**CDS501**

**Principles & Practices of Data Science & Analytics**

**Assignment 2**

**Sem 1 2022-2023**

**Group Members**

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Matric number** |
| **1** | Chong Yong Ming | P-COM0108/22 |
| **2** | Tan Tee Yin | P-COM0090/22 |
| **3** | Looi Kah Fung | P-COM0049/22 |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Group Name** | **:** |  |
| **Project Dataset** | **:** | **Triage** |

Table of Contents

[CHAPTER 1 1](#_Toc124109540)

[BACKGROUND STUDY AND AIM 1](#_Toc124109541)

[1.1 PROBLEM DEFINITON 1](#_Toc124109542)

[PART B. (UNSUPERVISED MODELLING) 2](#_Toc124109543)

[REFERENCES 7](#_Toc124109544)

# PART A. (Cross Validation)

1. **Part A Cross Validation**

Cross-validation is prevalent in classification problems and used the data resampling method to assess the generalization of a predictor model in unseen data. In our case, determining suboptimal performance estimation is our primary goal, and the size of our dataset is considered when choosing a model validation. Decision tree is selected as our model because the Triage dataset is a multi-classification problem. To make the comparison square and fair, the hyperparameter of the decision tree is pre-set in constant where the complexity parameter is fixed at 0.001, min\_bucket, and pruning is set at 5. k-Fold Cross Validation and Leave-One-Out Cross Validation are compared and evaluated thru performance estimation based on the nature of our dataset Triage. Both cross validation techniques employ a different approach, with the Leave one out cross validation is bluntly straightforward by considering each of the observations as the validation set in the sense that it trains and validates the model for all possible combinations while K-fold cross validation is only considering the number of splits in the dataset by pre-specifying the K-value, for example, k=5, the dataset will be split into 5 groups of equal proportion of size, the first group is treated as a validation, and fitted on the remaining k-1 groups.

**Methodology**

*#Triage is a multi-classification dataset with inscript 5 factors; thus decision tree ML is deployed.*

*>model <- rpart(triage~., cp=0.001,maxdepth=5,minbucket=5,method='class',data=triageData)*

*# cp - complexity parameter*

*# maxdepth - max tree depth*

*# minbucket - min number of obs in leaf nodes*

*# method - return classification*

*options(repr.plot.width = 6, repr.plot.height = 6)*

*prp(model, space=4,split.cex=1.2,nn.border.col=0)*

*#Make prediction onto the dataset*

*train\_preds <- predict(model, newdata=triageData, type="class")*

*#Evaluation metric*

*confusionMatrix(factor(train\_preds), factor(triageData$triage))*

*#Cross-validation [k-fold CV vs LOOCV]*

*#k-fold CV=10*

*train\_control\_cv<- trainControl(method="cv",number=10)*

*#LOOCV*

*train\_control\_loocv <- trainControl(method = "LOOCV")*

*#set required parametes for the model type*

*tune\_grid = expand.grid(cp=c(0.001))*

*# Use the train() function to create the model*

*validated\_tree\_cv <- train(triage~.,method='rpart',data=triageData,maxdepth=5,minbucket=5, trControl=train\_control\_cv,tuneGrid=tune\_grid)*

*validated\_tree\_loocv <- train(triage~.,method='rpart',data=triageData,maxdepth=5,minbucket=5, trControl=train\_control\_loocv,tuneGrid=tune\_grid)*

*#summary of the model*

*validated\_tree\_cv*

*validated\_tree\_loocv*

**Model Evaluation**

Performance estimation of both cross-validation techniques is tabulated in table 1. As our dataset is a multi-classification problem set, thus it is evaluated in a contingency way.

Table 1.0: performance estimation of K-Fold Cross Validation and LOOCV.

|  |  |  |
| --- | --- | --- |
|  | 10-Fold Cross Validation | Leave-One-Out Cross Validation |
| Accuracy | 0.9987085 | 0.9961208 |
| Kappa | 0.9985232 | 0.9955609 |

