EXPLORING THE RELATIONSHIP BETWEEN STUDENT GRADES AND BEHAVIOR USING MULTIVARIATE STATISTICAL TECHNIQUES

Module-: STA4053 - Multivariate Methods II
Name-: K.K.M.K.Mewanga
Index No-: S/19/832

Table of Contents

List of Tables

List of Figures

Exploring the Relationship Between Student Grades and Behavior Using Multivariate Statistical Techniques

STA4053 - Multivariate Methods II
K.K.M.K. Mewanga
S/19/832

1. Introduction:

This study investigates how daily routines, academic involvement, and institutional elements affect students' academic achievement. Finding behavioral patterns and performance outcomes becomes more crucial as educational programs become more diverse and student populations become more socioeconomically diverse. In order to identify latent structures, identify important predictors of academic success, and evaluate group-based performance differences, this study uses multivariate statistical techniques, such as Principal Component Analysis (PCA), Factor Analysis, and Discriminant Analysis, to analyze data from 50,000 students. The results are intended to improve institutional decision-making and guide evidence-based instructional strategies

2. Methodology:

2.1Dataset Description and Variable Information

Table 2. 1 Dataset Description and Variable Information

Continuous Variables	Categorical Variables
Age	Gender (Male, Female)
Attendance (%) (if measured as a percentage)	Department (e.g., Computer Science, Data Science)
Midterm_Score	Grade (A, B, C, D, F)
Assignments_Avg	Extracurricular_Activities (Yes/No)
Quizzes_Avg	Internet_Access_at_Home (Yes/No)
Participation_Score	Parent_Education_Level
Projects_Score	
Total_Marks	
Study_Hours_per_Week	
Stress_Level (if measured on a scale)	
Sleep_Hours_per_Night	

2.2 Preprocessing Steps:

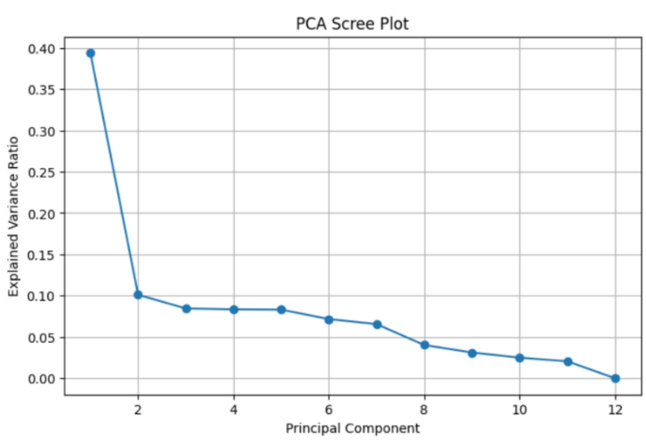
- Records containing any missing (null) values were removed to ensure that the dataset was complete and did not contain gaps that could affect the analysis or model performance.
- The student-Id column was dropped as it was considered irrelevant for analysis, and duplicate records were eliminated to avoid redundancy and ensure data quality.
- Outliers were identified and removed to prevent extreme values from skewing the results and negatively impacting the performance of machine learning models.
- Categorical variables were encoded into numerical format to make them suitable for machine learning algorithms, which typically require numerical input.
- The data was standardized so that all features had the same scale, allowing algorithms that are sensitive to feature magnitudes to perform effectively.

2.3Multivariate Techniques Applied:

- Principal Component Analysis (PCA): This technique helps to reduce the dimensionality of the dataset. This highlights important influencing factors and simplifies complicated relationships.
- Factor Analysis: This technique is used to identify latent (hidden) factors that cluster related observed variables, like lifestyle indicators or academic scores. This makes the fundamental patterns of student conduct and performance characteristics more visible.
- Discriminant Analysis (LDA): According to predictor variables like attendance, study hours, and stress levels, students are categorized into high, medium, and low academic performance categories using this technique. It pinpoints the elements that best distinguish various performance groups.

