

# Towards Improving Fairness in Tabular Data through Class Balancing Using GPT-4o

KoMeScVr: Amber Koelfat (6467296), Simon Menke (2510405), Max Schaffelder (1219200), Olivier Vroom (5315956)  
a.j.koelfat@students.uu.nl, s.m.menke@students.uu.nl, m.schaffelder@students.uu.nl, o.f.e.vroom@students.uu.nl  
Utrecht University  
Utrecht, the Netherlands

## ABSTRACT

Real-world datasets are typically unbalanced, with few large groups making up the bulk of the data and many minority groups being underrepresented. Machine learning models trained on these datasets will often perform suboptimally and could benefit from an increase in data for the minority groups. We experiment with synthetic data generation using a large language model (LLM), specifically GPT-4o, using different prompts to produce datasets that can be used to increase the in-dataset representation of minority groups and subsequently improve model performance and fairness. We find mixed results: While synthetic data generation using LLMs seems to be feasible, in-depth prompt engineering is needed to produce realistic feature distributions, and the production of data in this way takes a long time, especially because of maximum token output limits of web-interfacing LLMs like ChatGPT. Furthermore, the levels of classification fairness do not improve significantly when training with the synthetic data, which makes our results overall inconclusive.

## KEYWORDS

machine learning, fairness, LLM, synthetic data generation

### ACM Reference Format:

KoMeScVr: Amber Koelfat (6467296), Simon Menke (2510405), Max Schaffelder (1219200), Olivier Vroom (5315956). 2024. Towards Improving Fairness in Tabular Data through Class Balancing Using GPT-4o. In *Proceedings of Utrecht University (INFOMHCML'2023)*. Utrecht University, 7 pages.

## 1 INTRODUCTION

In modern data science, the availability of high-quality data is just as important for effective knowledge production and decision-making as choosing suitable algorithms and machine learning models [9]. While the quantity and granularity of collected data keep increasing, this trend holds to different degrees for different societal groups, meaning that much more data is available for some groups than for others. Unequal representation can lead to unequal performance for different groups [6], potentially producing unjustified negative outcomes. Different approaches are used to combat this, from reweighing [4] to data augmentation [10]. In this study, we attempt to create synthetic tabular data samples for underrepresented groups

This paper is published under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International (CC-BY-NC-ND 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

*INFOMHCML'2023, April 2023, Utrecht, the Netherlands*

© 2024 Utrecht University

ACM ISBN xxxxxxxx.

<https://doi.org/xxxxxxx>

using a large language model (LLM), specifically GPT-4o, one of the most capable LLMs at the time of writing. We attempt to answer the following research question: What LLM prompting strategies are most effective in generating synthetic tabular data to balance dataset classes and improve model fairness? To investigate this, we use different prompts to generate hundreds of synthetic data points each and compare classifiers trained on the resulting datasets as well as trained on the original data only. Additionally, we train on synthetic data generated using baseline approaches. We compare the models both in terms of performance (accuracy, F1) and fairness metrics (true positive rate, false positive rate). We chose equalized odds as our target fairness metric because we wanted to minimize disparate impact by making sure no group would be disproportionately harmed by the errors of the model. We hypothesize that using prompts that encourage the LLM to do step-by-step thinking will lead to more realistic data generation compared to less structured approaches. As we iteratively made the prompts more explicit in asking for step-by-step thinking, we hypothesize that each successive prompt will be more effective in terms of realism and in terms of fairness.

## 2 RELATED WORK

The field of synthetic data generation has been active for over a decade [3], and different approaches have been developed, with the most influential one likely being generative adversarial networks (GANs). GANs essentially consist of two parts, namely a generator model that creates new data samples, and a discriminator model that attempts to distinguish the generated samples from real data. The adversarial interplay between these two models allows GANs to iteratively optimize the data generation and acquire the ability to generate realistic data in a range of different formats, including images, text, tabular data, and others [3]. We used a GAN model for this study, specifically a conditional tabular GAN (CTGAN) [9], to generate our synthetic data baseline. While GANs have proven to be highly useful for some tasks, they have drawbacks as well, mainly concerning their training which involves making many decisions about loss functions and hyperparameters as well as preventing the generator from creating data that satisfies the discriminator's detection algorithm while not aligning with the demands of what humans need it for.

