Step-by-Step Data Quality Augmentation, Bias Mitigation, and Implementing Fairness Metrics using Synthetic data

The aim of this report is to explain the functionalities of the software modules related to the deliverable on bias mitigation and fairness metrics of the TIME project, the successful WomenTechEU submission of Clearbox AI. The open source work done during the span of the project can be found in the following repositories:

1. Structured Data Profiling library of Clearbox AI: https://github.com/Clearbox-AI/StructuredDataProfiling
2. Bias Mitigation and Fairness module of Clearbox AI: https://github.com/Clearbox-AI/Bias-Detection-Mitigation

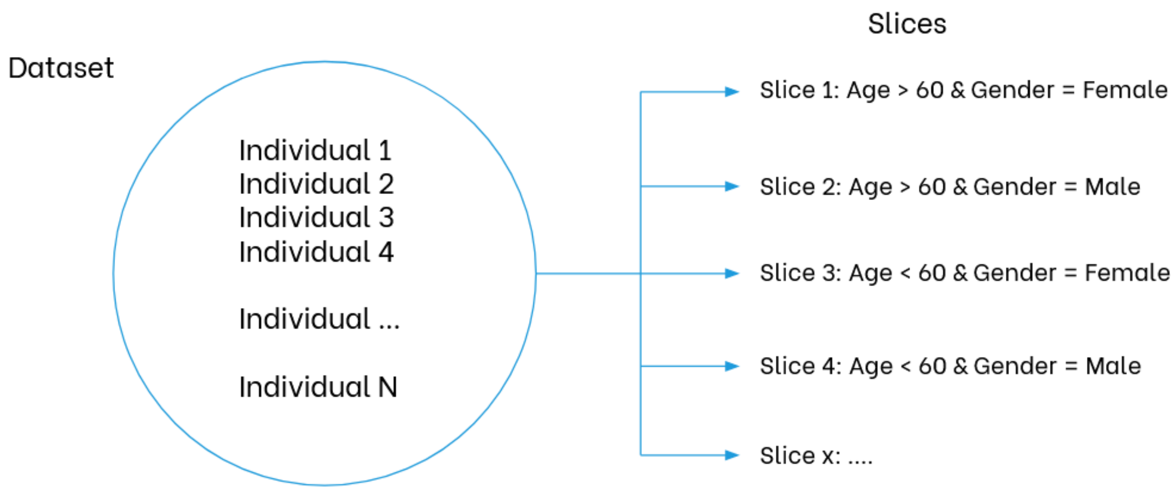
Introduction

Machine Learning models can efficiently detect patterns within historical data, making them a practical choice for decision-making tools. Unfortunately, historical data is often affected by issues that can propagate into unacceptable model behavior. Such issues can arise, for example, because of the underrepresentation of certain labels or the presence of noise for certain data segments. Data augmentation corresponds to detecting and mitigating our dataset's issues by generating synthetic examples.

Step 1 - Data profiling

The first step towards effective data augmentation requires quantitatively identifying our dataset's specific problems. It is not a trivial task as most machine learning metrics, such as MSE or F-1 score, provide data scientists with a global picture of the situation, i.e. they are not very useful at highlighting model limitations concerning different data points.

A good approach to highlight data issues reflecting a machine learning model performance is to start reasoning in terms of data slices. This corresponds to analyzing each metric for subgroups of points defined by certain rules or characteristics. These subgroups are usually defined as data slices. Data slices are data partitions obtained according to specific queries, as shown in the following Figure.