Accuracy and Kappa are used to assess the inter-rater reliability of agreement. The Kappa coefficient indicates the interest in the magnitude of Kappa, values greater than 0.75 or so may be taken to present excellent agreement beyond chance. Simply put, the evaluation is deemed almost perfect for our model. While comparing 2 crossing over approaches, k-Fold Cross Validation and LOOCV, the difference between accuracy and kappa is minimal, which is less than 0.001. The advantage of LOOCV is a less biased model as almost every data point is used for training, wherein training takes place in the model on N-1 data points and testing the model against that one data point which was left in the previous step until all the data points are covered while its disadvantage is its expensive computation time. In another way, k-Fold Cross Validation offers a faster computation speed wherein the dataset has been split randomly into 10 subsets, one subset is used as a validation set and the remnants are involved in training the model. The disadvantage of k-Fold Cross Validation has been hedged by pre-specifying 10-folds as 10 is the most desired k value in most of the studies [1]. Thus, the 10-Fold Cross Validation technique is opted for our small dataset because of speedy computation time, promising performance estimation, and low bias a modest variance.

# PART B. (UNSUPERVISED MODELLING)

**UNSUPERVISED MODELLING: HIERARCHICAL CLUSTERING**

Unsupervised modelling is a modelling technique in which learns patterns from unlabelled data. Unsupervised modelling is mainly applied for clustering, density estimation and dimensionally reduction, unlike supervised modelling which is usually implemented for classification and regression.

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. (S. Patel et al., 2015). There are two hierarchical clustering method, divisive (top-down) and agglomerative (bottom-up). For this assignment, agglomerative hierarchical clustering is chosen to do grouping for triage dataset. An agglomerative clustering starts with a singleton (one object) cluster and then successively merges pairs of clusters until all clusters have been merged into one big cluster containing all objects. (El Bouchefry & de Souza, 2020)

Hierarchical clustering has advantages among other unsupervised modelling. It does not require to specify number of clusters for the algorithm because of the dendrogram can help for the analysis to choose the number of clusters. Besides, it also easy to be implemented with any attribute type and further interpret the analysis result. However, hierarchical clustering will take more time to apply by comparing to K-means so it will not suitable for large dataset. Moreover, it also sensitive to outliers and missing data so data cleaning is a strong role for hierarchical cluster. Furthermore, final results of the hierarchical clustering are highly relied on the initial seeds.

**METHODOLOGY**

First, do set\_seed for maintaining reproducibility of the results. Next, data cleaning is performed after importing the dataset to R. Based on command below, the duplicated or unimportant attribute is removed afterwards.

*> set.seed(354)*

*> triageData <- read.csv('triage.csv')*

*> triageData = subset(triageData, select = -c(X)*

Then, all the categorical attribute are converted to numerical data as the hierarchical clustering only able to process only numeric data else it will get error in R.

*> triageData <- triageData%>%*

*mutate(Residence\_type=case\_when(*

*.$Residence\_type=="Urban" ~ 1,*

*.$Residence\_type=="Rural" ~ 2,*

*))*

*> triageData <- triageData%>%*

*mutate(smoking\_status=case\_when(*

*.$smoking\_status=="never smoked" ~ 1,*

*.$smoking\_status=="formerly smoked" ~ 2,*

*.$smoking\_status=="smokes" ~ 3,*

*))*

*> triageData <- triageData%>%*

*mutate(triage=case\_when(*

*.$triage=="green" ~ 1,*

*.$triage=="blue" ~ 2,*

*.$triage=="yellow" ~ 3,*

*.$triage=="orange" ~ 4,*

*.$triage=="red" ~ 5,*

*))*

After that, the rows, which have missing value, and the target attribute are removed.

*> triageData <- na.omit(triageData)*

*> triageData\_new <- triageData[,-17]*

For the training dataset of hierarchical clustering, there is no action required for data splitting from whole dataset because unsupervised learning does not require any train and test dataset. Therefore, *triageData\_new*is the final dataset and it can proceed to train hierarchical clustering.

To train hierarchical clustering, Euclidean distance is calculated for each instance.

> distance\_mat <- dist(triageData\_new, method = 'euclidean')

Once get Euclidean distance for each instance, it can be fit to the hierarchical clustering model with average method and it can be plotted using command below.

> Hierar\_cl <- hclust(distance\_mat, method = "average")

> plot(Hierar\_cl , main="Average Linkage")

By knowing the target attributes have 5 classes, the dendrogram can be cut into 5 clusters by using commands below.

