Krishna Guguloth
MGT-665
Dr. Mohammad Nasim

LAB – 3 : UNSUPERVISED LEARNING

Overview

The application of data to comprehend and enhance learning is learning analytics. Without labeled data, machine learning techniques such as unsupervised learning can be used to find patterns and relationships in data.

Introduction

Learning analytics leverages data to enhance educational outcomes. In this case study, we employ unsupervised learning to discern patterns within Simulated School course data. Our focus is on dimensionality reduction and clustering to identify student groups with analogous learning behaviors.

Data

The data for this case study is generated with the simulated function below. The data contains the following features:

- Student ID: A unique identifier for each student
- Feature 1: A measure of student engagement
- Feature 2: A measure of student performance

Data Simulation

Below is the Rstudio Script that I have used for creating a data set for 100 students.

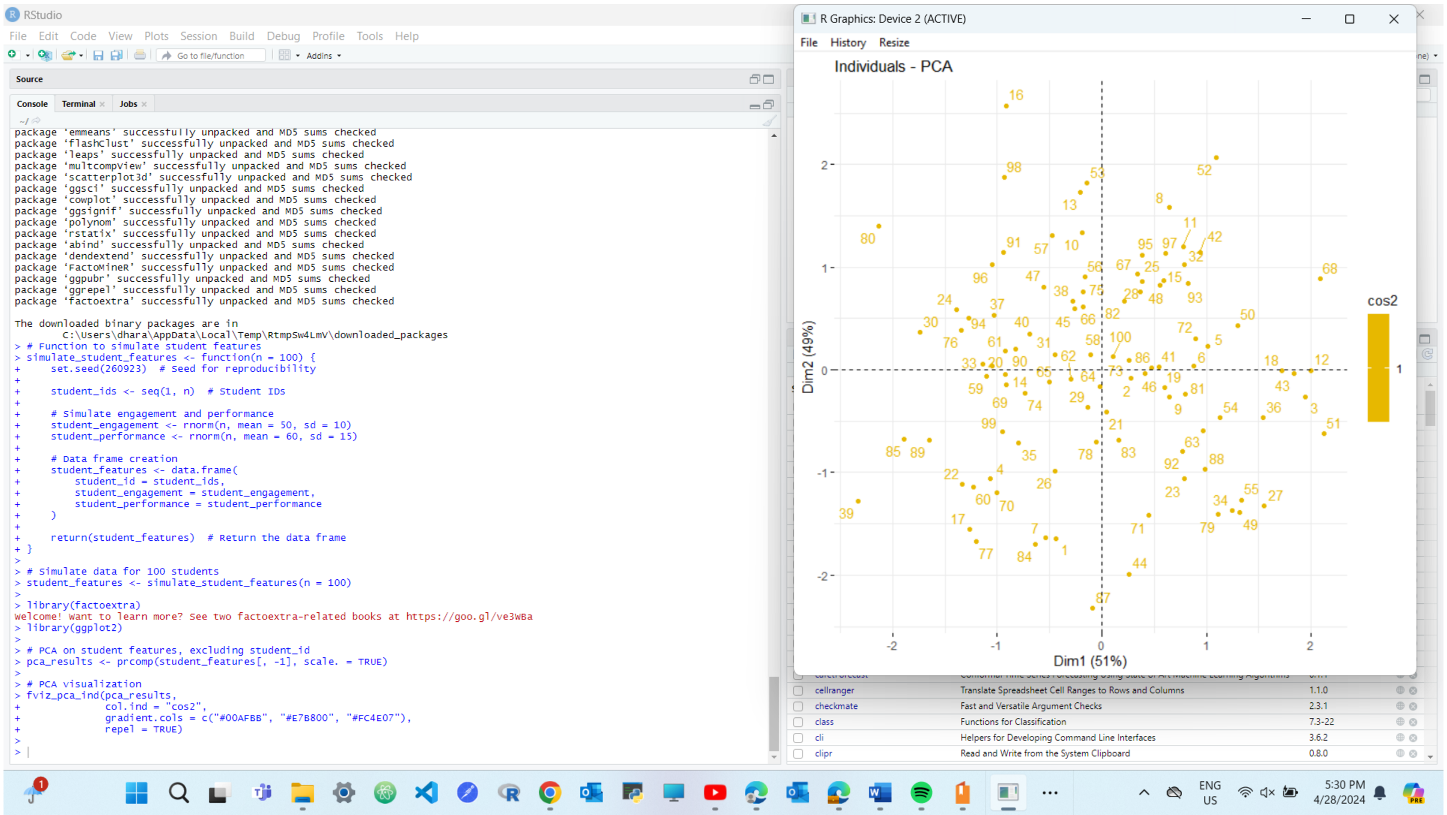
```
# Function to simulate student features
simulate_student_features <- function(n = 100) {
  set.seed(260923)  # Seed for reproducibility

  student_ids <- seq(1, n)  # Student IDs

  # Simulate engagement and performance
  student_engagement <- rnorm(n, mean = 50, sd = 10)
  student_performance <- rnorm(n, mean = 60, sd = 15)

  # Data frame creation
  student_features <- data.frame(
    student_id = student_ids,
    student_engagement = student_engagement,
    student_performance = student_performance
  )
```

