# REPORT PROJECT: Essay quality prediction.

**PROJECT objective:** Develop an end-to-end Machine Learning Pipeline

GROUP 4: Stéphanie AGBODJOGBE, Karl DO SANTOS ZOUNON, Hortis DOUMAKPE, Marie-Laure JOVIAL, Nada NAFIE
TUTOR: Assan SANOGO

Project available on github: Nadanafie1 (github.com)

# EXPLORING THE DATA

By **reading the README FIRST documents** that explain the dataset we started by creating a depurated dataset by **selecting only the useful columns** for exploratory analysis.

We kept the column: essay id, essay set, essay and domain1_score, the latter being our target.

Thanks to these documents we already knew that **the essays were distributed in 8 groups, focusing on different themes, written by students in different grades and graded by considering several items.**

We first try to understand the elements below:

- Make sure that there were no blank values or crucial data missing.
- Repartition of the number essays across the different essay set
- Distribution of the essays grade within the different essay set
- What were the @WORDS that seemed to appear in every essay?

These are the **first obvious traits that emerge** after some overall analysis:

## Trait 1: DIFFERENCE GRADING SCALE

Each **essay set having its own rating scale** means that 2 essays with the same grade but from 2 different **essay set are not comparable** which could lead to misinterpretation. In addition, the scales range can be totally different from one another. The smaller range goes from 0 to 4 and the largest from 0 to 60.
*(For e.g. essay set 1 is rated from 0 to 4* and essay set 8 is rated from 0 to 60, so an essay rate 4 can be considered exceptionally good or bad whether it is from essay set 1 or essay set 8).

To avoid this interpretability problem, **we will later normalize the column 'domain1_score' to put all grade on a scale between 0 and 1.**

## Trait 2: IMBALANCE ON ESSAY REPARTITION

A pie chart representing the quantity of essays within each essay set made us realize that essay **set 8 was seriously underrepresented in our dataset.** In average each essay set was containing about 12% of total essays (i.e. 1500+ essays) except essay set 8 which was 6% of our data.

Consequently, a model might have a hard time learning to properly rate essays coming from essay set 8 as it will be less trained on such data.

**To fix this imbalance, we will later use SMOTE and random under sampling to set each essay set category to 1200 essays**.

# SCALING

- SCALING THE TARGET COLUMN

We initially hesitated between standardization and normalization but went with normalization as the objective was to have all grades between a range of 0 and 1.

First, we normalized the entire column domain1_score independently of the essay set category meaning that the Xmax and Xmin applied during normalization was the same for all essays. But this method led to 85% of the essay being located between 0 and 0,2 on a scale ranging from 0 to 1.

**For this reason, we decided that the Xmax and Xmin considered during normalization should be specific to the essay set the essay belongs to. So, we normalized the target within its own range.**

# BALANCING THE DATA

The dataset will be **separated into 2 parts as essay set 8 has to be oversampled with SMOTE and essay set 1 to essay set 7 must be under sample**. Given that they will be appended, we must make sure that there is no discrepancy between them.

For SMOTE, we would need to vectorize or encode some data. And knowing that SMOTE cannot oversample a class that only contains one element, we must make sure that no grade contains only one essay

To limit the risk of discrepancies we applied all necessary transformation to the total dataset before splitting.

### *STEP 1: ADDING 'FORSMOTE' COLUMN AND DATA ENCODING*

We have grouped the essays from essay set 8 into 7 different groups to avoid any notes being isolated. We **encoded the nominal columns** using cat code when we wanted to keep the idea of hierarchical order('grades'), target encoding when the original column only had an identification purpose('essay_set), and tf-Idf when we wanted to have a vector reflective the content ('essay').

### *STEP 2: SPLITTING THE DATA*

Before performing any sampling transformation, we had to s**plit the data into training set and test set.**

To have a better repartition of our data between the training set and the test set, we have decided to use the stratify parameter on the target. As our target has become a continuous variable after standardization, and could not be used as a **stratify parameter**, we had to create a new column where we group our grades ranging from 0 to 1 into 10 bins. We called this column 'for grades.

