

**Informatics Institute of Technology**

**Department of Computing**

BSc in Computer Science

**Module: 5DATA001C**

**Machine Learning & Data Mining**

**Module Leader: Mr. Achala Aponso**

**COURSE WORK REPORT**

**Tutorial Group** : SE/CS Group B

**Student IIT ID** : 20191191

**Student UoW ID**  : W17903352

**Student First Name** : Mahfoos

**Student Surname** : Ahamed

Table of Contents

[**Objective 1 (Partitioning clustering)** 3](#_Toc70713689)

[**Methodologies used for reducing the input dimensionality**. 3](#_Toc70713690)

[**Pre- Processing tasks** 3](#_Toc70713691)

[**Code and plot of Outliers detection** 3](#_Toc70713692)

[**Bus Class** 3](#_Toc70713693)

[**Van Class** 4](#_Toc70713694)

[**Saab Class** 4](#_Toc70713695)

[Number of cluster centers 5](#_Toc70713696)

[K-means analysis for each attempt 5](#_Toc70713697)

[Evaluation of the produced outputs against 19th column 5](#_Toc70713698)

[Final Winner cluster case 5](#_Toc70713699)

[Illustrate of coordinates of each center for each clustering group! 5](#_Toc70713700)

[Objective 2 (MLP) 6](#_Toc70713701)

[Methods for defining the input vector in time-series problems. 6](#_Toc70713702)

[Evidence of various adopted Input vectors and the related input/output metrices 6](#_Toc70713703)

[Evidence of correct normalization 6](#_Toc70713704)

[Implement Number of MLPs, using various structures (layers/nodes) 6](#_Toc70713705)

# **Objective 1 (Partitioning clustering)**

# **Methodologies used for reducing the input dimensionality**.

* Here is a brief review of techniques for dimensionality reduction:
* **Missing Values Ratio**. Data columns with too many missing values are unlikely to carry much useful information. Thus, data columns with a ratio of missing values greater than a given threshold can be removed. The higher the threshold, the more aggressive the reduction.
* **Principal Component Analysis (PCA)**. Principal component analysis (PCA) is a statistical procedure that orthogonally transforms the original n numeric dimensions of a dataset into a new set of n dimensions called principal components. As a result of the transformation, the first principal component has the largest possible variance each succeeding principal component has the highest possible variance under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding principal components. Keeping only the first m < n principal components reduce the data dimensionality while retaining most of the data information, i.e., variation in the data. Notice that the PCA transformation is sensitive to the relative scaling of the original columns, and therefore, the data need to be normalized before applying PCA. Also notice that the new coordinates (PCs) are not real, system-produced variables anymore. Applying PCA to your dataset loses its interpretability. If interpretability of the results is important for your analysis, PCA is not the transformation that you should apply.

# **Pre- Processing tasks**

* In the preprocessing task first, identify and detection the outlier of each class (attribute).
* After that removed the outliers

## **Code and plot of Outliers detection**

### **Bus Class**

vehicles\_original %>%

pivot\_longer (2:19, names\_to = "labels") %>%

filter (class == "bus") %>%

mutate (class = fct\_reorder (class, value, median)) %>%

ggplot (aes (class, value, fill = reorder (labels, value))) +

geom\_boxplot() +

labs(title = "Outlier Detection for class: 'bus'")

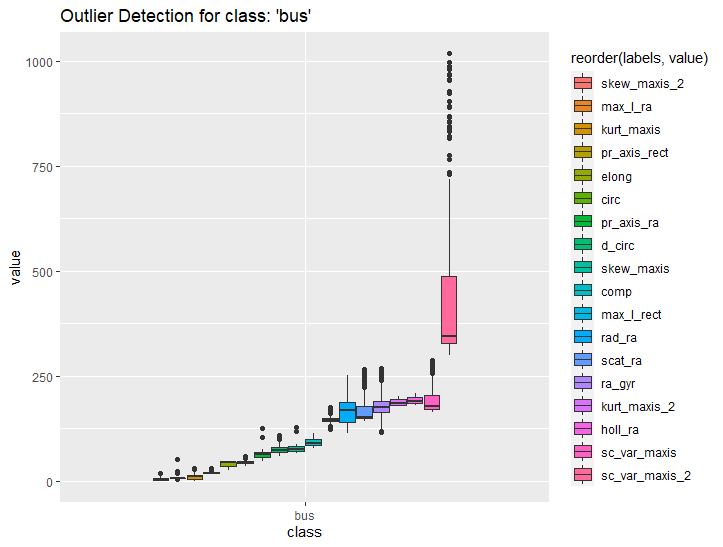


Figure 1 Outlier Detection bus

### **Van Class**

vehicles\_original %>%

pivot\_longer (2:19, names\_to = "labels") %>%

filter (class == "van") %>%

mutate (class = fct\_reorder (class, value, median)) %>%

ggplot (aes (class, value, fill = reorder (labels, value))) +

geom\_boxplot () +

labs (title = "Outlier Detection for class: 'van'")

