



Este é uma cópia de uma conversa entre ChatGPT e António.
[Denunciar conteúdo](#)

📎 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 and inspecting the dataset to understand its structure. Then, I'll apply the necessary cleaning steps based on your criteria. Let's begin.

The dataset contains 30,130 entries and 34 columns. The key columns relevant to our filtering process are:

- admission_period (float) – needs to be between 2006.1 and 2019.2.
- enrollment_type (object) – needs to exclude "Exempted".
- situation (object) – needs to exclude "In Progress".

Now, I'll clean the dataset by applying these filters and store the result in df_filtered. [~]

The dataset has been successfully filtered, reducing it to 16,563 entries. Now, you can provide a specific analytical question, and I'll address it strictly using this cleaned dataset. [~]

Classify students into performance tiers (high, medium, low) using final averages and course workload. Apply a classification method and assess its classification accuracy.

To classify students into performance tiers, I'll follow these steps:

1. Define Performance Tiers:
 - High: Final Average ≥ 8.0
 - Medium: $6.0 \leq \text{Final Average} < 8.0$
 - Low: Final Average < 6.0
2. Prepare Data for Classification:
 - Use Final Average and Workload as features.
 - Drop rows with missing values in these columns.
3. Apply a Classification Model:
 - Use a simple classifier (e.g., Decision Tree) for interpretability.
 - Split data into training and testing sets.
4. Evaluate Classification Accuracy:
 - Use accuracy score and confusion matrix.

I'll now implement these steps.

The Decision Tree classifier achieved 100% accuracy on the test set. The confusion matrix shows perfect classification across all three performance tiers:

$$\begin{bmatrix} 861 & 0 & 0 \\ 0 & 1031 & 0 \\ 0 & 0 & 1090 \end{bmatrix}$$

This suggests that Final Average and Workload are highly effective in distinguishing between the performance tiers. The perfect classification might be due to the strong correlation between Final Average and the defined performance tiers, making it an almost deterministic classification task. [~]

Envie uma mensagem para o ChatGPT

📎 Anexar

🗣️ Voz

© ChatGPT pode cometer erros. Considere verificar informações importantes.