Modelling Report

Team 2

1 Objective

The goal of this project is to fit a good predictor model for this dataset. We chose **Tuition Payment 2023** as the target variable because of the existence of **Tuition Payment 2022**, which is highly correlated with the target. Upon further analysis, it became clear that other variables exhibited very low correlation and offered minimal insight for the prediction task. This reinforces the choice of **Tuition Payment 2022** as a strong predictor, helping us effectively analyze the payment trends.

2 Feature Engineering

In this section, we detail the selection of features used for both the regression and classification models. The choice of features is driven by their correlation with the target variable, **Tuition Payment March 2023**, which is the main focus of the model. The feature correlation analysis reveals the strength of relationships between different features and the target, which guides the selection process.

The correlation values with the target variable **Tuition Payment March 2023** are as follows:

Feature	**Correlation with Tuition Payment March 2023**
TUITION PAYMENT MARCH 2022	0.922731
ENROLLMENT	0.327749
AGE RANGE OF ENROLLED STUDENT	0.061663
GENDER	0.043202
NUMBER OF ENROLLED COURSES	0.033821
SHIFT/SCHEDULE	0.030778
STUDY MODE	0.014902
BENEFIT DISCOUNTS	0.014032
DEPARTMENT	0.012320
PROGRAM/MAJOR	0.001469

Table 1: Correlation of Features with Target Variable (Tuition Payment March 2023)

From this analysis, we observe that the top features with the strongest correlation to the target variable are:

- **Tuition Payment March 2022**: With a strong correlation of **0.922731**, this feature captures previous payment trends, which strongly influence the prediction of future tuition payments.
- **Enrollment**: This feature has a moderate correlation of **0.327749**, suggesting that the number of enrolled students is somewhat related to the tuition payment.
- **Age Range of Enrolled Student**: Although this feature shows a weaker correlation of **0.061663**, it still provides some useful information, particularly when considering demographic influences on tuition payments.
- **Gender, Shift/Schedule, Study Mode, Benefit Discounts, Department, Program/Major**: These features have weak correlations with the target variable, but they were still considered for

inclusion in the model as they could provide secondary insights that might complement the primary features.

3 Data Overview

- **Target variable**: Tuition Payment 2023 (binary classification: paid vs. not paid)
- **Primary features used**: Tuition Payment 2022, Enrollment
- **Feature types**: All features are categorical
- **Total samples**: 36,584
- **Data split**: 80% training, 10% validation, 10% testing
  - **Training set size**: 29,267 samples
  - **Validation set size**: 3,658 samples
  - **Testing set size**: 3,659 samples
- **Note on data splits**: We experimented with various conventional dataset splits for training, validation, and testing—including the standard 80-10-10, as well as 60-20-20 and other common proportions. Across all these configurations, the results remained consistent: all models reliably converged to a high level of accuracy, typically exceeding 97%, regardless of the exact split. This consistency suggests the task is relatively easy and well-defined, with highly separable features and low ambiguity. The saturated predictions imply that the dataset is not only balanced but also exhibits patterns that models can learn with high confidence, resulting in minimal variance across splits.

4 Regression Model Performance

Model	MSE	R^2 Score
Linear Regression	0.0156	0.8801
Ridge Regression	0.0156	0.8801
Lasso Regression	0.0261	0.7995

Table 2: Regression performance metrics

The Ridge Regression model was trained with `alpha = 1`, while the Lasso Regression used `alpha = 0.1` and `max_iter = 10000`.
Note: Both Linear and Ridge Regression models achieved identical and superior performance compared to Lasso Regression. This suggests that regularization with Ridge did not significantly alter model behavior relative to standard Linear Regression, while Lasso's feature selection introduced sparsity at the cost of predictive accuracy.

5 Classification Model Performance

Model	Accuracy
Linear Regression + Classifier	98.57%
Ridge + Classifier	98.57%
Lasso + Classifier	98.57%
Logistic Regression	98.57%
Random Forest	98.57%
XGBoost	98.57%
Deep Neural Network	98.57%
Ensemble	98.57%

Table 3: Classification accuracy across models

Key Insight: All models converge to approximately **98.578% accuracy** and produce *exactly the same classification report*, suggesting they make **identical predictions on the test data**, with only a few misclassified instances. Reducing the test size to 5% pushes these models to **100% accuracy**, indicating near-perfect fit on seen patterns.

