**INFS7203 Assignment Report**

**Task 1 - Data Preparation**

1.1. Extract the data into an R data frame.

Explain :

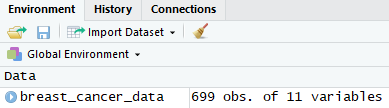
Read the data from raw file into R language system as table, with the separator as comma.

Code :

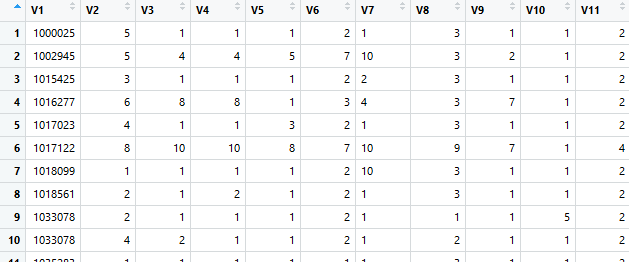


Result :

(1) database



(2) data

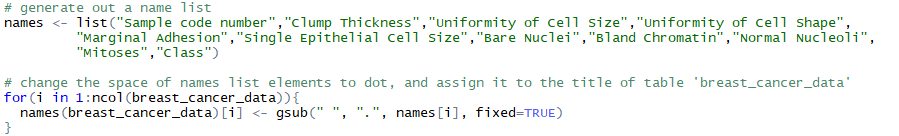


1.2. Assign the following names to the 11 different columns in your dataset.

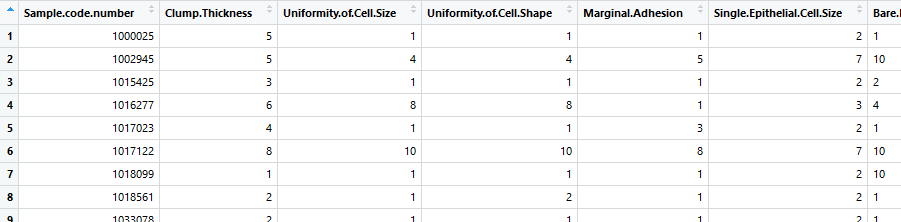
Explain :

Firstly, generate a list contains all the column names provides; Secondly, for each column of breast\_cancer\_data, change the space in its name from space to dot by function ‘gsub’, then assigns it as the column name of ith column in ‘breast\_cancer\_data’ by function ‘names’.

Code :



Result :

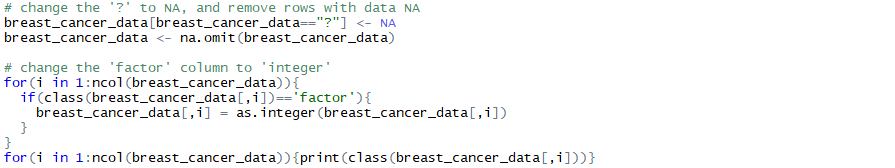


1.3. Remove all rows with missing values. Notice that R might define a column with missing data as “*factor*”. If such column is supposed to be integer, then your need to change the column type from factor to integer.

Explain :

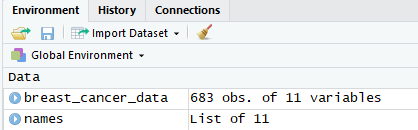
Firstly, update all the cells where shows as ‘?’ in table ‘breast\_cancer\_data’ to ‘NA’, and remove the rows has ‘NA’ with the function na.omit; Secondly, for each column in the table, check whether the column category is ‘factor’, and if it is, resign its category to integer with function ‘as.integer’.

Code :

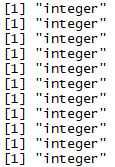


Result :

(1) database



(2) class of each column



1.4 Remove the first column (Sample code number) as it is not useful for our next tasks.

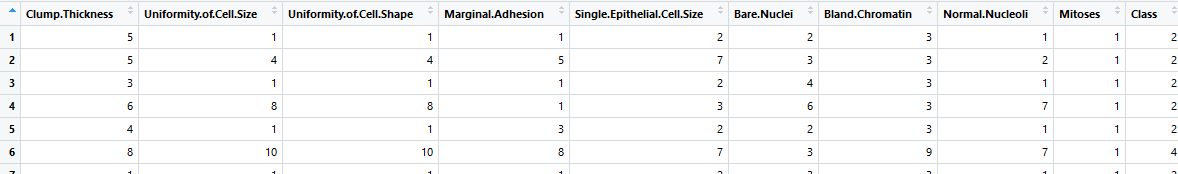
Explain :

Choose all the rows of the table, and remove the first column by choose the column as ‘-c(1)’, which adopted to remove first column when choosing data.

Code :



Result :



1.5 Notice that R might define the “class” column as integer. In that case, change its type from integer to factor.

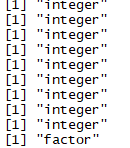
Explain :

Choose the ‘Class’ column by ‘breast\_cancer\_data$Class’, and define it as factor by function ‘as.factor’, then assign the data back to that column.

Code :



Result :

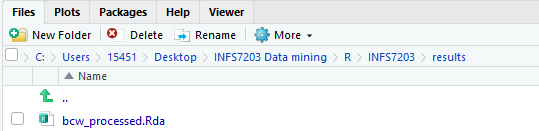


1.6. Save the data frame into a file with filename bcw\_processed.Rda .

Code :



Result :



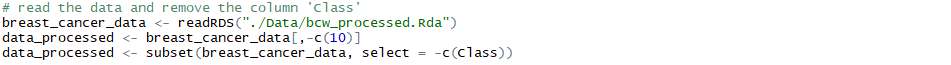
**Task 2 – Clustering**

2.1. Load the preprocessed data file from Task 1 into a data frame. Please note that for this set of clustering tasks, you should not include the Class column.

Explain :

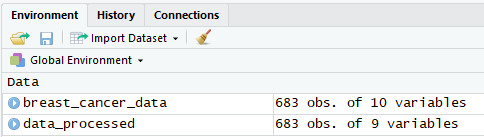
Read the data from ‘bcw\_processed.Rda’ obtained from the task1. There are two method could be used to remove the ‘Class’ column. For the first method, chooses all the rows and removes the last column by –c(10); for the second method, generate a subset by removing the column ‘Class’.