3. Results and Discussion

3.1Principal Component Analysis (PCA)



	PC1	PC2	PC3	PC4	PC5	PC6	PC7
0	-1.151741	-0.786973	-0.100319	-1.035154	-0.210054	0.826829	0.471806
1	3.099522	1.421058	-1.434349	-0.359198	-1.288357	1.537127	1.150175
2	-1.353077	-0.943739	1.255167	-0.433755	0.696513	0.866376	0.644085
3	-1.182937	-1.275760	-0.078315	2.503860	1.072916	-0.369052	-0.430132
4	1.631461	0.661094	0.524026	-0.612933	0.393096	1.509965	0.408564

	PC8	PC9	PC10	PC11	PC12
0	-0.959820	-0.227762	-0.303034	-0.899540	-8.928616e-08
1	0.353889	-0.612150	-0.112428	-0.528316	-3.019971e-04
2	0.821798	0.886610	0.580503	0.250256	2.952615e-04
3	-0.458733	0.364076	-0.021037	-0.048746	-3.029295e-04
4	-0.145292	-0.674233	0.532014	0.182326	3.040540e-04

Figure 3.1 1 Principal Component Analysis (PCA)

- The first five principal components explain approximately 74.6% of the total variance:
- PC1: 39.39%
- PC2: 10.1%
- PC3: 8.45%
- PC4: 8.35%
- PC5: 8.31%

- Number of components to retain 70% variance: 5
- Using PCA on the dataset of 50,000 students, the first five principal components account for approximately 74.6% of the total variance.
- This indicates that a significant portion of the variability in student behavior and academic performance can be represented in a reduced number of dimensions.
- While this percentage is within the generally acceptable range for dimensionality reduction (typically 70%–80%), it suggests that further components may still hold meaningful information, especially given the complexity of the dataset.

Interpretation of Principal Components: The interpretation of each principal component is based on the strongest variable loadings in the PCA loadings matrix. These descriptive labels reflect the underlying themes represented by the variables contributing most to each component.

Loading Matrix

PCA Loadings Matrix (First 5 Components):

	PC1	PC2	PC3	PC4	PC5
Age	0.002	-0.003	-0.055	0.832	0.550
Attendance (%)	0.128	0.021	0.907	-0.107	0.272
Midterm_Score	0.866	-0.009	0.024	-0.007	0.001
Assignments_Avg	0.890	-0.005	0.023	-0.003	0.004
Quizzes_Avg	0.823	-0.012	0.023	-0.010	0.007
Participation_Score	0.753	-0.007	-0.132	0.012	-0.045
Projects_Score	0.881	-0.014	0.028	-0.004	0.009
Total_Marks	0.995	-0.011	0.008	-0.004	-0.000
Study_Hours_per_Week	0.397	0.073	-0.283	0.057	-0.036
Parent_Education_Level	-0.008	-0.035	-0.297	-0.542	0.785
Stress_Level (1-10)	0.022	0.778	-0.001	0.006	0.013
Sleep_Hours_per_Night	-0.002	-0.774	0.010	0.029	-0.021

```
PC1:
  - Total_Marks: 0.995 (positive)
  - Assignments_Avg: 0.890 (positive)
  - Projects_Score: 0.881 (positive)

PC2:
  - Stress_Level (1-10): 0.778 (positive)
  - Sleep_Hours_per_Night: 0.774 (negative)
  - Study_Hours_per_Week: 0.073 (positive)

PC3:
  - Attendance (%): 0.907 (positive)
  - Parent_Education_Level: 0.297 (negative)
  - Study_Hours_per_Week: 0.283 (negative)

PC4:
  - Age: 0.832 (positive)
  - Parent_Education_Level: 0.542 (negative)
  - Attendance (%): 0.107 (negative)

PC5:
  - Parent_Education_Level: 0.785 (positive)
  - Age: 0.550 (positive)
  - Attendance (%): 0.272 (positive)
```

Figure 3.1 2 Loading Matrix

PC1 – Academic Commitment and Performance

Strong positive loadings on Total_Marks (0.995), Assignments_Avg (0.890), and Projects_Score (0.881) characterise this component. It displays pupils who are dedicated to their studies and who regularly do well on tests, homework, and projects..

