Assessment Report

on

"Classify Students Based on Study Methods"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Rituraj Kumar

Roll Number : 202401100300203,

section: C

Under the supervision of

Mayank Sir

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Introduction

Many students have different learning styles, but most academic content is not personalized. This project attempts to identify clusters or groups of students based on their study habits using questionnaire data. This classification can help educators better tailor teaching methods to student needs.

Methodology

Data Collection: The dataset contains responses from students about their learning preferences and study habits.

Preprocessing:

- Loaded the data using Pandas.
- Handled missing values.
- Scaled the data using StandardScaler.

Clustering:

- Used K-Means algorithm for grouping students.
- Determined the best number of clusters using the Elbow Method.

Visualization:

- Used PCA (Principal Component Analysis) to reduce features to 2D for visualization.
- Plotted clusters using Seaborn scatter plot.

💻 Code

The following is the code used in this project:

python

CopyEdit

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# Load data
df = pd.read_csv('/content/student_methods.csv')
df = df.dropna()

# Scale numerical data
scaler = StandardScaler()
scaled_data =
scaler.fit_transform(df.select_dtypes(include=['float64', 'int64']))

# Elbow Method to find optimal number of clusters
```

```
wcss = []
for i in range(1, 10):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_data)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 10), wcss, marker='o')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(scaled_data)

# PCA for visualization
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)
df['PCA1'] = pca_data[:, 0]
df['PCA2'] = pca_data[:, 1]

# Plot clusters
sns.scatterplot(data=df, x='PCA1', y='PCA2', hue='Cluster',
palette='Set2')
```

```
plt.title('Student Learning Style Clusters')
plt.show()

