TITLE PROJECT : Essay quality prediction

About : The objective is to create a model predicting the grade of an essay.

- **READ ME FILE**

TITLE : Essay quality prediction

ABOUT : The objective is to create a model predicting the grade of an essay.

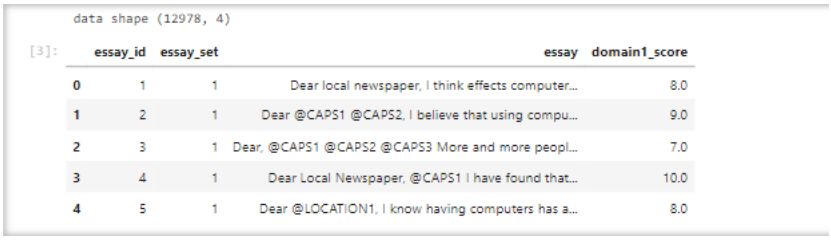
INSTALLATION : our project was built in an environment using the packages mentioned in the text file 'requirement.txt'. This can be easily installed using the command line pip install -r REQUIREMENTfile.txt

DATA DESCRIPTION

- **essay_id**: A unique identifier for each individual student essay
- **essay_set**: 1-8, an id for each set of essays
- **essay**: The ascii text of a student's response
- **rater1_domain1**: Rater 1's domain 1 score; all essays have this
- **rater2_domain1**: Rater 2's domain 1 score; all essays have this
- **rater3_domain1**: Rater 3's domain 1 score; only some essays in set 8 have this.
- **domain1_score**: Resolved score between the raters; all essays have this
- **rater1_domain2**: Rater 1's domain 2 score; only essays in set 2 have this
- **rater2_domain2**: Rater 2's domain 2 score; only essays in set 2 have this
- **domain2_score**: Resolved score between the raters; only essays in set 2 have this
- **rater1_trait1 score - rater3_trait6 score**: trait scores for sets 7-8

```
Index(['essay_id', 'essay_set', 'essay', 'rater1_domain1', 'rater2_domain1',
       'rater3_domain1', 'domain1_score', 'rater1_domain2', 'rater2_domain2',
       'domain2_score', 'rater1_trait1', 'rater1_trait2', 'rater1_trait3',
       'rater1_trait4', 'rater1_trait5', 'rater1_trait6', 'rater2_trait1',
       'rater2_trait2', 'rater2_trait3', 'rater2_trait4', 'rater2_trait5',
       'rater2_trait6', 'rater3_trait1', 'rater3_trait2', 'rater3_trait3',
       'rater3_trait4', 'rater3_trait5', 'rater3_trait6'],
      dtype='object'): hypothesis section (not obligatoire)
```

(subpart in assumption) Code chuck : example of data I am using . And this is what Y my model had to assess (



data shape (12978, 4)

[3]:

	essay_id	essay_set	essay	domain1_score
0	1	1	Dear local newspaper, I think effects computer...	8.0
1	2	1	Dear @CAPS1 @CAPS2, I believe that using compu...	9.0
2	3	1	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7.0
3	4	1	Dear Local Newspaper, @CAPS1 I have found that...	10.0
4	5	1	Dear @LOCATION1, I know having computers has a...	8.0

FUNCTIONALITY :

We approached this project by correcting the several imbalances by using random under sampling and SMOTE.

We also encode our dataset with tf-idf encoder, target encoder , cat.cod encoder and count.vectorizer encoder.

We then create our model with *XGBOOSTRegressor*.

As result we got : 65% of accuracy | mean squared error (MSE): 0,02and | R^2 : 0,65.

Before feature engineering, with our *XGBOOSTRegressor* we had an accuracy score of 55%.

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.

ENHANCEMENT :

Future step to improve our performance score could reside in:

- Improving our features

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.

- Changin our model hyper parameters

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.

- Totally changing 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.

REFERENCES :

Below a list of resources used during this project especially for dealing with imbalance and choosing a model

https://ichi.pro/fr/standardiser-ou-normaliser-exemples-en-python-250626184732156

https://www.geeksforgeeks.org/stratified-sampling-in-pandas/

https://www.youtube.com/watch?v=irHhDMbw3xo&ab_channel=DataSchool

https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html

https://machinelearningmastery.com/xgboost-for-imbalanced-classification/

https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e

https://www.youtube.com/watch?v=irHhDMbw3xo&ab_channel=DataSchool (Title: How do I encode categorical features using scikit-learn?)

- **FILE WITH ORIGINAL DATASET**
- **FILE REQUIREMENT TXT**
- **FINAL JUPYTER NOTEBOOKS**