```
# Technique: Create 'Avg_Grade' and 'Total_Approved'
# Justification: Semester-specific data (e.g., grades, approved units) lacks hc

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from google.colab import drive
import os
```

```
# Mount Google Drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Define paths and create directories
data_path = '/content/drive/My Drive/rawDataset.csv'
results_path = '/content/drive/My Drive/results/'
eda_vis_path = results_path + 'eda_visualizations/'
os.makedirs(eda_vis_path, exist_ok=True)
```

```
# Load dataset
columns = ['Marital status', 'Application mode', 'Application order', 'Course',
           'Previous qualification', 'Previous qualification (grade)', 'Nacionа
           "Father's qualification", "Mother's occupation", "Father's occupatic
           'Educational special needs', 'Debtor', 'Tuition fees up to date', 'G
           'Age at enrollment', 'International', 'Curricular units 1st sem (cre
           'Curricular units 1st sem (enrolled)', 'Curricular units 1st sem (ev
           'Curricular units 1st sem (approved)', 'Curricular units 1st sem (gr
           'Curricular units 1st sem (without evaluations)', 'Curricular units
           'Curricular units 2nd sem (enrolled)', 'Curricular units 2nd sem (ev
           'Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (gr
           'Curricular units 2nd sem (without evaluations)', 'Unemployment rate
df = pd.read_csv(data_path, sep=';', names=columns, header=0)
```

```
# Feature Engineering: Creation of Aggregated Features
df['Avg_Grade'] = (df['Curricular units 1st sem (grade)'] + df['Curricular unit
df['Total_Approved'] = df['Curricular units 1st sem (approved)'] + df['Curricul
```

```
# Handle missing or invalid values (e.g., NaN or infinite values from division)
df['Avg_Grade'] = df['Avg_Grade'].replace([np.inf, -np.inf], np.nan).fillna(0)
df['Total_Approved'] = df['Total_Approved'].fillna(0)
```

```
# Feature Engineering: Dimension Reduction using PCA on Curricular Units Featur
# Select relevant features for PCA (focusing on curricular units to reduce dime
curricular_features = [
    'Curricular units 1st sem (credited)', 'Curricular units 1st sem (enrolled)
```

```
    'Curricular units 1st sem (evaluations)', 'Curricular units 1st sem (approv
    'Curricular units 1st sem (grade)', 'Curricular units 1st sem (without eval
    'Curricular units 2nd sem (credited)', 'Curricular units 2nd sem (enrolled)
    'Curricular units 2nd sem (evaluations)', 'Curricular units 2nd sem (approv
    'Curricular units 2nd sem (grade)', 'Curricular units 2nd sem (without eval
]

X = df[curricular_features]
```

```
# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Apply PCA to reduce to 2 components for visualization and efficiency
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
```