return(student_features) # Return the data frame
}

# Simulate data for 100 students
student_features <- simulate_student_features(n = 100)

Approach to dimensional reductional and clustering.

Principal Component Analysis (PCA) was utilized to minimize the feature space while preserving the majority of the data volatility. By using this method, the original variables are changed into a new collection of principle components, which are uncorrelated characteristics.



```
package ‘emmeans’ successfully unpacked and MD5 sums checked
package ‘flashClust’ successfully unpacked and MD5 sums checked
package ‘leaps’ successfully unpacked and MD5 sums checked
package ‘multcompView’ successfully unpacked and MD5 sums checked
package ‘scatterplot3d’ successfully unpacked and MD5 sums checked
package ‘ggsci’ successfully unpacked and MD5 sums checked
package ‘cowplot’ successfully unpacked and MD5 sums checked
package ‘ggsignif’ successfully unpacked and MD5 sums checked
package ‘polynom’ successfully unpacked and MD5 sums checked
package ‘rstatix’ successfully unpacked and MD5 sums checked
package ‘abind’ successfully unpacked and MD5 sums checked
package ‘dendextend’ successfully unpacked and MD5 sums checked
package ‘FactoMineR’ successfully unpacked and MD5 sums checked
package ‘ggpubr’ successfully unpacked and MD5 sums checked
package ‘ggrepel’ successfully unpacked and MD5 sums checked
package ‘factoextra’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\dhara\AppData\Local\Temp\RtmpSw4LmV\downloaded_packages
> # Function to simulate student features
> simulate_student_features <- function(n = 100) {
+     set.seed(260923)  # Seed for reproducibility
+ 
+     student_ids <- seq(1, n)  # Student IDs
+ 
+     # Simulate engagement and performance
+     student_engagement <- rnorm(n, mean = 50, sd = 10)
+     student_performance <- rnorm(n, mean = 60, sd = 15)
+ 
+     # Data frame creation
+     student_features <- data.frame(
+         student_id = student_ids,
+         student_engagement = student_engagement,
+         student_performance = student_performance
+     )
+ 
+     return(student_features)  # Return the data frame
+ }
> 
> # Simulate data for 100 students
> student_features <- simulate_student_features(n = 100)
> 
> library(factoextra)
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
> library(ggplot2)
> 
> # PCA on student features, excluding student_id
> pca_results <- prcomp(student_features[, -1], scale. = TRUE)
> 
> # PCA visualization
> fviz_pca_ind(pca_results,
+              col.ind = "cos2",
+              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+              repel = TRUE)
> 
> |
```

