Height” can help identify the distance between two clusters.

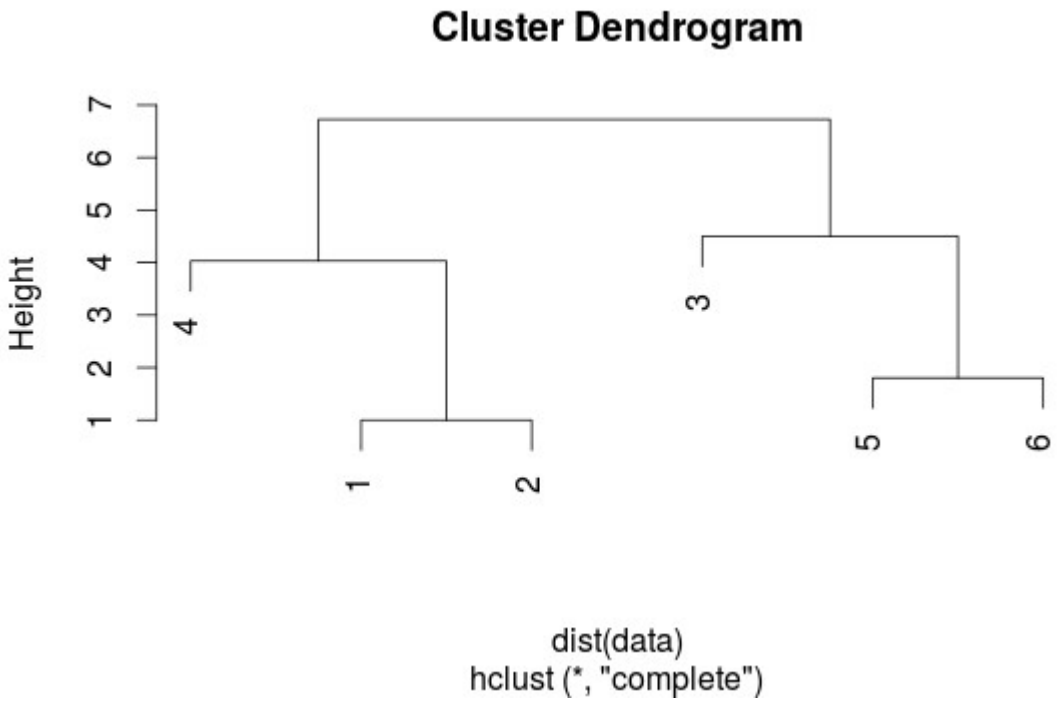
(1) Single linkage (minimum distance):

```
> plot(hclust(dist(data), method="single"))
```



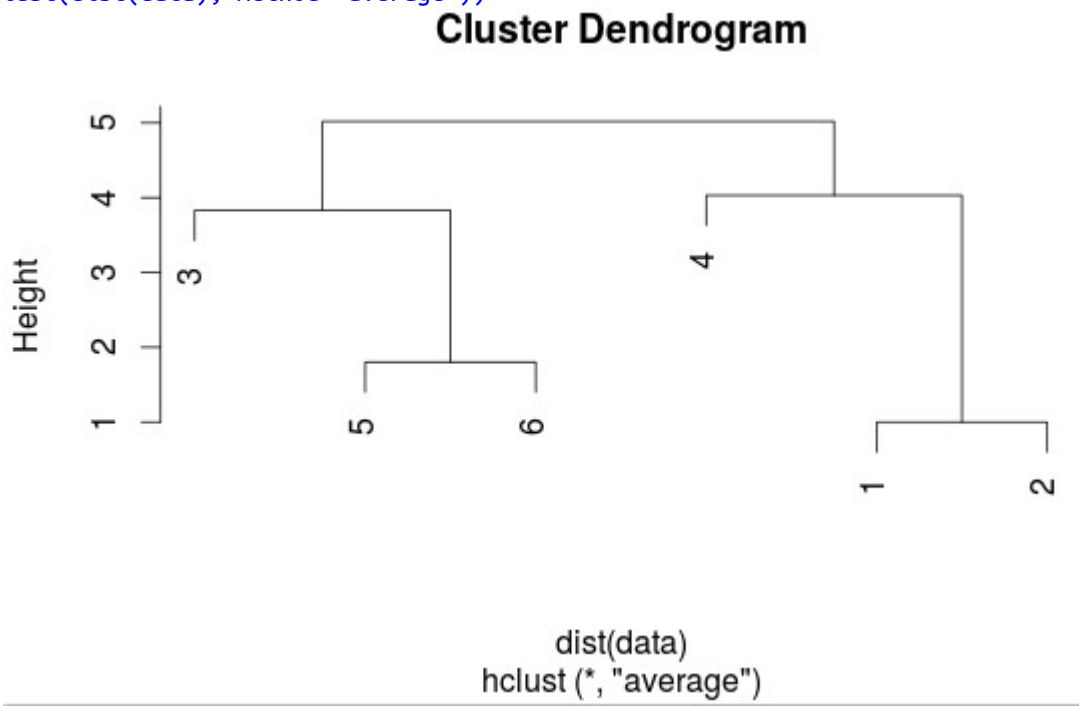
(2) Complete linkage (maximum distance):

```
> plot(hclust(dist(data), method="complete"))
```