Code :

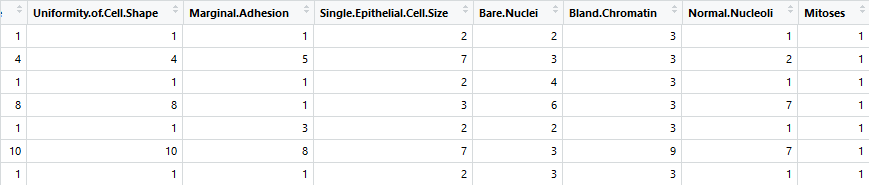


Result :

(1) database



(2) last column screenshot

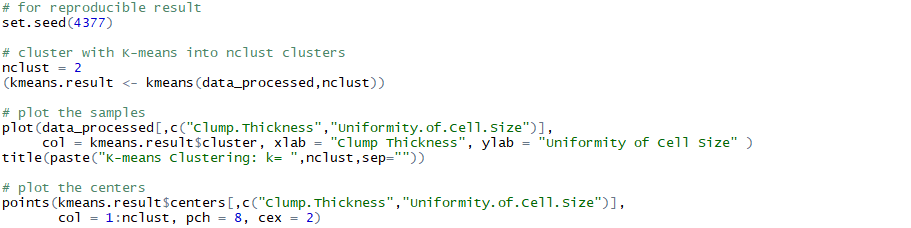


2.2. Cluster the data into 2 clusters using K-Means clustering, using the default parameters for the kmeans function. Plot the results of the clusters as a 2D plot where the x-axis is Clump Thickness and the y-axis is Uniformity of Cell Size.

Explain :

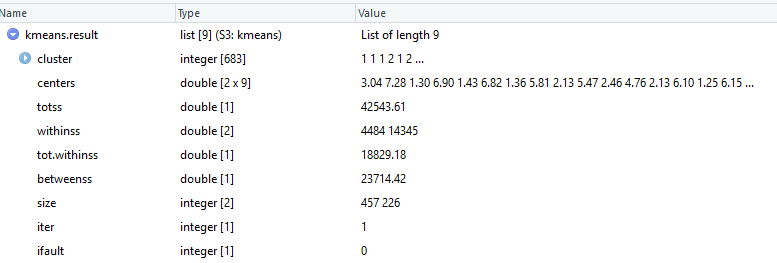
Firstly, set the seed to random generate data; secondly, set the cluster number and generate out the k-means clustering by assign the data and clustering number; thirdly, plot the 2 dimensions sample plot and add the title as “K-means Clustering: 2”; finally, draw the centroids on the graph by choose the data from “kmeans.result$centers”, and set the color as 1 and 2.

Code :



Result :

(1) k-means result



(2) Graph

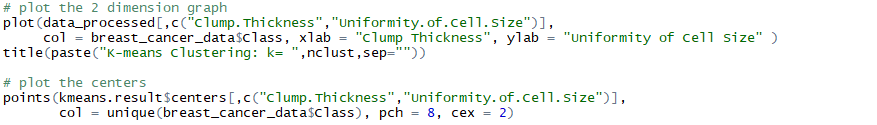


2.3. Plot another 2D plot with the same dimensions above, but color the points according to the Class column.

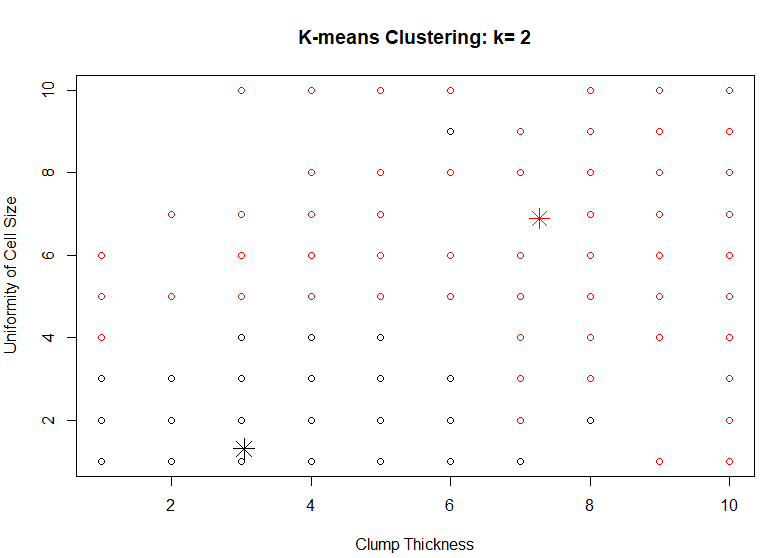
Explain :

Similar as the previous actions, and what is different is that setting the color of points and centroids as ‘’breast\_cancer\_data$Class”.

Code :



Result :



2.4. Compare the 2 plots obtained in the previous two tasks – do the clusters visually represent the benign vs malignant classes?

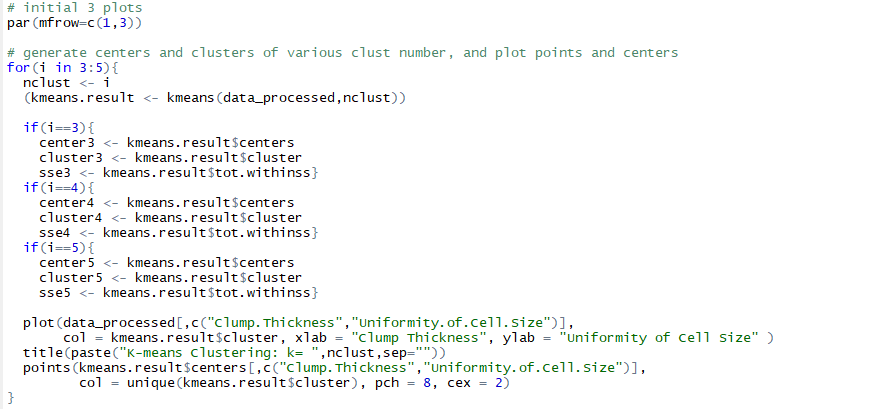
According to the comparing of the 2 plots, the entire trend of cluster results represent the benign and malignant classes from the raw data, although some points assign to various class.

2.5. Cluster the data into more than 2 clusters (i.e., k = 3, 4, 5) using K-Means clustering and plot all the clustering results.

