Self-Reflection on **Predict Students' Dropout and Academic Success** Dataset Project

In this project, I aimed to predict student outcomes—whether they would drop out, continue enrollment, or graduate—using a dataset that included demographic, socio-economic, and academic factors. I implemented various machine learning models, including Logistic Regression, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Random Forest, to assess their performance in predicting student success.

# Insights

**Data Preprocessing Importance:** During the preprocessing phase, I encountered several missing values, which were addressed using mean imputation for numerical columns. The focus on proper data preprocessing was crucial, as errors during this phase can impact the entire model pipeline.

**Model Performance:** The Logistic Regression, Decision Tree, and Random Forest models performed similarly well, with an accuracy of 75.91%. This reinforced the idea that selecting appropriate models for a given dataset is important, as even simple models like Logistic Regression can perform well for binary or multi-class classification tasks.

# Skills

**Data Handling:** I worked with a diverse dataset containing categorical, integer, and real values, which required careful handling of missing data and the identification of important features for model predictions.

**Model Selection and Tuning:** I became proficient in selecting hyperparameters for various models, particularly in models like Decision Trees and Random Forests, where hyperparameter tuning (such as using ccp\_alpha for pruning in Decision Trees) helped improve model performance.

**Feature Importance Evaluation:** I was able to visualize feature importance, which helped me understand which factors (e.g., previous qualifications, age, and tuition fees) contributed the most to predicting student outcomes.

# Challenges

The most challenging aspect of this project was the data preprocessing phase, where I encountered an error that kept me stuck for hours. Despite trying various solutions, I couldn't pinpoint the issue initially. I used Gemini in Colab, which suggested fixes, but the errors persisted even after applying its suggestions. The issue stemmed from a small oversight: I forgot to set *inplace=True* while modifying the dataset, a simple but crucial detail that I missed in my haste.

# New Learnings

1. **Attention to Detail:** I learned that sometimes the simplest solutions can be the hardest to spot, especially when dealing with complex workflows. In this case, a basic one-line solution (inplace=True) solved my issue. This experience reinforced the importance of checking even the most basic steps when troubleshooting.
2. **Perseverance and Patience:** This project taught me the importance of persistence and the value of stepping back and re-evaluating when things aren't working. I realized that getting too caught up in complex solutions can sometimes obscure simpler fixes.
3. **Using Tools Effectively:** I gained experience in using external tools like Gemini to troubleshoot, though I learned the importance of applying my own critical thinking and not relying solely on suggestions.