Msid – Final Report

Author: Kamil Borusiak 280447

Spis treści

Part I - Preliminary Student Data Analysis – Predicting Academic Success and Dropout Risk

Introduction

The analyzed problem is the relationship between students' academic success and demographic, economic, and educational factors. Using real data from a Portuguese university, the analysis aims to determine which student characteristics and environmental factors influence decisions to graduate, drop out, or continue studies.

Dataset

The data comes from the UCI Machine Learning Repository and includes details of 4424 students, such as demographic information, education background, and economic situation. Each row represents a student and their status: graduated, dropped out, or still enrolled.

Analysis

Descriptive Statistics – Numerical Features

Attribute	Mean	Median	Min	Max	Std	5th Percentile	95th Percentile	Missing Values	Unique
Application order	1,73	1,00	0,00	9,00	1,31	1,00	5,00	0,00	8
Previous qualification (grade)	132,61	133,10	95,00	190,00	13,19	110,00	157,00	0,00	101
Age at enrollment	23,27	20,00	17,00	70,00	7,59	18,00	41,00	0,00	46
Admission grade	126,98	126,10	95,00	190,00	14,48	103,42	153,50	0,00	620
Curricular units 1st sem (credited)	0,71	0,00	0,00	20,00	2,36	0,00	6,00	0,00	21
Curricular units 1st sem (enrolled)	6,27	6,00	0,00	26,00	2,48	4,00	11,00	0,00	23
Curricular units 1st sem (evaluations)	8,30	8,00	0,00	45,00	4,18	0,00	15,00	0,00	35
Curricular units 1st sem (approved)	4,71	5,00	0,00	26,00	3,09	0,00	9,00	0,00	23
Curricular units 1st sem (grade)	10,64	12,29	0,00	18,88	4,84	0,00	14,86	0,00	805
Curricular units 1st sem (without evaluations)	0,14	0,00	0,00	12,00	0,69	0,00	1,00	0,00	11
Curricular units 2nd sem (credited)	0,54	0,00	0,00	19,00	1,92	0,00	4,00	0,00	19
Curricular units 2nd sem (enrolled)	6,23	6,00	0,00	23,00	2,20	5,00	10,00	0,00	22
Curricular units 2nd sem (evaluations)	8,06	8,00	0,00	33,00	3,95	0,00	15,00	0,00	30
Curricular units 2nd sem (approved)	4,44	5,00	0,00	20,00	3,01	0,00	8,00	0,00	20
Curricular units 2nd sem (grade)	10,23	12,20	0,00	18,57	5,21	0,00	14,98	0,00	786
Unemployment rate	11,57	11,10	7,60	16,20	2,66	7,60	16,20	0,00	10
Mother's qualification level	16,17	19,00	0,00	33,00	8,58	3,00	27,00	0,00	29
Father's qualification level	15,02	19,00	0,00	33,00	8,57	3,00	27,00	0,00	34

Sample insights from the data:

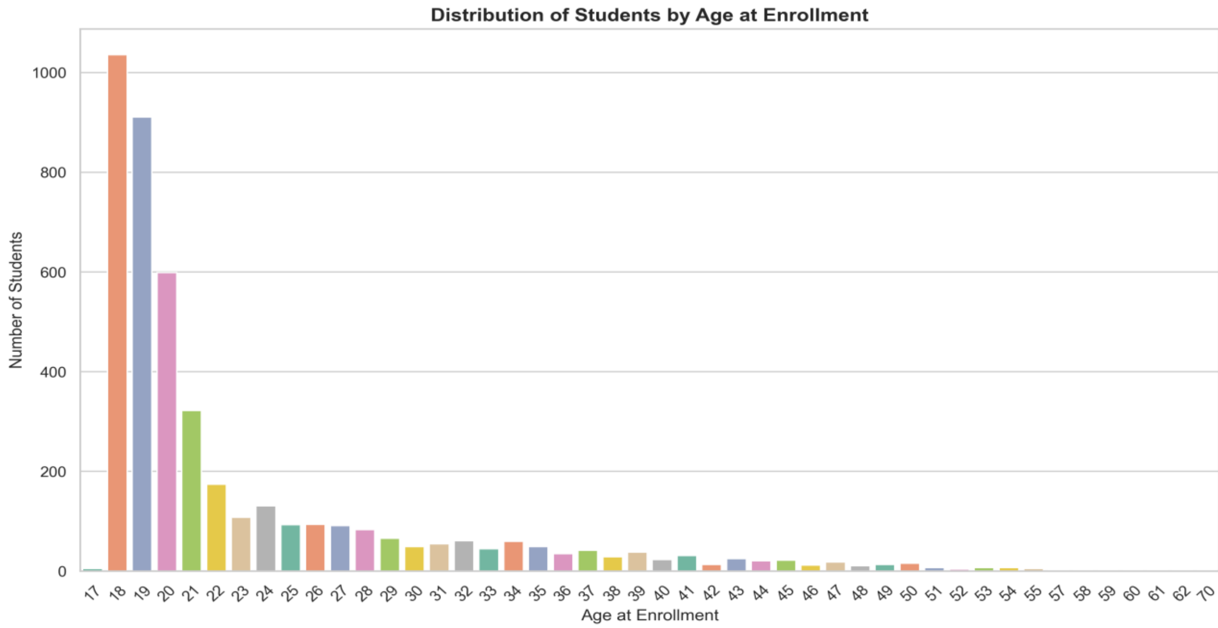
- The average age at enrollment is about 23 years.
- 5% of students are 41 or older

Descriptive Statistics – Categorical Features

Attribute	Unique	Missing	Class Proportions
Displaced	2	0	{1: 0.548372513562387, 0: 0.451627486437613}
Educational special needs	2	0	{0: 0.9884719710669078, 1: 0.011528028933092224}
Debtor	2	0	{0: 0.8863019891500904, 1: 0.11369801084990959}
Tuition fees up to date	2	0	{1: 0.8806509945750453, 0: 0.11934900542495479}
Gender	2	0	{0: 0.6482820976491862, 1: 0.35171790235081374}
Scholarship holder	2	0	{0: 0.7515822784810127, 1: 0.24841772151898733}
International	2	0	{0: 0.9751356238698011, 1: 0.024864376130198915}
Unemployment rate	10	0	{7.6: 0.12906871609403256, 9.4: 0.1204792043399638
Inflation rate	9	0	{1.4: 0.20185352622061484, 2.6: 0.1290687160940325
Course Text	17	0	{'Nursing': 0.17314647377938516, 'Management': 0.0
GDP	10	0	{0.32: 0.12906871609403256, -3.12: 0.1204792043399
Target	3	0	{'Graduate': 0.4993218806509946, 'Dropout': 0.3212
Mother's qualification category	7	0	{'Secondary Education': 0.46971066907775766, 'Basi
Marital Status Text	6	0	{'Single': 0.8858499095840868, 'Married': 0.085669
Application mode Text	18	0	{'1st phase - general contingent': 0.3860759493670
Daytime/evening attendance Text	2	0	{'Daytime': 0.8908227848101266, 'Evening': 0.10917
Previous qualification Text	17	0	{'Secondary education': 0.8401898734177216, 'Techn
Nacionality Text	21	0	{'Portuguese': 0.9751356238698011, 'Brazilian': 0.
Mother's qualification Text	29	0	{'Secondary Education - 12th Year of Schooling or
Father's qualification Text	34	0	{'Basic Education 1st cycle (4th/5th year) or equi
Mother's occupation Text	10	186	{'Unskilled Workers': 0.3721094856064181, 'Adminis
Father's occupation Text	10	443	{'Unskilled Workers': 0.2537050992213012, 'Skilled

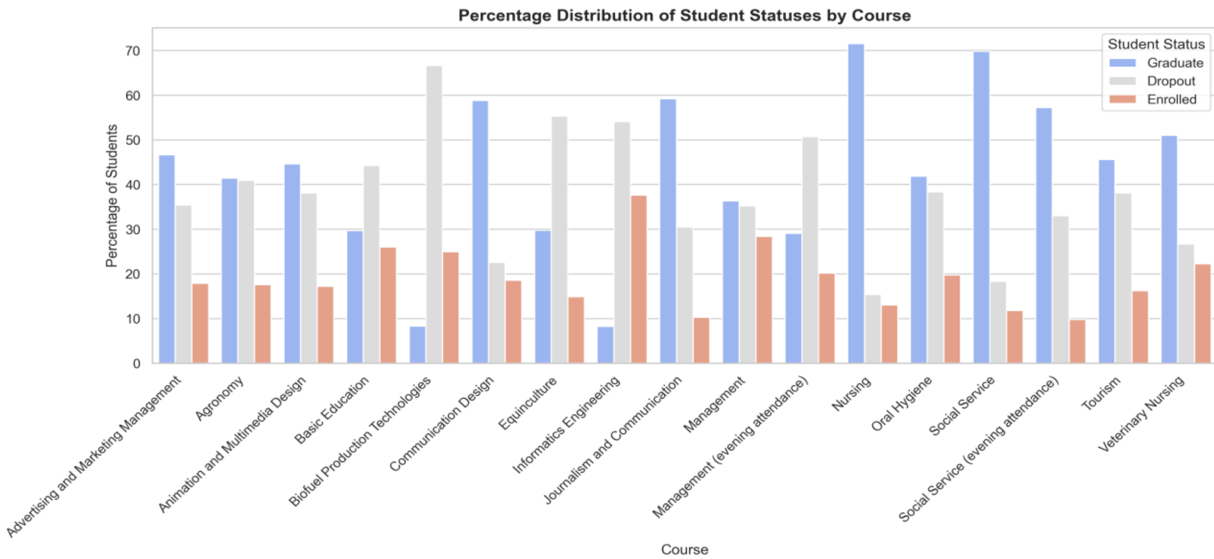
- Most students are single.
- The second most common country of origin after Portugal is Brazil.