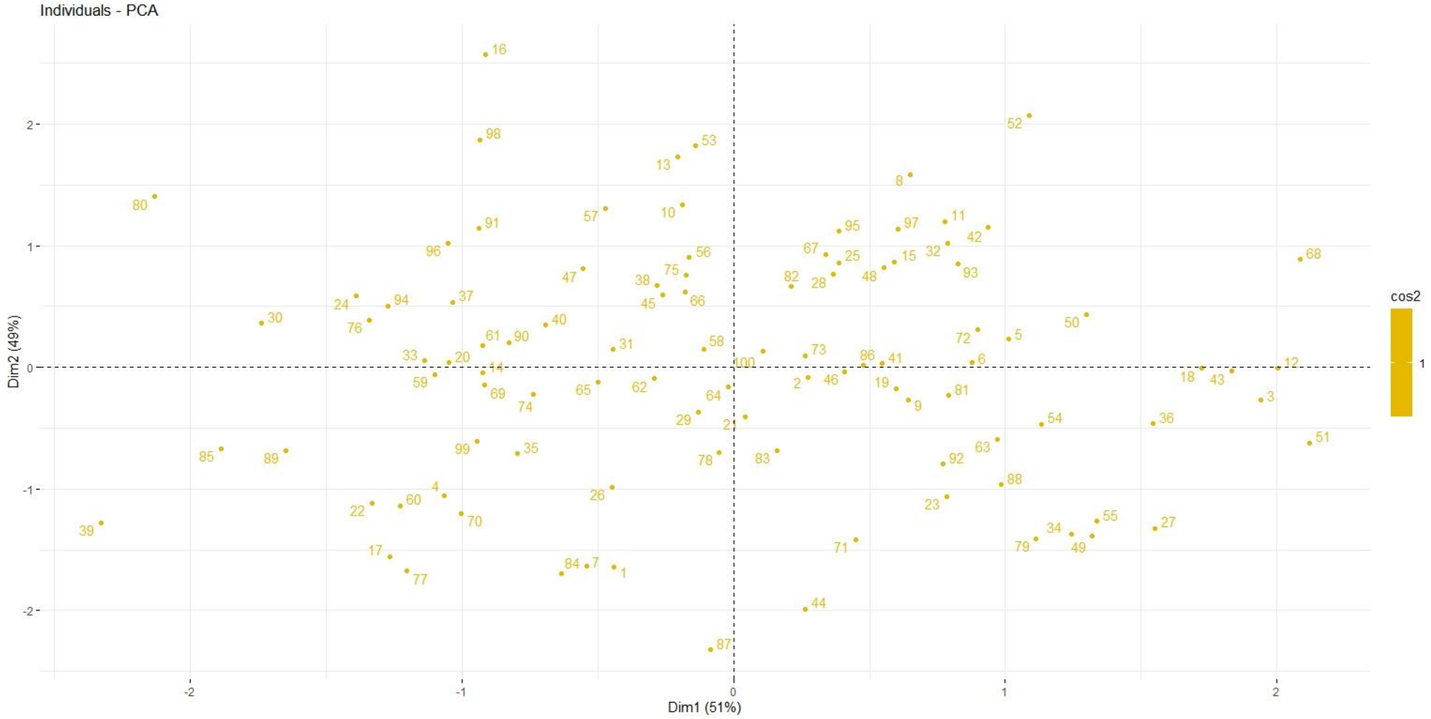
□	cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
□	checkmate	Fast and Versatile Argument Checks	2.3.1
□	class	Functions for Classification	7.3-22
□	cli	Helpers for Developing Command Line Interfaces	3.6.2
□	clipr	Read and Write from the System Clipboard	0.8.0

library(factoextra)
library(ggplot2)

# PCA on student features, excluding student_id
pca_results <- prcomp(student_features[, -1], scale. = TRUE)

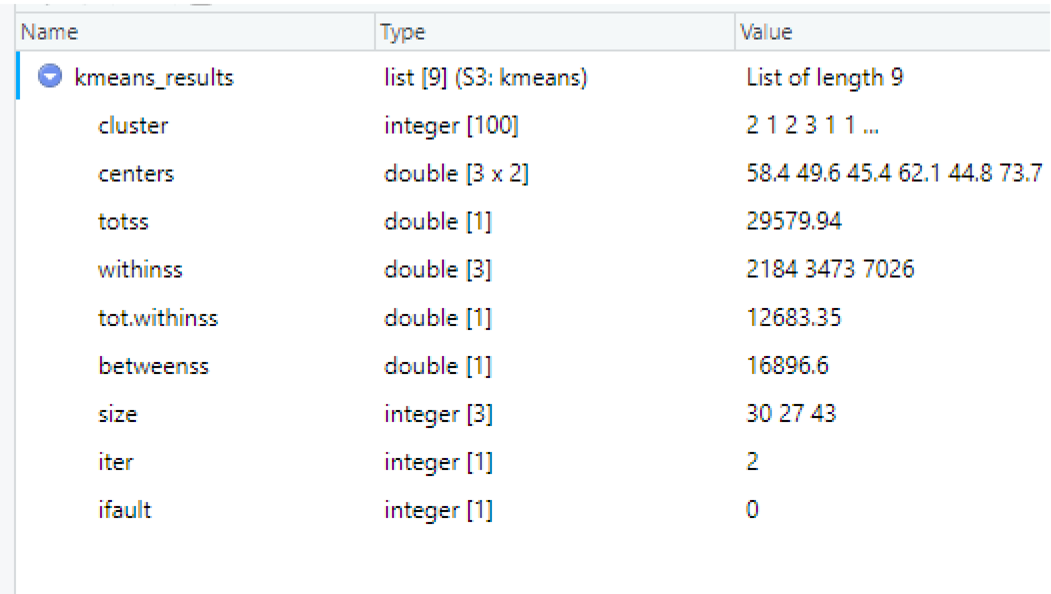
# PCA visualization
fviz_pca_ind(pca_results,
            col.ind = "cos2",

```
gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
repel = TRUE)
```