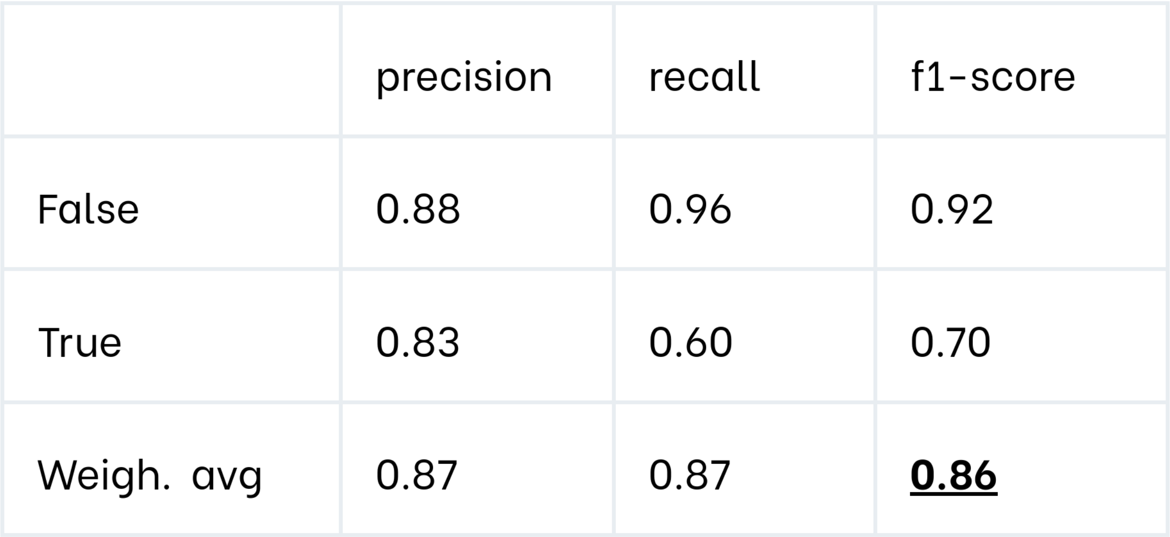

Dataset
Individual 1
Individual 2
Individual 3
Individual 4
Individual ...
Individual N
Slices
Slice 1: Age > 60 & Gender = Female
Slice 2: Age > 60 & Gender = Male
Slice 3: Age < 60 & Gender = Female
Slice 4: Age < 60 & Gender = Male
Slice x: ....

Performing such slicing and calculating metrics over each slice will give us a much more granular overview of our dataset's issues. Once we possess this information, we can augment our dataset by targeting specific data slices, making them less noisy or more represented within the whole dataset.

An example

To better understand the concepts introduced so far, let's look at a practical example of detecting and mitigating data issues using synthetic data. We will use a binary classification dataset used in the marketing domain to predict churn rates. The task is to predict whether an individual will terminate a subscription based on their characteristics. It's a mixed-type dataset as it includes both numerical and categorical variables.

To set up this example, we trained an XGBoost classifier. We first analyze model performance on the whole dataset, calculating a classification report. The next table shows the metrics achieved on hold-out test data.

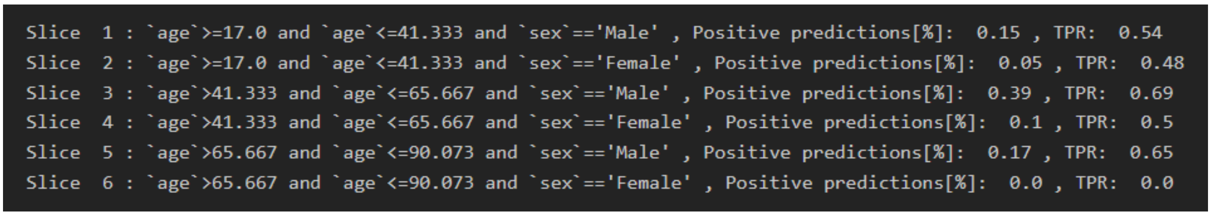


	precision	recall	f1-score
False	0.88	0.96	0.92
True	0.83	0.60	0.70
Weigh. avg	0.87	0.87	**0.86**

Table 1. Accuracy metrics for our original model.

On the surface, the global metrics appear to be already fairly OK. However, we want to dig deeper into the dataset by analyzing the model performance on several data slices.