The main methods used to create synthetic tabular data before the advent of generative deep-learning models included various versions of oversampling methods, where individual data points are duplicated or approximated and added to the dataset. Another method is Gaussian mixture models (GMMs), using weighted sums

of Gaussian distributions fitted to the existing data and then sampling points from those distributions [3].

There is limited research on the feasibility of LLMs for generating synthetic data. Kim et al. [5] used a "completion triggering" approach where they provided only data without explicit instructions and let the model generate new instances, avoiding the need for lengthy prompt engineering. They achieved promising results using models like GPT-3.5 and GPT-3.5-turbo, Mistral-7b, and Llama-2-7b. Their approach differs from ours in two major ways: Firstly, instead of using the web interface of a language model and using prompt engineering to achieve their goal, they made use of an API and Python to generate their data, creating the need for a sufficient coding skillset to implement the code, as well as for monetary resources, as the most advanced models typically demand money on a per-token basis for access to their APIs. Secondly, the models used by Kim et al. are not state of the art (SOTA), but downscaled, "lighter" versions of the largest models available to date. For our study, we used GPT-4o, one of the most advanced models at the time of writing, and used prompt engineering and its publicly available web interface to create synthetic data samples, representing an approach requiring lower coding skills and producing no additional monetary cost.

## 3 METHODOLOGY

### 3.1 Study design

The current study set out to find whether prompting ChatGPT-4o to generate more instances of a certain underrepresented class within a biased data set would be useful in debiasing said dataset. We did so by creating four different prompts in which we asked ChatGPT to generate more samples. These new instances were then merged with the existing data set to create four new datasets on which a random forest model was fitted to the data. We then judged the models on both performance and fairness, to be compared with model predictions prior to debiasing. Although this experiment can also be done for other protected features including gender and marital status, in order to limit the scope of our research we decided to focus on race as the feature to differentiate between underrepresented groups. When analyzing the distribution of instances across different races, we observed that Race 1 was significantly over-represented, see Fig. 1. Due to the constraints on the maximum tokens that ChatGPT can generate per output, we decided to focus on a race with a relatively smaller number of instances to allow for a meaningful increase. Races 4, 5, and 7 had too few instances to identify any trends reliably. Therefore, we selected race 3 (American Indian) for generating synthetic data. Race 3 had a sufficient number of instances for ChatGPT to discern trends while still being small enough to benefit from a substantial enlargement of the dataset.

We chose to compare race 3 (American Indians) with race 6 (Asian) because they were closest in terms of size. Although races 8 (Some Other Race Alone) and 9 (Two or More Races) were even closer in size to race 3, they did not represent specific, concrete races. For practical and intuitive reasons, we aimed to compare the fairness between two distinct races.

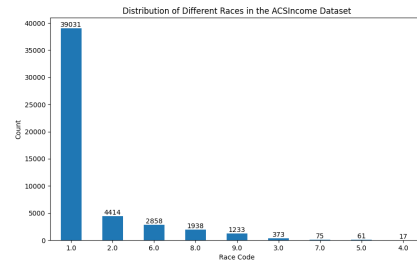


Figure 1: Distribution of Races

### 3.2 Dataset and preprocessing

For our study, we used the ACSIncome dataset [2], which is a larger and more recent alternative to the popular UCI Adult dataset [1]. It contains demographic data on 1,664,500 individuals. It contains 12 attributes including age, work class, education, marital status, occupation, race, sex, and income. In order to make the data more manageable to work with, we reduced the data set to 50,000 instances by randomly sampling from it.