> rect.hclust(Hierar\_cl , k = 5, border = 2:6)

> abline(h = 5, col = 'red')

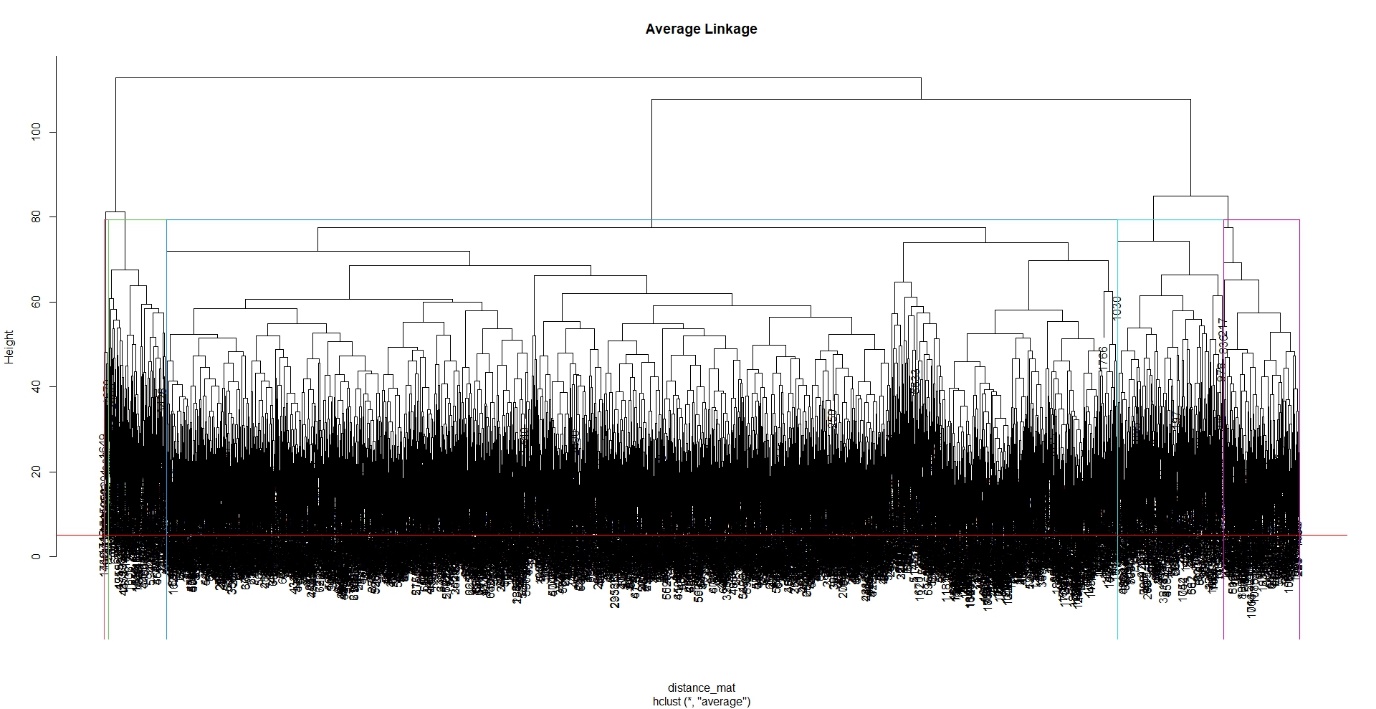


Figure 1: Dendrogram with average method and 5 cuts.

After hierarchical modelling is completed, the result of the model is extracted and implement confusion matrix to evaluate the performance of the model by using commands below.

> a <- cutree(Hierar\_cl,5)

> confusionMatrix(as.factor(a), as.factor(triageData$triage))

**RESULT AND DISCUSSION**

Based on Figure 1, it can be clearly seen that the triage data can be clustered into different groups based on different heights. At the height of 79, the triage data can be grouped into 5 clusters which are match to the number of classes of the target attribute. The number of the instance for each cluster are shown as Figure 2 and Figure 3. By visualizing two plots, it can be seen that hierarchical clustering manage to group most instances into group 3 which is highly similar to the actual clustering and it can be categorized the group 3 in the hierarchical clustering to yellow (potentially unstable).