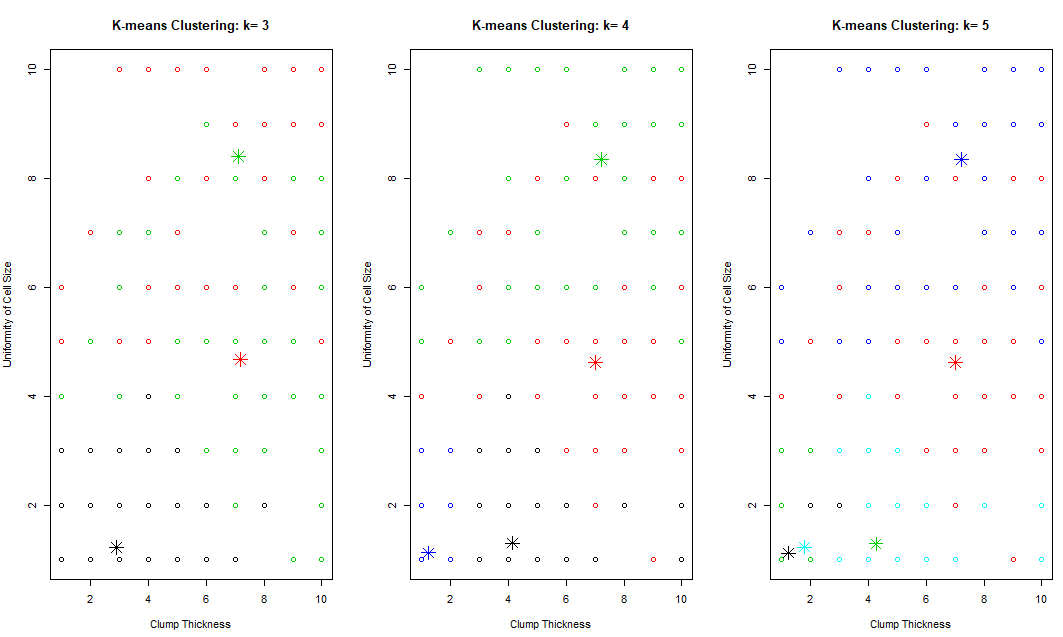
Explain :

Adopting ‘for loop’ to do various number K-Means clustering, and plot the graph with points distribution and centroids. Besides, save the result of centroids, clusters, and SSE into various varieties.

Code :



Result :



2.6. Compare the plots and SSEs obtained in the previous task, and provide your comments on the quality of clustering.

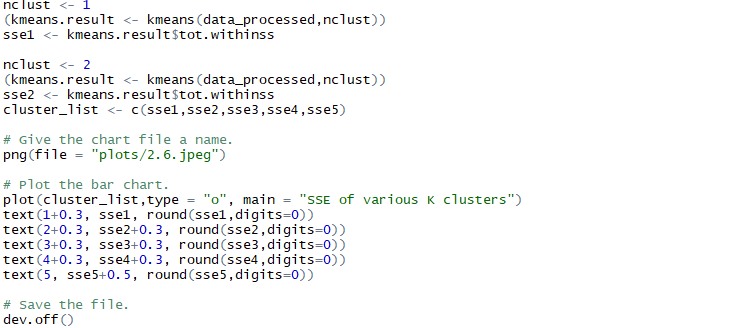
Comment :

As a conclusion, the SSE is decreased according to the increase of n, which is the number of clusters. It indicates that the quality of clustering is improved by choose larger n.

Explain :

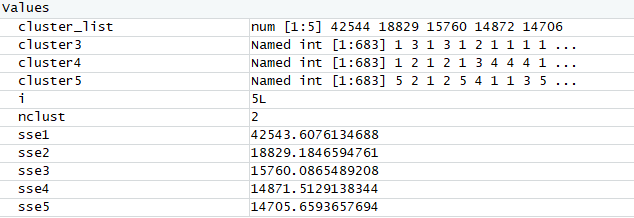
Calculate the kmeans result of 1 and 2 clusters, and save all the SSE by adopting function kmens.result$tot.withinss into the cluster\_list. Draw the graph of the SSE result on the canvas. As a conclusion, the SSE is decreased according to the increase of n, which is the number of clusters. It indicates that the quality of clustering is improved by choose larger n.

Code :

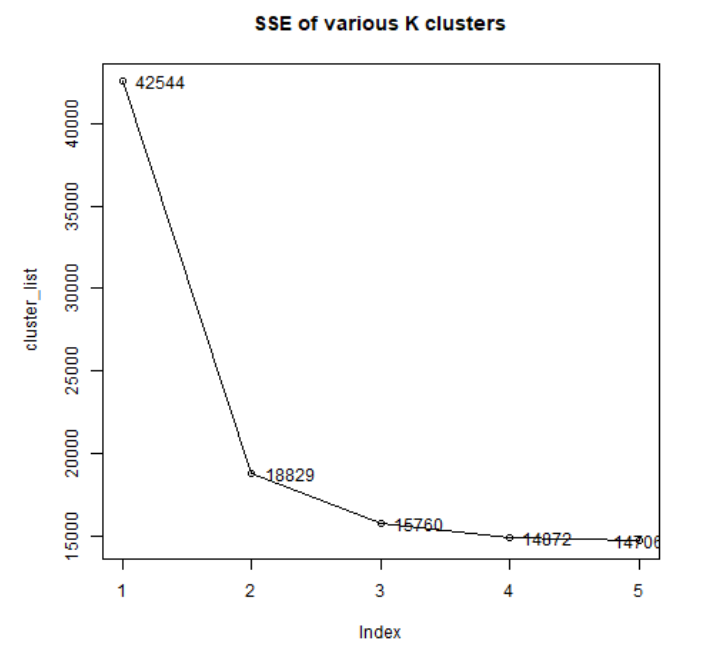


Result :

(1) SSE result



(2) Graph

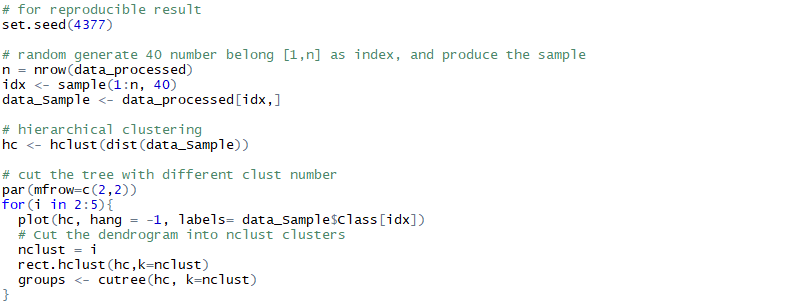


2.7. Apply hierarchical clustering to the data using the hclust function with default parameters and plot the corresponding dendrogram. Particularly, cluster the dendrogram into 2, 3, 4, and 5 clusters and plot all of them.

