Description for functions

Functions in *myPreparation.R*:

sapply(bcw, class): check all columns' types in a specified dataset
as.integer(): change original type to integer
as.factor(): change original type to factor
na.omit(): remove all rows with missing values
saveRDS(): save the data frame into a file

Functions in *myClustering.R*:

(kmeans.result <- kmeans(bcw2,nclust)): cluster a specified dataset into n clusters with K-means
hclust(dist(bcwSample), method = myMethod): apply hierarchical clustering to the data with a specified agglomeration method
rect.hclust(): cut the dendrogram into n clusters
jpeg() and dev.off(): save the plot into a file
hcluster(n): (custom function) input the k value and plot the clustering result with the specified k
hcluster2(n, myMethod): (custom function) input the k value and the name of agglomeration method, then plot the clustering result with the specified k and method

Functions in *myClassification.R:*

sample(2, m, replace = TRUE, prob = c(training_percentage, test_percentage)): select randomized samples as index
ctree(myFormula, data = training_data): build the classification tree with default parameters
predict(bcw_ctree, newdata = test_features): predict test labels
as.matrix(table(Actual = test_labels, predicted = ctree_pred)): create the confusion matrix
ctree(myFormula, data = training_data, controls = ctree_control(minbucket = 3L, minsplit = 7L, testtype = "Bonferroni", mincriterion = 0.1)): set lower values of

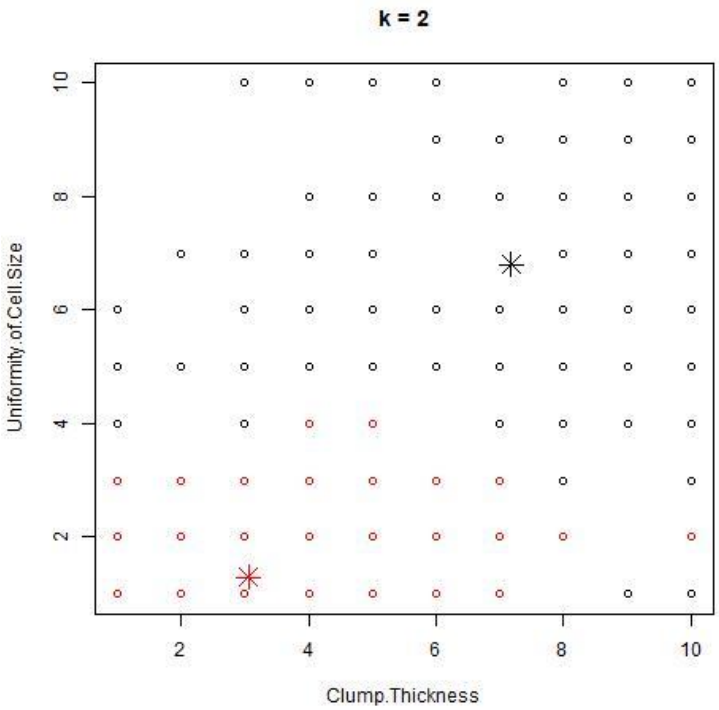
minbucket, minsplit and mincriterion in order to build a classification tree with a greater value of depth

knn(train = training_features, test = test_features, cl = training_labels, k = my_k): apply K-NN classification to predict the labels in the test subset

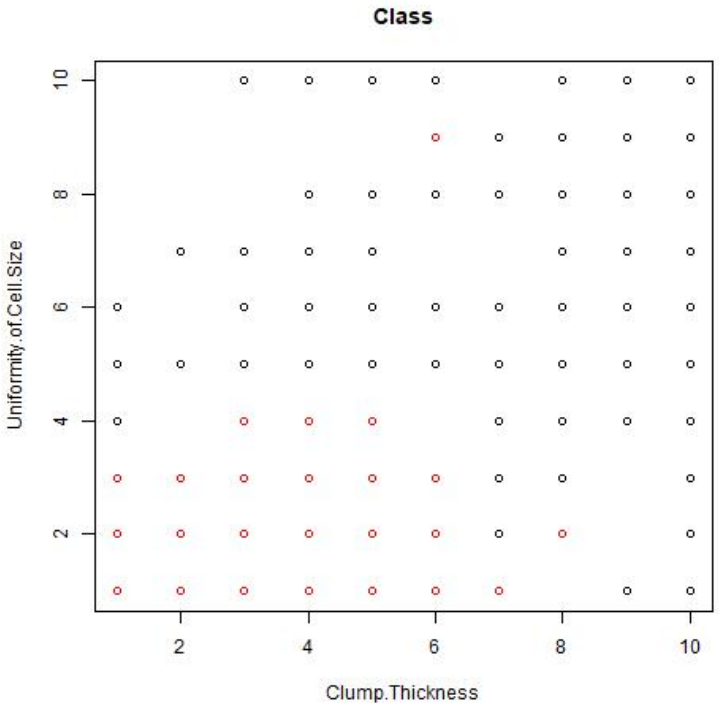
Calculation(my_k): (custom function) input the predicted results and return the values of accuracy, precision, recall and F1