Age Distribution of Students



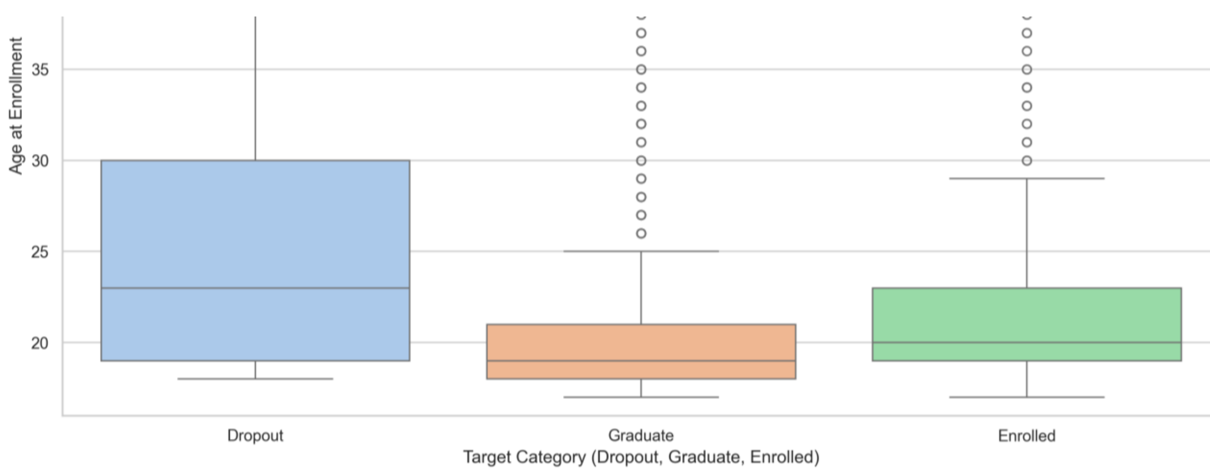
- Most students enroll between the ages of 17–19.
- Enrollment decreases sharply after age 20.
- Students aged 30+ are rare.

Status Distribution by Study Program



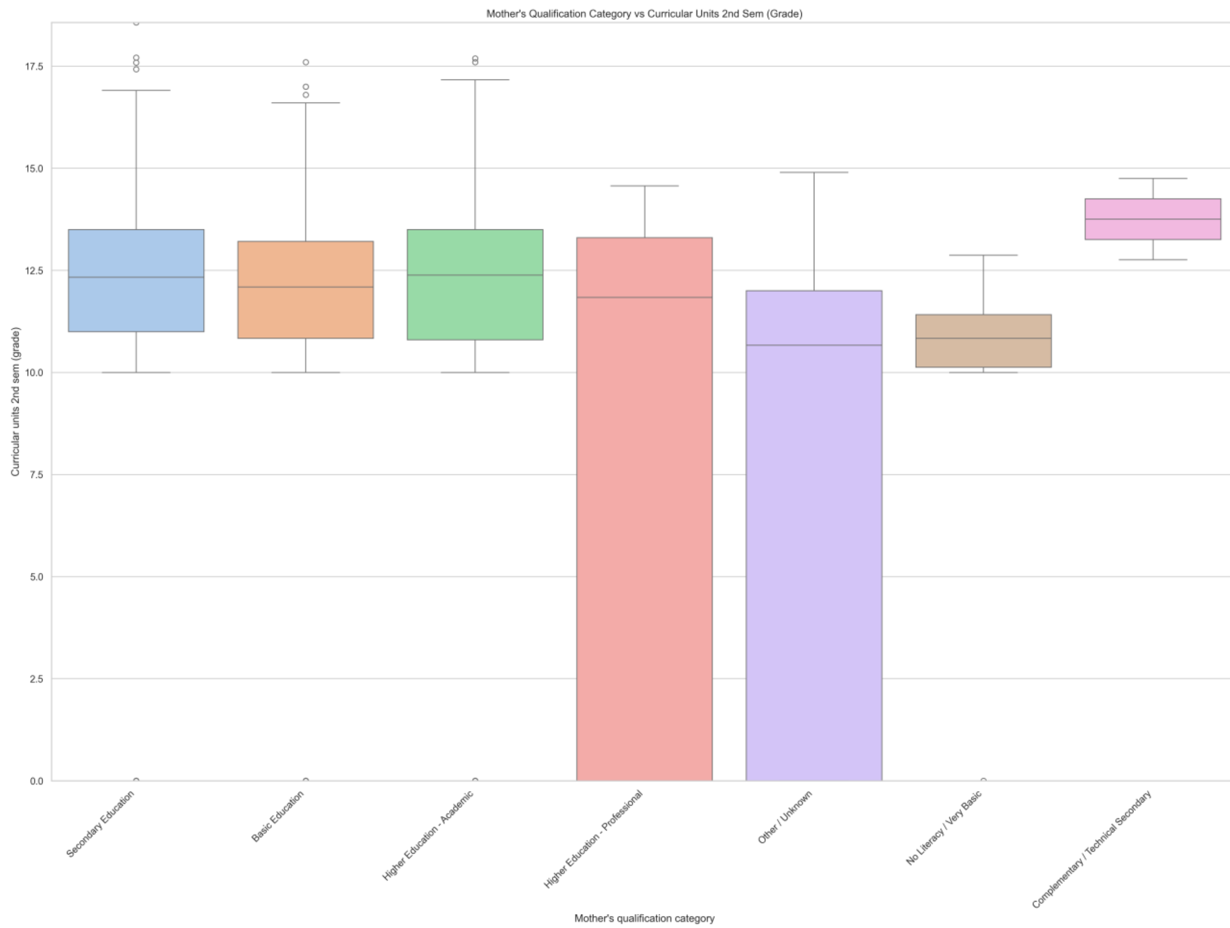
- Highest graduation rates: Nursing, Social Work, Journalism and Communication.
- Highest dropout rate: Biofuel Production Technologies.
- High proportion of ongoing studies in Computer Science, suggesting a recent interest in this field.

Age vs. Status



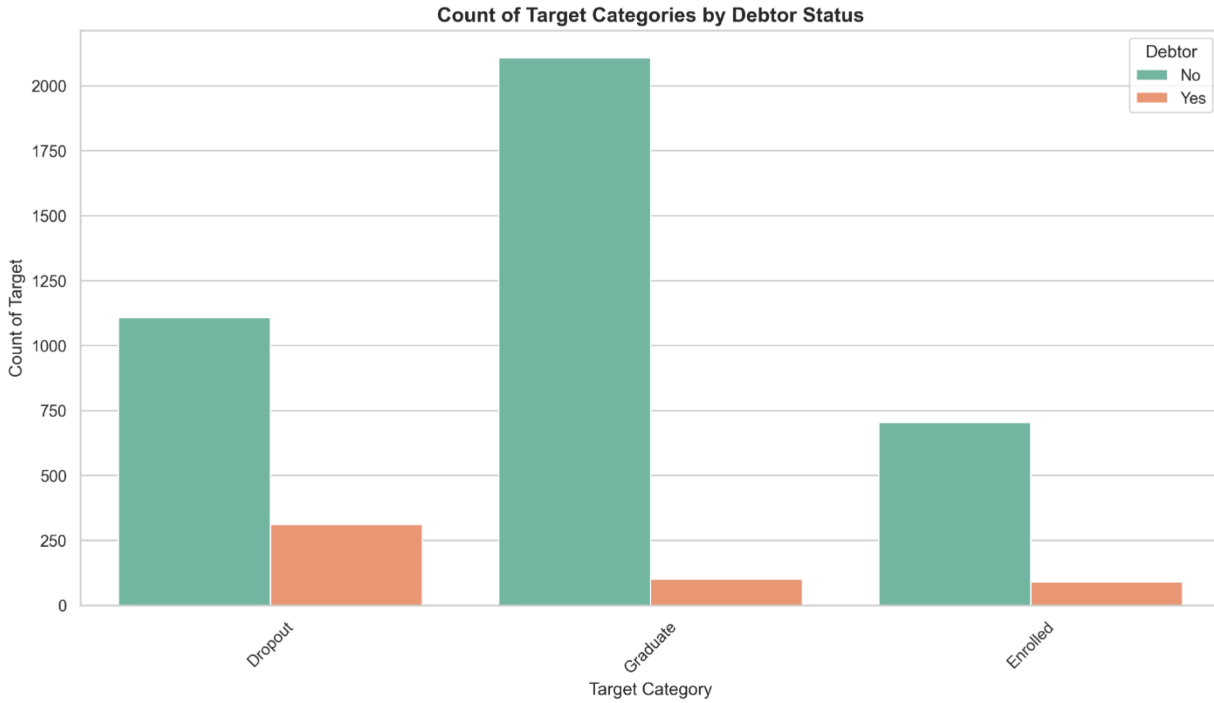
- Dropouts are generally the oldest, with the highest age median and spread.
- Graduates tend to enroll between 18–22 years, showing the lowest median and variation.
- Currently enrolled students have mixed ages.