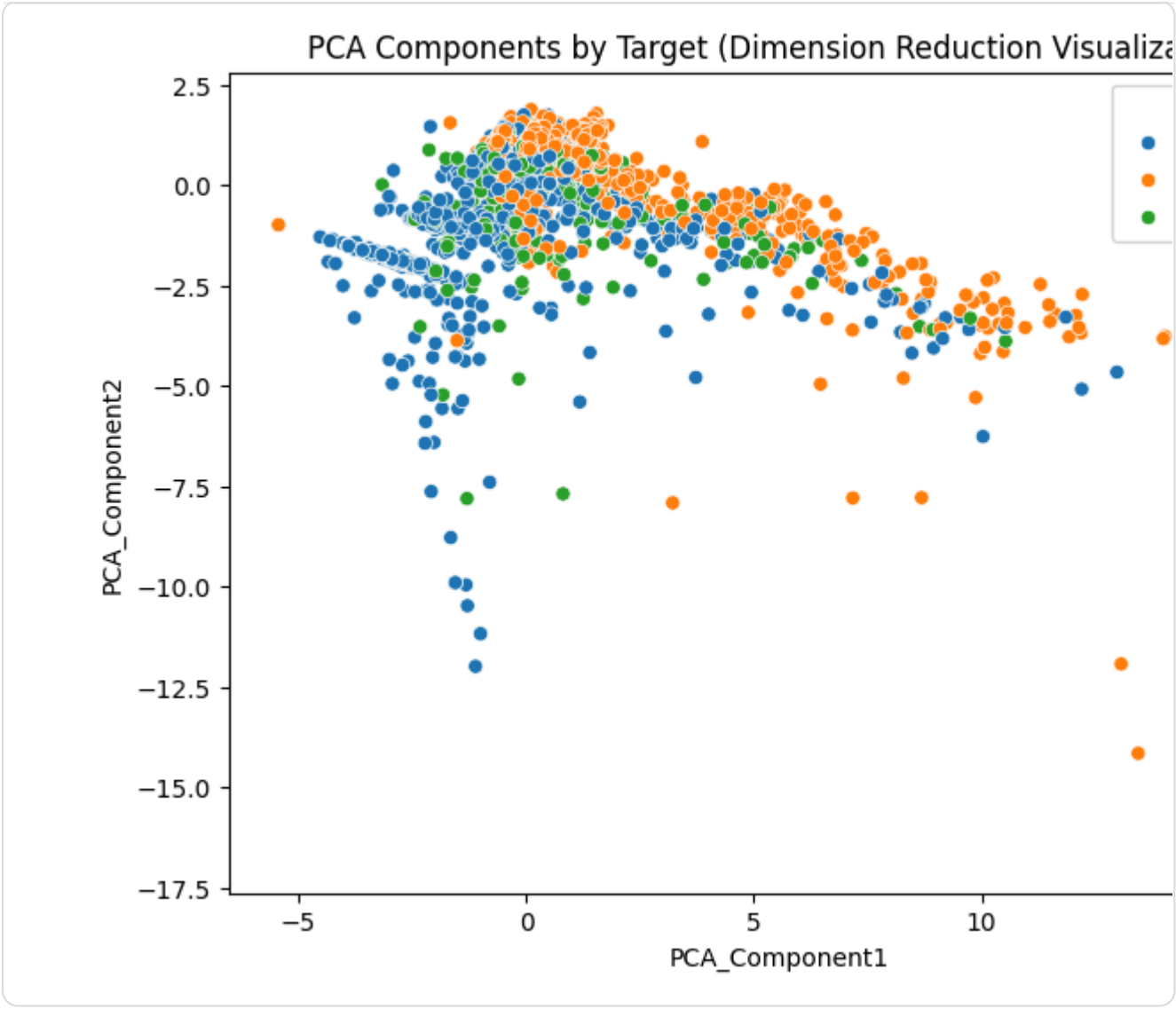
```
# Add PCA components back to dataframe
df['PCA_Component1'] = X_pca[:, 0]
df['PCA_Component2'] = X_pca[:, 1]

print("\nPCA Explained Variance Ratio:", pca.explained_variance_ratio_)
print("PCA Components Sample:\n", df[['PCA_Component1', 'PCA_Component2']].heac
```

```
PCA Explained Variance Ratio: [0.51379728 0.16979443]
PCA Components Sample:
    PCA_Component1  PCA_Component2
0       -5.436567       -0.986890
1        0.085205        1.224100
2       -3.589884       -1.621948
3        0.325992        0.884045
4        0.064950        0.867917
```

```
# Save the processed dataset (for group pipeline integration)
processed_path = results_path + 'outputs/member5_processed_features.csv'
# Create the directory if it doesn't exist
os.makedirs(os.path.dirname(processed_path), exist_ok=True)
df.to_csv(processed_path, index=False)
print(f"Processed features saved to: {processed_path}")
```

```
Processed features saved to: /content/drive/My Drive/results/outputs/member5_pro
```

```
# EDA Visualization: Scatterplot of PCA Components colored by Target
plt.figure(figsize=(8, 6))
sns.scatterplot(x='PCA_Component1', y='PCA_Component2', hue='Target', data=df)
plt.title("PCA Components by Target (Dimension Reduction Visualization)")
plt.savefig(eda_vis_path + 'member5_pca_scatterplot.png')
plt.show()
```



Interpretation

This PCA plot shows that while graduates and dropouts have some separation, their points still overlap a lot, and enrolled students are mostly scattered among them, meaning the groups are not clearly distinct.