5.1 Hyperparameters (via scikit-learn):

- Ridge: `alpha = 1`
- Lasso: `alpha = 0.1`, `max_iter = 10000`
- Logistic Regression: `multi_class = 'multinomial'`, `max_iter = 1000`
- XGBoost: `eval_metric = 'mlogloss'`

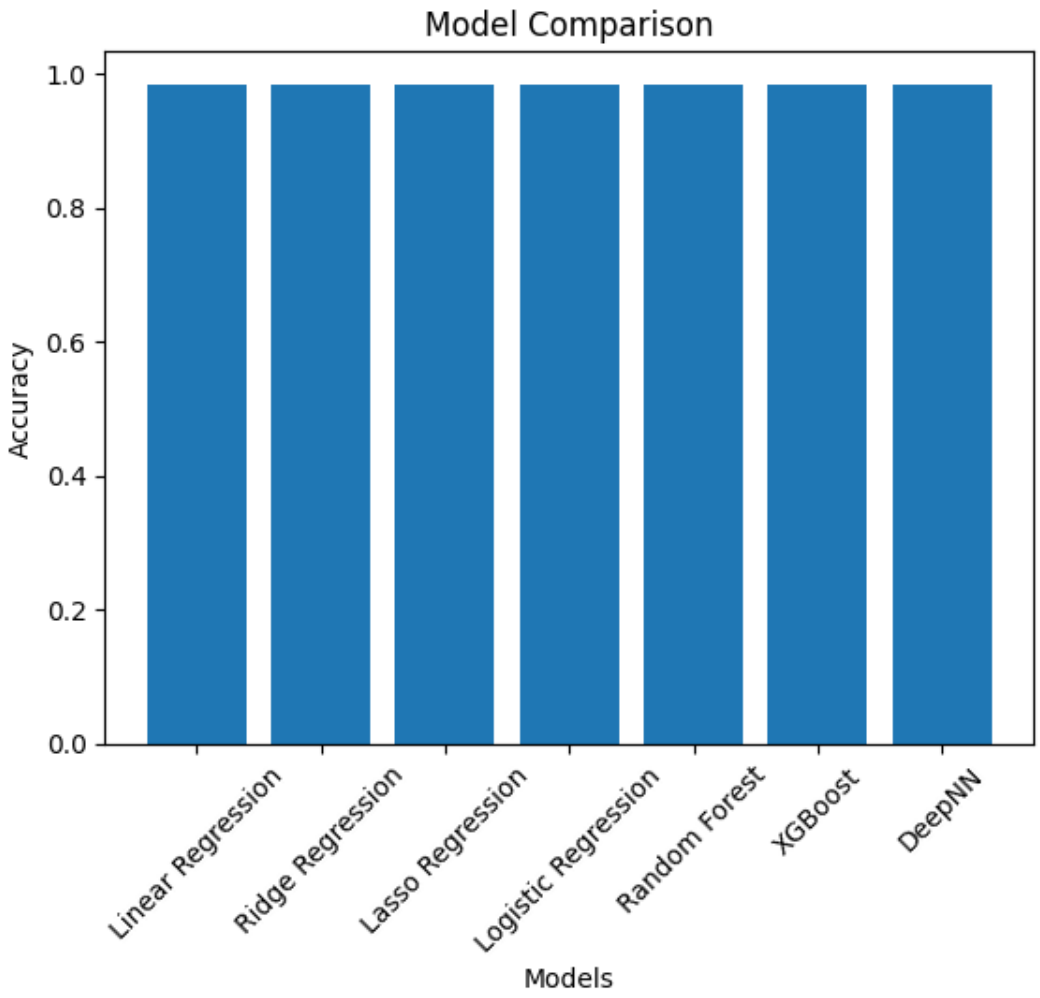

Figure 1: Model comparison chart across classifiers

5.2 Deep Neural Network Architecture

```
nn.Sequential(
    nn.Linear(input_dim, 256),
    nn.BatchNorm1d(256),
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Linear(256, 128),
    nn.BatchNorm1d(128),
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Linear(128, 64),
    nn.ReLU(),
    nn.Linear(64, num_classes)
)
```

Explanation: The architecture of the Deep Neural Network (DNN) includes multiple layers of batch normalization and dropout to enhance performance, generalization, and training stability. Below is a breakdown of why each component is included:

- **Batch Normalization:**
  - Normalizes the inputs of each layer to maintain a stable distribution during training.
  - Speeds up convergence by allowing higher learning rates.
  - Helps mitigate the **vanishing gradient problem** by keeping activations in a stable range.
  - Reduces internal covariate shift, leading to more consistent gradient flow.
- **Dropout:**
  - Randomly deactivates a portion of neurons (30% in this model) during training.
  - Prevents overfitting by encouraging the network to learn redundant and robust representations.
  - Improves generalization performance on unseen data.
- **ReLU Activation Functions:**
  - Avoid the saturation issues found in sigmoid or tanh activations.
  - Allow gradients to propagate more effectively in deep networks.
  - Further reduce the risk of the **vanishing gradient problem**.
- **Vanishing Gradient Problem:**
  - In deep networks, gradients can diminish as they backpropagate through many layers.
  - This causes very slow or stalled learning in earlier layers.
  - The use of **ReLU**, **Batch Normalization**, and good initialization practices help prevent this issue.

Overall, this architecture balances depth and stability using best practices like batch normalization and dropout, ensuring the network learns effectively without overfitting or suffering from unstable gradient flow. mitigating the vanishing gradient problem that can hinder the performance of deep neural networks.