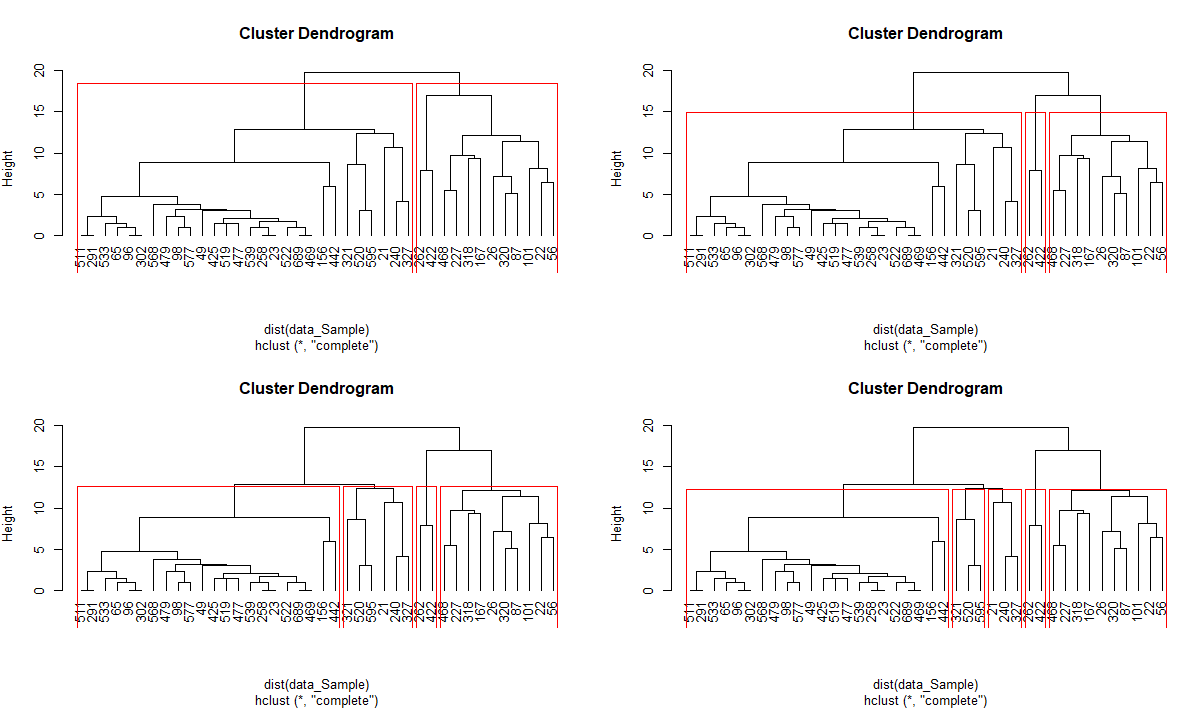
Explain :

Firstly, choose random data by setting the seed; secondly, random generate out 40 data between 1 to rows of data as index, and use the index to random choose 40 samples from ‘data\_processed’ table; thirdly, generate out the hierarchical clustering with the data sample; finally, draw the rectangle to split the hierarchical clustering graph into many parts according to the clustering number.

Code :



Result :



2.8. Compare the plots obtained in the previous task and provide your observations on the achieved clusters - should we have a new subtype of diseases?

Answer :

Comparing the hierarchical clustering of 2 clusters with 3, 4, 5 clusters, only 262 and 422 on the right side has be divide into a new category, and most of the data still belong to the same cluster. Based on it, there is no need to have a new subtype of diseases.

2.9. Try different agglomeration methods in hierarchical clustering (i.e. “single”, “complete”, and “average”). Plot the resulting dendrograms and provide your comments on the quality of clustering - is the data sensitive to the used agglomeration method? Based on your results, what do you think is the default agglomeration method used in Task 2.7?

Answer :

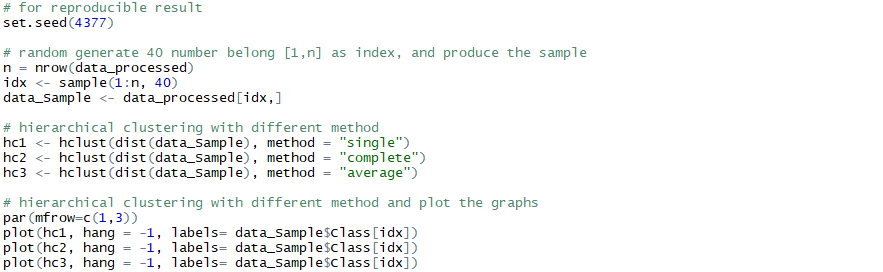
(1) The data is sensitive to the used agglomeration method, because that the clustering results is different based on various agglomeration method.

(2) The second method, “complete” method is the default agglomeration method. Comparing the clustering result graph, the distribution result of “complete” method is same as the result from default method in task 2.7.

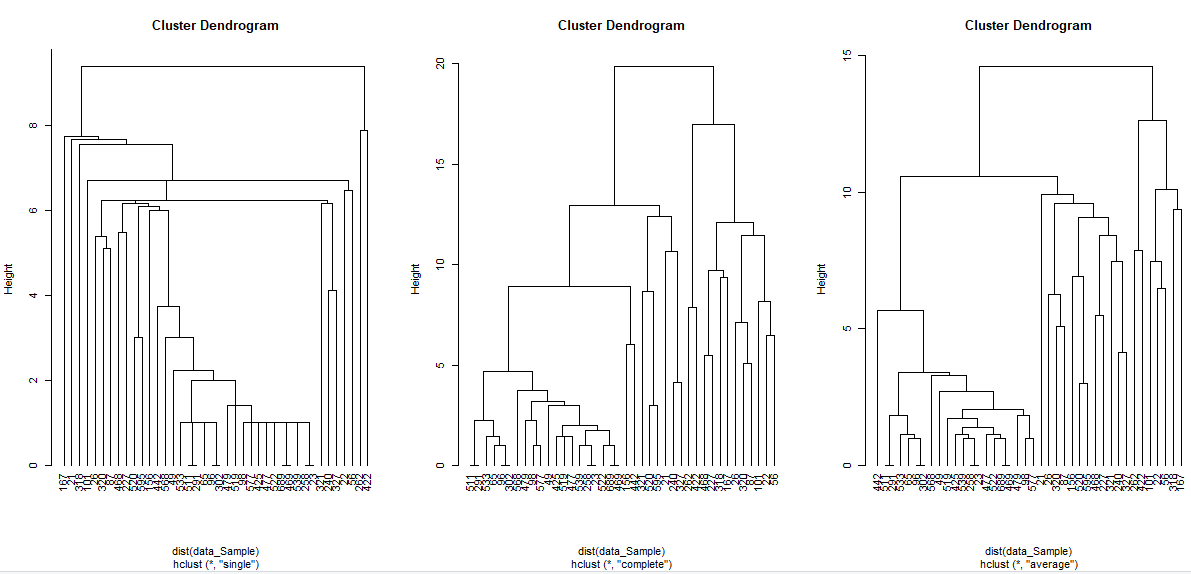
Explain :

Similar as the previous steps, generate out different types of hierarchical clustering by setting the method of hclust function. Besides, plot the hierarchical clustering graphs of various methods.