Mother's Education vs. Number of Passed Courses



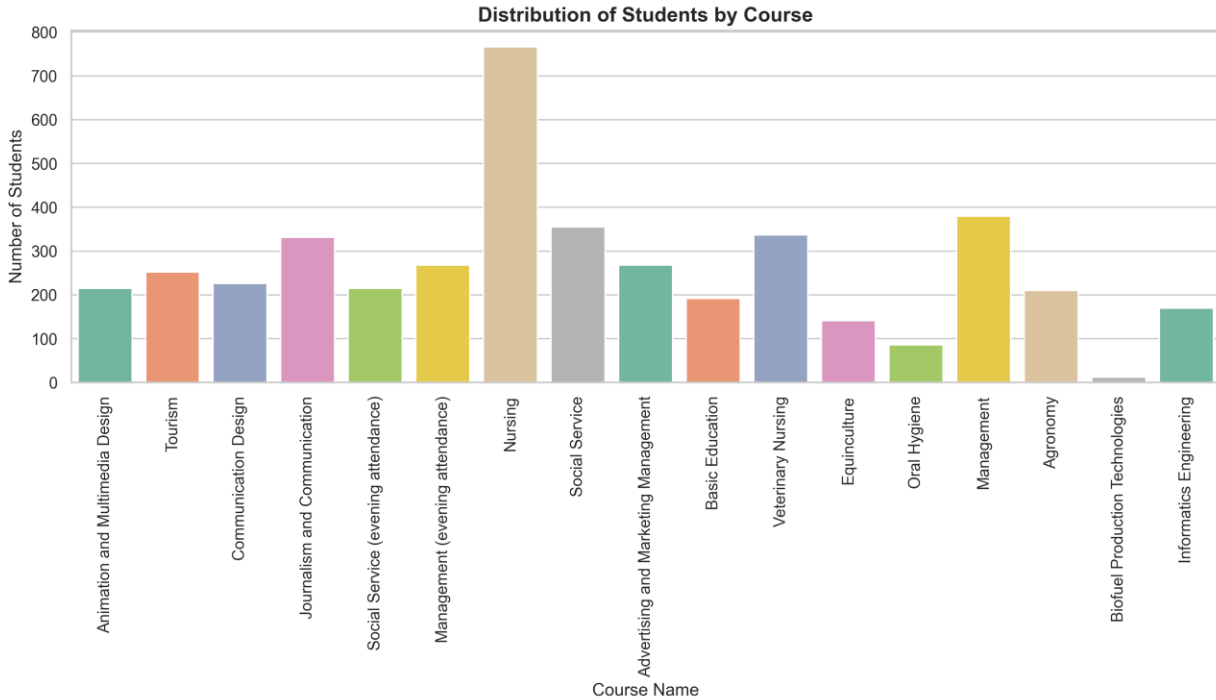
- Students whose mothers have the lowest education levels perform worst.
- Oddly, vocational school graduates underperform compared to primary education.
- Overall, mother's education level doesn't significantly affect student performance

Impact of Debt on Student Status



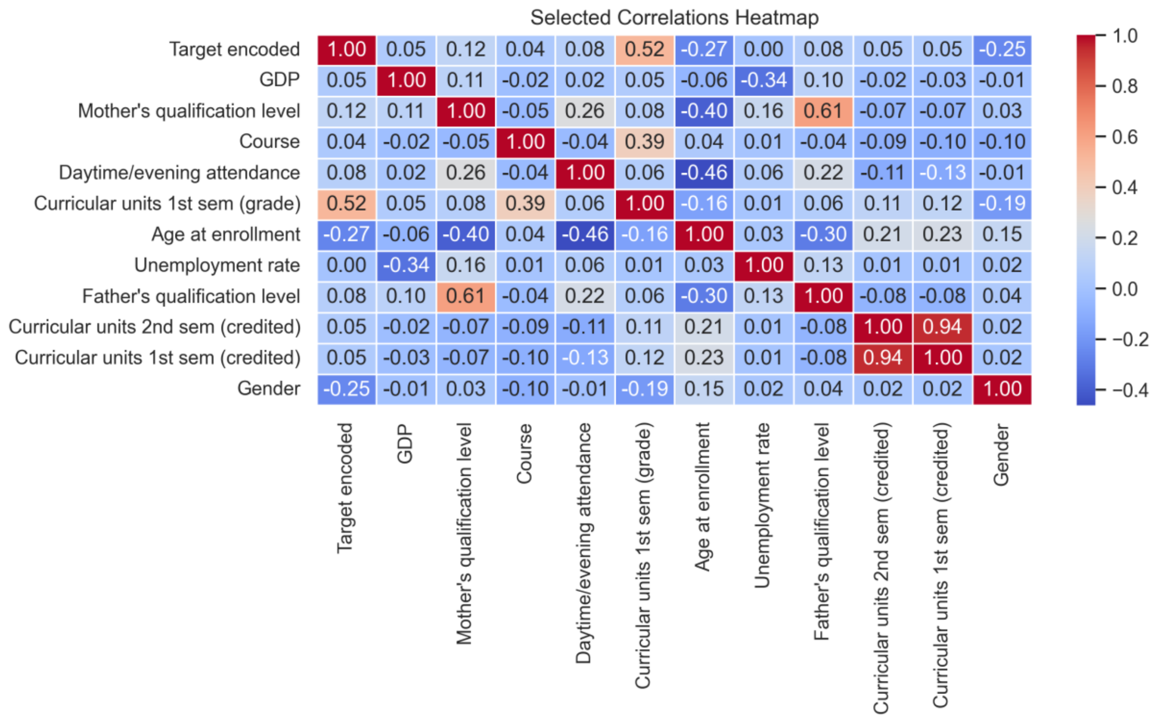
- Most graduates had no debt.
- Students with debt are more likely to drop out.
- Debt negatively affects the chances of completing studies.

Student Count per Study Program



- Nursing has the most students.
- Other popular programs include Social Services, Journalism amd Communication, Veterinary Nursing, and Management.
- Biofuel Production Technologies has the fewest students.
- Most programs have between 200–300 students, indicating balanced interest.

Correlations



Positive:

- 1st and 2nd semester passed units (r = 0.94)
- Mother's and father's qualification levels (r = 0.61)
- Graduation status and 1st semester grades (r = 0.52)

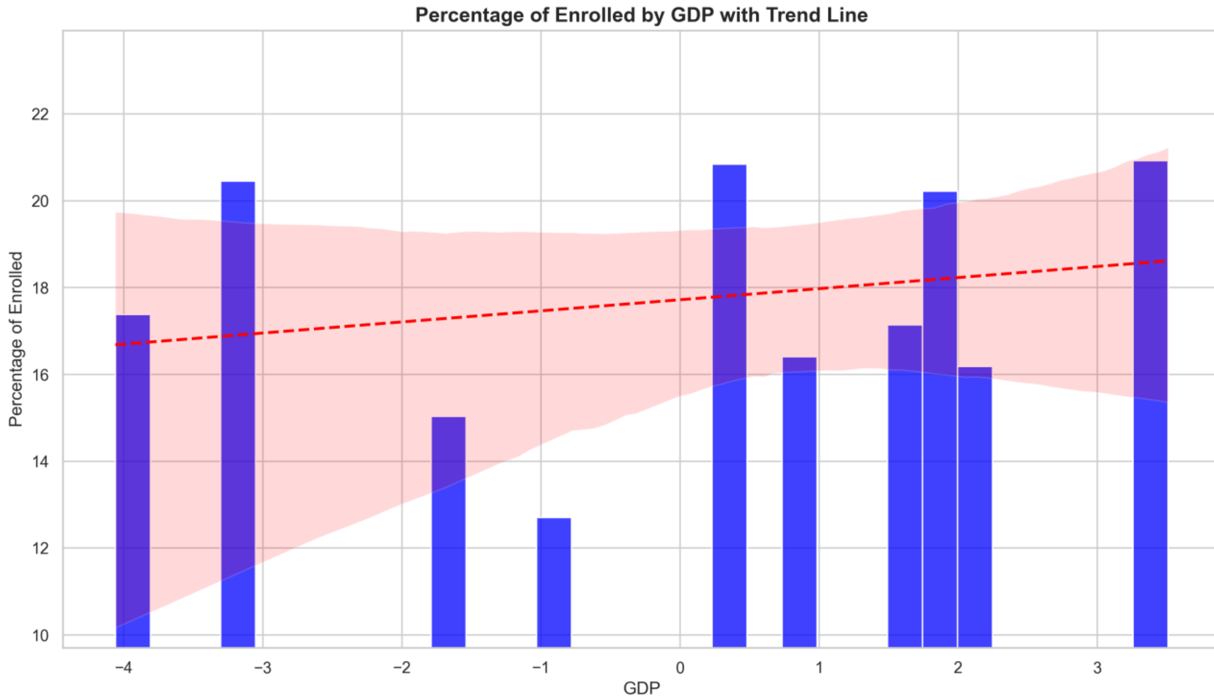
Negative:

- Graduation and age at enrollment (r = -0.27)
- Age at enrollment and daytime attendance (r = -0.46)
- Mother's qualification level and age at enrollment (r = -0.40)

Zero correlation:

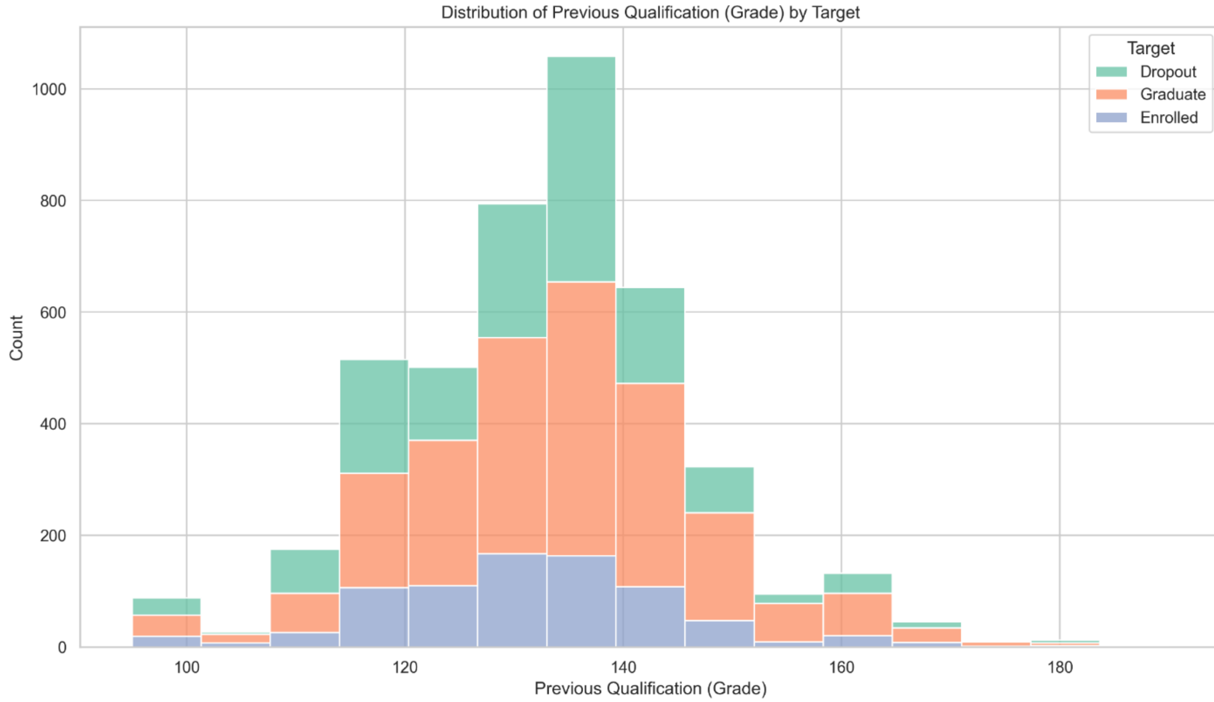
- Graduation and GDP (0.05), Course (0.04), Daytime attendance (0.08), Unemployment rate (0.00)

GDP vs. Student Enrollment Rate



- Weak positive trend: higher GDP - higher enrollment rate.
- Data is dispersed; other factors like education policy may influence outcomes.

Past Grades vs. Student Status

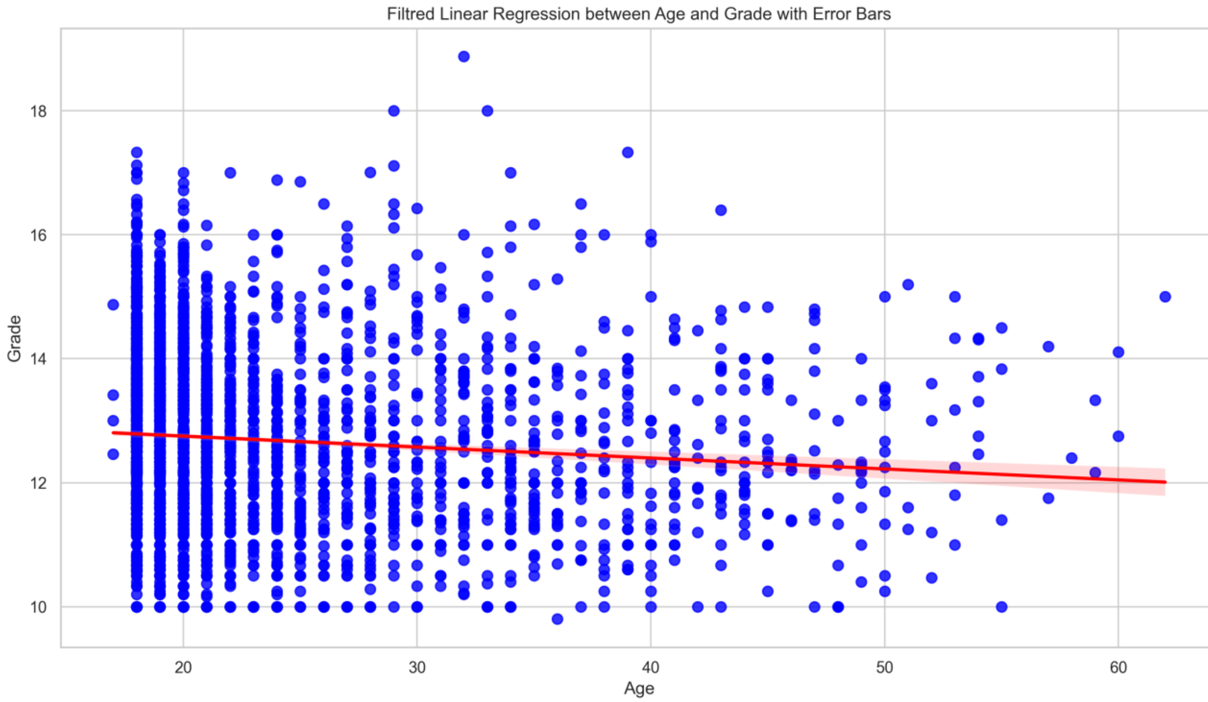


- Graduates mostly had grades between 130–160.
- Dropouts were more common with grades between 100–130.
- Enrolled students are clustered around 120–140.
- Most students have grades in the 120–140 range – considered typical entry level.

Age at Enrollment vs. Graduation Chance

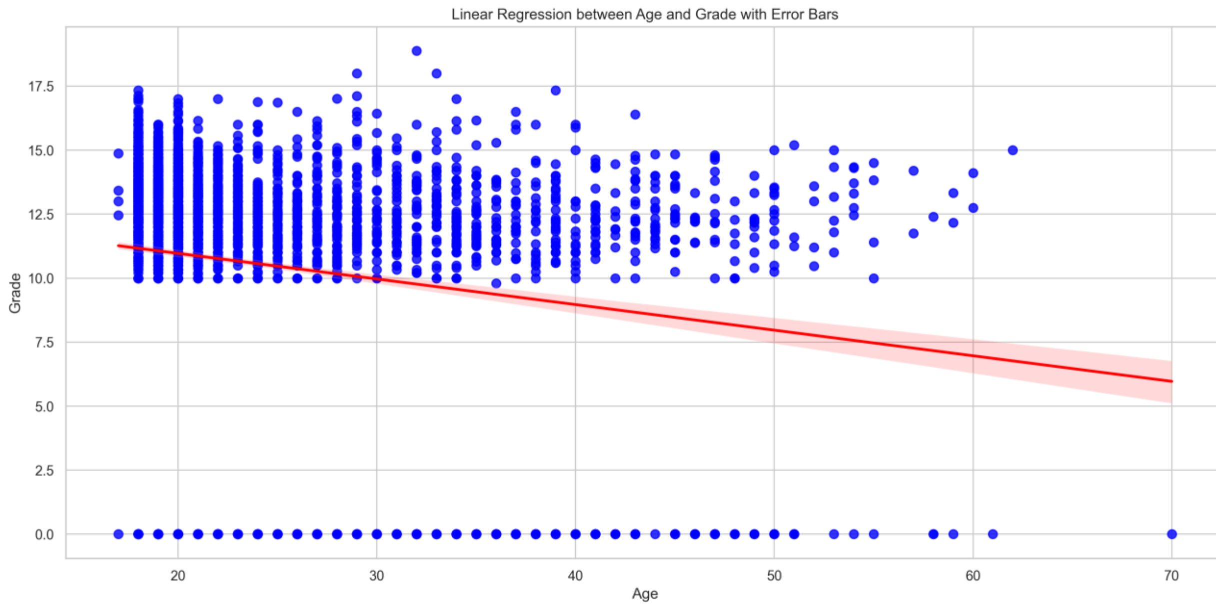
- Negative correlation: older age = lower chance of graduation.
- Students under 25 graduate more often.
- Wide confidence interval suggests variability and individual differences.

Age vs. Grades – Linear Regression



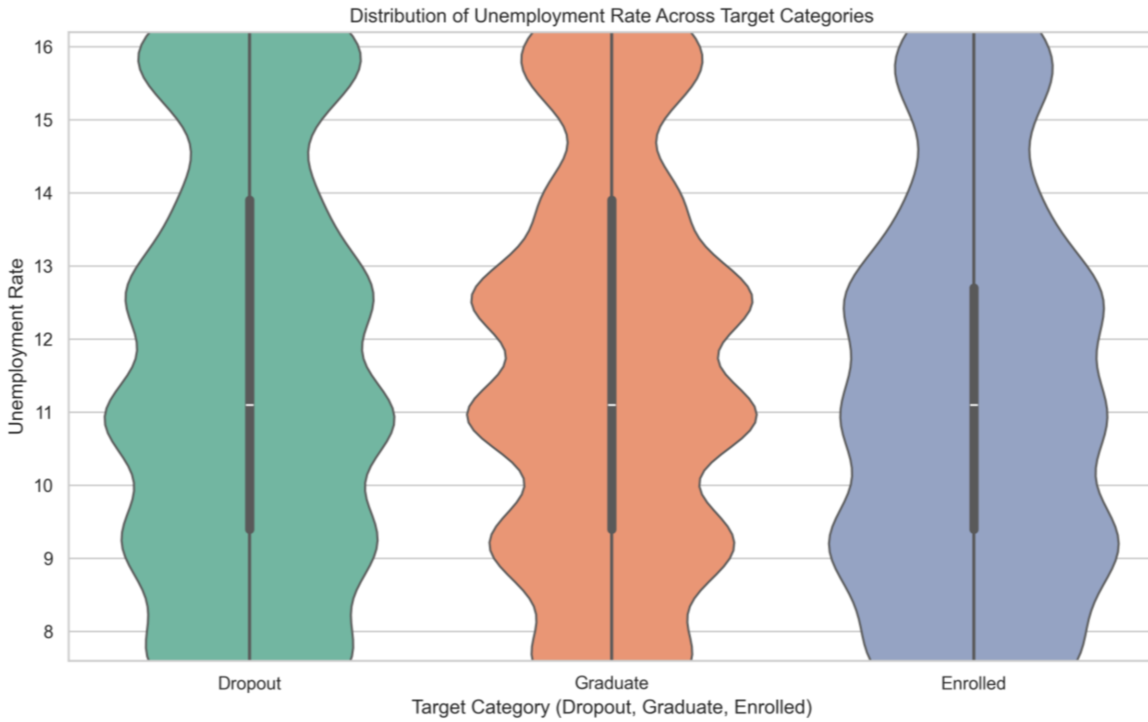
- Very weak negative correlation: older students slightly tend to have lower grades.
- Students aged 40+ perform similarly to younger peers.
- Large variability in each age group suggests age isn't a strong performance predictor.