Plots and results

Task 2.2



Task 2.3

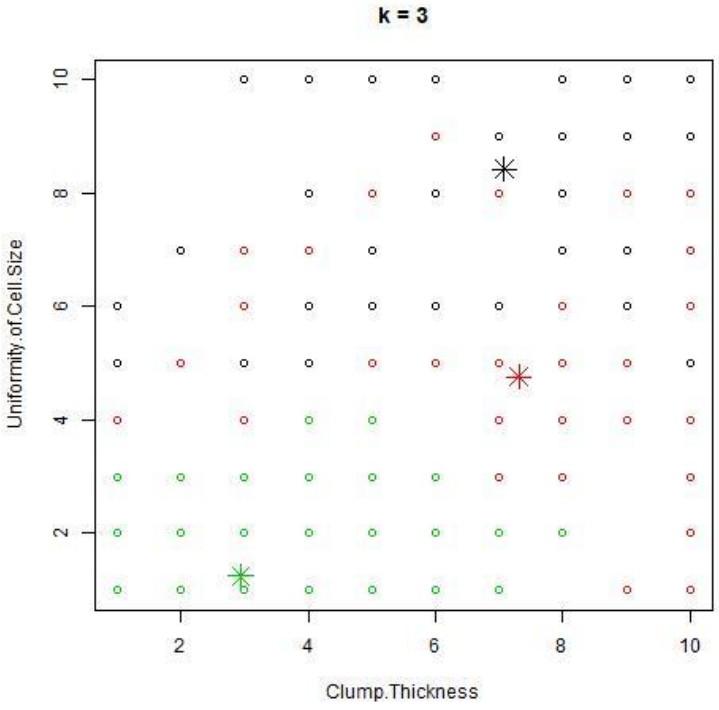


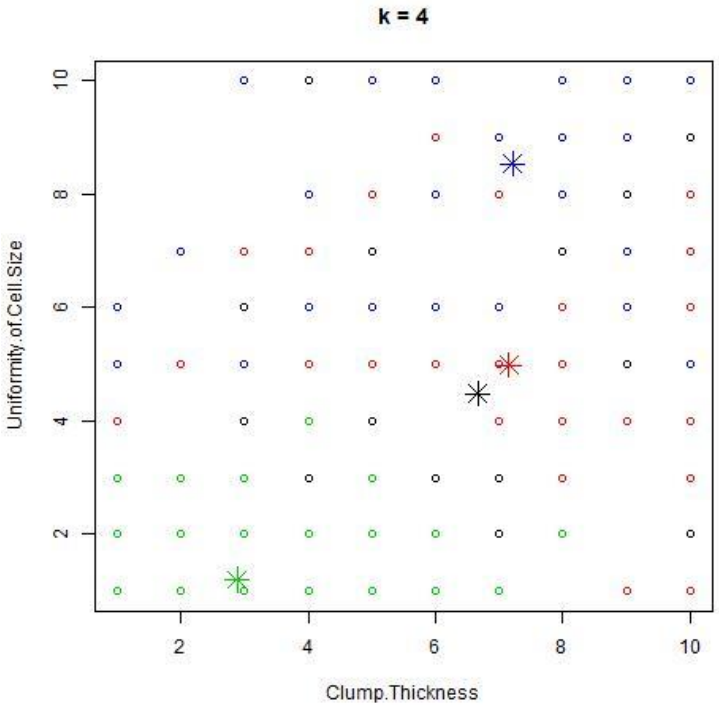
In this task, palette(c("red","black")) was used to set the same color combination as task 2.2.

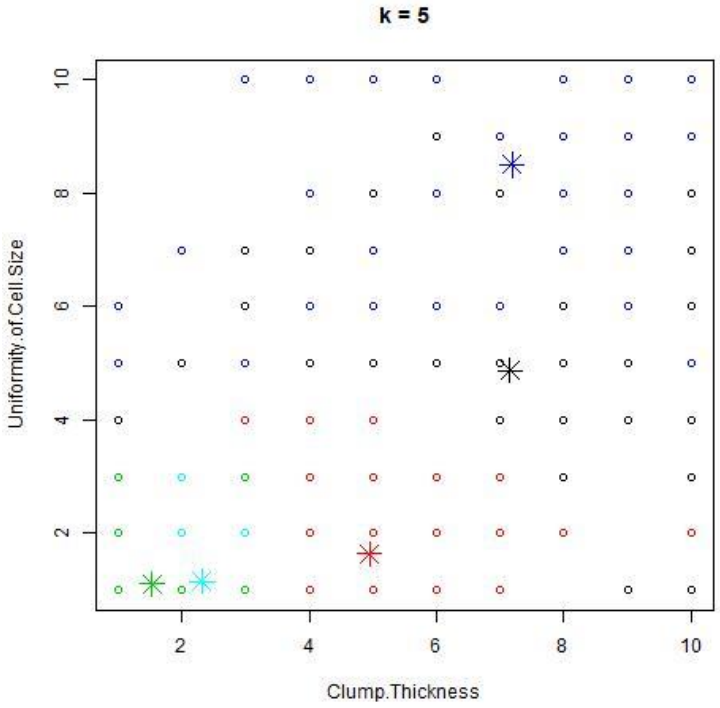
Task 2.4

The clusters is very similar with the results of benign vs malignant classes, except for some points lied in the bottom right corner and some noises.

Task 2.5







Task 2.6

2 clusters:

Within cluster sum of squares by cluster: [1] 14938.609, 4384.565 SSE = 19323.17 (between_SS / total_SS = 60.1 %)

3 clusters:

Within cluster sum of squares by cluster: [1] 7347.724, 5464.890, 3443.303 SSE = 16255.92 (between_SS / total_SS = 66.4 %)

4 clusters:

Within cluster sum of squares by cluster: [1] 2307.980, 3597.161, 2753.301, 6294.649 SSE = 14953.09 (between_SS / total_SS = 69.1 %)

5 clusters:

Within cluster sum of squares by cluster: [1] 4417.8679, 2294.7022, 636.8986,

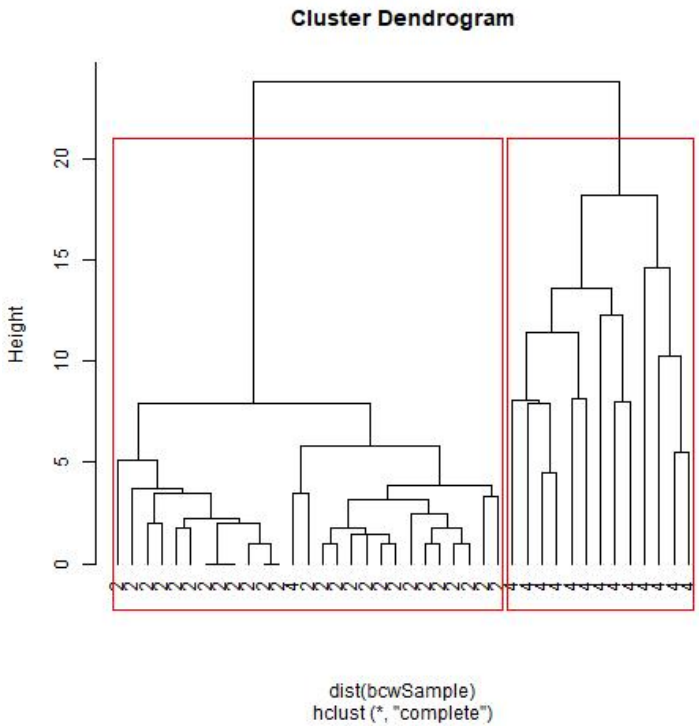
7250.0661, 402.9769 SSE = 15002.51 (between_SS / total_SS = 69.0 %)

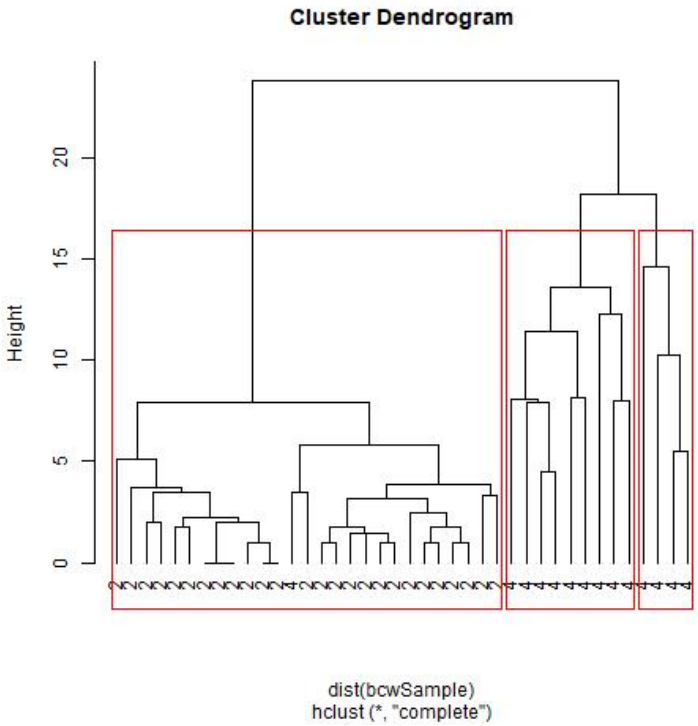
Summary:

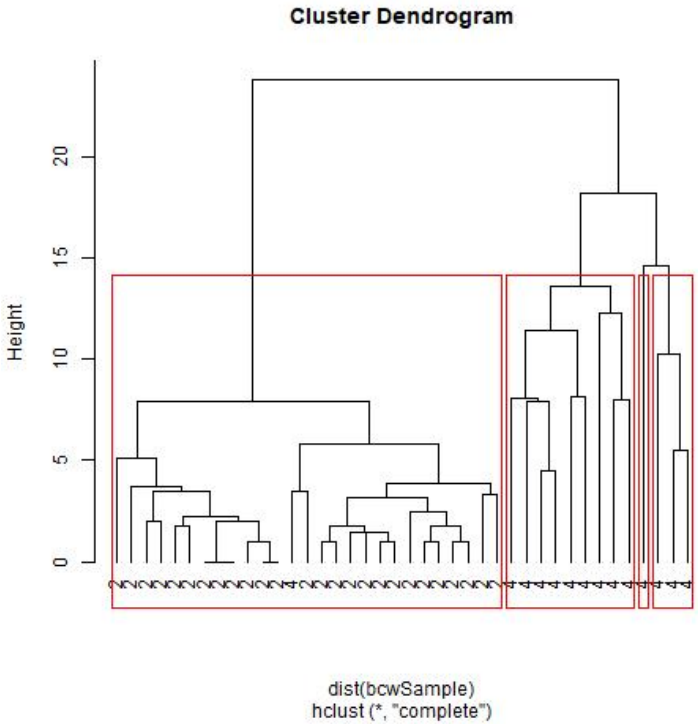
The SSE in the 4-cluster group is the smallest (14953.09) among 4 groups, and most of the total variance (69.1%) is explained by the variance between groups. Therefore, the 4-cluster group is a good fit and the quality of clustering is highest among 4 groups.

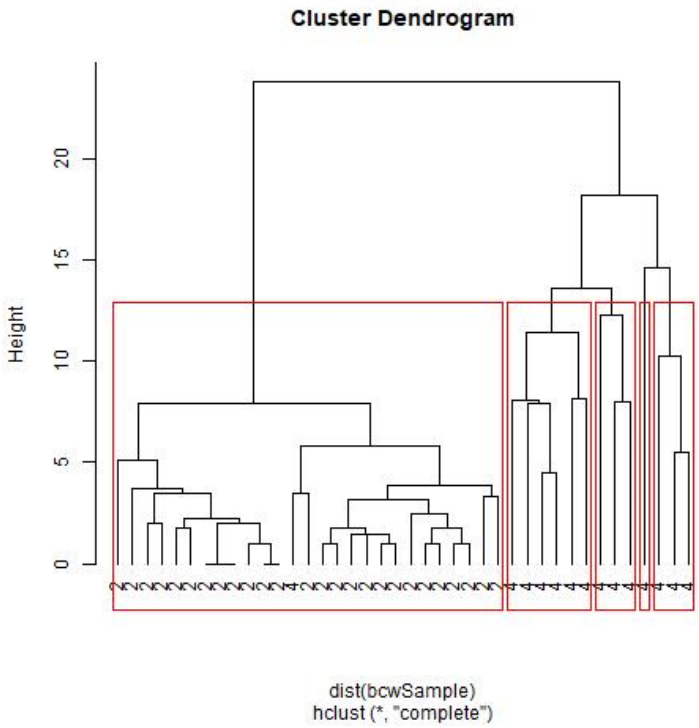
Task 2.7

In this task, a random sample (40) of the dataset was used to generate clear dendrograms in an acceptable time because the implementation of hierarchical clustering is expensive for really big datasets that do not fit in memory.




Cluster Dendrogram
Height
0
5
10
15
20
dist(bcwSample)
hclust (*, "complete")


Cluster Dendrogram
Height
0
5
10
15
20
dist(bcwSample)
hclust (*, "complete")



Task 2.8

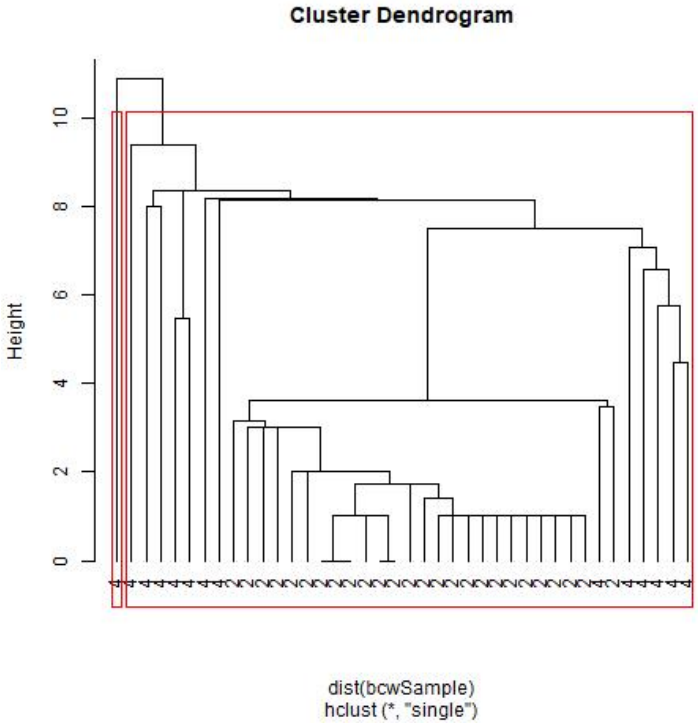
We should have a new subtype of disease because the SSE in the 3-cluster group is lower than that of 2-cluster and the size of each cluster in the 3-cluster group is acceptable. However, we should not have more than one new subtype of disease because the size of the fourth cluster is too small, and it may reflect the noises or outliers.

