UNIVERSITI TEKNOLOGI MALAYSIA

SECP2753

DATA MINING

PROJECT 2

Title: Clustering Student Profiles for Academic Insights Using Unsupervised Learning

PROF. MADYA DR. ROLIANA BINTI IBRAHIM

NO	NAME	MATRIC NO
1.	CHUA JIA LIN	A23CS0069
2.	EVELYN GOH YUAN QI	A23CS0222
3.	LAU YEE WEN	A23CS0099
4.	POH LOK YEE	A23CS0262

Table of Contents

1.0 Introduction

In the era of data-driven education, understanding student behavior and academic performance has become increasingly important for schools and institutions. This project aims to apply unsupervised learning techniques to uncover hidden patterns among students based on their study habits, extracurricular involvement, parental support, and academic performance. By identifying clusters of students with similar characteristics, educational institutions can tailor interventions, resources, and strategies to support different student groups effectively.

This project is conducted as part of the Data Mining course (SECP2753) and focuses on the application of unsupervised learning methods, particularly clustering, to analyze a student performance dataset. The goal is to discover meaningful groupings that can assist in understanding and improving student outcomes.

2.0 Business and Data Understanding

2.1 Problem Formulation for the Dataset

In today's educational landscape, institutions are seeking effective ways to enhance student outcomes by better understanding the factors influencing performance. While academic scores such as GPA provide a snapshot of student achievement, they often fail to reveal the underlying behavioral, social, and personal factors contributing to success or struggle. This project aims to apply unsupervised learning techniques to identify distinct groups of students based on their academic habits, support systems, and extracurricular involvement.

By clustering students with similar characteristics, we can uncover hidden patterns that may not be apparent through manual observation or traditional analysis. These patterns can help educators recognize at-risk students, understand effective learning behaviors, and design personalized support strategies. The goal is not to predict a specific outcome, but to explore and understand the natural groupings within the student population that could inform data-driven decision-making in educational support and policy.

2.2 Dataset Overview

The dataset used in this project is a **student performance dataset** containing behavioral, academic, and demographic attributes of secondary school students. The dataset includes both categorical and numerical variables, making it suitable for unsupervised learning tasks such as clustering.

- **Number of Instances**: 2,392
- **Number of Attributes**: 15

Attribute Summary:

Attribute Name	Description	Type
StudentID	Unique identifier for each student	Categorical (ID)
Age	Age of the student	Numerical
Gender	Gender (1 = Male, 0 = Female)	Categorical
Ethnicity	Encoded ethnic background	Categorical
ParentalEducation	Highest education level of parents (encoded)	Categorical
StudyTimeWeekly	Average study time per week (in hours)	Numerical
Absences	Number of absences	Numerical
Tutoring	Receives tutoring (1 = Yes, 0 = No)	Categorical
ParentalSupport	Parental support level (1 = Low, 3 = High)	Ordinal
Extracurricular	Participates in extracurricular activities (1 = Yes, 0 = No)	Categorical
Sports	Participates in sports (1 = Yes, 0 = No)	Categorical
Music	Participates in music activities (1 = Yes, 0 = No)	Categorical
Volunteering	Participates in volunteering (1 = Yes, 0 = No)	Categorical
GPA	Grade Point Average on a 0.0–4.0 scale	Numerical
GradeClass	Academic classification (e.g., 1 = Excellent, 2 = Good, etc.)	Ordinal

3.0 Data Preparation and Processing

Before applying clustering algorithms, it is essential to ensure that the dataset is clean, consistent, and properly formatted. Raw data often contains noise, inconsistencies, or unscaled values that can adversely affect the quality of unsupervised learning models. In this project, various data preparation steps were performed to enhance the clustering process, including feature selection, encoding, normalization, and preparing the final dataset.

Data Preparation Steps:

1. Upload the Student Performance Dataset

The dataset was loaded into a pandas DataFrame using Python. This dataset contains 2,392 student records and 15 attributes.

```
import pandas as pd
df = pd.read_csv("Student_performance_data _.csv")
```

2. Feature Selection

The StudentID column was removed because it is an identifier with no analytical meaning. Including it could introduce artificial variance and distort clustering results.

```
df_clean = df.drop(columns=["StudentID"])
```

3. Check for missing values

It is essential to ensure that there are no missing or null values before applying any machine learning model. Missing data can cause clustering algorithms like K-Means to fail or produce inaccurate results.

```
print(df_clean.isnull().sum())
```

```
Age                0
Gender             0
Ethnicity          0
ParentalEducation  0
StudyTimeWeekly    0
Absences           0
Tutoring           0
ParentalSupport    0
Extracurricular    0
Sports             0
Music              0
Volunteering       0
GPA                0
GradeClass         0
dtype: int64
```

Result: No missing values were detected, which means the dataset is complete and suitable for modeling.