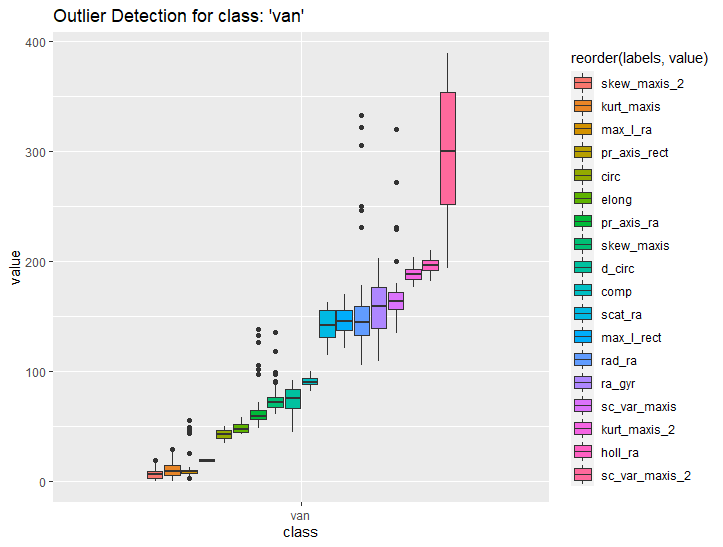


Figure 2 Outlier Detection van

### **Saab Class**

vehicles\_original %>%

pivot\_longer (2:19, names\_to = "labels") %>%

filter (class == "saab") %>%

mutate (class = fct\_reorder (class, value, median)) %>%

ggplot (aes (class, value, fill = reorder (labels, value))) +

geom\_boxplot () +

labs (title = "Outlier Detection for class: 'saab'")

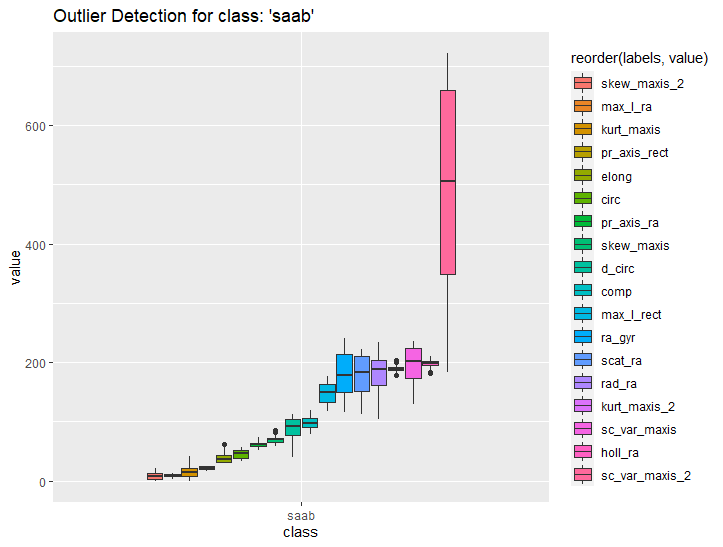


Figure 3 Outlier Detection saab

### **Opel Class**

vehicles\_original %>%

pivot\_longer (2:19, names\_to = "labels") %>%

filter (class == "opel") %>%

mutate (class = fct\_reorder (class, value, median)) %>%

ggplot (aes (class, value, fill = reorder (labels, value))) +

geom\_boxplot () +

labs (title = "Outlier Detection for class: 'opel'")