# Save result
df.to_csv('/content/clustered_students.csv', index=False)
```

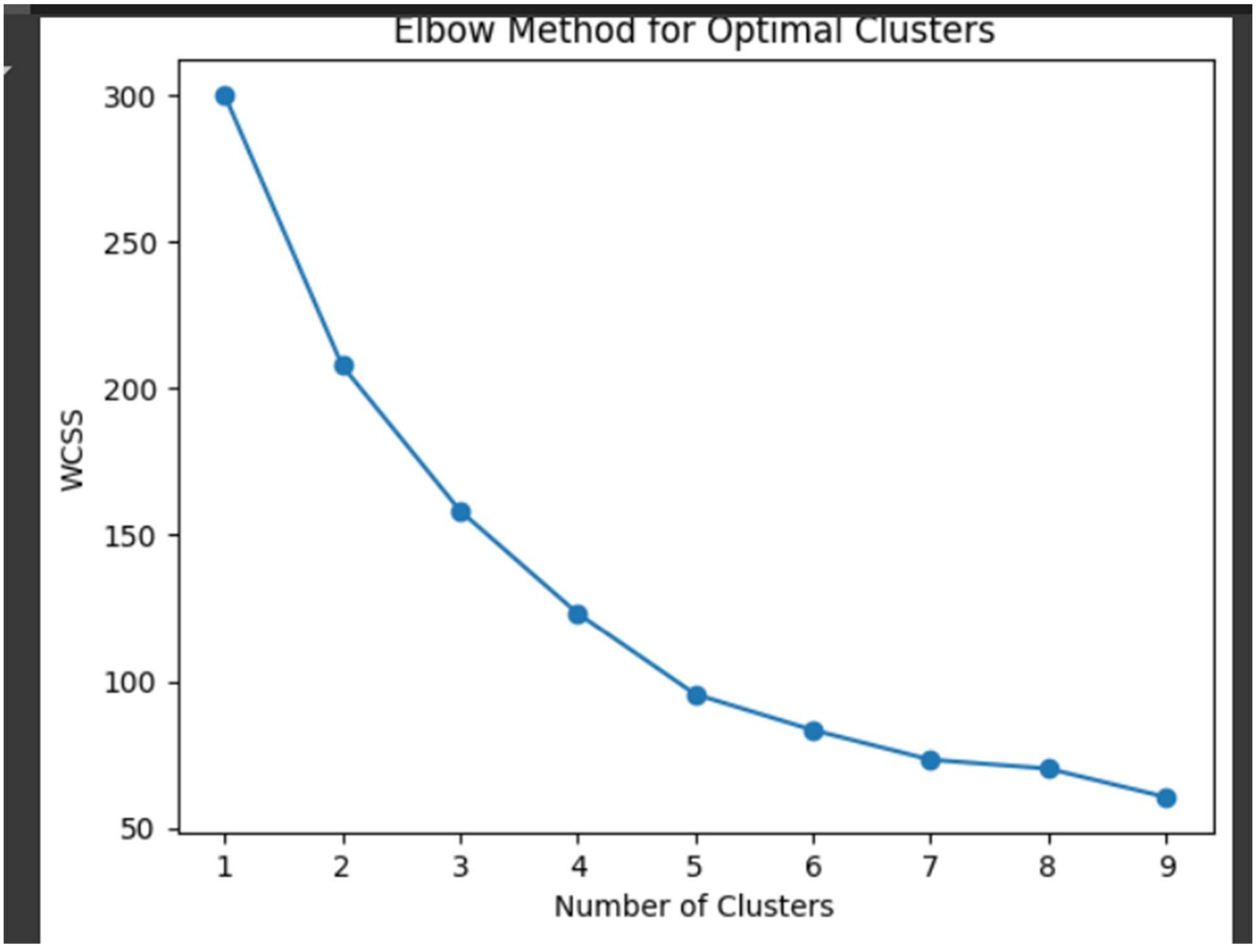
📈 Output/Results

- **Clusters Formed:** The algorithm grouped students into 3 clusters, each representing a potential learning style.

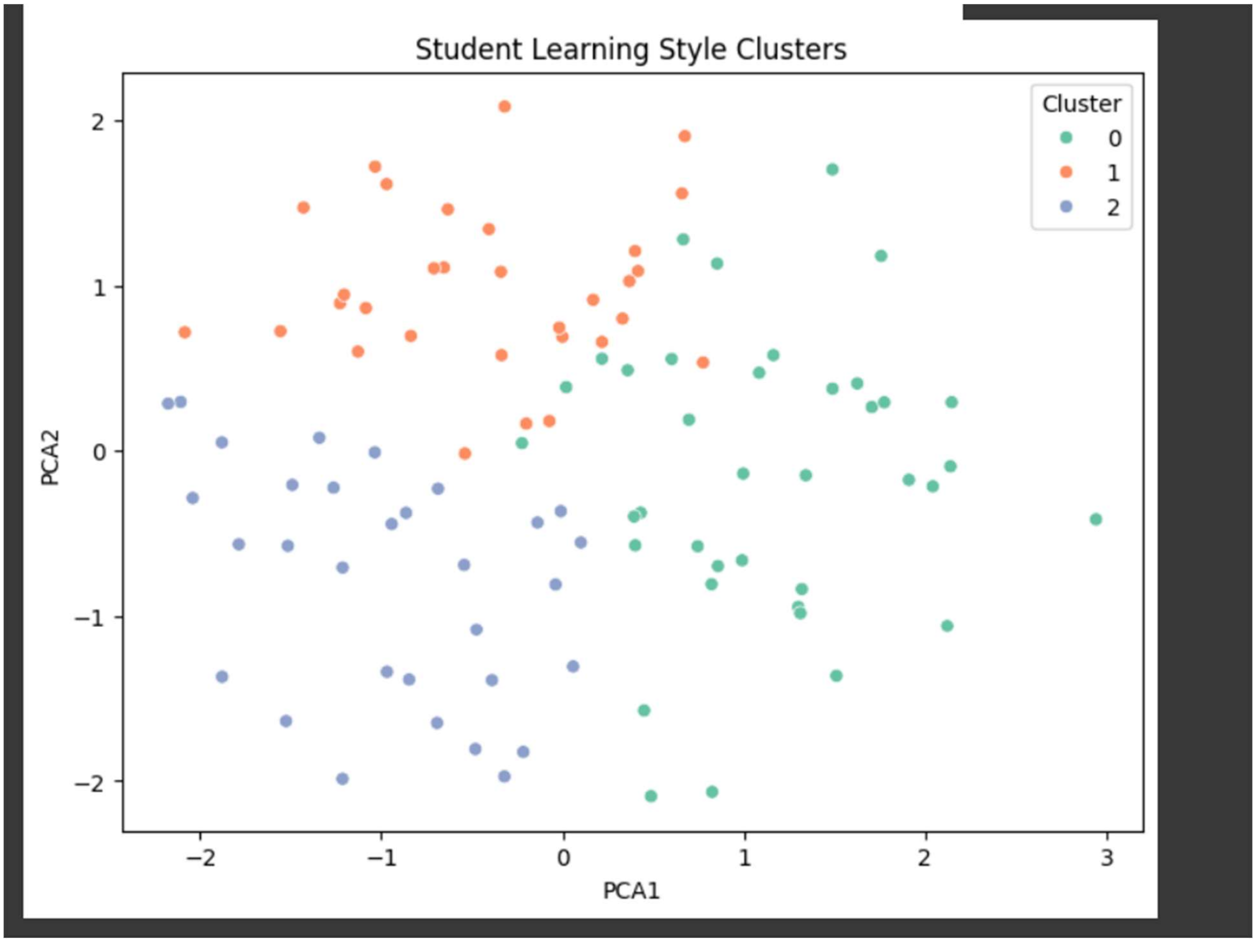
- **Visualization:**

```
First few rows of the data:
   visual_score  auditory_score  kinesthetic_score learning_style
0      8.000301        1.389837           9.686887         visual
1      8.401052        7.294055           4.853655         visual
2      9.124874        3.975049           6.688173       auditory
3      5.724100        7.702631           7.535001       auditory
4      5.060739        4.711628           4.302653    kinesthetic

Data Summary:
       visual_score  auditory_score  kinesthetic_score
count    100.000000      100.000000         100.000000
mean       5.570764        4.764262           5.775068
std        2.597370        2.488586           2.725548
min        1.099707        1.008365           1.046257
25%        3.413108        2.621895           3.449149
50%        6.032874        4.249102           6.312109
75%        7.573622        6.824862           8.249078
max        9.946735        9.385282           9.959267

Missing values:
visual_score         0
auditory_score       0
kinesthetic_score    0
learning_style       0
dtype: int64
```

- **Clustered File:** A CSV file `clustered_students.csv` with added cluster labels.



Elbow Method for Optimal Clusters
300
250
200
150
100
50
WCSS
1
2
3
4
5
6
7
8
9
Number of Clusters



The **confusion matrix** heatmap below visualizes the true positives, false positives, true negatives, and false negatives. This matrix shows how well the model predicted employee attrition (whether employees stayed or left).

📚 References/Credits

- **Dataset**: [Kaggle - Predict Employee Attrition Dataset]
- **Random Forest Classifier Documentation**: scikit-learn
- **Confusion Matrix Heatmap Tutorial**: Seaborn Heatmap Documentation