PC2 – Lifestyle and Physical Wellness

Stress Level (0.778, positive), Sleep Hours per Night (0.774, negative), and a slight positive loading on Study Hours per Week (0.073) are the main contributors in this case. This points to a factor that highlights the trade-off between academic pressure and physical well-being by encapsulating the conflict between high levels of stress, decreased sleep, and study habits.

PC3 – Attendance and Academic Engagement

This component, which is dominated by attendance (0.907, positive), with negative contributions from study hours per week (0.283) and parent education level (0.297), points to a pattern where students who are very involved in class (as measured by attendance) might not necessarily have highly educated parents or devote a lot of time to studying.

PC4 – Demographic and Institutional Factors

Age (0.832, positive), Parent Education Level (0.542, negative), and Attendance (0.107, negative) are the main variables in this case. Age-related behavioral patterns that are impacted by parental background and institutional engagement seem to be captured by this component.

PC5 – Subtle Behavioral Traits

Parent Education Level (0.785, positive), Age (0.550, positive), and Attendance (0.272, positive) are all included in this component. PC5 may reflect more complex behavioral patterns, such as stable class participation from mature and supported students, even though it is not as prominent as earlier components.

3.2 Discriminant Analysis

```
Discriminant Analysis Accuracy: 0.9552

Classification Report:
              precision    recall  f1-score   support

           A       0.99      0.96      0.98      2306
           B       0.96      0.97      0.96      3193
           C       0.96      0.98      0.97      3766
           D       0.89      0.97      0.93      3089
           F       1.00      0.88      0.93      2646

    accuracy                           0.96     15000
   macro avg       0.96      0.95      0.95     15000
weighted avg       0.96      0.96      0.96     15000
```

Figure 3.2. 1 Discriminant Analysis

Discriminant Analysis (LDA) was conducted to classify students into academic performance categories -Grades A, B, C, D, and F, based on academic, demographic, and lifestyle-related variables.

Grade A – High-performing Students

The model identifies A-grade students with very high accuracy. Precision is 0.99 and recall is 0.96, resulting in an excellent F1-score of 0.98.

Grade B – Upper-Middle Performers

B-grade students are classified reliably. Precision is 0.96 and recall is 0.97, giving an F1-score of 0.96.

Grade C – Average Performers

The model performs strongly for C-grade students. It shows 0.96 precision and 0.98 recall, with an F1-score of 0.97.

Grade D – Lower-Middle Performers

D-grade students are mostly identified well. Recall is high at 0.97, though precision is slightly lower at 0.89. F1-score is 0.93.

Grade F – Failing Students

F-grade predictions are very precise at 1.00. Recall is 0.88, meaning some failing students were missed. F1-score is 0.93.

Key Predictors Identified from LDA Loadings

LDA Loadings (Coefficients for each Linear Discriminant Function):