Code :



Result :



**Task 3 –Classification**

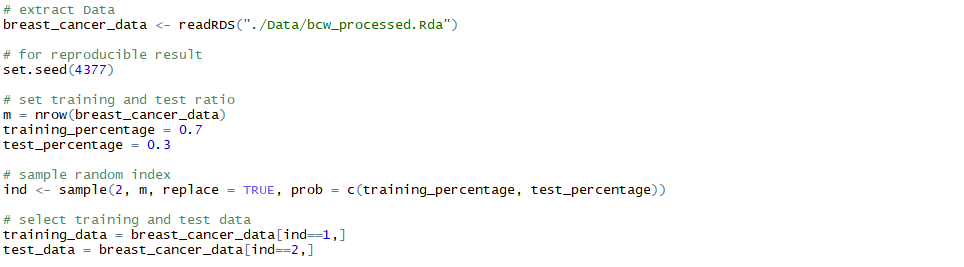
Assumption : For the classification part, it assumes that independence among different attributes.

3.1. Load the preprocessed data file from Task 1 into data frame. Divide the dataset into “training” and “test” subsets randomly (70% and 30% respectively).

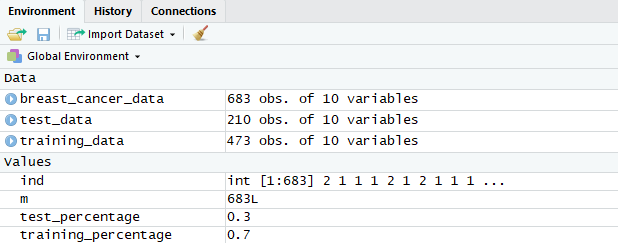
Explain :

Firstly, read the data from Rda file with commend readRDS; secondly, set the reproducible result seed; thirdly, set the training and test data percentage as 0.3 and 0.7, respectively; fourthly, by sample function, randomly generate out m (row number) numbers belong to 1 or 2 with replacement as index, and whose ratio is 0.3 and 0.7, respectively; finally, use the index to generate out training\_data and test\_data by choosing the rows owns the 1 or 2 index.

Code :



Result :



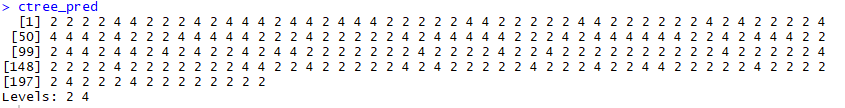
3.2. Learn a classification tree from the training data using the default parameters of the ctree function from the “party” library. Plot that classification tree and provide your comments on its structure (e.g., what are the important/unimportant variables? Is there any knowledge we can infer from the tree representation that helps in differentiating between the classes?). Using the learned tree, predict the class labels of the test data. Calculate the accuracy, precision, and recall.

Answer :

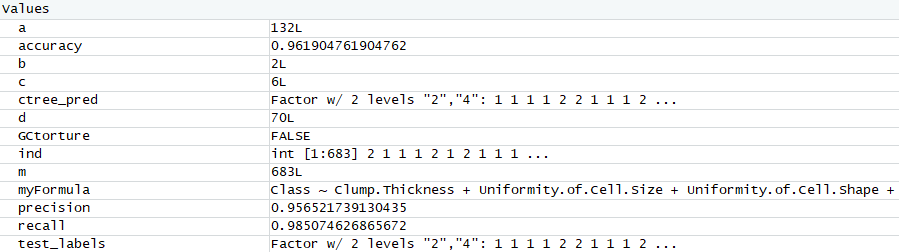
(1) “Uniformity of cell shape” is important variables, because that it almost divided all the samples belong to various categories separately. “Bland Chromatin” is unimportant variables, because that it has not divided the data with labels 2 and 4, and both its subsets has a great amount of data belong to category 4.

(2) Uniformity of cell shape is a critical factor in making category decision and it could be used as the first few steps.

(3) Class labels prediction of the test data



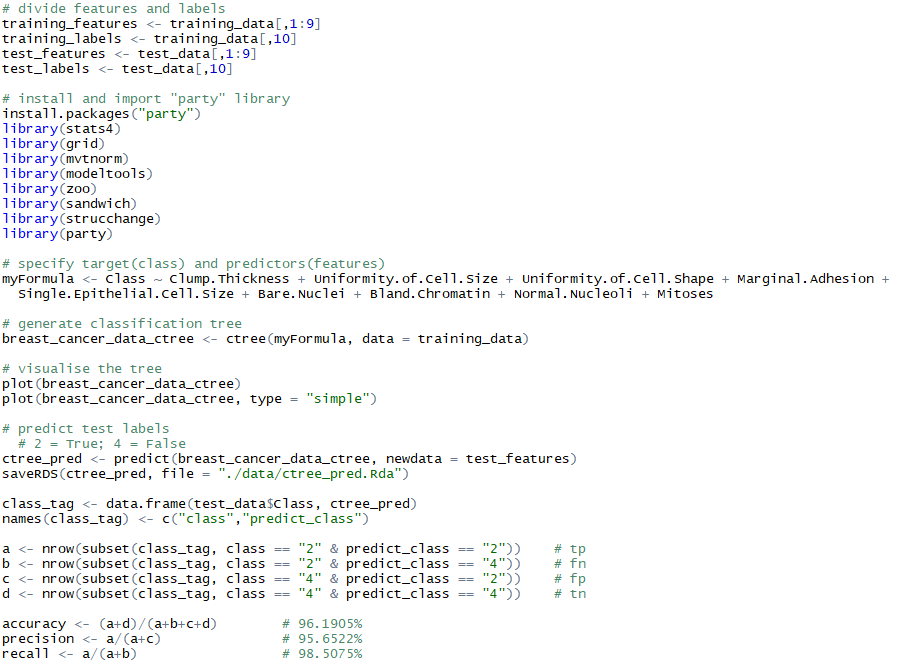
(4) Accuracy/ precision/ recall



Explain :