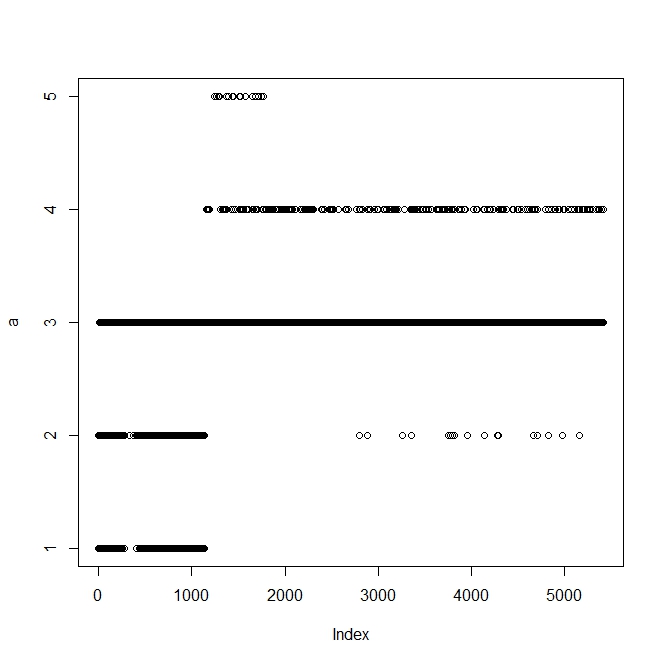


Figure 2: Actual clustering

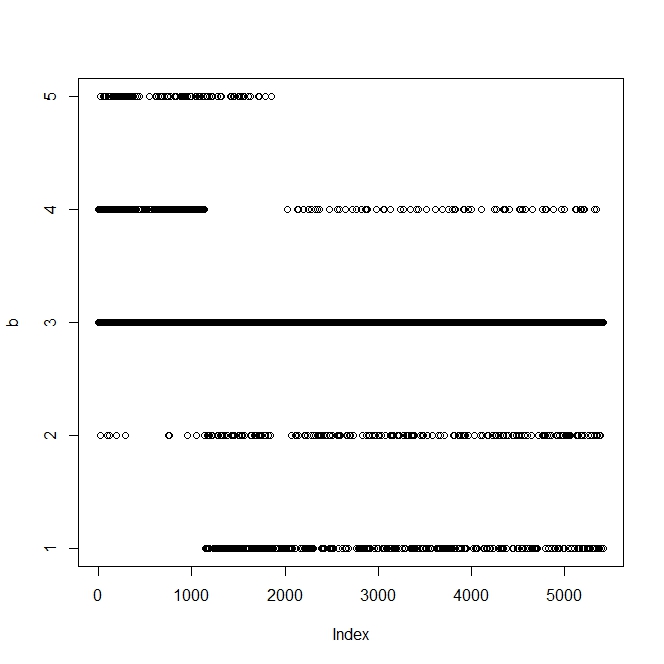


Figure 3: Hierarchical clustering

For the performance of the model, confusion matrix is done and shown in the table 1.

Based on the result of the confusion matrix, the accuracy of the hierarchical clustering is 69.54%. The model is still can be improved by assigning the cluster from the hierarchical model to match with the actual data because hierarchical model only provides the cluster number without the class information. From table 1, the cluster 4 in hierarchical model mostly predicted class 1(non-urgent) so it might belong to class 1. However, this model has poor performance to do clustering for class 2 (minor injuries or complaints) as it only managed to cluster 4 instances to class 2 and the actual number of class 2 is 422. For the overall performance, it can be concluded that hierarchical clustering has an acceptable performance model to perform clustering for the triage data.

*Table 1: Confusion Matrix of hierarchical clustering*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Actual | | | | | |
| Hierarchical |  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 2 | 268 | 53 | 21 |
| 2 | 0 | 4 | 355 | 100 | 22 |
| 3 | 94 | 252 | 3763 | 115 | 86 |
| 4 | 257 | 1 | 3 | 0 | 0 |
|  | 5 | 19 | 0 | 2 | 0 | 0 |

# REFERENCES

S. Patel, S. Sihmar and A. Jatain, "A study of hierarchical clustering algorithms," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 537-541.

El Bouchefry, K. and de Souza, R.S. (2020) “Learning in big data: Introduction to machine learning,” Knowledge Discovery in Big Data from Astronomy and Earth Observation, pp. 225–249. Available at: <https://doi.org/10.1016/b978-0-12-819154-5.00023-0>.

[1] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation," *International Journal of Information Technology and Computer Science,* vol. 13, no. 6, pp. 61-71, 2021.