(3) Average linkage (average distance):

```
> plot(hclust(dist(data), method="average"))
```



As the three dendrograms show, using different methods to compute the distance between clusters can result in different outcomes.

Question 4

Parameters:

$$\epsilon = \sqrt{2}, MinPts = 3$$

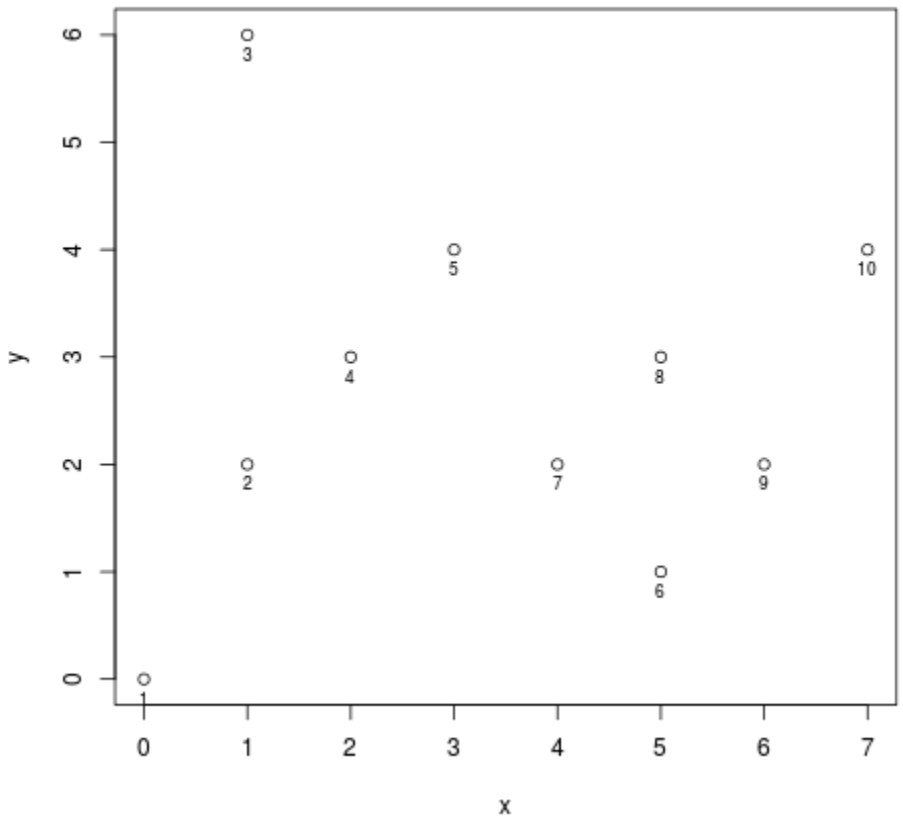
Given the following points:
(0, 0), (1, 2), (1, 6), (2, 3), (3, 4), (5, 1), (4, 2), (5, 3), (6, 2), (7, 4).