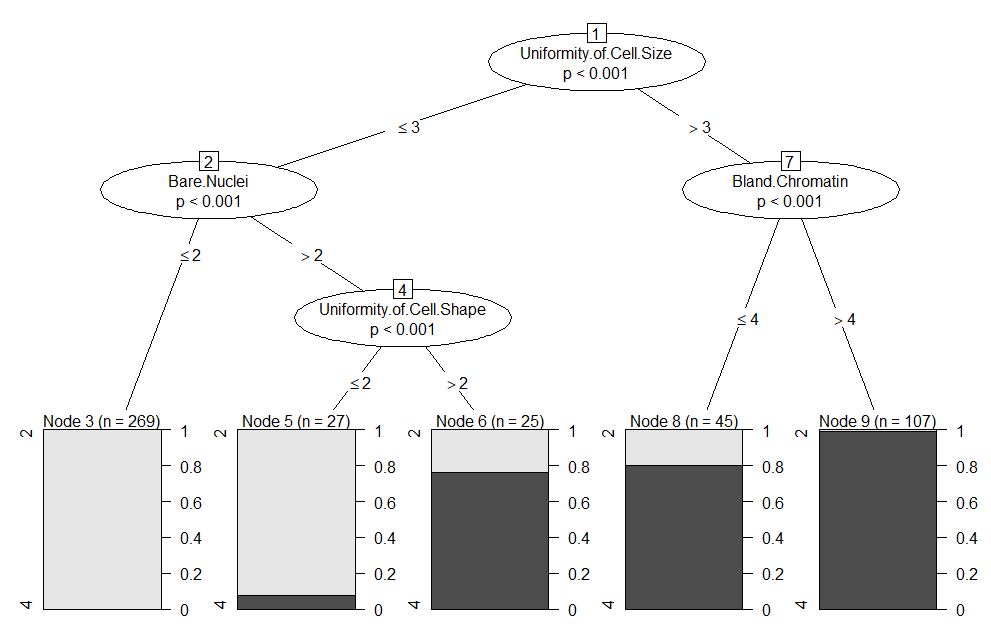
Firstly, generate out the training data and test data; secondly, set up “myFormula” to finger out the predictor attributes and the target attribute; thirdly, using the training data to set up the classification tree model; fourthly, use the model to prediction the class of the test data, and Ctree\_pred is the prediction result of test data; finally, comparing the class of the data with the predicted class of the data, in the aiming of calculate out the accuracy, precision, and recall, which used to evaluate the effectiveness of the model.

Code :

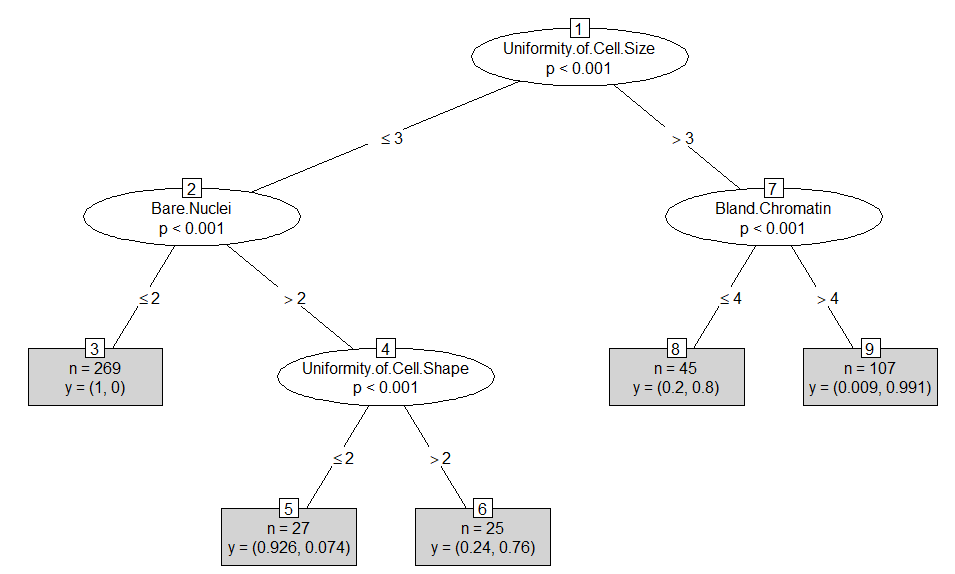


Result :