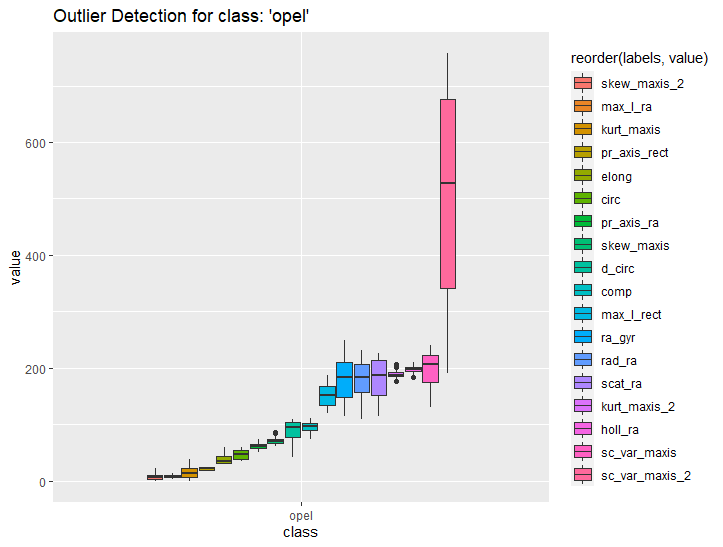


Figure 4 Outlier Detection opel

## **Code of Removing the outliers**

# Remove the Outlier

vehicles\_bus = vehicles\_original %>%

filter(class == "bus") %>%

mutate(across(2:19, ~squish(.x, quantile(.x, c(.05, .95)))))

vehicles\_van = vehicles\_original %>%

filter(class == "van") %>%

mutate(across(2:19, ~squish(.x, quantile(.x, c(.05, .95)))))

vehicles\_opel = vehicles\_original %>%

filter(class == "opel") %>%

mutate(across(2:19, ~squish(.x, quantile(.x, c(.05, .95)))))

vehicles\_saab = vehicles\_original %>%

filter(class == "saab") %>%

mutate(across(2:19, ~squish(.x, quantile(.x, c(.05, .95)))))

# Combining all class

combined = bind\_rows (list (vehicles\_bus, vehicles\_opel, vehicles\_saab, vehicles\_van)) %>%

arrange(samples)

### **Scaling the data**

* When you have variables, which are measured in different scales it is useful to scale the data.

vehicles\_scaled = vehicles\_data\_points %>%

mutate (across (everything (), scale))

# **Number of cluster centers**

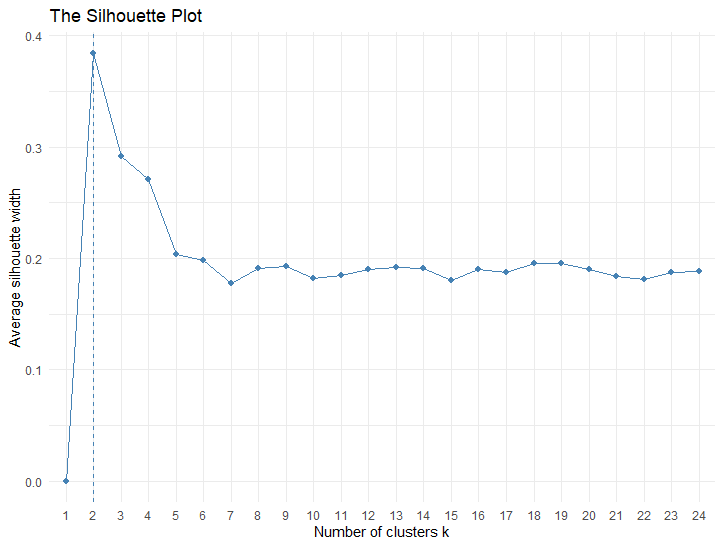
* A variety of measures have been used in the partition clustering for evaluating clustering results. The term **clustering validation** is used to design the procedure of evaluating the results of a clustering algorithm. There are more than thirty indices and methods for identifying the optimal number of clusters.

Silhouette Method

* This method can help determine the optimal number of clusters is called a silhouette method. Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for **k**.

Code

fviz\_nbclust (vehicles\_scaled, kmeans, method = "silhouette", k.max = 24) + theme\_minimal () + ggtitle ("The Silhouette Plot")



## **NbClust**

* The **NbClust** package provides 30 indices for determining the relevant number of clusters and proposes to users the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods.

# Use Euclidean for distance

cluster\_euclidean = NbClust (vehicles\_scaled, distance="euclidean", min.nc=2, max.nc=10, method="kmeans”, index="all")

# Plot the best cluster

factoextra::fviz\_nbclust(cluster\_euclidean) + theme\_minimal() +

ggtitle("NbClust's optimal number of clusters")

# 

# **K-means analysis for each attempt**

# **Evaluation of the produced outputs against 19th column**

# **Final Winner cluster case**

# **Illustrate of coordinates of each center for each clustering group**

# Objective 2 (MLP)

# Methods for defining the input vector in time-series problems.

# Evidence of various adopted Input vectors and the related input/output metrices

# Evidence of correct normalization

# Implement Number of MLPs, using various structures (layers/nodes)