(a) List the clusters in terms of their points:
The distance matrix:

```
> dist(data)
          1        2        3        4        5        6        7        8        9
2  2.236068
3  6.082763 4.000000
4  3.605551 1.414214 3.162278
5  5.000000 2.828427 2.828427 1.414214
6  5.099020 4.123106 6.403124 3.605551 3.605551
7  4.472136 3.000000 5.000000 2.236068 2.236068 1.414214
8  5.830952 4.123106 5.000000 3.000000 2.236068 2.000000 1.414214
9  6.324555 5.000000 6.403124 4.123106 3.605551 1.414214 2.000000 1.414214
10 8.062258 6.324555 6.324555 5.099020 4.000000 3.605551 3.605551 2.236068 2.236068
```

The scatter plot of the datapoints:

```
> plot(data)
> text(data, labels=row.names(data), cex=0.7, pos=1)
```



After calculation, the clusters are listed below:
C_1 = {(5, 1), (4, 2), (5, 3), (6, 2)}
C_2 = {(1, 2), (2, 3), (3, 4)}
C_3 = {(0,0)} // Outlier
C_4 = {(1,6)} // Outlier
C_5 = {(7,4)} // Outlier

(b) What are the density-connected points?

1	2	3	4	5	6	7	8	9	10
(0, 0)	(1, 2)	(1, 6)	(2, 3)	(3, 4)	(5, 1)	(4, 2)	(5, 3)	(6, 2)	(7, 4)

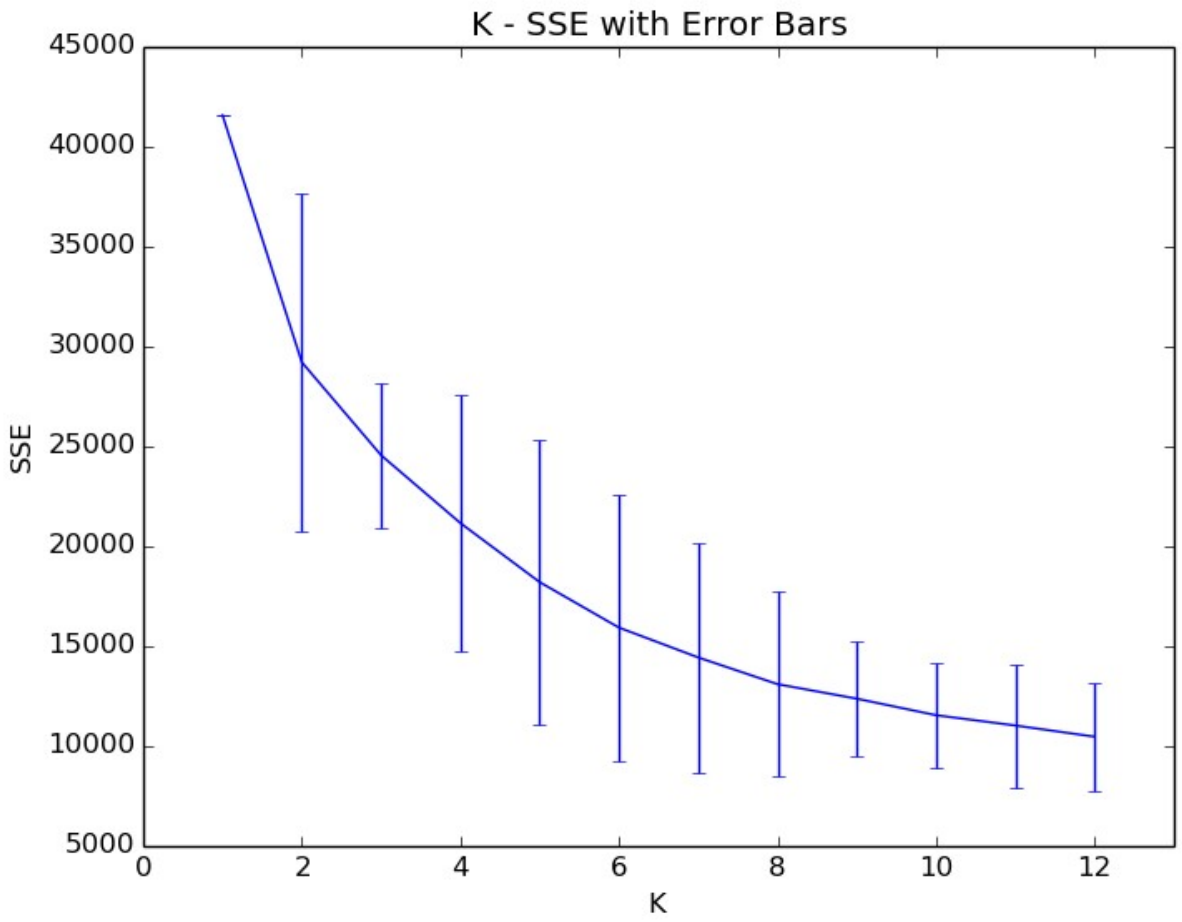
Point (5,1) is density-connected with point (5,3) through (4,2).
Point (4,2) is density-connected with point (6,2) through (5,3).
Point (1,2) is density-connected with point (3,4) through (2,3).