Effect of Outliers



- Outliers distorted the regression and falsely suggested older students perform worse.
- Removing them shows age has little effect on grades.
- Highlights the importance of data quality and cleaning.

Unemployment Rate vs. Student Status

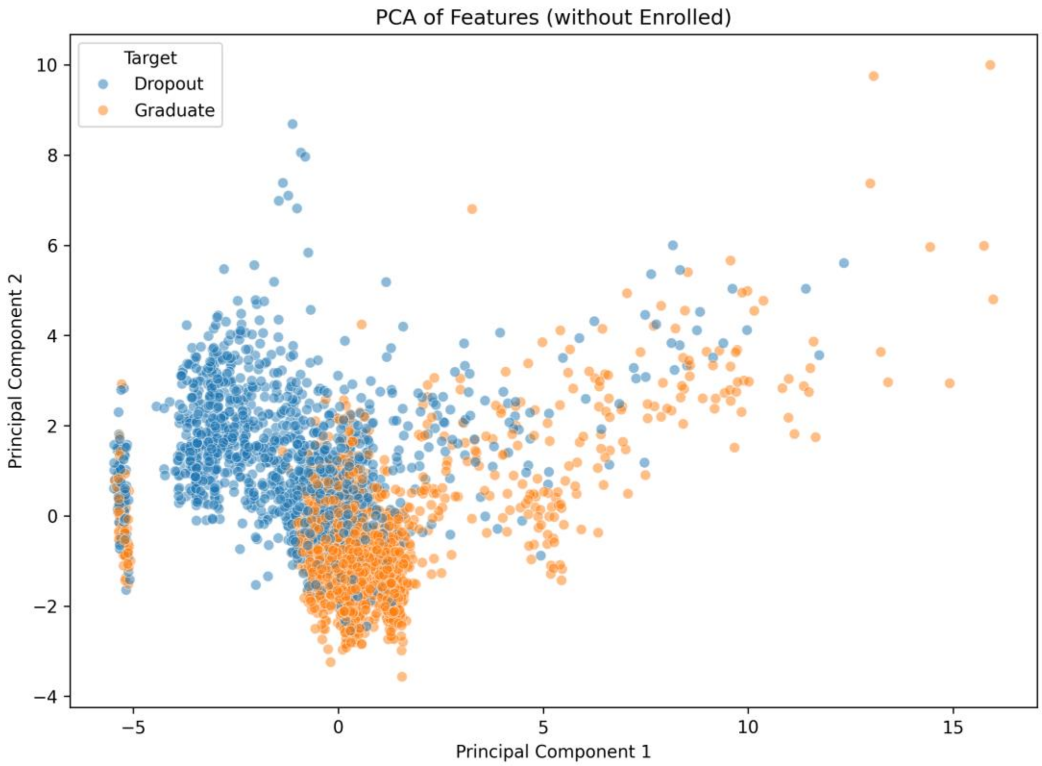


- Dropout rate increases with higher unemployment.
- Fewer graduates when unemployment is low.
- Most applications occur when inflation is ~8–10%.

1st Semester Grades by Gender

- Similar grade distributions for both genders, most scores between 11–14.
- Men more often have extremely low grades.
- Women have slightly higher median grades.

PCA Results – Visualizing Students (excluding Enrolled)



- Dropouts cluster on the left side of the plot.
- Some overlap between Dropouts and Graduates.
- Vertical line on the left indicates students with similar feature profiles.

Part II - Models training and evaluation

Introduction

The dataset concerns university students and includes information such as entrance scores, average grades, semester activity, and socio-economic details (e.g., parents' qualifications, age, unemployment rate). The goal of the analysis was to build classification models that predict whether a student will graduate ("Graduate") or drop out ("Dropout").

The data was preprocessed as follows:

- The "Enrolled" class (students still in progress) was removed.
- The target variable was encoded as 0/1.
- Numerical and categorical features were identified and prepared.
- Missing values were imputed.

Data Split

The dataset was divided into three parts:

- Training set: 70%
- Test set: 20%
- Validation set: 10%

Evaluation Metrics

The following metrics were used to evaluate model performance:

- Accuracy: The ratio of correctly predicted observations to the total observations.
- Precision: The ratio of correctly predicted positive observations to the total predicted positives.
- Recall: The ratio of correctly predicted positives to all actual positives.
- F1-score: Harmonic mean of precision and recall.
- Support: The number of actual occurrences of each class in the dataset.
- AUC: Area under the Receiver Operating Characteristic curve, measuring the model's ability to distinguish between classes.
- MSE (Mean Squared Error): The average of squared differences between actual and predicted values
- R^2: Proportion of variance in the dependent variable explained by the model (ranges from 0 to 1).

Classification Model Performance

Logistic Regression

Set	Class	Precision	Recall	F1-score	Support
Test	-1.0	0.92	0.86	0.89	217
	1.0	0.91	0.95	0.93	328
Accuracy				0.92	545
Validation	-1.0	0.93	0.83	0.87	214
	1.0	0.90	0.96	0.93	330
Accuracy				0.91	544
Train	-1.0	0.94	0.85	0.89	990
	1.0	0.91	0.97	0.94	1551
Accuracy				0.92	2541

Decision Tree

Set	Class	Precision	Recall	F1-score	Support
Test	-1.0	0.85	0.83	0.84	217
	1.0	0.89	0.90	0.90	328
Accuracy				0.87	545
Validation	-1.0	0.82	0.80	0.81	214
	1.0	0.87	0.89	0.88	330
Accuracy				0.85	544
Train	-1.0	1.00	1.00	1.00	990
	1.0	1.00	1.00	1.00	1551
Accuracy				1.00	2541

SVM

Set	Class	Precision	Recall	F1-score	Support
Test	-1.0	0.94	0.86	0.90	217
	1.0	0.91	0.97	0.94	328
Accuracy				0.92	545
Validation	-1.0	0.95	0.82	0.88	214
	1.0	0.89	0.97	0.93	330
Accuracy				0.91	544
Train	-1.0	0.98	0.86	0.92	990
	1.0	0.92	0.99	0.95	1551
Accuracy				0.94	2541

Closed-Form Linear Regression

Set	MSE	R^2
Train	0.2272	0.9617
Test	0.2266	0.9657
Validation	0.2396	0.9641

Logistic Regression Summary

Custom Logistic Regression

Set	Accuracy	F1-score	AUC
Train	0.921	0.937	0.961
Test	0.919	0.935	0.961
Validation	0.912	0.930	0.960

Scikit-learn Logistic Regression

Set	Accuracy	F1-score	AUC
Train	0.921	0.937	0.964
Test	0.916	0.931	0.962
Validation	0.906	0.925	0.960

CPU vs GPU Training Comparison

Training time is longer on GPU due to the small model size and dataset. GPU has a higher overhead from transferring data between CPU and GPU memory, and initializing CUDA kernels, which outweighs the parallel processing benefits for this simple task.

Device	Training Time	Set	Accuracy	F1-score	AUC
CPU	4.81 s	Train	0.916	0.934	0.959
		Test	0.930	0.943	0.960
		Validation	0.914	0.931	0.958
GPU	6.97 s	Train	0.917	0.934	0.959
		Test	0.930	0.943	0.960
		Validation	0.914	0.931	0.958

Part III – Optimalization

Introduction

In the final stage of the project, we focused on improving model performance through systematic optimization techniques. This included the application of regularization methods (L1 and L2) to prevent overfitting, the use of ensemble approaches (such as voting and stacking classifiers) to combine the strengths of multiple models, and the implementation of a custom Mixture of Experts strategy using KMeans clustering.

Additionally, we conducted hyperparameter tuning using grid search for selected models (e.g., logistic regression and random forest) to identify the best-performing configurations. This phase also included an ablation study, in which optimization techniques were applied incrementally to observe their individual and combined effects on model performance.

Cross-validation

To assess the stability and generalization of the model, we used **StratifiedKFold.**
Each iteration provided a breakdown of training, validation, and test accuracy, as well as precision, recall, and F1-score.

First Run:

Fold	Train Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1-score
1	0.927	0.926	0.906	0.894	0.959	0.925
2	0.925	0.921	0.906	0.897	0.955	0.925
3	0.919	0.911	0.909	0.898	0.959	0.928

Second Run:

Fold	Train Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1-score
1	0.924	0.909	0.904	0.895	0.954	0.924
2	0.915	0.909	0.919	0.907	0.966	0.936
3	0.931	0.919	0.896	0.900	0.932	0.916

Third Run:

Fold	Train Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1-score
1	0.923	0.919	0.914	0.902	0.963	0.932
2	0.930	0.895	0.912	0.897	0.967	0.931
3	0.923	0.924	0.903	0.902	0.944	0.922

- The results are **consistent and stable** across all three runs, which indicates that the model's performance is not strongly dependent on the specific training subset used.
- Both **accuracy** and **f1-score** on the test sets remain in the range of **0.90–0.93**, indicating that the model generalizes well to unseen data.