### 3.3 Data generation

The race codes are as follows:

- 1: White alone
- 2: Black or African American alone
- 3: American Indian alone
- 4: Alaska Native alone
- 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
- 6: Asian alone
- 7: Native Hawaiian and Other Pacific Islander alone
- 8: Some Other Race alone
- 9: Two or More Races

**3.3.1 Prompts.** In order to get ChatGPT to generate synthetic data, we first uploaded all instances within the 50,000 dataset that had the race class equal to 3, which was 373 instances in total. By ensuring ChatGPT only saw instances where the race was equal to 3, we aimed to generate synthetic data that closely resembled this specific race class, rather than mixing characteristics from other races.

**3.3.2 Baselines.** In order to have suitable baselines of synthetic data to compare the LLM-generated data with, we used two separate methods. Firstly, we created a baseline using random oversampling (ROS), where we sampled from the training set with replacement and added those duplicated instances to the training set. For our more sophisticated baseline, we used CTGAN [9], a generative adversarial network built specifically for creating synthetic tabular data matching a preexisting data distribution. We made use of the Python library Synthetic Data Vault (SDV) [7] to create 364 synthetic data samples, using the default hyperparameter settings for the CTGAN synthesizer. The number of synthetic samples was chosen as the mean number of samples generated using GPT-4o in order to enable comparison of performance and fairness metrics.

Prompt	Details	Prompt Analysis/Explanation
<b>Prompt 1</b>	<b>1.1:</b> "You are a professional data analyst now, take a look at the following dataset. I am interested in the RAC1P column, more specifically in the value/class 3. Can you generate 30 more rows in text format with this value. Create synthetic data for other features than RAC1P while keeping in mind the distribution of these features. Don't use Python code, but fill the features yourself. Please output in raw csv code instead of tabular. Use seed XX." <b>1.2:</b> "Continue" - repeated until at least 300 instances were generated.	This prompt instructs the LLM to generate synthetic data while keeping in mind the distribution of the features. Initially, we used a random seed to avoid generating duplicate synthetic data from the same prompt, but later split the prompts between group members and dropped the random seed.
<b>Prompt 2</b>	<b>2.1:</b> "You are a professional data analyst now, take a look at the following dataset. Examine the distribution of feature values using bar plots and then generate additional statistics for each feature using pandas describe()." <b>2.2:</b> "Generate 60 more instances, making sure that RAC1P is equal to 3. Create synthetic data for other features than RAC1P while keeping in mind the distribution of these features. Don't use Python code, but fill the features yourself. Please output in raw csv code instead of tabular."	Prompt 2 builds upon the first prompt, but this time instructs the LLM to explicitly analyze the distribution of values.
<b>Prompt 3</b>	<b>3.1:</b> "You are a professional data analyst now, take a look at the following dataset. Examine the distribution of feature values using bar plots and then generate additional statistics for each feature using pandas describe()." <b>3.2:</b> "Analyze the correlations between the features. How do the features impact the PINCP (Income)?" <b>3.3:</b> "Generate 60 more instances, making sure that RAC1P is equal to 3. Create synthetic data for other features than RAC1P while keeping in mind the distribution and the correlations between the features. Don't use Python code, but fill the features yourself. Please output in raw csv code instead of tabular."	Prompt 3 extends prompt 1 and 2 by including correlation analysis in order to generate data with more realistic inter-feature relationships. By analyzing the correlations between features and their impact on income (PINCP), we hoped to encourage ChatGPT to consider these relationships when creating synthetic data. This step-by-step approach is similar to how solving problems incrementally can improve overall performance and accuracy.
<b>Prompt 4</b>	<b>4.1:</b> Researcher: "I've compiled your output into 'prompt 3 - 410 samples.csv' and attached it. Compare the feature distributions of it with the original 'ACS_50k_RACE_3.csv'." <b>4.2:</b> Researcher: "How would you suggest making the distributions even more similar to the original?" ChatGPT: "Preserve correlations: use techniques like multivariate normal distributions or copulas to generate data that preserves the correlations between features. This ensures that not just individual feature distributions, but also the relationships between features, are similar." ChatGPT then used the covariance matrix to capture relationships between the features in order to create synthetic data, and then set minimum and maximum values for the features to increase realism.	The last prompt aims to get the LLM to analyze its output in order to see how it differs from the original data, in order to generate even more realistic data. We tried to make it as interactive and iterative as possible. Due to its length, the conversation has been summarized. For a more detailed version of the conversation, see the appendix.