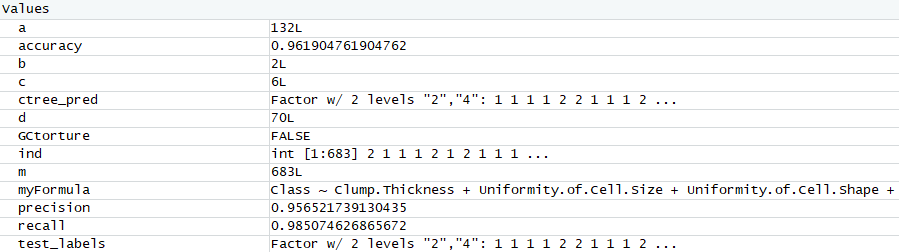
* Graph1



* Graph2



* Accuracy / precision / recall

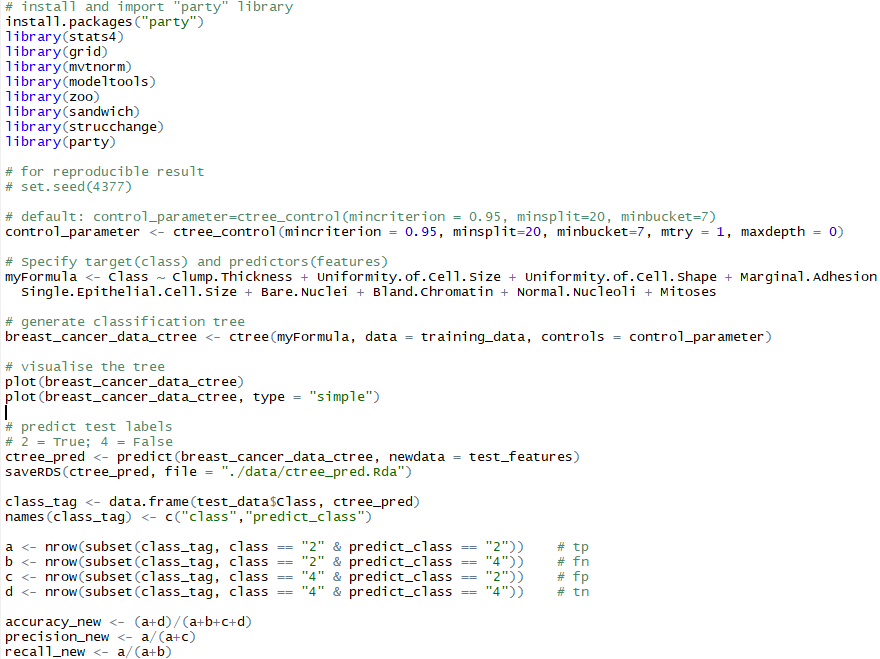


3.3. Try building you classification tree again via the ctree function but using parameters that are different from the default settings. Can you achieve better accuracy or more meaningful representation by tuning some parameters? (Note that in the ctree function from “party” library, you can modifiy ctree\_control parameters. Execute?ctree form RStudio Console for the detailed documentation.)

Answer :

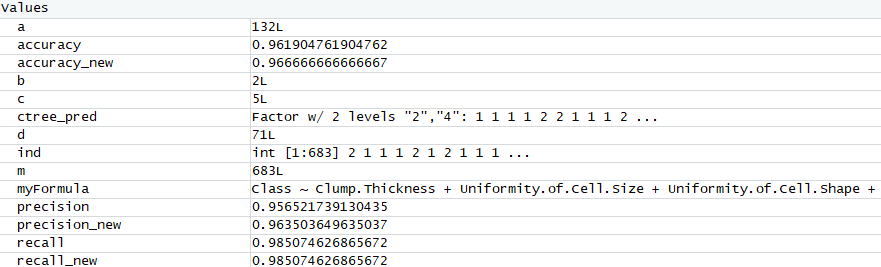
Reset the ctree\_control parameter, and change the mtry equal to 1 or some other random number, except 0. Run the code many times, and it will achieve better accuracy and precision sometime.

Code :

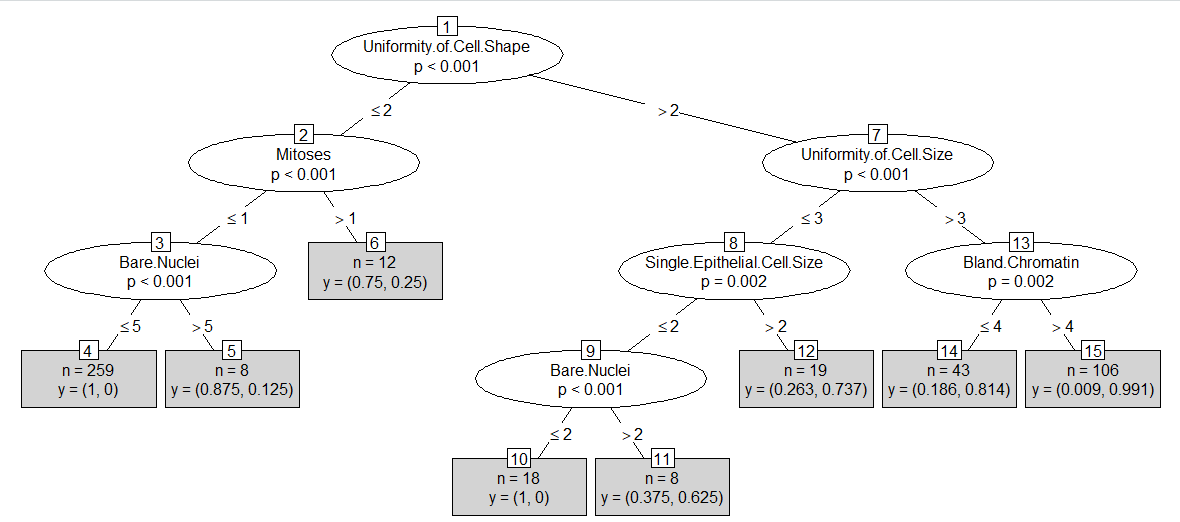


Result :

(1) accuracy / precision / recall



(2) Graph 2



3.4 Apply K-NN classification to predict the labels in the test subset and calculate the accuracy, precision and recall. Particularly, try different values of K (e.g. K = 1, 2, 3, 4, 5), and report your observations on the achieved classification.

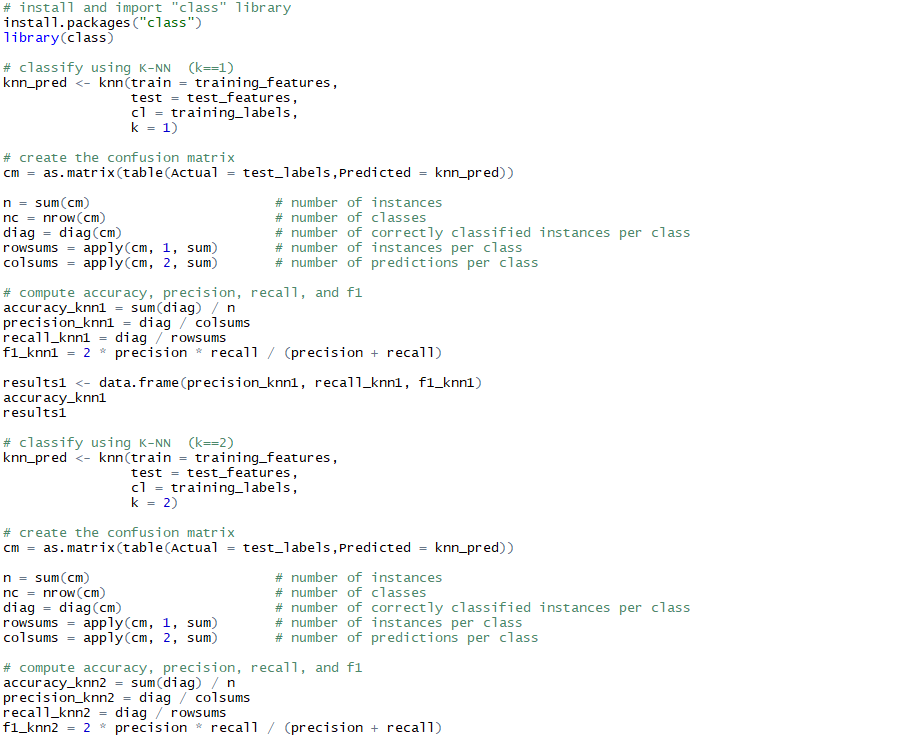
Answer :