6 Classification Reports

Same across all Models

	precision	recall	f1-score	support
0	1.00	0.91	0.95	575
1	0.98	1.00	0.99	3084
accuracy			0.99	3659
macro avg	0.99	0.95	0.97	3659
weighted avg	0.99	0.99	0.99	3659

Interpretation:

- The classification report shows performance across both classes: **0** (did not pay tuition) and **1** (paid tuition).
- The high **precision** (1.00 for class 0, 0.98 for class 1) indicates that when the model predicts either class, it is almost always correct.
- The **recall** of 0.91 for class 0 suggests the model correctly identifies 91% of all actual unpaid cases, while recall of 1.00 for class 1 shows that all paid instances are captured.
- The **f1-score**, a harmonic mean of precision and recall, reflects overall balance between the two:
  - Class 0: f1 = 0.95 (solid despite fewer samples)
  - Class 1: f1 = 0.99 (very strong)
- **Accuracy** of 0.99 over 3,659 test samples confirms that the models nearly always predict correctly.
- The **macro average** gives equal weight to both classes and indicates some disparity (due to smaller class 0), with recall at 0.95 and f1 at 0.97.
- The **weighted average**, which considers class imbalance, is uniformly high (0.99 across metrics), affirming consistent and reliable classification performance.

Conclusion: Despite the class imbalance (class 1 being significantly more prevalent), the models generalize exceptionally well, demonstrating high precision and recall for both classes. The nearly identical results across all models reinforce the observation that the task is highly learnable—potentially indicating a saturated classification problem where simple or complex models reach the same decision boundary.

7 Clustering Analysis

Motivation for PCA in Clustering

A natural idea when visualizing clusters might be to plot the two most important features: **Enrollment** and **Tuition Payment March 2022**. However, this approach is flawed because both features are **binary or low-cardinality categorical variables**. As a result, the data points would collapse onto a few discrete coordinates on a 2D plane (e.g., points like (0,0), (1,1), (0,1), etc.), preventing the identification of meaningful clusters. This would result in poor separation, excessive overlap, and little to no structure in the scatter plot.

To overcome this limitation, we use all available features and apply **Principal Component Analysis (PCA)** to reduce the dimensionality to two components. PCA captures the directions of maximum variance in the data, projecting it onto a 2D space that is much more suitable for visualizing the inherent structure and clustering patterns.

KMeans Clustering (PCA-Reduced Data)

- **K = 2**: Purity = 0.8429
- **K = 3**: Purity = **0.9419** (best result)
- **K = 4**: Purity = 0.9340

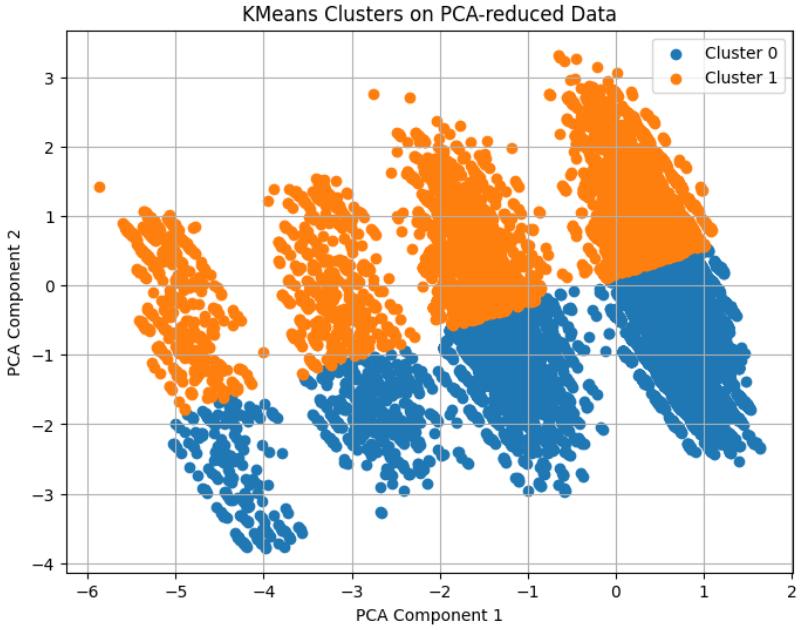

Figure 2: KMeans Clustering (2 Clusters)