Table 1: Table of Prompts Used in the Study

### 3.4 Model training

For our model we chose a random forest regressor to create predictions and run fairness metrics. Out of all 11 features, we only selected 5 features for training the model. Other features such as age or sex are deemed protected attributes and therefore not included in the training data. The selected features can be seen in Table 2. See

the PUMS documentation<sup>1</sup> for more information about the features and corresponding codes. As visible in the table, we have three nominal features with a high number of feature values. Due to the structure of random forests, they cannot intrinsically capture nominal values and therefore we decided to one-hot encode all nominal

<sup>1</sup><https://www.census.gov/programs-surveys/acs/microdata/documentation.html>

Feature code	Full feature name	Type (#unique)
COW	Class of worker	nominal (8)
SCHL	Educational attainment	ordinal (24)
OCCP	Occupation	nominal (528)
WKHP	Usual hours worked per week past 12 months	numerical (0-99)
ST	State/Country	nominal (51)

**Table 2: Selected features**

features. This led to a high number of final features, however with over 50,000 instances the dimension was still manageable. The target variable to predict is income, which is represented as integers. As the problem is a classification, after prediction we converted the target variable to a binary variable using a threshold where 0 represents below 50,000 and 1 represents greater than or equal to 50,000. The random forest consists of 200 estimators and we set the maximum samples for each estimator to 0.8 to achieve horizontal randomization. We decided against vertical randomization as the columns are one-hot encoded. Avoiding vertical randomization when using one-hot encoded data helps maintain the integrity, completeness, and context of the original features, leading to better model performance and more accurate results.

Otherwise no hyper-parameters were used since focus of this study was fairness, as opposed to the machine learning model or its performance. This also makes the experiment more standardised and reproducible. The random forest model is then trained and evaluated on each dataset using a train/test split of 0.7/0.3.

### 3.5 Analysis

The performance of the model is analysed based on both fairness and performance measures. The fairness measure used in the current research is equalized odds. We will thus be comparing to what extent the true positive rate and the false positive rate are different between groups. For the fairness metrics we used the Fairlearn library [8]. The performance is measured based on accuracy and F1. The models compared here are all random forest models trained on the original data, the baseline generated data and prompts 1-4.

## 4 RESULTS

### 4.1 Fairness

The fairness results are presented in Tables 3 and 4. Table 3 shows the TPR and FPR scores per dataset. For both classes, TPR and FPR rates seem to remain similar, except for ROS. TPR values are consistently 0.464 for class 3, except for ROS for which it jumps to 0.714. The TPR values for class 6 are within a range of 0.799 and 0.813 for all datasets. The FPR values for class 3 lie in the range of 0.111 and 0.148 for all datasets except for ROS which has a much lower value of 0.086. The FPR values for class 6 are all within the range between 0.299 and 0.303, so they stay essentially the same.

In order to compare levels of fairness, we calculated the difference in TPR and FPR rates between the two races. Table 4 shows these results. All datasets except for ROS yielded similar differences

in TPR within a range of 0.335 to 0.348, with ROS achieving a significantly lower value of 0.096. Most of the FPR differences are in the range of 0.149 to 0.188, with the only outlier again being ROS with an FPR difference of 0.215.