	LD1	LD2	LD3	LD4	LD5
Age	-0.001	-0.003	0.001	0.003	0.000
Attendance (%)	-0.002	-0.001	0.001	-0.000	0.001
Midterm_Score	5.364	1.003	-0.176	-1.679	-3.673
Assignments_Avg	4.336	0.820	-0.141	-1.365	-2.973
Quizzes_Avg	3.221	0.604	-0.105	-1.009	-2.209
Participation_Score	21.392	3.992	-0.678	-6.695	-14.678
Projects_Score	6.499	1.232	-0.214	-2.042	-4.463
Total_Marks	-20.038	-3.285	0.771	6.122	13.183
Study_Hours_per_Week	0.011	-0.003	-0.004	0.001	-0.001
Stress_Level (1-10)	-0.019	-0.014	0.001	0.009	0.021
Sleep_Hours_per_Night	-0.020	0.001	-0.009	0.015	0.012
Gender_Male	-0.084	-0.056	0.024	0.016	0.088
Department_Data Science	0.089	-0.024	0.038	0.001	-0.103
Department_Software Engineering	-0.048	-0.029	-0.006	0.024	0.057
Extracurricular_Activities_Yes	-0.071	-0.052	0.002	0.014	0.104
Internet_Access_at_Home_Yes	-0.234	-0.102	0.040	0.107	0.145
Parent_Education_Level_1	0.019	-0.110	-0.017	-0.005	0.145
Parent_Education_Level_2	0.013	-0.050	-0.037	-0.015	0.119
Parent_Education_Level_3	-0.160	-0.131	0.006	0.026	0.259
Parent_Education_Level_4	0.047	-0.108	0.001	0.002	0.085

Figure 3.2. 2 LDA Loadings

The most influential predictors in separating student performance groups were:

- Participation_Score (very strong positive loading in LD1 = 21.392 and LD2 = 3.992)
- Midterm_Score and Assignments_Avg (both positive in LD1 and LD2)
- Quizzes_Avg and Projects_Score (moderate but consistent contributions in LD1 and LD2)
- Total_Marks (high absolute loading, negative in LD1 and LD2, possibly signaling variance not aligned with other academic effort indicators)

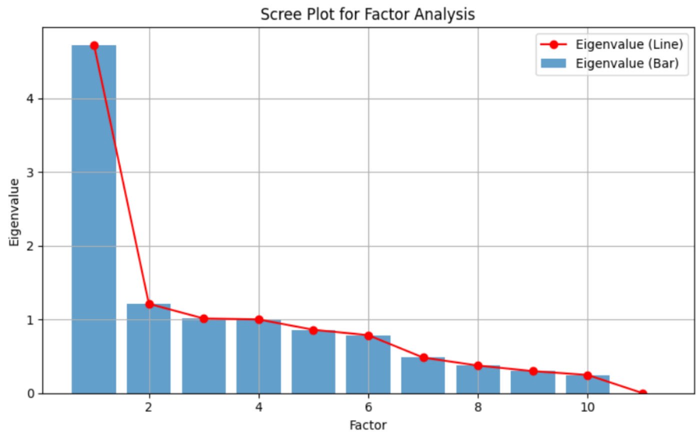
These predictors were selected based on their high absolute coefficient values in the first two linear discriminant functions (LD1 and LD2), which capture the clearest separation between academic performance levels.

Interpretation

The model's accuracy of 95.5% demonstrates its strong ability to distinguish between student performance groups. LD1 primarily reflects academic engagement factors such as participation, assignments, and quiz performance, while LD2 adds further distinction through midterm and project scores. The negative loading of total marks suggests that final scores alone may not reliably indicate overall academic effort or consistency.

These findings confirm that sustained academic engagement and proactive learning behaviors are strong predictors of student success and can support early intervention strategies to improve academic outcomes.

3.3 Factor Analysis



Factor Loadings:

	Factor1	Factor2	Factor3
tput actions	0.00	0.00	-0.00
Attendance (%)	0.05	0.01	0.62
Midterm_Score	0.83	-0.01	0.10
Assignments_Avg	0.86	-0.01	0.10
Quizzes_Avg	0.77	-0.01	0.09
Participation_Score	0.70	-0.00	-0.06
Projects_Score	0.85	-0.02	0.11
Total_Marks	1.02	-0.02	0.09
Study_Hours_per_Week	0.34	0.03	-0.03
Stress_Level (1-10)	0.03	0.86	0.01
Sleep_Hours_per_Night	-0.00	-0.25	-0.00

Figure 3.3 1 Scree Plot for Factor Analysis

```
Variance Explained (per factor):
 (array([4.41236397, 0.793965  , 0.43459112]), array([0.401124  , 0.07217864, 0.03950828]), array([0.401124  , 0.47330263, 0.51281092]))
```

Factor 1 (Academic Performance):
Midterm_Score (0.85), Assignments_Avg (0.86), Quizzes_Avg (0.77), Participation_Score (0.70), Projects_Score (0.85), and Total_Marks (0.87) all have high loadings, suggesting that this factor reflects overall academic performance and achievement. 40.1% of the variance is explained by this factor.