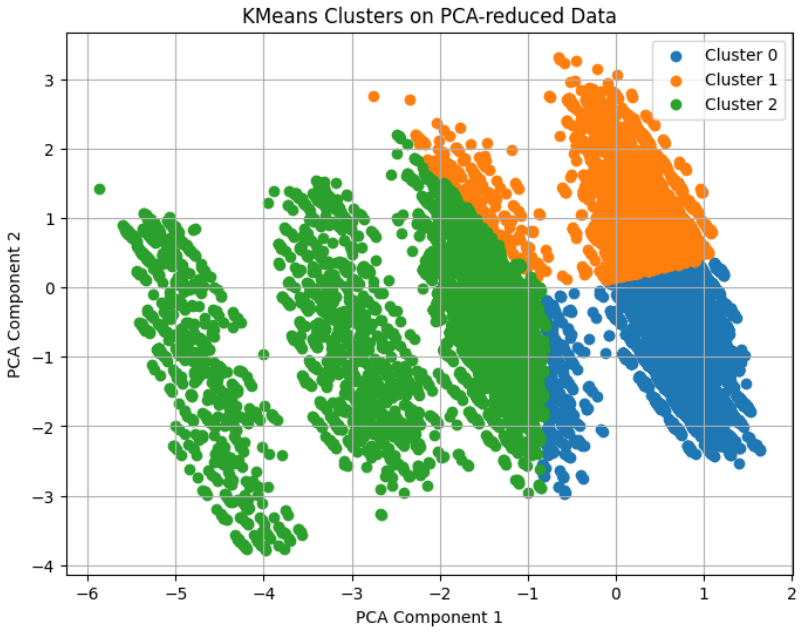

Figure 3: KMeans Clustering (3 Clusters)

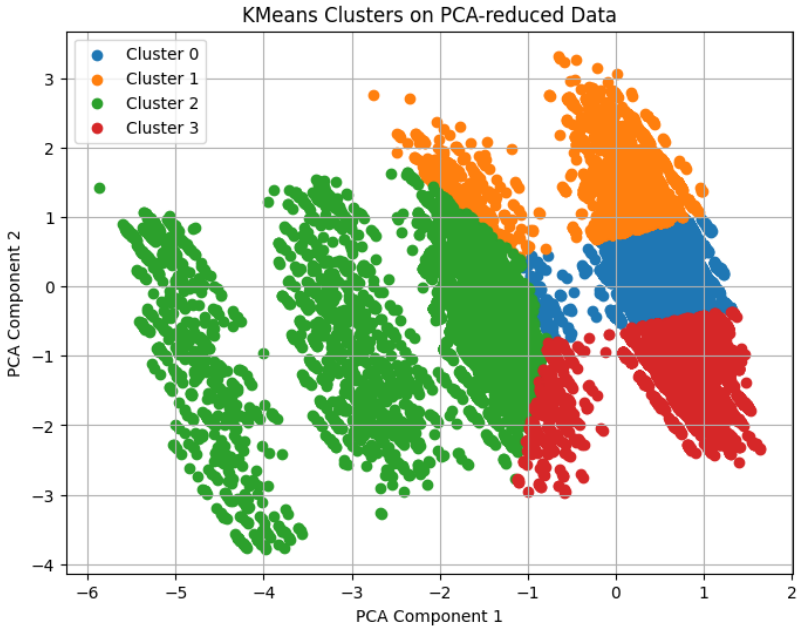

Figure 4: KMeans Clustering (4 Clusters)

Interpretation:

- **Purity** measures how well the clustering aligns with true class labels. Higher purity indicates better separation.
- While increasing the number of clusters generally improves purity, it also risks **overfragmentation**.
- In this case, $\mathbf{K = 3}$ achieves the highest purity but does not correspond to the binary classification nature of the data.
- Hence, $\mathbf{K = 2}$ is more appropriate and interpretable, closely matching the underlying label structure.

DBSCAN Clustering (PCA-Reduced Data)

Table 4: DBSCAN Clustering Results on PCA-Reduced Data

Epsilon	**Min Samples**	**# Clusters**	**Purity**
1	5	338	0.0433
1	10	331	0.04333
2	5	23	0.4970
2	10	18	0.4981
3	5	5	0.8275
3	10	4	0.8278
4	5	3	0.8437
4	10	3	0.8437
5	5	1	0.8429
5	10	1	0.8429