No Signs of Overfitting

- The difference between training and validation accuracy is very small indicating that the model is not overfitting the training data.
- The similarity between validation and test performance suggests that the model maintains its quality on truly unseen data.

Loss analysis

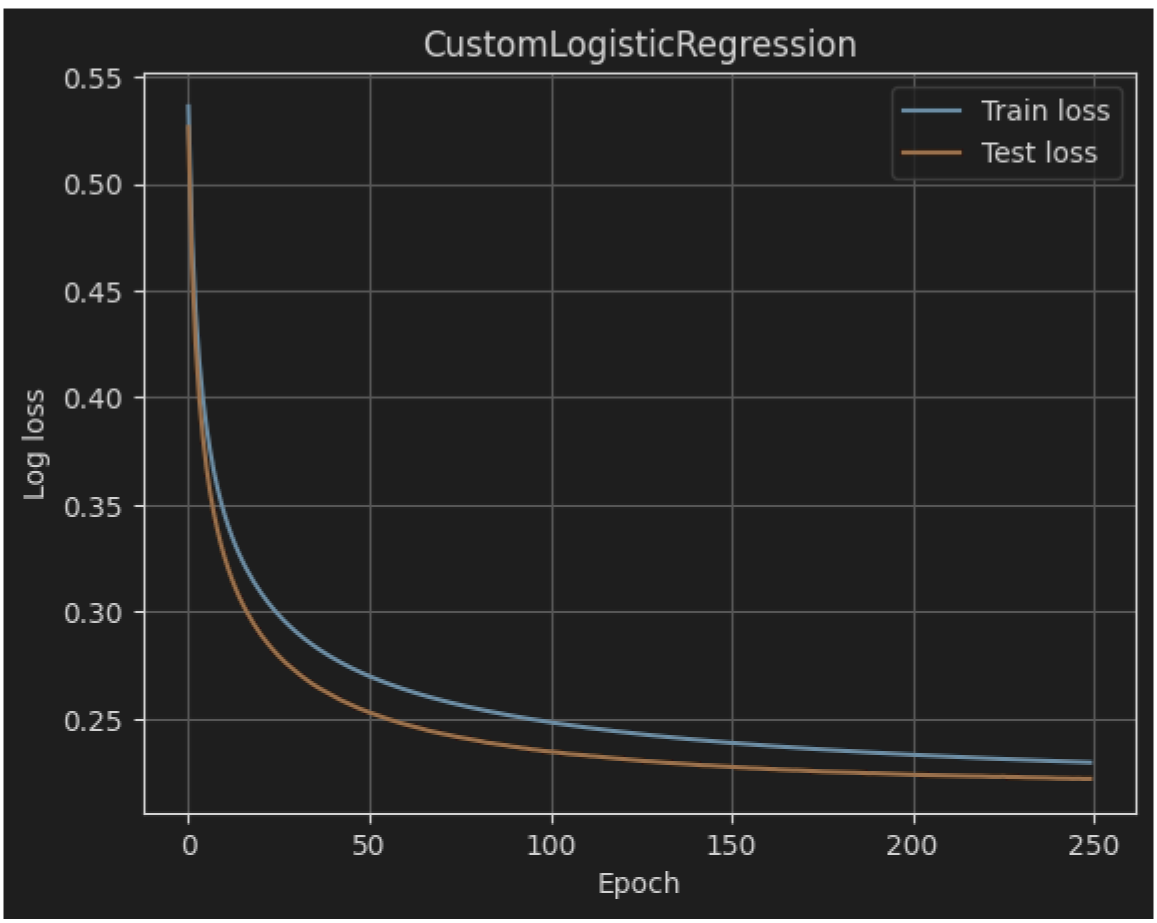
To assess the behavior of our custom logistic regression model trained with gradient descent, we monitored the **log loss** over 250 epochs on both training and validation/test subsets. Three key training scenarios were compared:

Base model (all features)

The first convergence plot shows a **steady and consistent drop in log loss** for both the training and validation sets. Both curves flatten after approximately 150 epochs, maintaining a small and stable gap. This indicates:

- Good convergence of the optimization process,
- No overfitting symptoms,
- Sufficient model capacity to capture the data structure.

This baseline model achieves a good balance between fit and generalization.



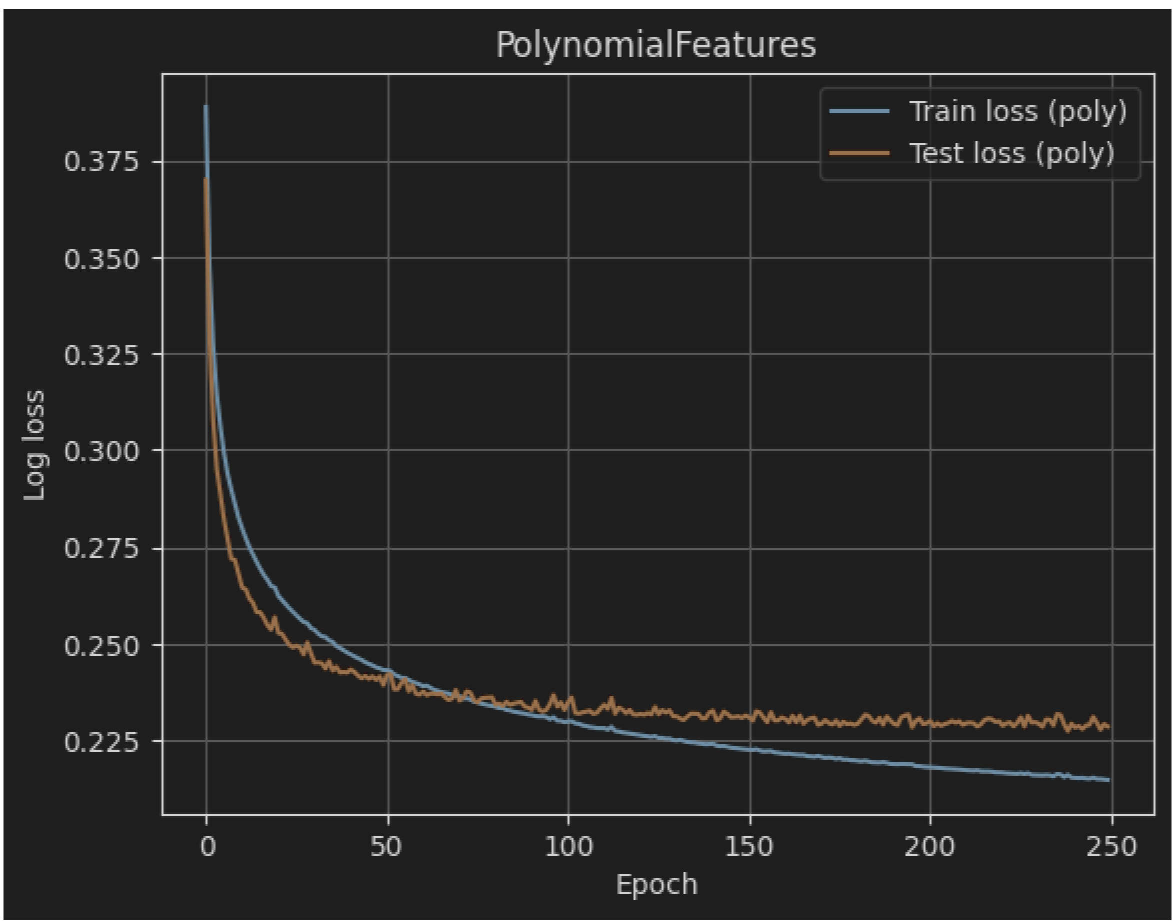
Model with PolynomialFeatures (degree = 2)

In the second experiment, we introduced **polynomial feature expansion**, increasing the number of features by including all pairwise combinations and squared terms of numeric variables.

The convergence plot reveals:

- A much steeper drop in training loss, which continues to decrease smoothly,
- A **flat, noisy test loss** that plateaus early and does not improve.

This clearly indicates **overfitting**: the model is overly flexible and fits the training data too closely, while failing to generalize to unseen data.
This demonstrates that **increasing model complexity (e.g., with PolynomialFeatures) can hurt generalization** when not controlled (e.g., via regularization or feature selection).