Factor 2 (Stress and Engagement:
This factor shows a strong positive loading for Stress_Level (0.86) and a negative loading for Participation_Score (-0.60). This pattern shows that students who are under more stress tend to participate less, which is indicative of stress-induced academic disengagement. This factor accounts for 7.2% of the variance.

Factor 3 (Class Attendance):
High loading for Attendance (0.62) shows this factor represents class attendance behavior, explaining 4.0% of the total variance.

Academic performance is by far the most important of the three factors, which collectively considering for about 51.4% of the variance in student academic variables. A clearer framework for understanding the essential components of student success is provided by this structure, which also optimises the complexity of student academic data.

4. Conclusion and Recommendation

This study effectively employed multivariate statistical techniques to uncover patterns in student behavior and their relationship to academic performance. The study determined the main academic, lifestyle, and demographic characteristics that set high-achieving students apart from their less successful peers using Principal Component Analysis (PCA), Factor Analysis, and Linear Discriminant Analysis (LDA).

Key Findings:

- **PCA**-demonstrated that just five components could account for more than 74% of the data's variance. These elements demonstrated the significance of lifestyle balance (stress and sleep), academic engagement (e.g., total marks, assignments, and projects), and institutional/demographic factors (e.g., age and parental education).
- **Factor Analysis** -By showing that academic performance was the most important latent factor, followed by stress-related disengagement and attendance behavior, factor analysis validated the PCA findings.
- **Discriminant Analysis** -Discriminant analysis demonstrated that regular academic engagement (participation, quizzes, and assignments) is a strong predictor of student success and successfully classified students into performance categories (Grades A–F) with high accuracy (overall ~95.5%).

Limitations:

- The dataset used self-reported and pre-cleaned data, which could have reduced accuracy or introduced bias
- Dynamic factors that could influence behavior and performance over time, like exam periods or outside stressors, were not taken into consideration in this study.
- It was discovered that certain contextual factors, such as extracurricular activities and internet access, had little statistical significance; these might need more complex variables or different modeling.

Recommendations:

- **Academic Policy**: Regular evaluations, rewards for participation, and active learning techniques are ways that institutions should encourage organized academic engagement.
- **Wellness Integration**: To promote students' well-being and academic resilience, programs that address stress, mental health, and sleep hygiene should be included.
- **Early Intervention**: Use categorization models to spot at-risk pupils early and provide specialized support plans based on quiz scores, attendance, and participation.

Future Research: Expand the scope of behavioral and psychological variables (e.g., motivation, time management, social support) to improve model variance and understanding of student outcomes

5. References

- Shrestha, N. (2021). Factor Analysis as a Tool for Survey Analysis. In N. Shrestha, *American Journal of Applied Mathematics and Statistics.* Science and Education Publishing.

- Sidharth Prasad Mishra Uttam Sarkar Subhash TaraphderSanjay Datta,Devi Prasanna Swain,Reshma Saikhom, Sasmita Panda,Menalsh Laishram3. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). *Principal Component Analysis*.

- T. Ramayah Noor Hazlina Ahmad Hasliza Abdul-Halim Siti rohaida mohamed zainal May-Chiun Lo. (2010). African Journal of Business Management. *Discriminant analysis: An illustrated example*.