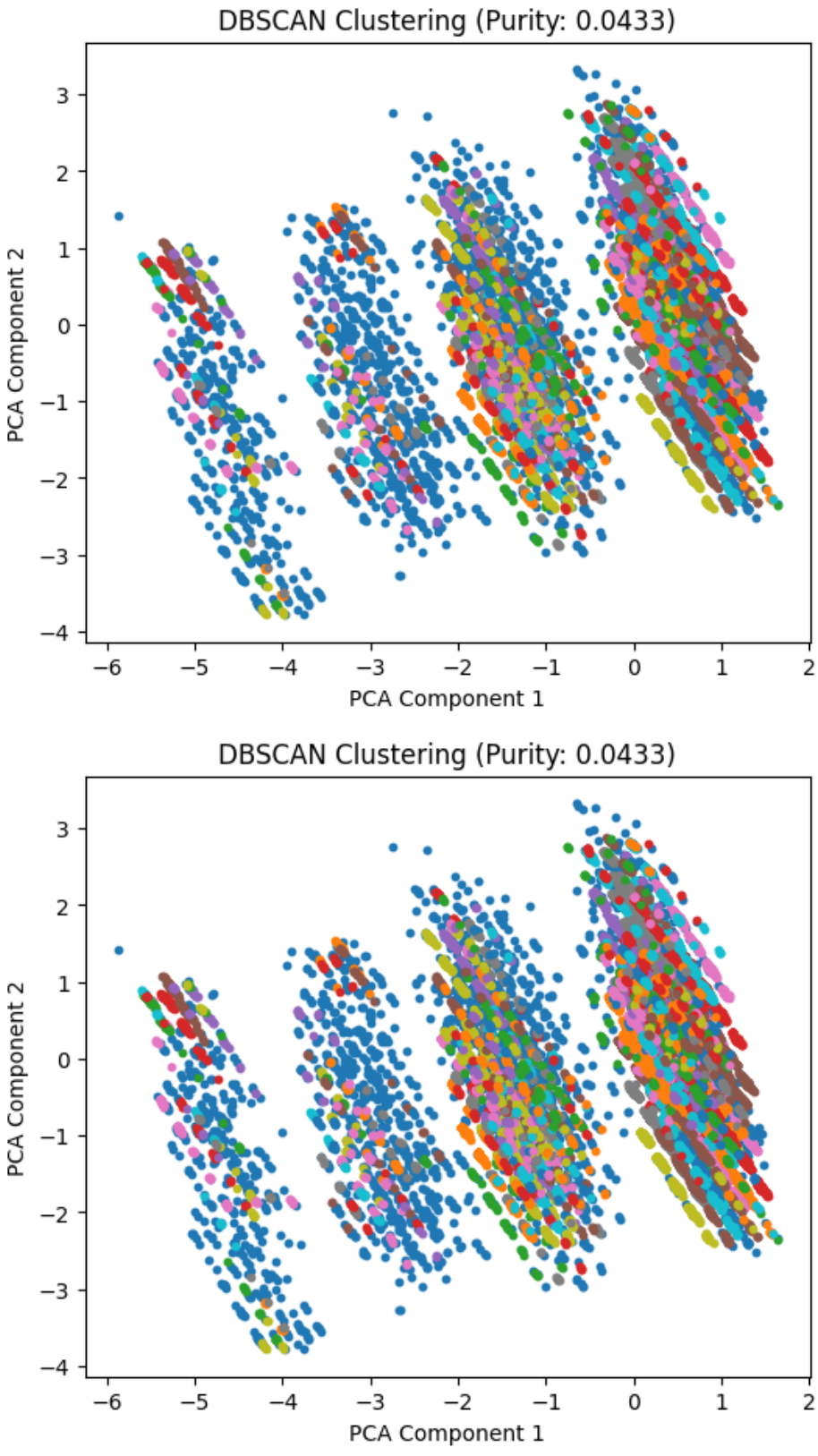

Figure 5: DBSCAN Clustering Results (Epsilon = 1, Min Samples = 5 and 10)