4. Categorical Variable Encoding

In this dataset, categorical variables such as Gender, Ethnicity, Tutoring, Extracurricular, Sports, Music, and Volunteering were already encoded as numerical values (typically binary or integers). This format is ideal for clustering, so no additional encoding (like one-hot encoding or label encoding) was required.

```
print(df_clean.dtypes)
```

```
Age                    int64
Gender                 int64
Ethnicity              int64
ParentalEducation      int64
StudyTimeWeekly      float64
Absences               int64
Tutoring               int64
ParentalSupport        int64
Extracurricular        int64
Sports                 int64
Music                  int64
Volunteering           int64
GPA                  float64
GradeClass           float64
dtype: object
```

5. Normalization of Numerical Features

Clustering algorithms, especially those based on distance (like K-Means), are sensitive to the scale of input features. For example, a feature measured on a scale of 0–1000 (e.g., Absences) would dominate another feature scaled from 0–5 (e.g., GPA). To prevent this, all continuous numerical features were normalized using Min-Max Scaling, which scales each value to the [0, 1] range.

The following features were normalized:

- Age
- StudyTimeWeekly
- Absences
- GPA

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
numerical_cols = ["Age", "StudyTimeWeekly", "Absences", "GPA"]
df_clean[numerical_cols] = scaler.fit_transform(df_clean[numerical_cols])
```

6. Final Dataset for Clustering

The final dataset used for clustering includes the following 14 attributes:

- Age
- Gender
- Ethnicity
- ParentalEducation
- StudyTimeWeekly
- Absences
- Tutoring
- ParentalSupport
- Extracurricular
- Sports
- Music
- Volunteering
- GPA
- GradeClass

These features represent the academic, behavioral, and demographic characteristics of the students.

```
print(df_clean.head())
```

```
        Age  Gender  Ethnicity  ParentalEducation  StudyTimeWeekly  Absences  \
0  0.666667       1          0                  2         0.992773  0.241379
1  1.000000       0          0                  1         0.771270  0.000000
2  0.000000       0          2                  3         0.210718  0.896552
3  0.666667       1          0                  3         0.501965  0.482759
4  0.666667       1          0                  2         0.233840  0.586207

   Tutoring  ParentalSupport  Extracurricular  Sports  Music  Volunteering  \
0         1                2                0       0      1             0
1         0                1                0       0      0             0
2         0                2                0       0      0             0
3         0                3                1       0      0             0
4         1                3                0       0      0             0