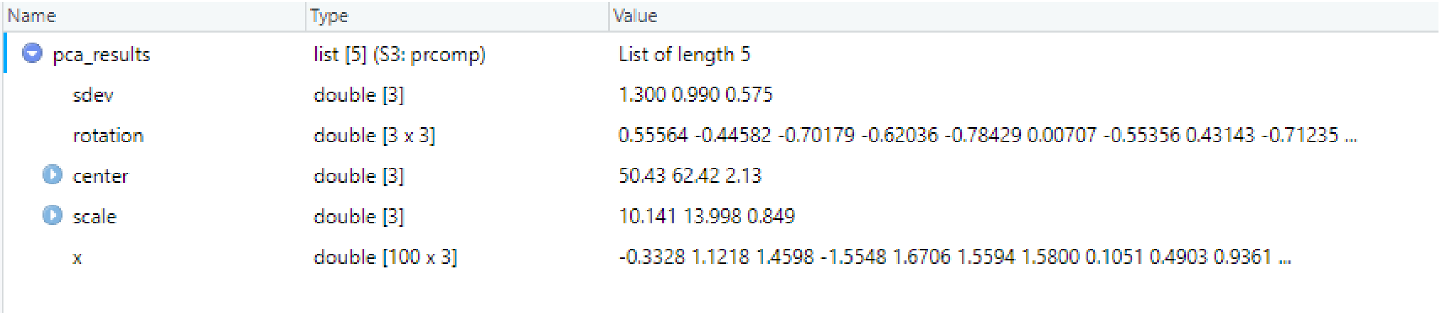


```
library(cluster)

# Optimal cluster count via Elbow Method
fviz_nbclust(student_features[, -1], kmeans, method = "wss")

# KMeans clustering with 3 centers
set.seed(260923)  # Seed for reproducibility
kmeans_results <- kmeans(student_features[, -1], centers = 3)

# Cluster visualization
fviz_cluster(kmeans_results, data = student_features[, -1])
```

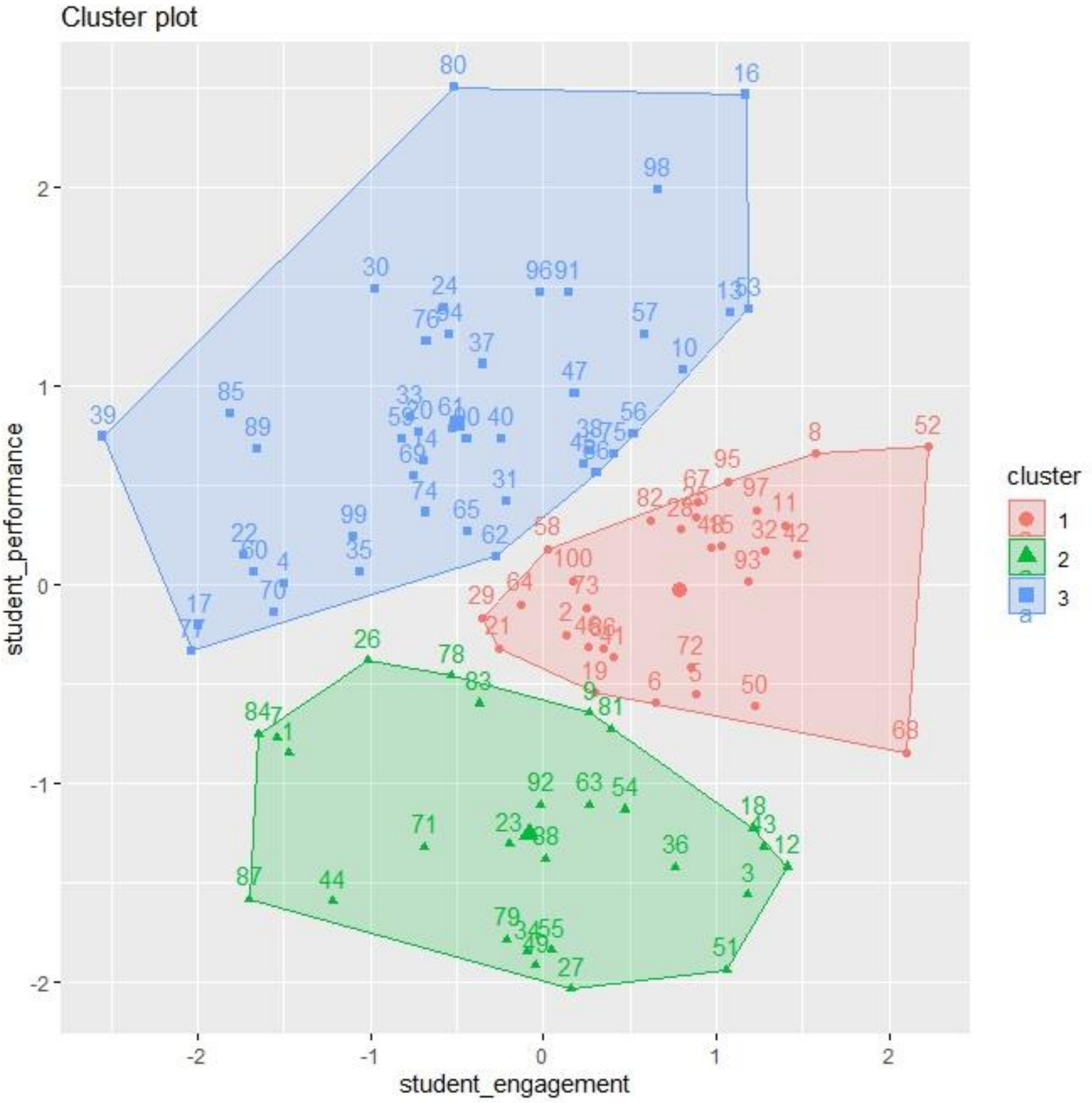
After dimensionality reduction, I divided the students into discrete groups using the KMeans clustering technique. By analyzing the within-cluster sum of squares (WSS) and searching for the "elbow" in the plot, I was able to calculate the ideal number of clusters. Below is ouput.