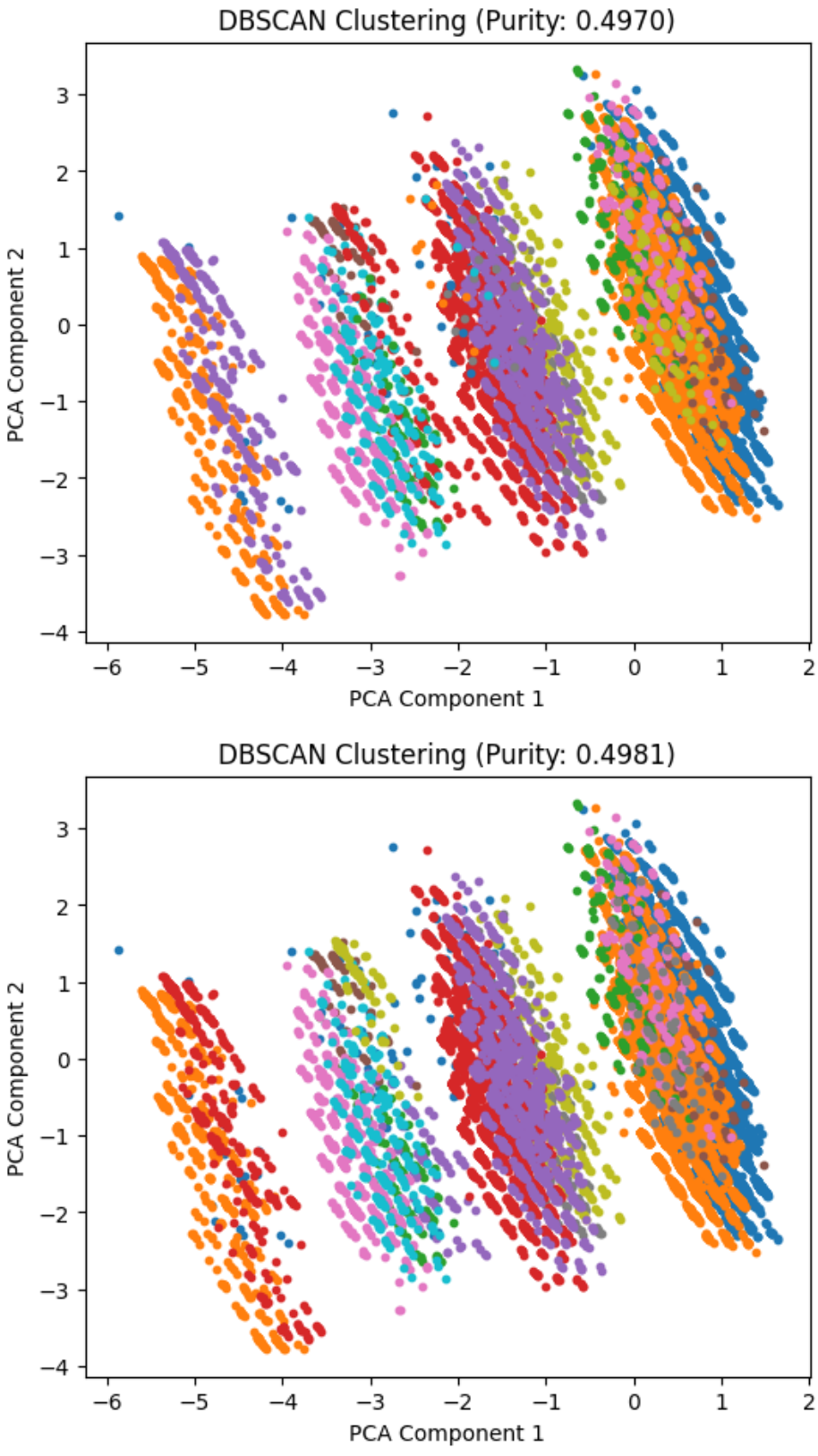

Figure 6: DBSCAN Clustering Results (Epsilon = 2, Min Samples = 5 and 10)

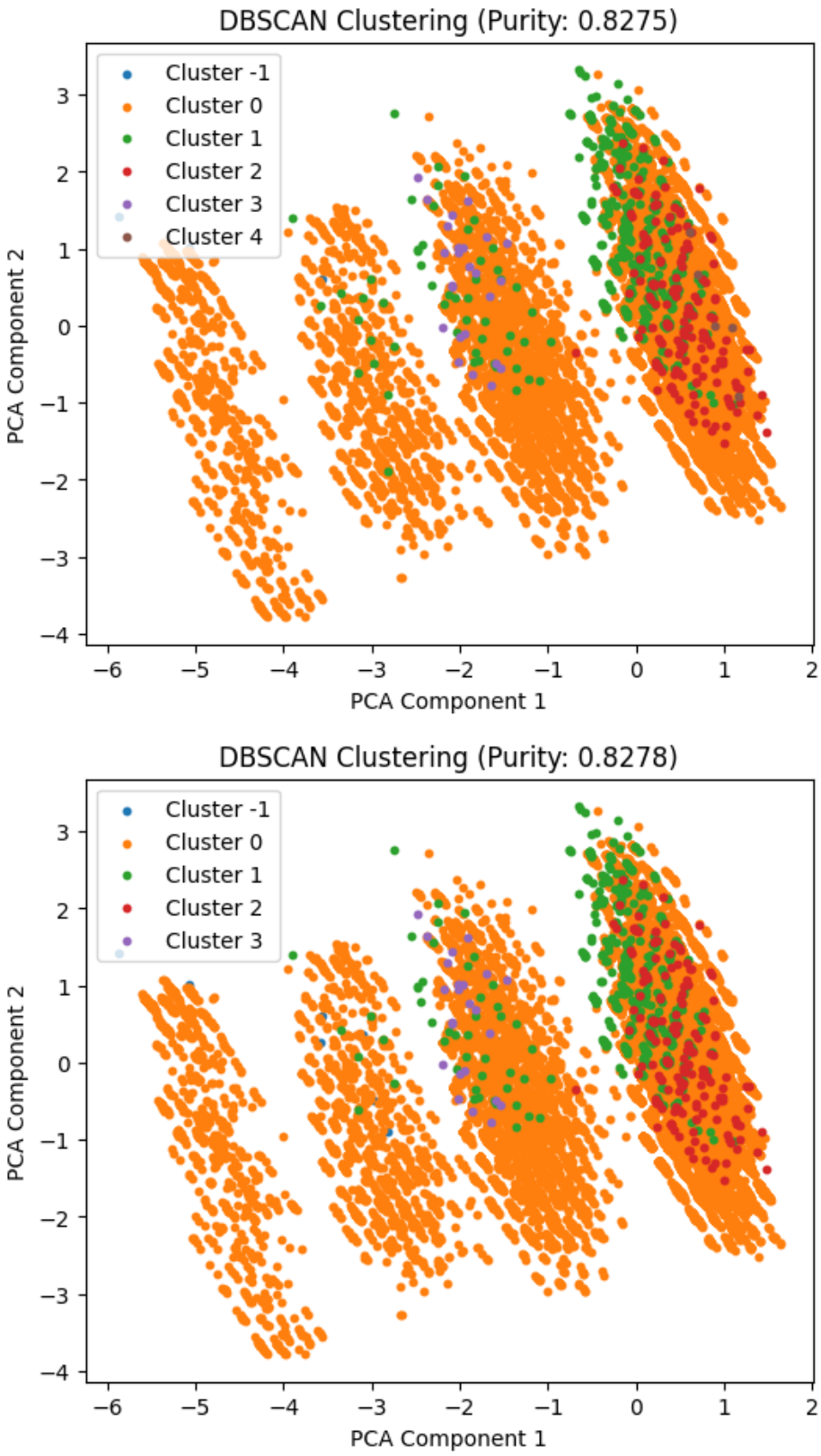

Figure 7: DBSCAN Clustering Results (Epsilon = 3, Min Samples = 5 and 10)