Point (1,2) is density-connected with point (2,3).
Point (2,3) is density-connected with point (3,4).
Point (5,1) is density-connected with point (4,2).
Point (4,2) is density-connected with point (5,3).
Point (5,3) is density-connected with point (6,2).
Point (6,2) is density-connected with point (5,1).

(c) What points (if any) does DBSCAN consider as noise?
Here DBSCAN consider (0,0), (1,6), (7,4) as noise, since they are not main points and cannot be density-reached by any other points.

Question 5

(1) The line plot showing the mean SSE as a function of k:



(2) The table got by running the run.py:

K	mean	mean - 2sd	mean + 2sd
1	41580.0	41580.0	41580.0
2	29217.6634841	24990.5649894	33444.7619788
3	24561.6490619	22745.128085	26378.1700389
4	21167.0217376	17968.0329335	24366.0105418
5	18211.2140117	14657.3570833	21765.07094
6	15935.2056614	12608.3831822	19262.0281405
7	14441.3823183	11570.2395989	17312.5250377
8	13103.7675171	10800.2766373	15407.258397
9	12377.3768077	10948.2028171	13806.5507983
10	11550.5133669	10244.1804033	12856.8463305
11	11035.4297127	9492.93422561	12577.9251997
12	10475.2585195	9124.22512762	11826.2919113

(3) As k increases and approaches the total number of examples N , what value does the SSE approach? What problems does this cause in terms of using SSE to choose an optimal k?

As k increases, the SSE tends to drop gradually, because as the size of each cluster decreases, the clusters become more compact, compared to the one with a larger size (with smaller k). If we use SSE to choose an optimal k, the k we use will be the number of instances, since it gives to lowest SSE, which is zero.

(4) Can you suggest another measure of cluster compactness and separation that might be more useful than SSE?

- Within cluster scatter matrix :

$$S_W = \sum_{i=1}^{K} S_i$$

- Between cluster scatter matrix :

$$S_B = \sum_{i=1}^{K} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

Above are the formulas found in professor's slides.
We can use the scatter matrix criterion to measure the cluster compactness and separation. If the k-means algorithm outputs the clusters with lower trace(S_W), then the cluster compactness is higher. If the algorithm outputs the clusters with higher trace(S_B), then the cluster separation is higher.

If you want to decide a k to use, run the analysis in the same way here and plot the graph showing SSE to the function of k. Look for the elbow point where the SSE drops significantly and that should be the appropriate k to use.