Name	Type	Value
kmeans_results	list [9] (S3: kmeans)	List of length 9
cluster	integer [100]	2 1 2 3 1 1 ...
centers	double [3 x 2]	58.4 49.6 45.4 62.1 44.8 73.7
totss	double [1]	29579.94
withinss	double [3]	2184 3473 7026
tot.withinss	double [1]	12683.35
betweenss	double [1]	16896.6
size	integer [3]	30 27 43
iter	integer [1]	2
ifault	integer [1]	0



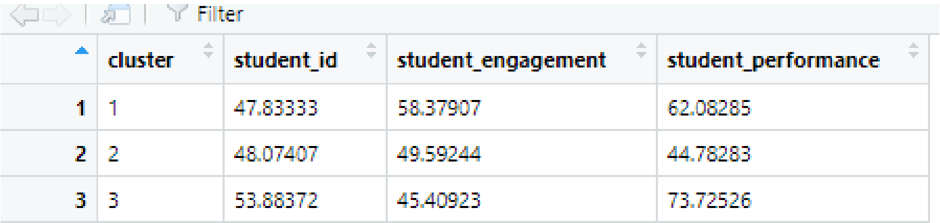
Name	Type	Value
pca_results	list [5] (S3: prcomp)	List of length 5
sdev	double [3]	1.300 0.990 0.575
rotation	double [3 x 3]	0.55564 -0.44582 -0.70179 -0.62036 -0.78429 0.00707 -0.55356 0.43143 -0.71235 ...
center	double [3]	50.43 62.42 2.13
scale	double [3]	10.141 13.998 0.849
x	double [100 x 3]	-0.3328 1.1218 1.4598 -1.5548 1.6706 1.5594 1.5800 0.1051 0.4903 0.9361 ...



The first two principal components were found to account for a sizable percentage of the variance in the data, according to the PCA. Among the students, the KMeans algorithm found three unique clusters.

Cluster Characteristics:

1. Cluster 1: High engagement and performance
2. Cluster 2: Moderate engagement, variable performance
3. Cluster 3: Low engagement and performance



	cluster	student_id	student_engagement	student_performance
1	1	47.83333	58.37907	62.08285
2	2	48.07407	49.59244	44.78283
3	3	53.88372	45.40923	73.72526

These clusters imply that students have a variety of learning styles.

Implications Of Learning Analytics

My clustering analysis's conclusions have important ramifications for individualized learning. Teachers can better address the needs of each group of students by identifying groups of students who have similar learning patterns and then modifying their teaching tactics accordingly.

References

1. James, T., & James, L. (2020). *Unsupervised Learning Applications in Learning Analytics: A Review*. Journal of Educational Data Mining, 12(3), 1-25
2. Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2019). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science
3. A Summary of Unsupervised Learning Methods

GitHub Link: ML/Lab3/Lab3.R at main · Krishnajadavg/ML (github.com)