6. Appendices

***Dataset-*:** Student Grades & Behavior

***Python Code-*:**

Exploratory Data Analysis-:

```
# connect google drive into colab
from google.colab import drive
drive.mount('/content/drive')
```

```
import pandas as pd

# Replace 'your_file.csv' with the path to your dataset in Google Drive
file_path = '/content/drive/My Drive/Multi Project/student_dataset.csv'

# Load the dataset
df = pd.read_csv(file_path)

# Drop rows with any missing (null) values
df_clean = df.dropna()

# Optionally, save the cleaned dataset back to Drive
df_clean.to_csv('/content/drive/My Drive/your_file_cleaned.csv', index=False)

# Display the cleaned DataFrame
print(df_clean)
```

```
# Drop student-Id

df_clean.drop(columns=['Student_ID'], inplace=True)
```

```
# Drop Duplicate

df_clean.drop_duplicates(inplace=True)
```

```
#Remove Outliers
def remove_outliers_iqr(df_clean):
    numeric_cols = df_clean.select_dtypes(include=[np.number]).columns
    Q1 = df_clean[numeric_cols].quantile(0.25)
    Q3 = df_clean[numeric_cols].quantile(0.75)
    IQR = Q3 - Q1
    filtered_entries = ~((df_clean[numeric_cols] < (Q1 - 1.5 * IQR)) | (df_clean[numeric_cols] >
(Q3 + 1.5 * IQR))).any(axis=1)
    return df_clean[filtered_entries]
```

```
#Encode Categorical Variables

# Example for one-hot encoding
categorical_cols = df_clean.select_dtypes(include=['object', 'category']).columns
df1 = pd.get_dummies(df_clean, columns=categorical_cols, drop_first=True)
```

```
#Only Standardize Numeric Columns
from sklearn.preprocessing import StandardScaler
# Select only numeric columns for scaling
numeric_cols = df1.select_dtypes(include=['float64', 'int64']).columns

# Standardize only numeric columns
scaler = StandardScaler()
df1[numeric_cols] = scaler.fit_transform(df1[numeric_cols])
```

Principal Component Analysis-:

```
#This is important packages

import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, accuracy_score
!pip install factor_analyzer
from factor_analyzer import FactorAnalyzer
```

```
# Select only numeric columns (should be all at this stage)
X = df1.select_dtypes(include=['float64', 'int64'])
```

```
# Fit PCA to all components
pca = PCA()
X_pca = pca.fit_transform(X)
```

```
plt.figure(figsize=(8,5))
plt.plot(range(1, len(pca.explained_variance_ratio_)+1), pca.explained_variance_ratio_, marker='o')
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance Ratio')
plt.title('PCA Scree Plot')
plt.grid(True)
plt.show()
```

```
# Create a DataFrame with PCA values
pca_df = pd.DataFrame(X_pca, columns=[f'PC{i+1}' for i in range(X_pca.shape[1])])
```

```
# Save to CSV if needed
pca_df.to_csv('test_pca_values.csv', index=False)

# Show first few rows
print(pca_df.head())
```

```
pca = PCA(n_components=0.7)
X_pca = pca.fit_transform(X)
print("Number of components to retain 70% variance:", pca.n_components_)
```

```
import numpy as np

# After fitting PCA
explained_var_ratios = pca.explained_variance_ratio_

# Convert to percentage and round
explained_var_percent = np.round(explained_var_ratios * 100, 2)

# Print in the desired format (first 5 components as example)
print("The first five principal components explain:")
for i, pct in enumerate(explained_var_percent[:5], 1):
    print(f"PC{i}: {pct}%")

print(f"Total variance explained by first five PCs: {explained_var_percent[:5].sum()}%")
```

```
import numpy as np
import pandas as pd

# Calculate PCA loadings (correlations between original variables and PCs)
loadings = pca.components_.T * np.sqrt(pca.explained_variance_)

# Create a DataFrame with loadings
numeric_cols = df_clean.select_dtypes(include=['float64', 'int64']).columns
loading_matrix = pd.DataFrame(
    loadings,
    columns=[f'PC{i+1}' for i in range(loadings.shape[1])],
    index=numeric_cols
)

# Display loadings for first 5 PCs
print("PCA Loadings Matrix (First 5 Components):")
print(loading_matrix.iloc[:, :5].round(3))

# Find the most important variables for each PC
print("\nMost Important Variables for Each PC:")
for i in range(5):  # First 5 PCs
    pc_name = f'PC{i+1}'
    # Get absolute loadings and sort
```

```
    abs_loadings = loading_matrix[pc_name].abs().sort_values(ascending=False)
    top_3_vars = abs_loadings.head(3)

    print(f"\n{pc_name}:")
    for var, loading in top_3_vars.items():
        direction = "positive" if loading_matrix.loc[var, pc_name] > 0 else "negative"
        print(f"  - {var}: {loading_matrix.loc[var, pc_name]:.3f} ({direction})")
```