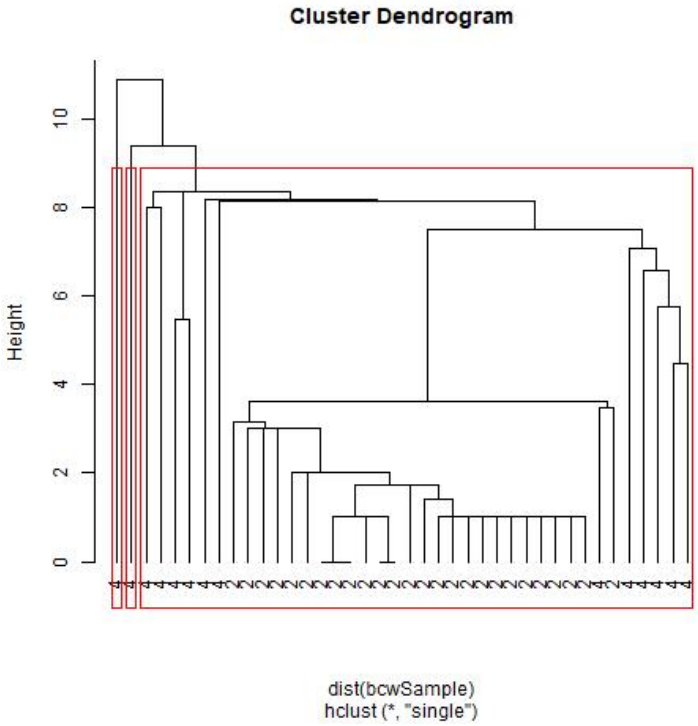
Task 2.9

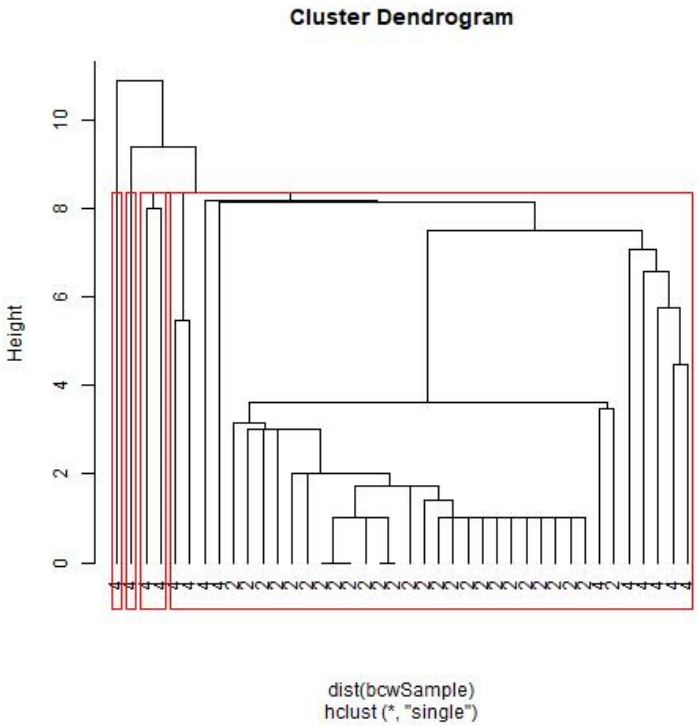
The data are sensitive to the used agglomeration method because when we implement different agglomeration methods we will get different structures of hierarchical clusters and results.

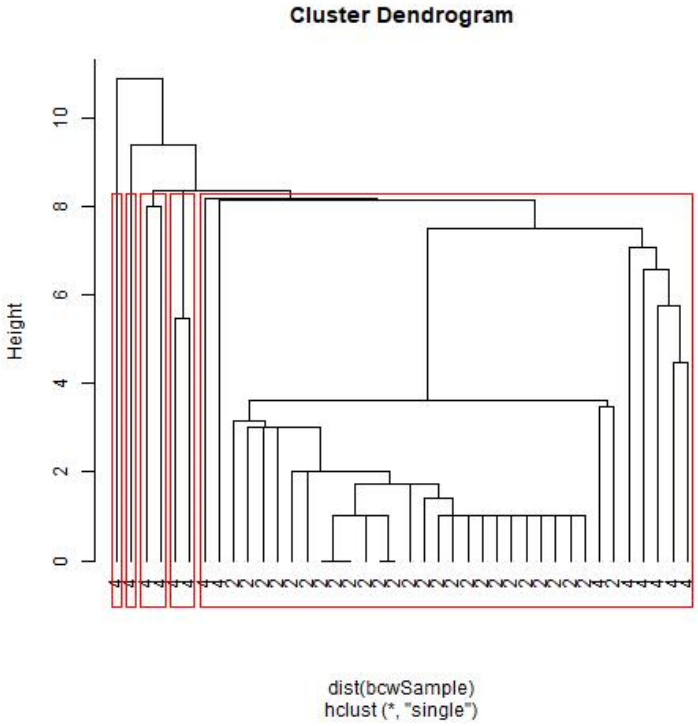
The default agglomeration method is “complete”.

“single” method:

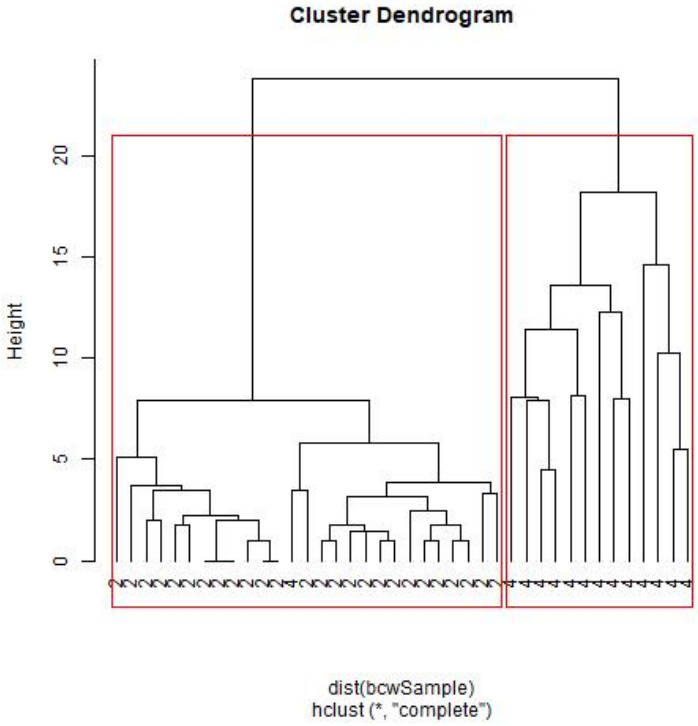


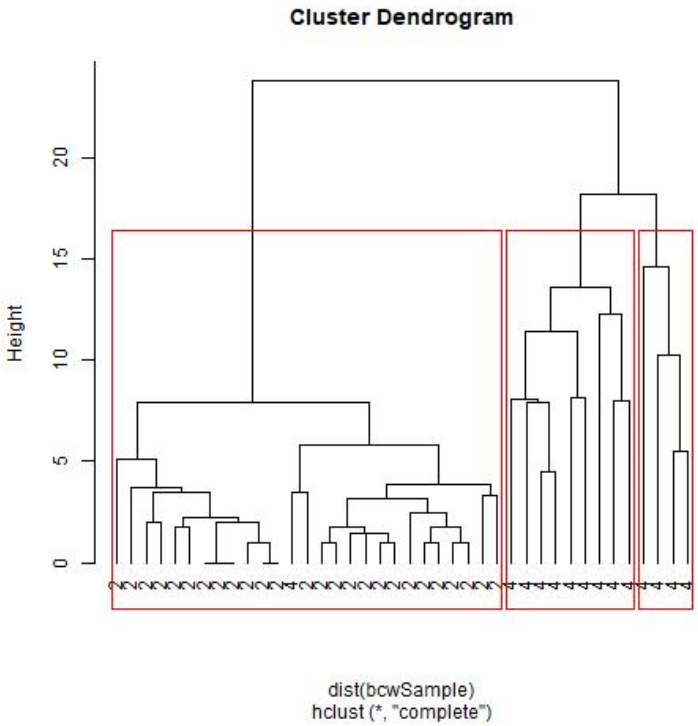


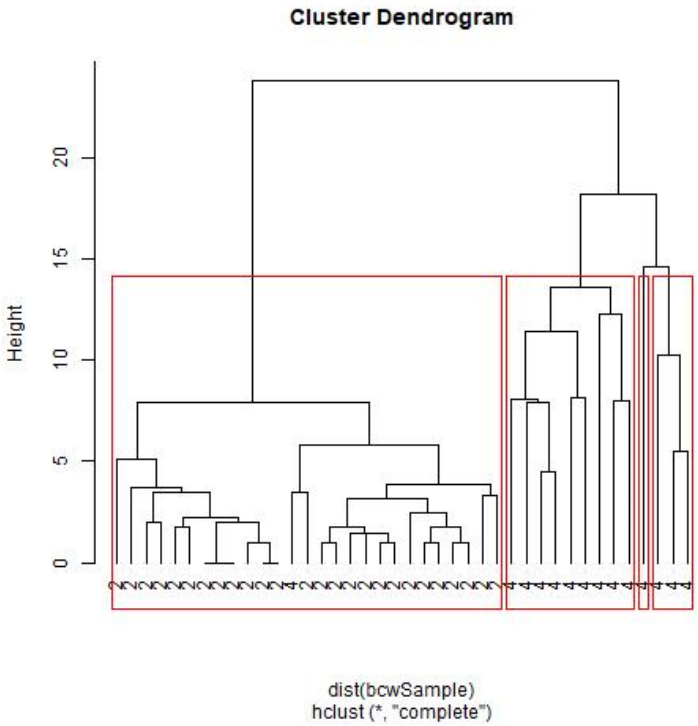

Cluster Dendrogram
Height
0
2
4
6
8
10
dist(bcwSample)
hclust (*, "single")

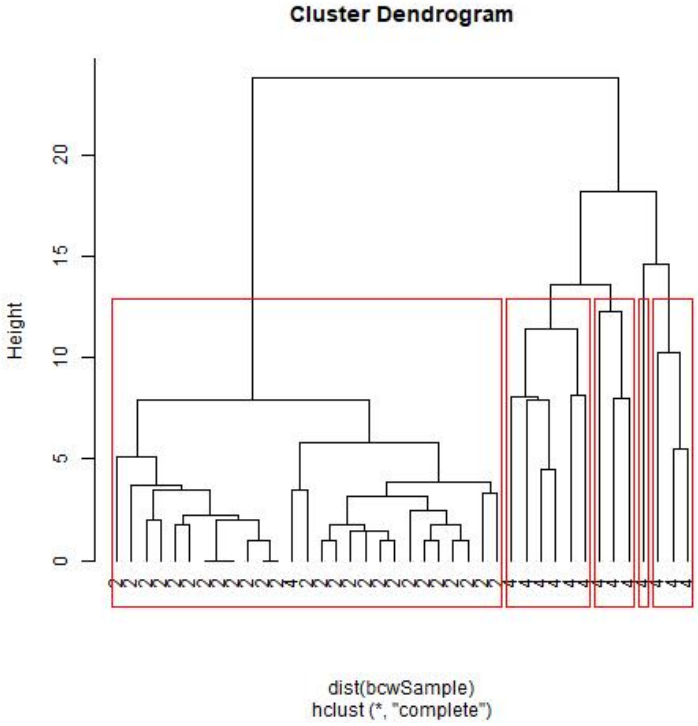

Cluster Dendrogram
Height
0
2
4
6
8
10
dist(bcwSample)
hclust (*, "single")

“complete” method:

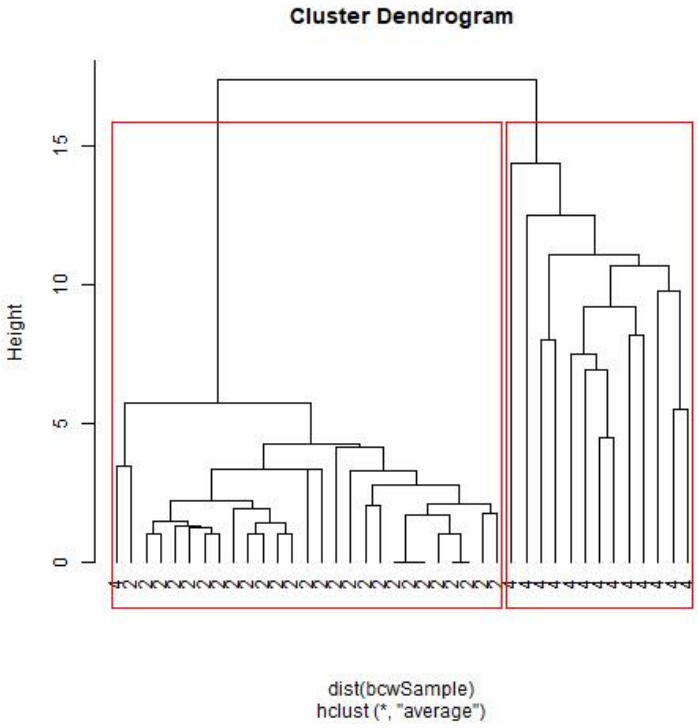


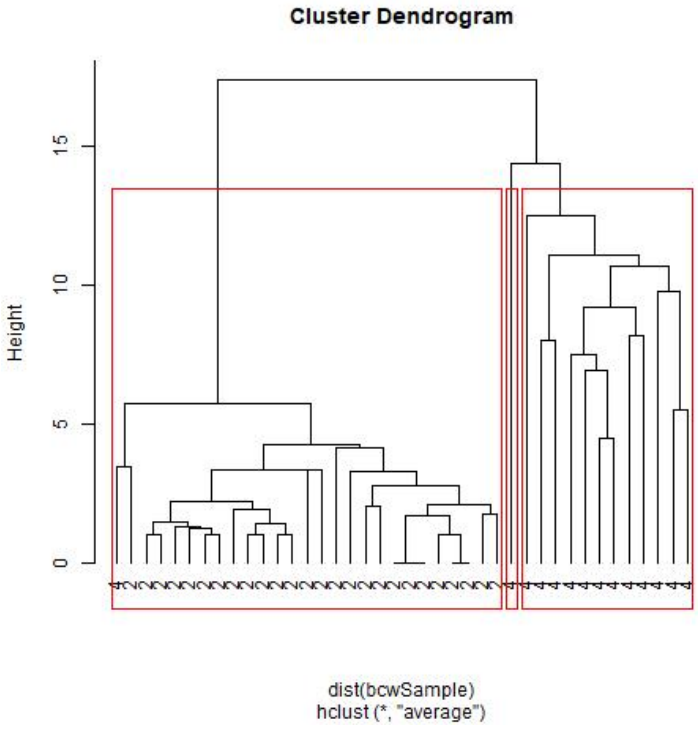


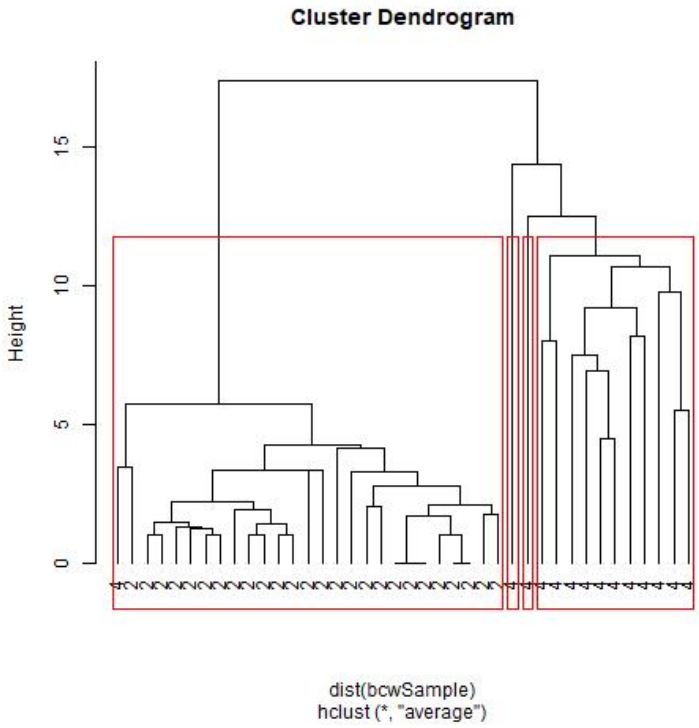

Cluster Dendrogram
Height
0
5
10
15
20
dist(bcwSample)
hclust (*, "complete")

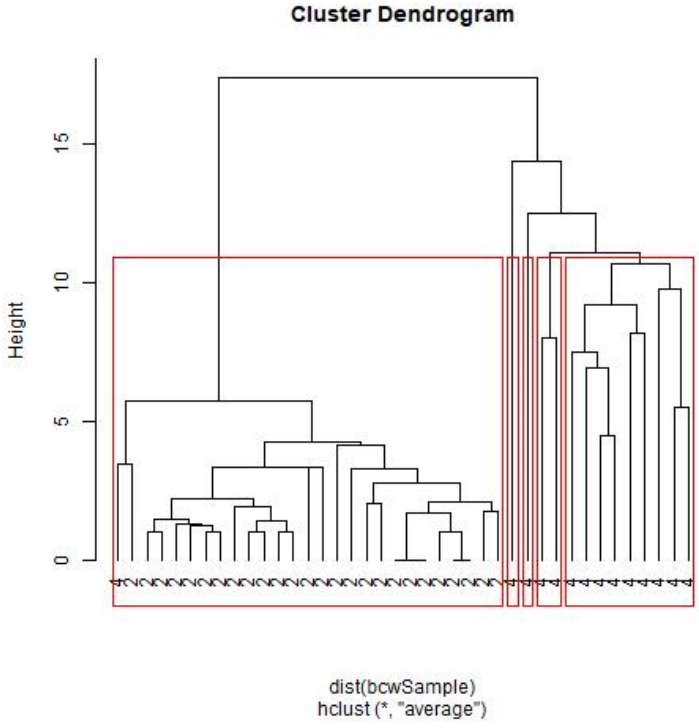

Cluster Dendrogram
Height
0
5
10
15
20
dist(bcwSample)
hclust (*, "complete")

“average” method:






Cluster Dendrogram
Height
0
5
10
15
dist(bcwSample)
hclust (*, "average")


Cluster Dendrogram
Height
0
5
10
15
dist(bcwSample)
hclust (*, "average")

Task 3.2

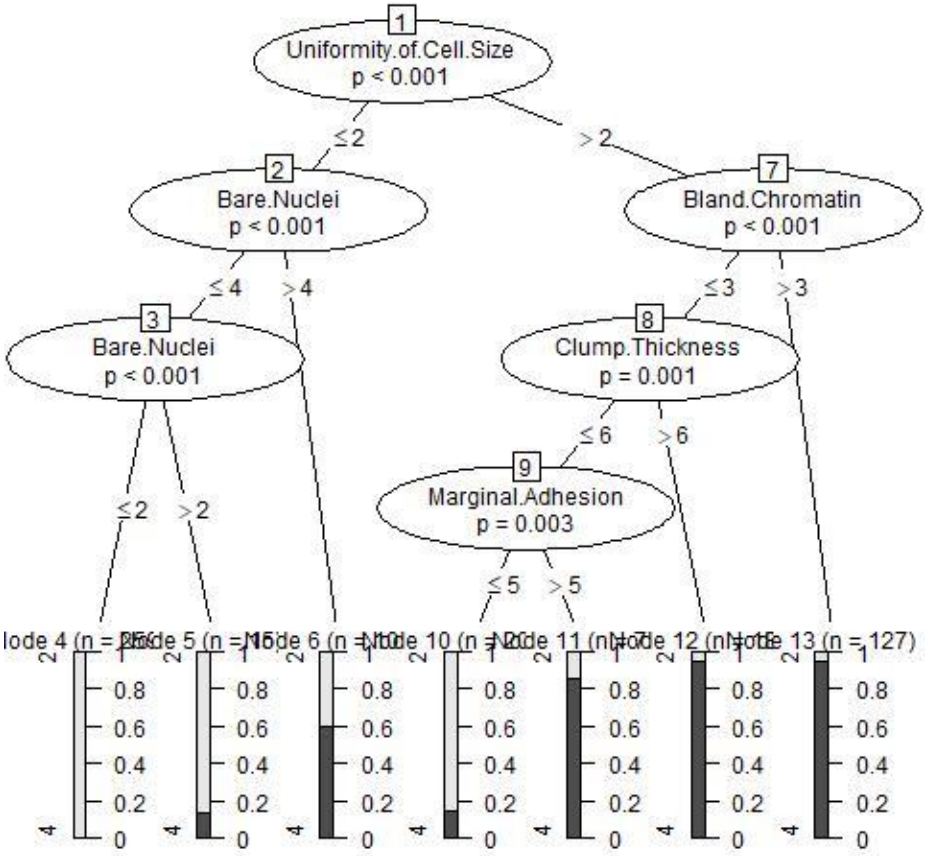
Important variables: Uniformity of Cell Size, Clump Thickness, Marginal Adhesion, Bare Nuclei, Bland Chromatin