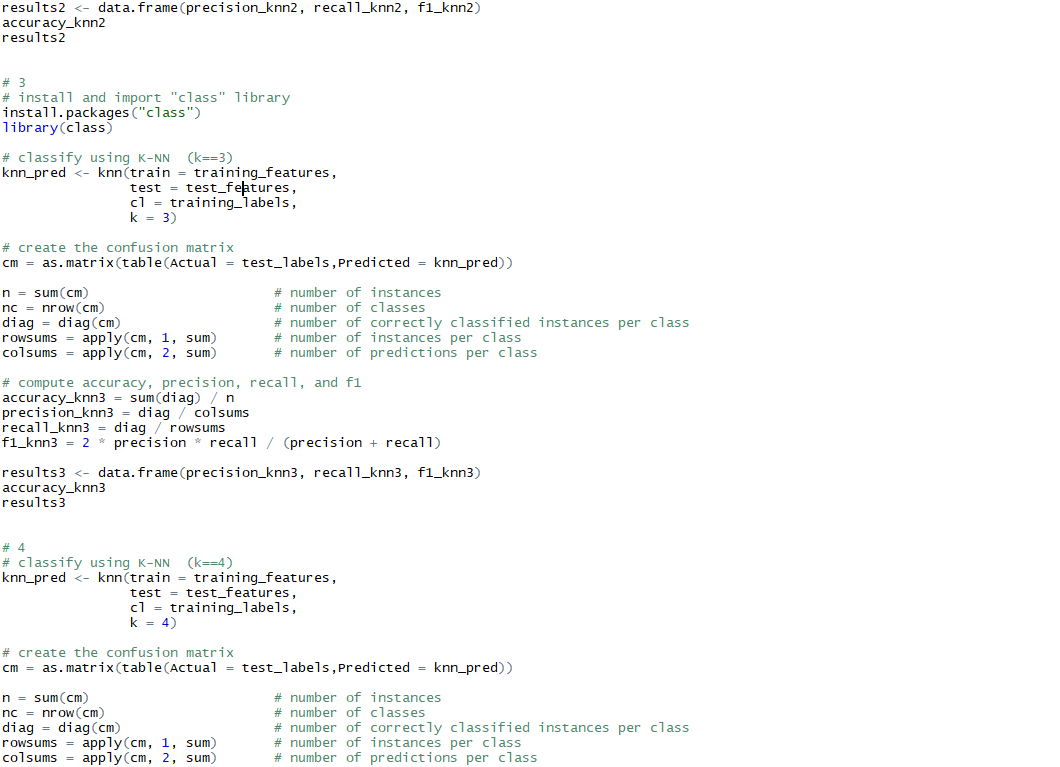
According to the records of knn accuracy, precision, recall, and f1 with various n, the entire trend of knn results is that accuracy and precision increased as the k value increased. The value of f1 kept constant as n increasing. The recall of label ‘4’ also increased as the increase of k. It could be explained by that, with more clusters, the points can be assigned to the centroids, which is nearer or more better explain it, which contributes to a better accuracy and precision.

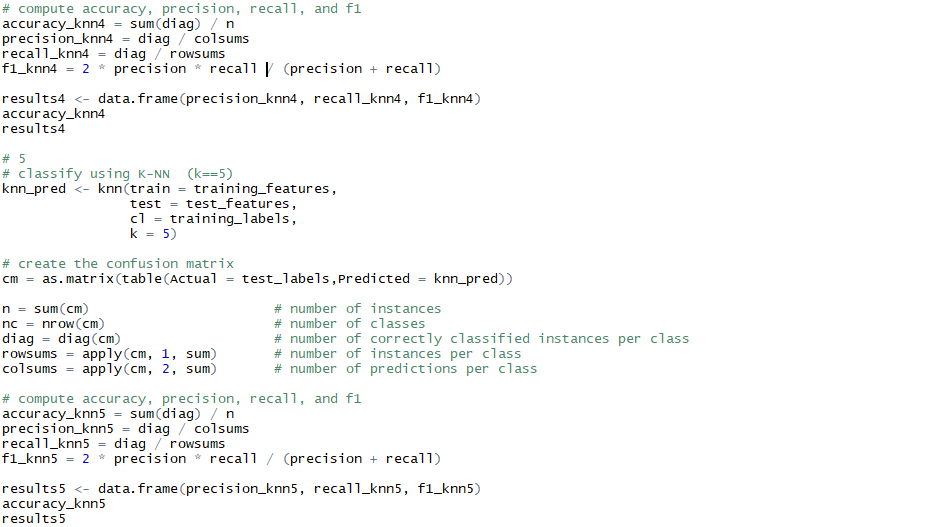
Explain :

Firstly, use for loop to work out the result of different K-NN classification. Use cm to generate the confusion matrix including true positive, false positive, true negative and false negative values. Secondly, calculate the n as the sum of TP, FP, TN and FN; diag as the sum of TP and TN; rowsums are the sum of each rows, and what worth mentioned here is that result of rowsums (TP+FN) has 2 value in the matrix, where the first one is used for category ‘2’ to calculate recall and the second one is used for category ‘4’ to calculate recall; similarly, colsums (TP+FP) is used to calculate precision.

Code :







Result :

|  |  |  |
| --- | --- | --- |
| K | Accuracy\_knn | results |
| 1 | 0.9428571 | precision\_knn1 recall\_knn1 f1\_knn1  2 0.9295775 0.9850746 0.9705882  4 0.9705882 0.8684211 0.9705882 |
| 2 | 0.9571429 | precision\_knn2 recall\_knn2 f1\_knn2  2 0.9496403 0.9850746 0.9705882  4 0.9718310 0.9078947 0.9705882 |
| 3 | 0.9619048 | precision\_knn3 recall\_knn3 f1\_knn3  2 0.9565217 0.9850746 0.9705882  4 0.9722222 0.9210526 0.9705882 |
| 4 | 0.9571429 | precision\_knn4 recall\_knn4 f1\_knn4  2 0.9496403 0.9850746 0.9705882  4 0.9718310 0.9078947 0.9705882 |
| 5 | 0.9714286 | precision\_knn5 recall\_knn5 f1\_knn5  2 0.9705882 0.9850746 0.9705882  4 0.9729730 0.9473684 0.9705882 |

* k=3 example :