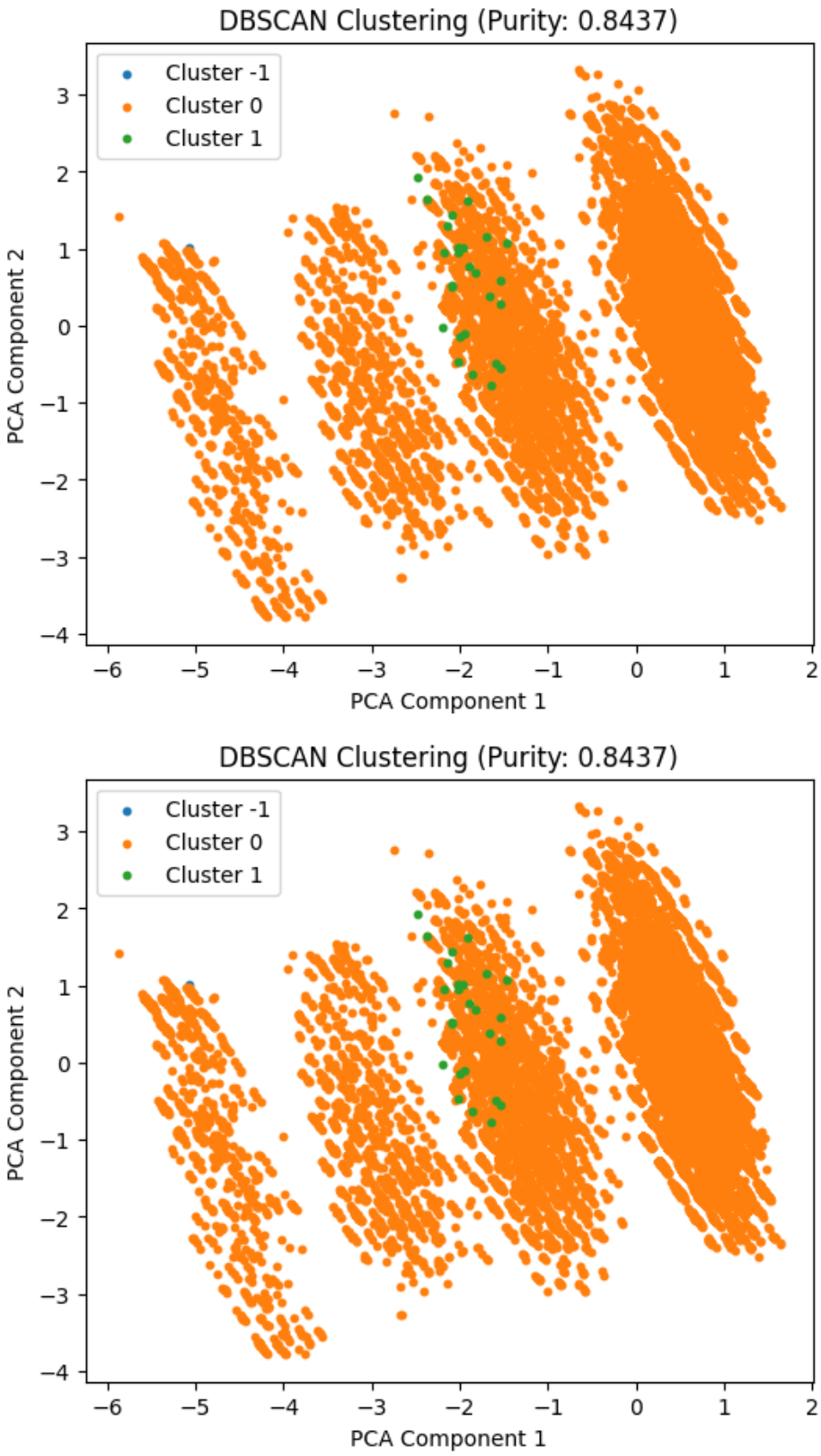

Figure 8: DBSCAN Clustering Results (Epsilon = 4, Min Samples = 5 and 10)