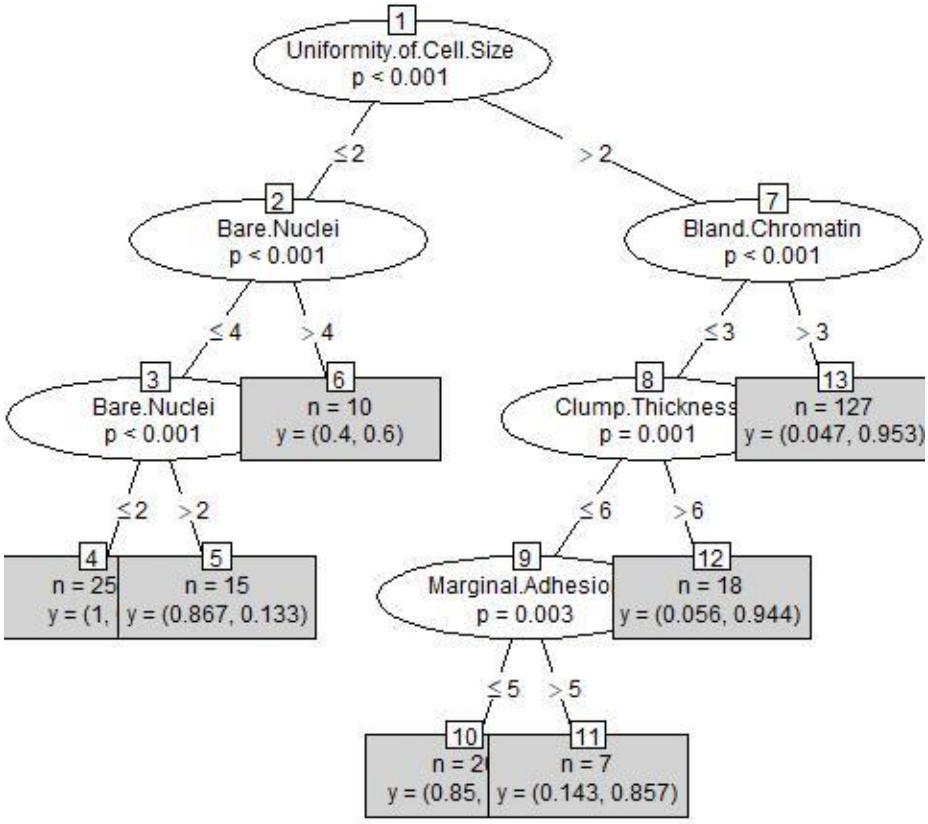
Unimportant variables: Uniformity of Cell Shape, Uniformity of Cell Shape, Normal Nucleoli, Mitoses

Depth: 5

Knowledge: For a breast cancer, if its Uniformity of Cell Size is less than 2 and Bare Nuclei is less than 4, it is more likely to be benign. However, if a breast cancer's Uniformity of Cell Size and Marginal Adhesion are greater than 2 and 5 respectively, it is more likely to be malignant.

```
> result
   accuracy precision    recall        f1
2 0.9471366 0.9712230 0.9440559 0.9574468
4 0.9471366 0.9090909 0.9523810 0.9302326
```