Dataset	Class	TPR	FPR
original	3	0.464	0.136
	6	0.807	0.301
CTGAN	3	0.464	0.136
	6	0.813	0.299
ROS	3	0.714	0.086
	6	0.810	0.301
prompt 1	3	0.464	0.148
	6	0.805	0.299
prompt 2	3	0.464	0.148
	6	0.799	0.303
prompt 3	3	0.464	0.148
	6	0.805	0.297
prompt 4	3	0.464	0.111
	6	0.807	0.299

**Table 3: TPR and FPR per dataset and race class**

Data	TPR difference	FPR difference
original	0.343	0.165
CTGAN	0.348	0.163
ROS	0.096	0.215
prompt 1	0.340	0.151
prompt 2	0.335	0.155
prompt 3	0.340	0.149
prompt 4	0.343	0.188

**Table 4: FPR and TPR differences**

### 4.2 Performance

The performance measures of the model for the different datasets are shown in Table 5. The accuracy ranges from 0.755 for prompt 1 to 0.758 to prompt 4, a difference of only 0.003. For F1 the scores range from 0.674 for the original dataset, prompt 1, and prompt 3, up to 0.678 for prompt 4. Overall both accuracy and F1 scores do not seem to differ greatly between datasets, meaning the overall performance of the model is stable across datasets.

Dataset	Accuracy	F1
original	0.756	0.674
CTGAN	0.756	0.675
ROS	0.757	0.676
prompt 1	0.755	0.674
prompt 2	0.756	0.675
prompt 3	0.756	0.674
prompt 4	0.758	0.678

Table 5: Accuracy and F1 Scores

### 4.3 Data distributions

The similarity of feature distributions between the original and the synthetic data differed widely depending on the variable and the prompt used. As can be seen in Fig. 2, for the "age" feature, the baselines stayed relatively close to the original distribution, while the LLM created data on a much smaller range, except for prompt 4, which created a very wide distribution. For the "sex" feature, the LLM data also stayed relatively close to the original distribution, with only prompt 4 deviating significantly by hallucinating two feature values that were not in the original data (Fig. 3). For some other features, like "state", capturing the state that the person was from, the LLM-generated data varied a lot between each prompt (Fig. 4).

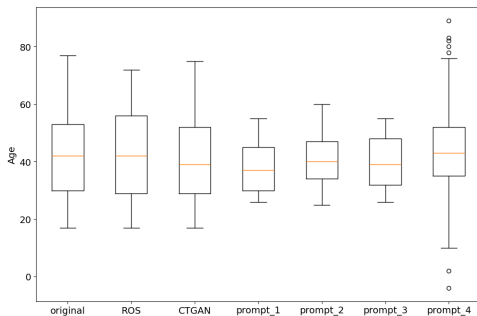


Figure 2: Box plots of age distributions for original and synthetic datasets

## 5 DISCUSSION

The study aimed to investigate the effectiveness of various prompting strategies for generating synthetic tabular data using GPT-4o to balance dataset classes and improve classifier fairness. Our hypothesis was that each successive prompt would make fairer and more realistic datasets. The results of this experiment failed to demonstrate that synthetic data generation using ChatGPT can significantly improve fairness between two social groups, regardless of prompt. However, it is worth noting that at least it did not decrease fairness. Further, the performance stays almost identical between prompts and also when compared to the original dataset. In terms of realism, the data generated from prompt 4 was the most unrealistic, including that it hallucinated two new sexes (3, 4, whereas male and female were 1 and 2 in the original dataset,

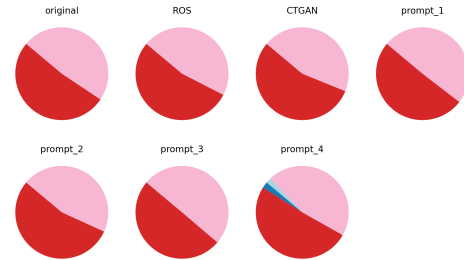


Figure 3: Pie charts of sex distributions for original and synthetic datasets; red=male, pink=female

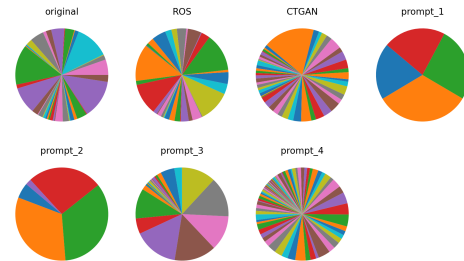


Figure 4: Pie charts of state distributions for original and synthetic datasets; feature values omitted for clarity

as mentioned in section 4.3). Thus, it is not clear what prompting strategies are more effective, although it seems that prompt 4 is not well suited for realistic synthetic data generation, contradicting our hypothesis, although further research is needed to draw any firm conclusions.