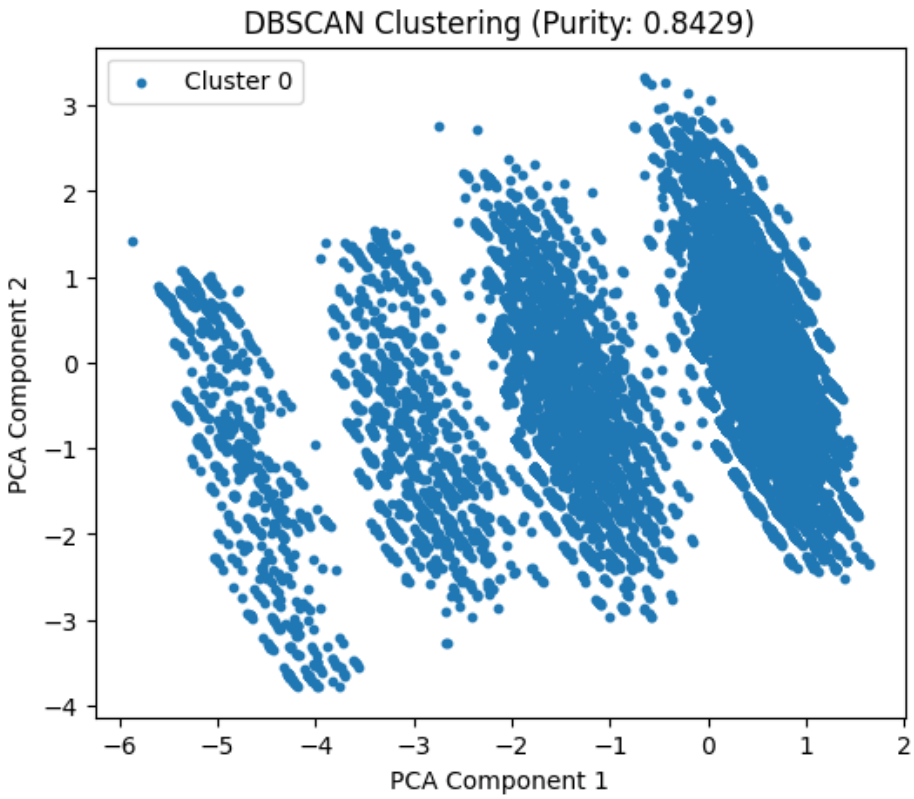

Figure 9: DBSCAN Clustering Results (Epsilon = 5, Min Samples = 5)

Interpretation:

- For small values of **epsilon** (1 or 2), DBSCAN generates an excessive number of micro-clusters, leading to **very low purity scores** (around 0.0433 to 0.49).
- As **epsilon** increases (especially at 3 and 4), DBSCAN begins to form larger, more meaningful clusters, achieving much better purity (**0.8275 to 0.8437**).
- At **epsilon = 5**, the algorithm merges all points into one cluster, collapsing the model and achieving the same purity as a trivial majority class guess.
- **Important caveat:** While DBSCAN shows high purity for epsilon = 4 or 5, this does *not* imply good clustering. The high score is driven by most data points being grouped into a single cluster, which happens to align well with the skewed distribution of the binary labels. This is a misleading outcome and reflects model collapse, not effective separation.

Conclusion:

- **KMeans** remains the superior clustering algorithm for this dataset, offering clean, high-purity clusters that align well with the binary classification task.
- Although **DBSCAN** can reach similar purity with careful tuning (e.g., epsilon = 4), it is more volatile and less interpretable than KMeans.
- Using PCA was crucial to enable effective clustering by reducing high-dimensional, categorical-heavy data into a usable form.

8 Conclusion

The modeling results demonstrate that both the regression and classification models effectively capture the observed trends in tuition payment behavior from 2022 to 2023. The features **ENROLLMENT** and **TUITION PAYMENT MARCH 2022** emerged as the most influential predictors, exhibiting strong correlation with the target variable, **Tuition Payment 2023**. Leveraging these features, Ridge and Linear Regression achieved high regression performance ($R^2 \approx 0.88$), while all classification models achieved a remarkable accuracy of approximately **98.6%**.

In contrast, alternative feature choices led to a significant drop in performance. For example, using the **Number of Enrolled Courses** as a predictor yielded poor results, with even advanced models like XGBoost and Deep Neural Networks reaching a maximum accuracy of only **45%**. This emphasizes the necessity of strong feature-target correlation for building reliable predictive models.

The clustering analysis further reinforced these findings. **KMeans clustering**—particularly with **K = 3**—achieved the highest purity score (**0.9419**), successfully uncovering structure aligned with the underlying class distribution. Although **DBSCAN** showed high purity for certain hyperparameter settings (e.g., epsilon = 4), this was a misleading result. The apparent performance was due to DBSCAN collapsing most points into a single cluster, which coincidentally aligned with the majority class label because of class imbalance. This outcome reflects a failure to identify meaningful structure rather than effective clustering.

Overall, the study highlights the central role of careful feature selection and model interpretability in achieving accurate and reliable predictions. It also underscores the importance of understanding clustering metrics in context—high purity alone does not guarantee good performance, especially in the presence of skewed data. By combining thoughtful data preprocessing, dimensionality reduction via PCA, and well-tuned modeling approaches, we obtained robust insights into tuition payment behaviors in the dataset.