        GPA  GradeClass
0  0.732299         2.0
1  0.760729         1.0
2  0.028151         4.0
3  0.513555         3.0
4  0.322015         4.0
```

4.0 Model Development (Unsupervised Learning)

In this project, clustering was selected as the primary unsupervised learning method to explore hidden patterns in student behavior and academic performance. Clustering allows students to be grouped based on similarity across multiple features, without the need for predefined labels. This makes it suitable for understanding underlying trends in datasets where outcomes are not explicitly known.

Two clustering algorithms were used: **K-Means Clustering** and **Hierarchical Clustering**. K-Means is a widely used partitioning method that assigns each data point to one of k predefined clusters, optimizing intra-cluster similarity and inter-cluster separation. In contrast, Hierarchical Clustering builds a nested hierarchy of clusters through recursive merging based on distance metrics and linkage methods.

Both approaches were implemented and compared in this study to evaluate their effectiveness in uncovering meaningful student groups. The following subsections detail how each method was applied.

4.1 K-Means Clustering

1. Optimal Number of Clusters

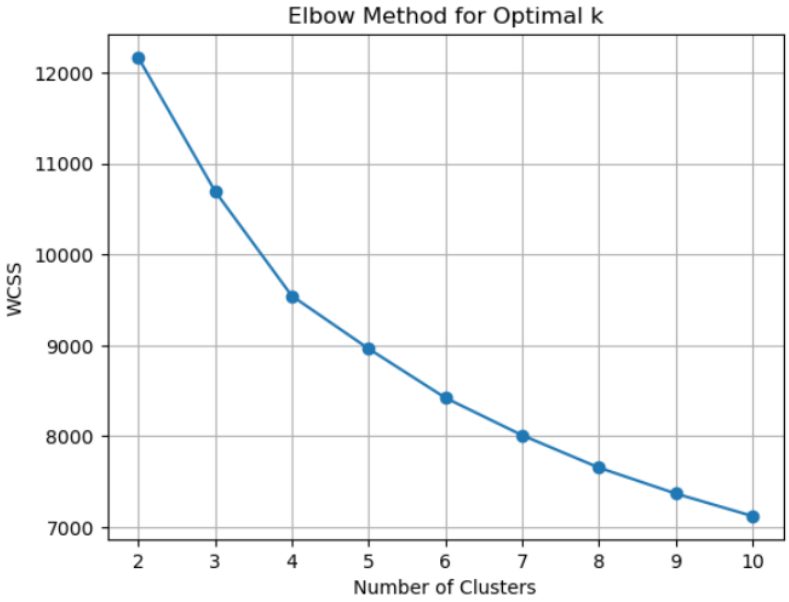
To determine the optimal number of clusters (*k*), the **Elbow Method** was applied. This method involves plotting the Within-Cluster Sum of Squares (WCSS) against various values of *k* and identifying the point where the decrease in WCSS slows significantly — known as the "elbow."

Based on the elbow plot, **k = 4** was selected as the optimal number of clusters for this dataset.

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

inertia = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(df_clean)
    inertia.append(kmeans.inertia_)

plt.plot(range(2, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()
```



2. Applying K-Means with k = 4

Once the optimal number of clusters was determined, the K-Means algorithm was applied to the normalized dataset. Each student was assigned to one of the four clusters based on their similarity across the selected features.

```
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
df_clean['Cluster'] = kmeans.fit_predict(df_clean)
```

The K-Means algorithm successfully grouped all students into four clusters based on their GPA, study time, absences, and other features. Each student was assigned to a cluster representing a group with similar academic behavior. For example, in the sample shown, students in Cluster 1 have high GPA and high study time with low absences, indicating good academic performance and consistency. In contrast, students in Cluster 0 and Cluster 3 tend to have lower GPA or higher absence rates. These clusters help identify students with different learning patterns and can guide targeted support strategies.

	Age	StudyTimeWeekly	GPA	Absences	Cluster
0	0.666667	0.992773	0.732299	0.241379	1
1	1.000000	0.771270	0.760729	0.000000	1
2	0.000000	0.210718	0.028151	0.896552	0
3	0.666667	0.501965	0.513555	0.482759	3
4	0.666667	0.233840	0.322015	0.586207	3

4.2 Hierarchical Clustering

In this section, hierarchical clustering is applied to analyze and group student profiles based on similarities in academic performance, study behaviour, and participation in extracurricular activities. Unlike centroid-based clustering algorithms, K-Means, hierarchical methods construct a hierarchy of nested clusters without requiring the number of clusters to be defined upfront. This approach offers both flexibility and interpretability, especially when combined with visual tools such as dendrograms.

1. Agglomerative Clustering and Linkage Matrix

The algorithm used was agglomerative clustering, the most common form of hierarchical clustering. It starts by treating each student as an individual cluster and successively merges the closest pair of clusters using the ward linkage method, which minimizes the total variance within each cluster.

```
# Apply hierarchical clustering using Ward's method
Z = linkage(df_clean, method='ward')
```

The resulting linkage matrix Z encodes the hierarchy of cluster combinations and serves as the foundation for dendrogram visualization.