We used the slicing tool from Clearbox AI's Structured Data Profiling library to do so. The tool can be automatically used to generate data slices that contain protected attributes. These slices are defined as SQL-like queries. We then calculated the fractions of positive predictions and the True Positive Rates for each slice, as shown in the next figure.



```
Slice  1 : `age`>=17.0 and `age`<=41.333 and `sex`=='Male' , Positive predictions[%]:  0.15 , TPR:  0.54
Slice  2 : `age`>=17.0 and `age`<=41.333 and `sex`=='Female' , Positive predictions[%]:  0.05 , TPR:  0.48
Slice  3 : `age`>41.333 and `age`<=65.667 and `sex`=='Male' , Positive predictions[%]:  0.39 , TPR:  0.69
Slice  4 : `age`>41.333 and `age`<=65.667 and `sex`=='Female' , Positive predictions[%]:  0.1 , TPR:  0.5
Slice  5 : `age`>65.667 and `age`<=90.073 and `sex`=='Male' , Positive predictions[%]:  0.17 , TPR:  0.65
Slice  6 : `age`>65.667 and `age`<=90.073 and `sex`=='Female' , Positive predictions[%]:  0.0 , TPR:  0.0
```

As seen from these slices, the model performs poorly on some data slices describing certain types of individuals. For example, we can see that the model never assigns positive predictions to women older than 65, even when it should. It means that we identified a potential model limitation. With this information, we could either decide not to use the model for certain data slices or try to investigate whether data augmentation could come to our help.

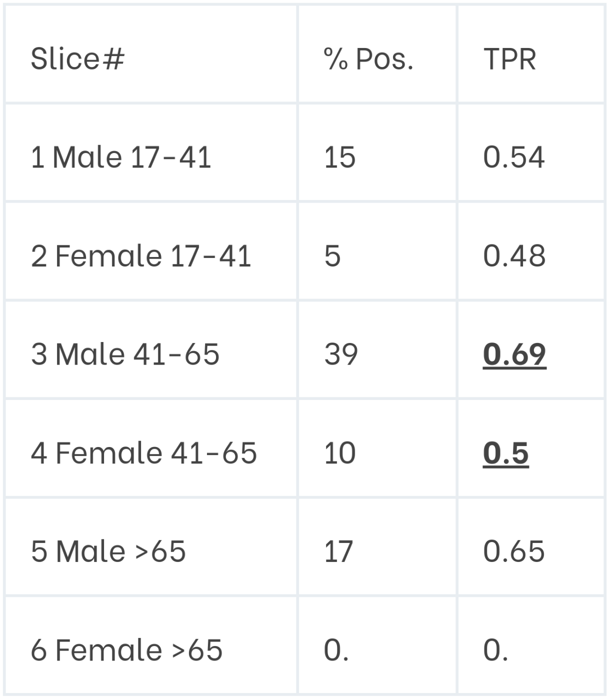
Step 2 - Mitigating data issues with synthetic data

We want to mitigate this aspect while still using the dataset. In this case, we can use synthetic data to inject our own bias into the dataset.

We can create additional examples of older women canceling their subscriptions to mitigate the fact that the model did not learn to make good predictions on this data slice.

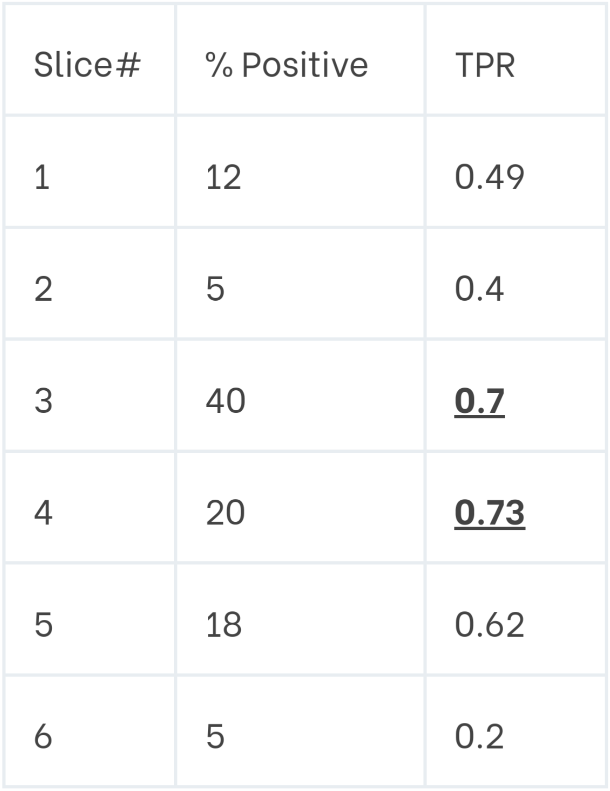
The operation corresponds to creating synthetic examples associated with a positive label for the data slices 4 and 6. We did so using our synthetic data engine, which allowed us to create new realistic data points for the slices in question.