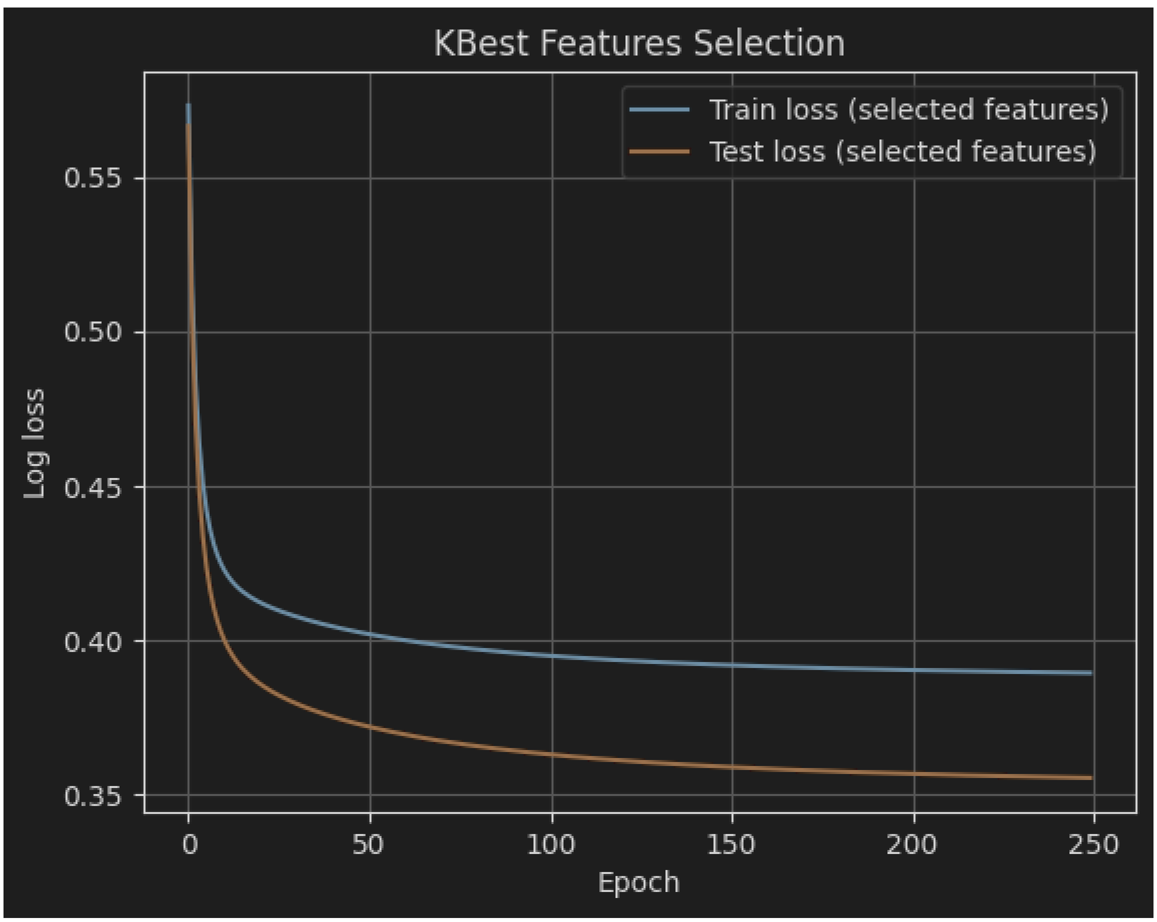


Model with selected top 5 features

In the third plot, we train the model on only the **top 5 features** selected via mutual information. This simplification leads to:

- While both training and test losses decrease steadily, they **flatten at higher levels** than in the baseline model.
- The **final test loss remains higher**, suggesting that too much relevant information may have been discarded during feature selection.

This suggests that **reducing input dimensionality can improve generalization**, especially when irrelevant or redundant features are removed.



Evaluation results for models with varying feature complexity

To complement the convergence analysis, we present the classification performance of the models evaluated on the test set:

Model 1 – Base model (all features)

Metric	Class 0	Class 1	Accuracy	Macro F1
Precision	0.9288	0.9217	0.9243	0.9196
Recall	0.8732	0.9571		
F1-score	0.9002	0.9391		

Best overall performance. Balanced and high precision and recall, indicating that full feature space enables the model to distinguish both classes effectively.

Model 2 – With PolynomialFeatures

Metric	Class 0	Class 1	Accuracy	Macro F1
Precision	0.9071	0.9127	0.9106	0.9052
Recall	0.8592	0.9436		
F1-score	0.8825	0.9279		

Slightly lower performance than the baseline. While recall for class 1 is high, class 0 recall drops. Indicates overfitting and poorer generalization despite increased complexity.

Model 3 – KBest (top 5 features)

Metric	Class 0	Class 1	Accuracy	Macro F1
Precision	0.9065	0.8732	0.8845	0.8753
Recall	0.7852	0.9481		
F1-score	0.8415	0.9091		

Lowest performance among all three. Reduced feature space appears to harm the model's ability to distinguish class 0, with recall dropping significantly. Suggests **underfitting** due to excessive simplification.

Regularization – L1 and L2

To improve model generalization and reduce overfitting, we extended our custom logistic regression implementation with **L1 (Lasso)** and **L2 (Ridge)** regularization. Both techniques penalize large weight magnitudes but operate differently:

- **L1** encourages sparsity, zeroing out irrelevant features,
- **L2** smooths and shrinks all weights without eliminating them.

Regularization terms were added directly into the gradient update rule:

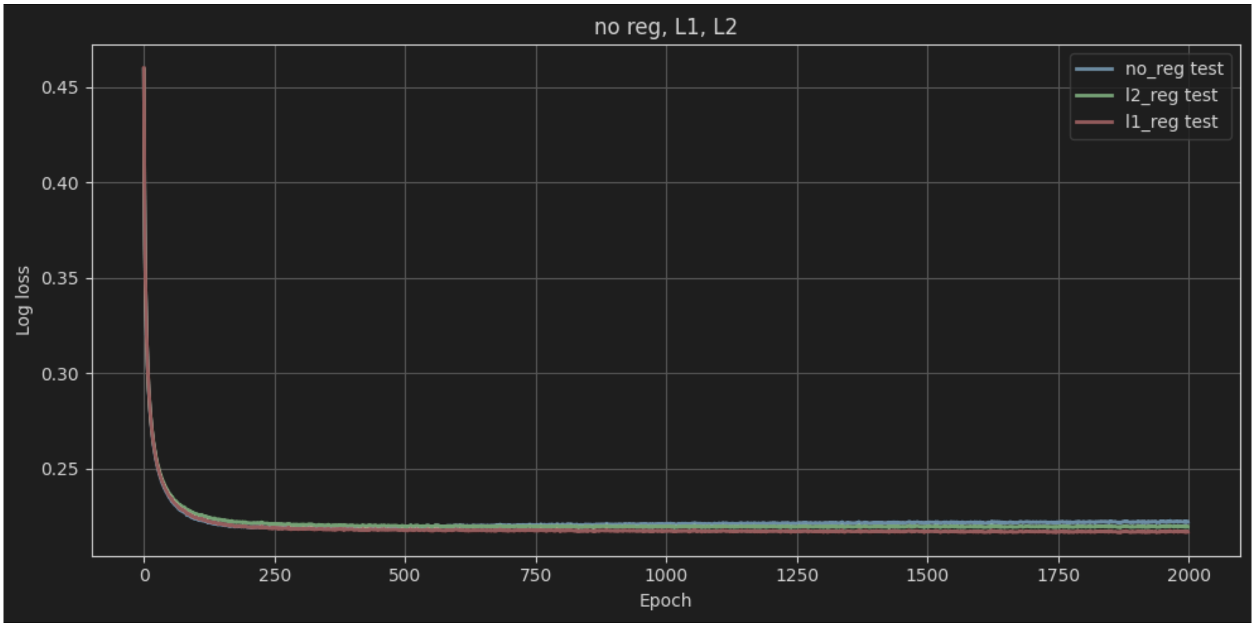
- L2: grad += λ * 2 * w
- L1: grad += λ * sign(w)

Test Loss Convergence

The convergence plot for test loss over **2000 epochs** shows that:

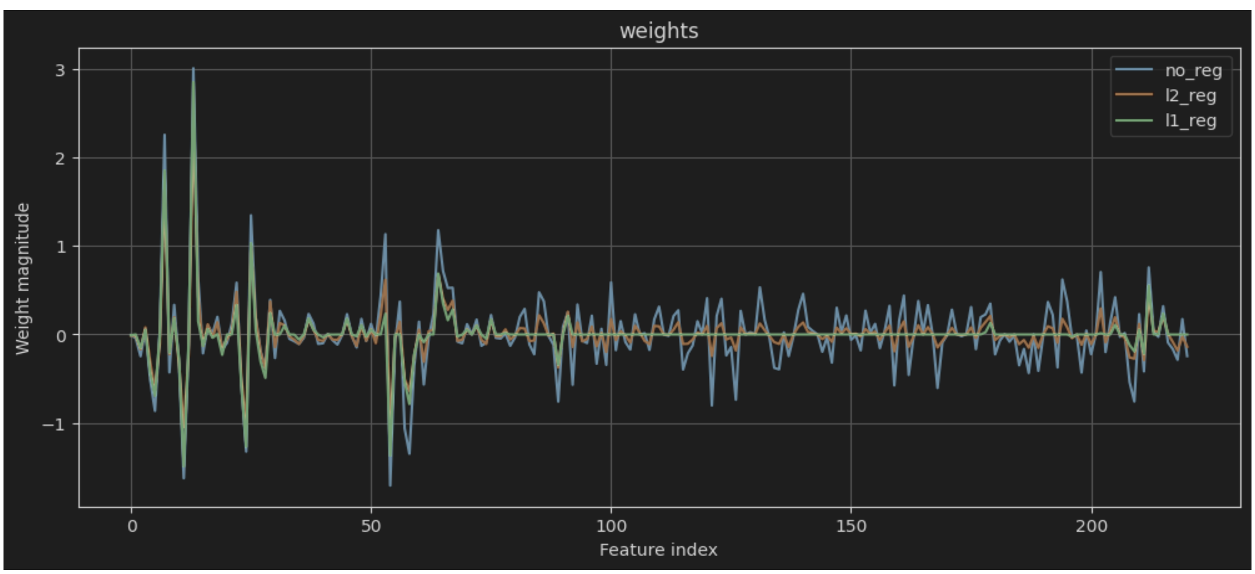
- All three models (no regularization, L1, L2) stabilize after ~250 epochs.
- The differences are subtle, but **regularized models produce smoother and more stable curves**.

Regularization helps maintain generalization and reduce fluctuations in the loss curve.



The second plot visualizes the learned weight magnitudes for each feature:

- The **non-regularized model** exhibits many high-magnitude weights, especially in less informative areas.
- **L2 regularization** keeps most weights small but still non-zero.
- **L1 regularization** significantly suppresses many weights close to zero.



