

## Data Cleaning & Preparation : Corrected Version Report.

Authors: Peter Amoye, Erick Banegas, Alhassane Samassekou

**Implicit Problem:** The initial project had fake numeric order because we used implicit encoding (.cat.codes) instead of OneHotEncoder (handle\_unknown='ignore') and therefore our code never encoded categorical variables; we only did manually. What we find out is that it imposed fake numeric order on nominal fields like gender, lunch\_type, or parent\_education.

In the corrected version, we applied OneHotEncoder to convert each category into its own binary column, removing any false ordering. handle\_unknown='ignore' ensures the model doesn't crash when new categories appear later.

**Data Leakage (Fitting Before Split):** In session two of the coding, we find out that we cleaned and visualized the entire dataset first and when we later fitted scalers/encoders before splitting, we leaked information. We corrected that, and we learned that splitting before applying encoders or scalers prevents the model from "seeing" data from the test set. It guarantees honest and realistic evaluation.

**Feature Scaling :** In session three, we discovered that feature scaling is missing for Logistic Regression/SVM because we never standardized numeric columns, which is crucial for algorithms like Logistic Regression or SVM. No scaling or normalization was applied, and hence large-valued columns dominate the model.

We corrected this by applying StandardScaler() which gives numeric columns zero mean and unit variance, so all features contribute equally.

**Feature Engineering :** The Feature Engineering has limited information in the initial project, which made it difficult to capture performance balance between STEM and language subjects. To correct this, we enriched the features by adding total\_score and stem\_bias to help capture performance balance between STEM and language subjects. This gives the model more meaningful patterns.

**Pipeline Integration:** There was a missing preprocessing pipeline; all cleaning was manual. In the corrected version, we integrated a pipeline to keep the workflow consistent and prevent data leakage.

This also ensures that transformations learned on training data are applied exactly the same way to new data.

**Model Output Issue:** In conclusion, we had a complete dataset failure, because it was not cleaned properly, everything in the model predicting 0.5 as a result of data mis-encoded (fake order) features not scaled and leakage confused the model. We cleaned the pipeline by splitting, applying OneHotEncoder and StandardScaler, which fixes all three issues.