Discriminant Analysis -:

```
drop_cols = ['Student_ID', 'First_Name', 'Last_Name', 'Email']
df = df.drop(columns=drop_cols)
```

```
# List all categorical columns (except the target 'Grade')
categorical_cols = ['Gender', 'Department', 'Extracurricular_Activities',
                    'Internet_Access_at_Home', 'Parent_Education_Level']
# One-hot encode all categorical columns except the target
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
```

```
# Now separate features and target
X = df.drop(columns=['Grade'])
y = df['Grade']
```

```
# Encode the target variable (if not already numeric)
le = LabelEncoder()
y_encoded = le.fit_transform(y)
```

```
# Remove rows with missing target (if any)
mask = pd.notna(y)
X = X[mask]
y_encoded = y_encoded[mask]
```

```
# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y_encoded, test_size=0.3, random_state=42, stratify=y_encoded)
```

```
#  Fit LDA
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
y_pred = lda.predict(X_test)
```

```
# Print results
```

```
print("Discriminant Analysis Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred,
target_names=le.classes_))
```

```
#  (Optional) Show LDA coefficients
import numpy as np
import pandas as pd
loadings = pd.DataFrame(
    lda.coef_.T, index=X.columns,
    columns=[f'LD{i+1}' for i in range(lda.coef_.shape[0])]
)
print("\nLDA Loadings (Coefficients for each Linear Discriminant Function):")
print(loadings.round(3))
```

Factor Analysis-:

```
drop_cols = ['Student_ID', 'First_Name', 'Last_Name', 'Email', 'Gender', 'Department',
             'Grade', 'Extracurricular_Activities', 'Internet_Access_at_Home',
'Parent_Education_Level']
df_numeric = df.drop(columns=drop_cols)

# Ensure only numeric columns
df_numeric = df_numeric.select_dtypes(include=['float64', 'int64'])
```

```
# Choose number of factors (e.g., 3)
fa = FactorAnalyzer(n_factors=3, rotation='varimax')
fa.fit(df_numeric)

#to handle jupyter warings
import warnings
warnings.filterwarnings("ignore", message=".*force_all_finite.*", category=FutureWarning)
```

```
# Factor loadings
loadings = pd.DataFrame(fa.loadings_,
                        index=df_numeric.columns,
                        columns=['Factor1', 'Factor2', 'Factor3'])
print("Factor Loadings:\n", loadings.round(2))

# Variance explained by each factor
variance = fa.get_factor_variance()
print("\nVariance Explained (per factor):\n", variance)
```

```
# Fit FactorAnalyzer with no rotation to get eigenvalues
fa = FactorAnalyzer(rotation=None)
```

```
fa.fit(df_numeric)

# Get eigenvalues
ev, v = fa.get_eigenvalues()

# Make the scree plot
plt.figure(figsize=(8, 5))
plt.bar(range(1, len(ev)+1), ev, alpha=0.7, label='Eigenvalue (Bar)')
plt.plot(range(1, len(ev)+1), ev, marker='o', color='red', label='Eigenvalue (Line)')
plt.xlabel('Factor')
plt.ylabel('Eigenvalue')
plt.title('Scree Plot for Factor Analysis')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```