Classification Results Summary

Model	Accuracy	F1-score (class 0)	Recall (class 0)
no_reg	0.9106	0.8820	0.8556
l2_reg	0.9175	0.8909	0.8627
l1_reg	0.9216	0.8958	0.8627

- Both L1 and L2 regularization **improved accuracy and F1-score** compared to the base model.
- The **L1-regularized model achieved the best overall performance**, likely due to feature selection and reduced noise.

Data Balancing – SMOTE and Undersampling

The dataset was initially **imbalanced**, with class "1" (Graduate) being overrepresented:

- Class distribution:
  - 0 (Dropout): 990
  - 1 (Graduate): 1550

To address this, we tested two common balancing techniques:

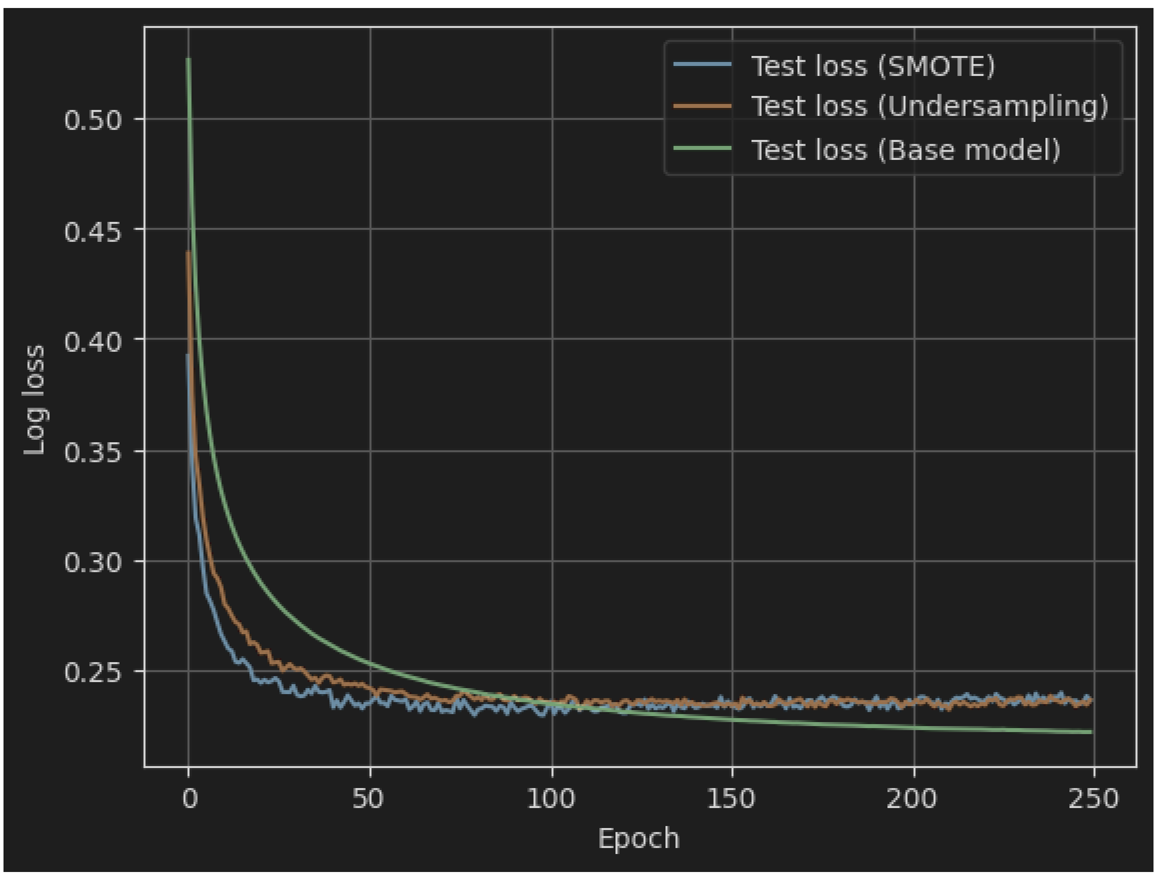
- **SMOTE (Synthetic Minority Oversampling Technique)** – oversamples the minority class using synthetic data.
- **Undersampling** – reduces the majority class by randomly removing samples.

Each model was trained using the same logistic regression architecture, and evaluated using metrics suited for imbalanced classification: **precision, recall, F1-score**, and **accuracy**.

Model	Accuracy	Recall (class 0)	F1-score (class 0)
Original (imbalanced)	0.9243	0.8732	0.9002
SMOTE (oversampled)	0.9120	0.8873	0.8873
Undersampling	0.9147	0.9014	0.8920

Observations

- The original imbalanced model **achieved the highest overall accuracy (92.4%)** and F1-score.
- **SMOTE** slightly reduced performance, likely due to **noisier synthetic examples**, although recall improved marginally.
- **Undersampling** yielded balanced recall (class 0: 90.1%) and competitive F1-score, while maintaining a solid accuracy of 91.5%.



- The **base model** (green curve) starts with a significantly higher loss and converges more slowly compared to the balanced variants.

- Both **SMOTE** (blue) and **undersampling** (orange) lead to **faster initial convergence** and lower test loss in early and mid-training stages (epochs 0–100).

While balancing improved **recall for the minority class**, it came at the cost of overall precision and F1-score. The **original model performed best overall**, though undersampling offered a viable alternative with slightly lower variance and better balance.

Hyperparameter Optimization

To further improve model performance, we conducted a **grid search** to find the optimal hyperparameters for two selected classifiers:

- **Logistic Regression**
- **Random Forest Classifier**

The optimization was performed using **GridSearchCV** with 5-fold cross-validation. The scoring metric used was the **F1-score**, as it provides a balanced evaluation for imbalanced classification problems.

Model	**Best Parameters**	**Best F1-score**
LogisticRegression	C = 0.1, penalty = 'l1', solver = 'liblinear'	0.9245
RandomForest	n_estimators = 100, max_depth = 20, min_samples_split = 5	0.9263

Hyperparameter search is challenging due to:

- **Combinatorial explosion**: the number of possible configurations grows exponentially with more parameters.
- **Training time**: each configuration requires model fitting and evaluation, which can be expensive.
- **Interactions** between parameters can be non-obvious — tuning one value might depend on the value of another.
- The optimal parameters may vary with different **datasets**, **random seeds**, or **preprocessing steps**.

Observations

- The grid search significantly improved both models' F1-scores compared to their default settings.
- Logistic Regression benefited from **L1 regularization** and a **lower C value**, which encouraged sparsity and helped reduce overfitting.
- Random Forest achieved its best performance with a **deeper tree** and more **splitting flexibility** (min_samples_split = 5).

Ensemble Methods

In this section, we explored ensemble techniques to improve model robustness and generalization. The methods tested included:

- **VotingClassifier** (majority voting),
- **StackingClassifier** (meta-model using predictions from base models),
- and a custom implementation of the **Mixture of Experts** architecture based on KMeans clustering and specialized sub-models.

Each method combined classifiers that previously made complementary errors, namely: **Logistic Regression**, **Random Forest**, and **SVC**. This was done to leverage their diverse perspectives on the data and potentially boost performance.

Model	**Accuracy**	**F1 (class 0)**	**Recall (class 0)**
Logistic Regression	0.9120	0.8836	0.8556
Random Forest	0.9108	0.8963	0.8521
SVC	0.9092	0.8782	0.8380
VotingClassifier	0.9287	0.9023	0.8803
StackingClassifier	**0.9271**	**0.9048**	**0.8873**
Mixture of Experts	0.9120	0.8845	0.8627

Analysis and Observations

- **StackingClassifier** delivered the best performance across most metrics. This indicates the effectiveness of combining heterogeneous base learners with a meta-classifier that learns how to best utilize their outputs.
- **VotingClassifier** also slightly improved performance over individual base models, showing that even simple ensemble logic (e.g., majority voting) helps stabilize predictions.
- The **Mixture of Experts** model, despite being a custom implementation, performed slightly worse than Voting or Stacking. This can be explained by:
    - The reliance on unsupervised KMeans to split the data, which may not align with optimal feature separation for classification.
    - Equal reliance on each cluster-specific expert, without a mechanism to weigh their confidence or accuracy adaptively.

Ensemble methods, particularly stacking, enhanced prediction quality by aggregating strengths of diverse models.

Best overall model

After comprehensive testing and evaluation, we developed a **final ensemble model** using a StackingClassifier that achieved the **highest accuracy of 92.98%**. The ensemble is composed of:

- **Base estimators**:
    - RandomForestClassifier with n_estimators=100, min_samples_split=5
    - LogisticRegression with L1 regularization (penalty="l1", C=0.01, solver="liblinear", max_iter=3000)
- **Final estimator**:
    - LogisticRegression (max_iter=1000)

This combination proved to be the most effective based on evaluation metrics:

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.9331	0.8838	0.9078	284
1	0.9279	0.9594	0.9434	443
Accuracy			**0.9298**	727
Macro avg	0.9305	0.9216	0.9256	727
Weighted avg	0.9300	0.9298	0.9295	727

Impact of Additional Techniques

- **PolynomialFeatures**:
  Adding polynomial features resulted in **lower overall accuracy (92.43%)**. Although it improved recall for class 1, it reduced precision and F1-score for class 0, indicating **increased complexity led to overfitting** or noise sensitivity.
- **SMOTE Oversampling**:
  Balancing the dataset using SMOTE led to **decreased accuracy (92.71%)**. Class 0 precision and F1-score dropped, suggesting that **oversampling disturbed natural feature distributions**.

- **PolynomialFeatures + SMOTE**:
  The combination of both transformations gave the **lowest performance (92.16%)**, confirming that additional synthetic data and nonlinear interactions negatively impacted generalization.

This report was created based on the dataset [“Prediction of students’ dropout and academic success”] from the UCI Machine Learning Repository.