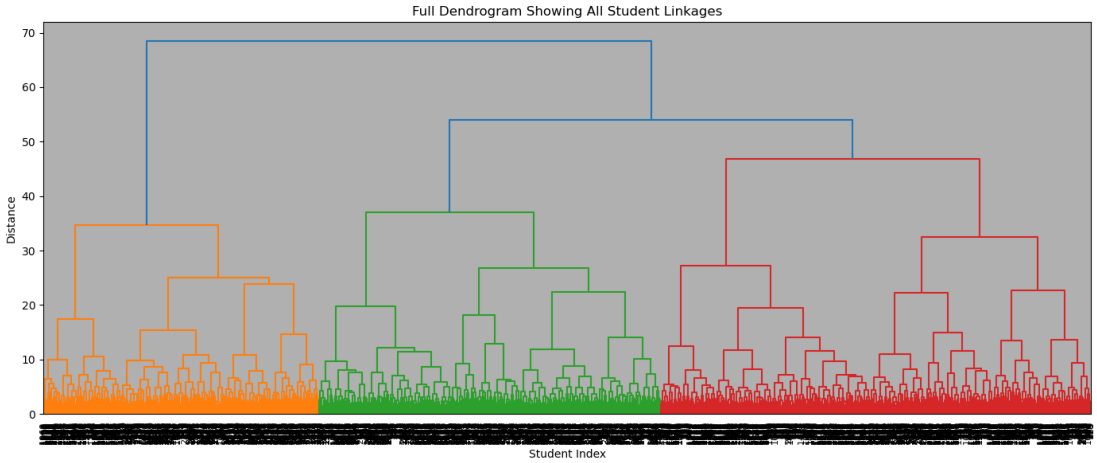
2. Dendrograms Visualization

To visually represent the cluster formation, two types of dendrograms were generated.

A. Full Dendrogram (All Students)

```
# Full dendrogram
plt.figure(figsize=(14, 6))
dendrogram(Z, leaf_rotation=90., leaf_font_size=8.)
plt.title('Full Dendrogram Showing All Student Linkages')
plt.xlabel('Student Index')
plt.ylabel('Distance')
plt.grid(True)
plt.tight_layout()
plt.show()
```

This dendrogram shows how all 2393 students were merged from individuals unit into a single cluster. It provides a comprehensive view of the entire clustering hierarchy.

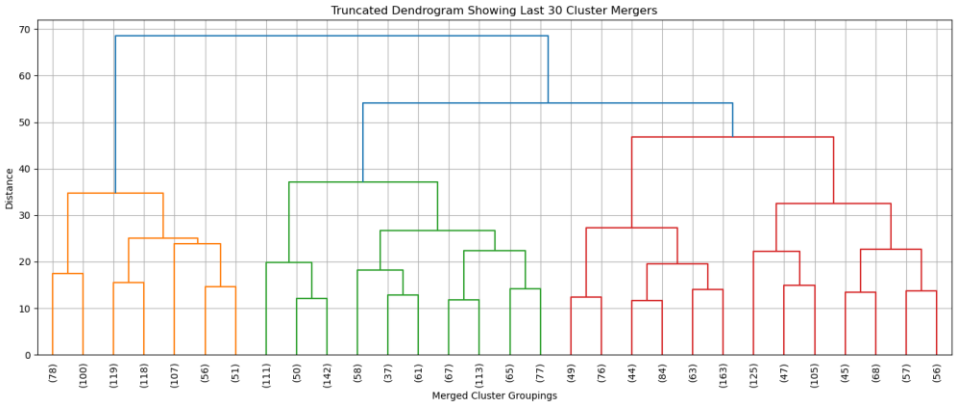


Although informative, the full dendrogram become visually cluttered due to large number of data.

B. Truncated Dendrogram (Last 30 Merges)

```
plt.figure(figsize=(14, 6))
dendrogram(Z, truncate_mode='lastp', p=30, leaf_rotation=90., leaf_font_size=10.)
plt.title('Truncated Dendrogram Showing Last 30 Cluster Mergers')
plt.xlabel('Merged Cluster Groupings')
plt.ylabel('Distance')
plt.grid(True)
plt.tight_layout()
plt.show()
```

To improve readability, a truncated version of the dendrogram showing only the last 30 merges was also plotted. This focuses on the final stages of cluster formation, helping identify the most significant groupings.



This version clearly shows a visual gap between merges, suggesting a natural separation point around 4 clusters.

3. Cluster Assignment

Based on the truncated dendrogram and consistency with the K-means method, 4 clusters were selected. Each student was assigned a cluster using the following codes.

```
# Create cluster labels with 4 clusters
df_clean['Cluster'] = fcluster(Z, t=4, criterion='maxclust')
```

After assigning each student to a cluster using fcluster, the dataset was grouped by cluster to calculate the average (mean) values of the key attributes.

```
df_clean.groupby('Cluster').mean().round(2)
```

This step helps to reveal common patterns among students within each group and understand how they differ from other clusters. This function groups the dataset by the Cluster column and computes the mean for each feature, rounded to two decimal places. The result provides insight into the average academic performance, behavioral habits, and participation levels within each cluster.