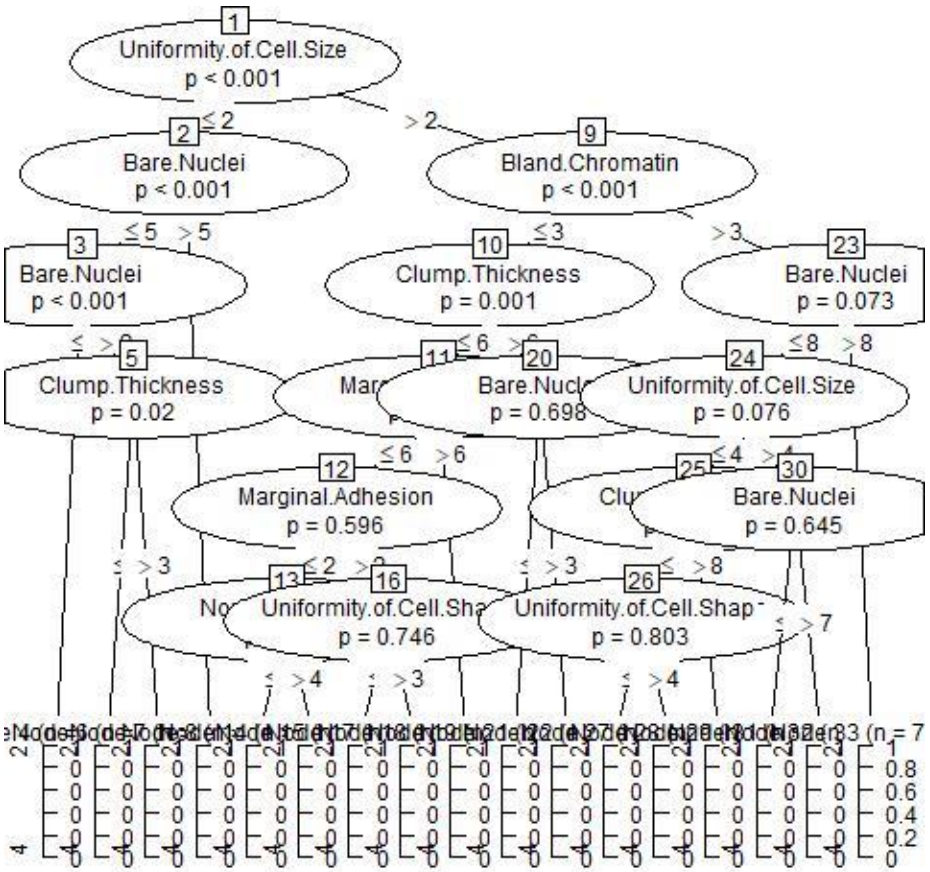
Task 3.3

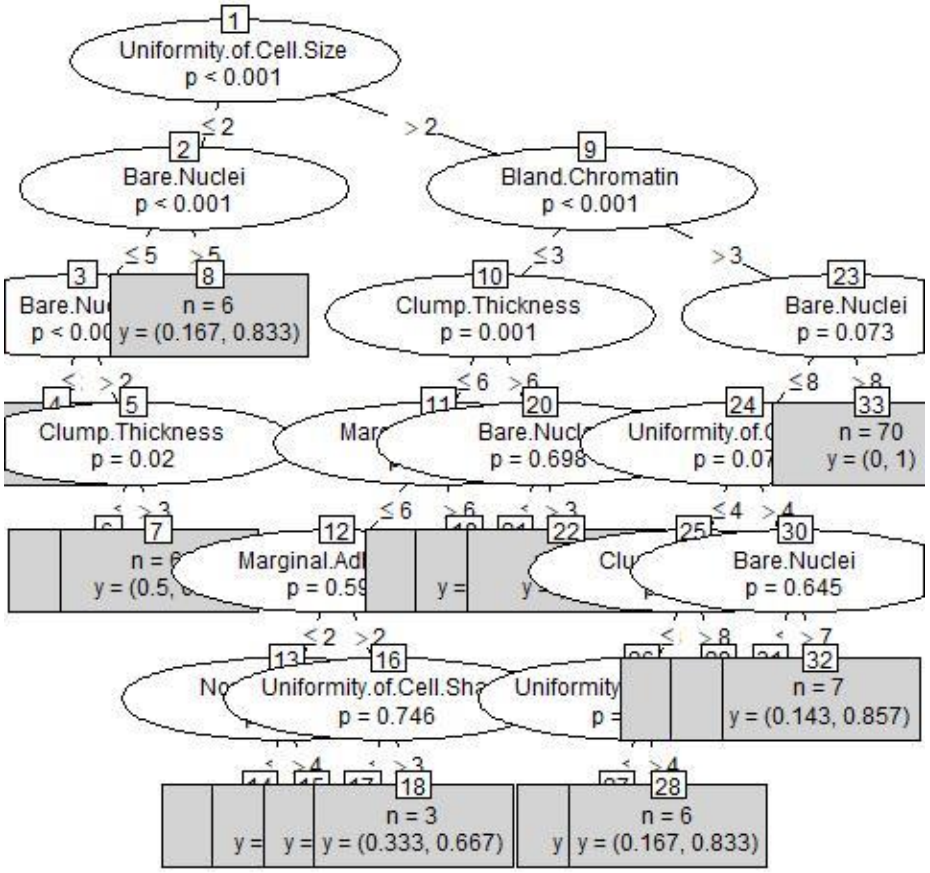
In my classification tree, minbucket = 3L, minsplit = 7L, testtype = "Bonferroni", mincriterion = 0.1, the lower values of these parameters result in a classification tree with higher depth and more branches.

Better accuracy was achieved in this case

```
> my_result
  my_accuracy my_precision my_recall     my_f1
2    0.969163    0.9594595 0.9930070 0.9759450
4    0.969163    0.9873418 0.9285714 0.9570552
```

```
> result
   accuracy precision    recall        f1
2 0.9471366 0.9712230 0.9440559 0.9574468
4 0.9471366 0.9090909 0.9523810 0.9302326
```


1
Uniformity.of.Cell.Size
p < 0.001
≤ 2
> 2
2
Bare.Nuclei
p < 0.001
9
Bland.Chromatin
p < 0.001
3
Bare.Nuclei
p < 0.001
≤ 5
> 5
10
Clump.Thickness
p = 0.001
≤ 3
> 3
23
Bare.Nuclei
p = 0.073
5
Clump.Thickness
p = 0.02
11
20
Bare.Nuclei
p = 0.698
≤ 6
> 6
24
Uniformity.of.Cell.Size
p = 0.076
≤ 8
> 8
12
Marginal.Adhesion
p = 0.596
≤ 6
> 6
25
30
Bare.Nuclei
p = 0.645
≤ 4
> 4
13
16
Uniformity.of.Cell.Shap
p = 0.746
≤ 2
> 2
26
Uniformity.of.Cell.Shap
p = 0.803
≤ 8
> 8
≤ 7
> 7
≤ 3
> 3
≤ 4
> 4
1
0.8
0.6
0.4
0.2
0


1
Uniformity.of.Cell.Size
p < 0.001
≤ 2
> 2
2
Bare.Nuclei
p < 0.001
9
Bland.Chromatin
p < 0.001
3
8
n = 6
y = (0.167, 0.833)
≤ 5
> 5
10
Clump.Thickness
p = 0.001
≤ 3
> 3
23
Bare.Nuclei
p = 0.073
5
Clump.Thickness
p = 0.02
11
20
Bare.Nuclei
p = 0.698
≤ 6
> 6
24
33
n = 70
y = (0, 1)
≤ 8
> 8
7
n = 6
12
Marginal.Adhesion
p = 0.596
22
25
30
Bare.Nuclei
p = 0.645
≤ 4
> 4
13
16
Uniformity.of.Cell.Shap
p = 0.746
≤ 2
> 2
32
n = 7
y = (0.143, 0.857)
≤ 8
> 8
≤ 7
> 7
18
n = 3
y = (0.333, 0.667)
28
n = 6
y = (0.167, 0.833)
≤ 3
> 3
≤ 4
> 4

Task 3.4

According to my test, the best K value in this scenario is 3. On one hand, when K equals to 3, the values of accuracy, precision, recall and F1 are highest among the 5 results. On the other hand, the other classifiers perform worse than that with K value of 3 (i.e. If k is too small, the classifier is sensitive to noise points. If k is too big, neighborhood may include points from other classes.).

```
> # k = 1
> calculation(1)
   accuracy precision    recall        f1
2 0.9559471 0.9463087 0.9860140 0.9657534
4 0.9559471 0.9743590 0.9047619 0.9382716
>
> # k = 2
> calculation(2)
   accuracy precision    recall        f1
2 0.9559471  0.952381 0.9790210 0.9655172
4 0.9559471  0.962500 0.9166667 0.9390244
>
> # k = 3
> calculation(3)
   accuracy precision    recall        f1
2 0.9867841 0.9861111 0.9930070 0.9895470
4 0.9867841 0.9879518 0.9761905 0.9820359
>
> # k = 4
> calculation(4)
   accuracy precision    recall        f1
2 0.9735683 0.9790210 0.9790210 0.9790210
4 0.9735683 0.9642857 0.9642857 0.9642857
>
> # k = 5
> calculation(5)
   accuracy precision    recall        f1
2 0.9735683 0.9790210 0.9790210 0.9790210
4 0.9735683 0.9642857 0.9642857 0.9642857
```