Arquivo carregado

Analyze the provided dataset to identify its key characteristics for data-driven decision-making. Upon receiving a specific analytical question, address it strictly using the dataset. Your deliverables must include a clear, concise answer to the question and the Python code that extracts and processes the data used to derive your answer. The dataset should be cleaned by filtering admission_period to include only values between 2006.1 and 2019.2, removing entries where enrollment_type is labeled as "Exempted", and excluding those where situation is "In Progress". The resulting cleaned dataset should be stored in a new DataFrame named df_filtered.

I'll start by loading the dataset and examining its structure to understand its contents. Then, I'll apply the specified cleaning steps before proceeding with any analysis. Let's begin.

The dataset contains 30,130 entries with 34 columns. Key columns relevant to cleaning include:

- **admission_period** (float): Needs filtering to keep values between 2006.1 and 2019.2.
- **enrollment_type** (object): Entries labeled "Exempted" should be removed.
- **situation** (object): Entries labeled "In Progress" should be excluded.

I'll now apply these cleaning steps and store the resulting dataset in `df_filtered`. [↓]

The cleaned dataset, `df_filtered`, now contains **16,563** entries after applying the filtering criteria. It is ready for analysis. What specific analytical question would you like to address? [↓]

Assess whether a student's exit mode (graduated or dropped out) can be predicted using features such as the number of credits taken, type of enrollment, and course status. Propose three predictive modeling approaches and justify the optimal approach using performance metrics.

To assess whether a student's **exit mode** (Graduated vs. Dropped Out) can be predicted using features like **credits taken, type of enrollment, and course status**, I'll follow these steps:

1. Data Preparation

- Define the target variable (`exit_mode`) as "**Graduate**" vs. "**Dropout**".
- Select relevant features:
  - **credits** (total credits taken)
  - **enrollment_type** (categorical)
  - **situation** (categorical course status)
- Encode categorical variables appropriately.
- Split the data into training and testing sets.

2. Model Selection

I'll consider three predictive modeling approaches:

1. **Logistic Regression**:
   - Suitable for binary classification.
   - Provides interpretable feature importance.
   - Assumes a linear relationship between predictors and the log-odds of the outcome.
2. **Random Forest**:
   - Captures nonlinear relationships.
   - Handles interactions between features well.
   - More robust to missing data and categorical variables.
3. **Gradient Boosting (XGBoost)**:
   - Often outperforms other models in structured data.
   - Strong at capturing complex feature interactions.
   - Can handle imbalanced datasets effectively.

3. Model Evaluation

- **Accuracy**: Measures overall correctness.
- **Precision & Recall**: Evaluate class-specific performance.
- **F1-Score**: Balances precision and recall.
- **ROC-AUC**: Measures model discrimination.

Now, I'll proceed with data preprocessing and modeling.

It looks like the training process took too long, particularly for the **XGBoost** model. I'll optimize the computation by reducing the number of trees for **Random Forest** and **XGBoost** to speed up execution while maintaining meaningful results. Let me retry the process. [↓]

The computation is still taking too long, especially for **XGBoost**. To proceed efficiently, I'll exclude XGBoost and focus on **Logistic Regression** and **Random Forest**, which are computationally lighter. Let me retry the process with just these two models. [↓]

Model Performance Comparison

Metric	Logistic Regression	Random Forest
Accuracy	61.2%	62.5%
Precision	53.6%	54.5%
Recall	69.5%	74.3%
F1-Score	60.5%	62.9%
ROC-AUC	62.3%	64.0%

Optimal Model Selection

- **Random Forest performs better** across all metrics, especially in recall (+5%), which is crucial for identifying students at risk of dropping out.
- It captures **nonlinear relationships** and **feature interactions** better than Logistic Regression.
- While computationally heavier, it provides better classification performance.

Conclusion

Random Forest is the optimal approach for predicting **student exit mode** based on credits, enrollment type, and course status. [↓]