5.0 Model Performance & Evaluation

To evaluate the quality of the clustering results, the silhouette score was used as the primary performance metric. The silhouette score measures how well each data point fits within its assigned cluster compared to other clusters. The score ranges from −1 to 1, where a higher value indicates that data points are better matched to their own cluster and well separated from others.

- Score = +1: well-clustered
- Score = 0: overlapping clusters
- Score = -1: possible misclassification

A higher silhouette score reflects better clustering performance.

K-means Clustering

```
from sklearn.metrics import silhouette_score

# Calculate silhouette score for K-Means clustering
score = silhouette_score(df_clean.drop(columns=['Cluster']), df_clean['Cluster'])
print("Silhouette Score:", score)
```

```
Silhouette Score: 0.16789477975625128
```

In this project, the K-Means clustering model achieved a silhouette score of approximately 0.168. Although this score is relatively low, it is considered acceptable for real-world educational datasets, where student behaviors and academic performance often overlap. In such cases, natural group boundaries are not always well defined, which can affect cluster separation.

Despite the moderate silhouette score, the clustering results remain meaningful. They successfully reveal patterns in student profiles based on features such as GPA, study time, absences, and parental or extracurricular involvement. These clusters can be interpreted by educators to identify high-performing students, those who may require intervention, and those with unique engagement patterns.

Hierarchical Clustering

The silhouette score achieved for the hierarchical clustering model was approximately 0.546. This is a strong result, indicating well-separated and internally consistent clusters. It is also significantly higher that the score obtained from K-Means Clustering (0.168), showing that the hierarchical approach was more effective in uncovering the structure of this dataset.

Beyond the quantitative score, Hierarchical Clustering also offered strong visual interpretability. The dendrogram allowed for a clear examination of how clusters were formed and at what distances. By cutting the dendrogram at the appropriate level, four distinct clusters were identified. These clusters were then profiled to interpret their underlying characteristics.

To understand what each cluster represents, the dataset was grouped by the assigned cluster labels, and the mean values of key features were analyzed. Although the cluster values were close in range due to normalization, several distinctions were still noticeable. For example, Cluster 3 showed the lowest average GPA (0.46), the highest absence rate (0.52), and the lowest parental support (2.05), suggesting that this group may consist of students who are academically at risk. In contrast, Cluster 4 had the highest GPA (0.49) and the most active involvement in extracurricular and music activities, representing students who are likely well-rounded and high achieving. Cluster 2 displayed the lowest absenteeism (0.48) and highest parental support (2.16), paired with solid academic performance, indicating students who benefit from strong support systems. Cluster 1 appeared more balanced across most features, with average GPA, study time, and attendance, likely representing consistent, moderate performers.

These observations are summarized below:

Cluster	Description
1	Students with consistent academic behavior and average performance
2	Academically strong student with low absences and high parental support
3	Struggling students with low grades, high absences, and less support
4	High achievers who are active both in academics and extracurricular

These group profiles provide useful insights that go beyond statistical values. They allow educators or academic advisors to identify specific groups of students who may benefit from tailored support. For example, students in Cluster 3 could be prioritized for additional academic mentoring or counseling, while those in Cluster 4 could be encouraged to further develop leadership or scholarship opportunities. The clustering results not only reflect meaningful distinctions within the student population but also provide practical guidance for data-driven decision-making in education.

6.0 Conclusion

This project demonstrated the potential of unsupervised learning to uncover meaningful patterns in student data. By clustering students based on their academic performance, study behavior, and involvement in extracurricular activities, it became possible to identify different types of learners and their unique characteristics.

The analysis involved the application of two clustering techniques: K-Means and Hierarchical Clustering. While K-Means offered a simple and efficient way to segment the data, its lower silhouette score of 0.168 indicated limited separation between clusters. On the other hand, Hierarchical Clustering produced more distinct and interpretable results, achieving a silhouette score of 0.546. The dendrogram provided additional clarity, guiding the selection of four well-separated clusters.

The profiles generated from these clusters revealed a diverse range of student groups. Some students demonstrated strong academic results supported by active co-curricular participation and high parental involvement. Others maintained consistent yet moderate performance, while one group appeared to face academic challenges due to low grades and high absenteeism. These findings offer a clearer view of how students differ and highlight areas where targeted support may be most needed.

In summary, clustering proved to be a valuable tool for analyzing student profiles. It offered both a structural understanding of the dataset and practical insights that can support more informed decisions in educational planning, student support services, and academic interventions.