- ### OVERSAMPLING ESSAY SET 8 CATEGORY

We therefore separated the essays in set 8 from the train dataset to increase the data. We agreed to increase or decrease the essays set to have 1200 essays in the 8 categories. On this basis and wanting to **keep the distribution of notes within each category** the same, we decided to calculate the percentage of each note in the current dataset and the number corresponding to this percentage in our target dataset using a function. This calculation is based on the 'forSMOTE' column, which helps us get around the problem of note uniqueness. This function returns the percentage of each 'SMOTECateg' and a dictionary containing the corresponding number of this 'SMOTECateg' in our target dataset.

After that, we moved on to SMOTE, taking the 'forSMOTE' column as the target, the rest of the dataset as features and our dictionary as the sampling strategy parameter. We thus obtained our 1,200 essays. while retaining the original distribution of 'SMOTECateg' (notes).

- ### UNDERSAMPLING ESSAY SET 1 TO ESSAY SET 7 CATEGORIES

We use random under sampling to address class imbalance in essay set_1 to 7 to avoid deleting essays that were already in an underrepresented grade interval within its essay set.

## CREATING A BASE MODEL/ RESULT OBTAIN

For this Machine Learning project, we need to predict the scores of each essay. We are therefore in a regression problem. We have therefore chosen to make a choice between 3 different models in terms of the way they predict. Our final choice will be based on the model with the best score.
We had done a LinearRegression(), RandomForestRegressor() and XGBOOST.
We chose to use *LinearRegression()* and *RandomForestRegressor()* because of their simplicity and ability to learn data as large as our dataset.

In RandomForestRegressor we chose the hyperparameter *Random_state = 100* to set the seed for the random generator so that we can ensure that the results that we get can be reproduced. Splitting dataset in train and test is randomized so we would get different data assigned to the train and test data but random_state help us to control the random factor.
XGBRegressor is one of the optimized models in regression problems. We use it because it performs on umbalancing dataset.

For Base Model approximately the average is around 53%. That means that with our model 1 prediction is true 1 prediction out of 2 is right.
Also, we use other metrics to be convinced about the performance of the model. Those metrics are the MSE and $R^2$.

- Mean Squared Error (MSE): This is the average of the squared differences between predicted and actual values. The lower the MSE, the better the model and we got 0,03

- Coefficient of determination (R-squared, $R^2$): This measures the proportion of the variance of the dependent variable that can be predicted from the independent variables. An $R^2$ close to 1 indicates a model that explains the variance of the data well. And we got 0,55

## FEATURE ENGINEERING

As we are in an NLP project, we try to get the most information about grading essays in general and about the items specifically graded in our essays thanks to the explanatory documentation provided for each essay set. For example, as a basic feature, we did consider the number of words and sentences being relevant as according to the student grade and the assignment, the range considered acceptable can vary. Ater get many variables based on semantics and vocabulary we checked the covariance between each of them and the target and selected the best one to make the final model.

In this part we plot a heatmap to visually represent data correlation between each column. It provides a quick and intuitive way to identify trends or relationships in large datasets.

## MODEL/RESULT /RECOMMANDATIONS for fine-tuning.

### MODEL AND RESULT

Our final model is a *XGBOOSTRegressor ()* as it was the one that performed the best during the base model trials. This time we got 65% of accuracy with a mean squared error of 0,02and $R^2$ equal to 0,65. After making feature engineering we progress from 55% for accuracy to 65% with a smaller mean squared error meaning that even when it predicts a false grade it is closer to the correct one than before. $R^2$ progress from 0,53 to 0,65 meaning that adding our features as variables helped the model predict the target.

### RECOMMANDATIONS

IMPROVE FEATURE ENGINEERING

Our $R^2$ not being optimal shows that the explanation of the target is only partially explained by our features. For a better result, one should create features exploring other aspects of the text such as readability, sentence structure, content analysis, theme similarity, etc.

## CHANGE ON MODELS

- HYPER PARAMETERS MODIFICATION

Modify hyperparameter like Number of Trees, Scale Pos Weight or Max Depth to help the model learn more accurately from the training data.

Increasing the number of trees and the Max depth parameters could help the model capture existing and non-obvious patterns in the data.

By changing the scale Pos Weight parameter, which is a parameter to handle imbalanced dataset, we could have given more weight to the essay_set- grades that were not as present as others.

- EXISTANCE OF MORE SUITABLE MODEL

We use the model that seems the most appropriate and easily understandable for everyone but here might exist better and more complex models to handle our prediction.