We then re-trained a new XGBoost model using the augmented dataset, containing the original data and the new examples. We tested this model on a hold-out dataset to ensure the bias metrics improved, as shown in the following tables.



Slice#	% Pos.	TPR
1 Male 17-41	15	0.54
2 Female 17-41	5	0.48
3 Male 41-65	39	**0.69**
4 Female 41-65	10	**0.5**
5 Male >65	17	0.65
6 Female >65	0.	0.

Table 2. Slice-by-slice metrics for the original data.



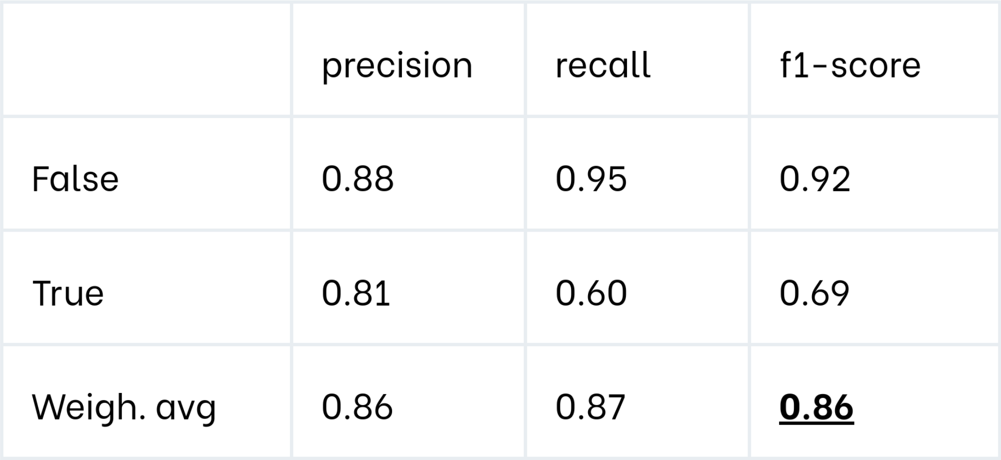
Slice#	% Positive	TPR
1	12	0.49
2	5	0.4
3	40	**0.7**
4	20	**0.73**
5	18	0.62
6	5	0.2

Table 3. Slice-by-slice metrics for the augmented data

The tables show that the model trained on the augmented data presented better metrics for the problematic slices. The model now makes positive predictions for women older than 65, and the TPR for middle-aged women is comparable to the TPR for middle-aged men. We can conclude that synthetic examples effectively mitigate a specific issue affecting our data.

The next question is, however, how the model performs on hold-out data, which is more representative of production data. Unfortunately, similar problems will likely affect this data, and the question is whether an augmented model will achieve decent performance.

The next table shows the classification metrics obtained by the improved model on untouched hold-out data.



	precision	recall	f1-score
False	0.88	0.95	0.92
True	0.81	0.60	0.69
Weigh. avg	0.86	0.87	**0.86**

Table 4. Accuracy metrics for the improved model.

In this case the new model achieves similar metrics on the hold-out dataset. This means we managed to improve model performances on problematic data slices while maintaining the same global performance.

Considerations- Synthetic data performance trade-off

Augmenting datasets is not a trivial task and usually corresponds to finding the ideal trade-off between local and global performance. However, we believe that synthetic data can be a great instrument in the hands of data scientists striving for better models, and we will continue to explore its potential even beyond the scope of the project.