### 5.1 Interpretation of results

The true positive rate (TPR) for race 3 is consistent across all datasets (0.464) except for the ROS dataset, which has a higher TPR of 0.714. This consistency is surprising because one would expect more variability. We hypothesize that the limited size of the synthetic data generated might explain this anomaly. The original dataset contains 50,000 instances, with 373 instances for race 3. In contrast, for each prompt, ChatGPT generated between 300 and 410 instances, and the models were tested on 112 samples for race 3. The relatively small synthetic dataset likely does not capture the variability present in the larger original dataset, leading to this unexpected uniformity in TPR values.

### 5.2 Implications

The inability to improve fairness using synthetic data generated by ChatGPT in our study does not necessarily mean it is unfeasible. Furthermore, the synthetic data generated seems to be realistic enough to not completely throw off the model training, shown by the stable performance. The distributions for some features of the synthetic data seem to be visually similar to the original data, as shown in Fig. 2 and Fig. 3. Overall, the implications of our study are limited given the inconclusive nature of the results.

### 5.3 Limitations

A limitation of this study is the uncertainty regarding the generalizability of the findings to other contexts. Specifically, we focused on augmenting the size of a single class, and it remains unclear how generating synthetic data for multiple classes would impact fairness. This narrow scope raises questions about the broader applicability of our results across different datasets and demographic groups. Further research is needed to explore the effects of synthetic data generation on fairness when applied to multiple classes. Additionally, another limitation stems from our method of uploading all instances with a race equal to 3 to ChatGPT for synthetic data generation. These instances were also included in the test set, introducing the possibility of data leakage. This could lead to ChatGPT inadvertently replicating data, thereby affecting the integrity of the results. Future studies should ensure that training and test data are strictly segregated to prevent such leakage and to validate the robustness of synthetic data generation methods.

### 5.4 Recommendations for Future Research

Future research should first focus on determining whether it is even possible to significantly improve fairness through synthetic data generation using LLMs. In order to do this, we recommend creating larger synthetic datasets than the ones generated in this study. Furthermore, future researchers could then compare the outputs of different LLMs to determine if one demonstrates a greater ability to generate synthetic data to improve fairness than the others. Additionally, training and testing the data using various machine learning models could provide deeper insights, as the selected random forest model might not be optimal for this particular scenario. Evaluating other models would provide insight into the impact of model selection and reveal any performance variations across models. While we chose to create synthetic data for individuals of a specific race in this study, the same principle could be applied to other protected attributes, such as gender and marital status. Ideally, the data should also be sanitized as the synthetic data contains made-up nominal values such as occupation codes that do not exist in the original dataset. The prompts could also be run multiple times to test the variance of the model and then take an average over multiple runs to get more representative data.

## 6 CONCLUSION

Our study tested the feasibility of generating synthetic tabular data using an LLM with access through its web interface. Additionally, we set out to investigate the impact of different prompting strategies on the quality of synthetic data being generated. Our findings are largely inconclusive, with accuracy, F1, and true positive and false positive rates staying surprisingly steady throughout. In the discussion, we provide several explanations of this finding. Additionally, the visualizations of the feature distributions show that some features are "understood" much better by the LLM than others. Our study illustrates that synthetic data generation using LLMs could be a feasible approach for increasing sample sizes of underrepresented groups, however, it is unclear how helpful it would be for fairness, and also includes significant technical challenges which could be addressed either by more precise prompting or by more

capable language models that might be developed in the coming years.

## 7 ETHICAL CONSIDERATIONS

There are several ethical considerations to take into account when using LLMs to generate synthetic data, including:

**Environmental costs:** Training and running an LLM like ChatGPT requires substantial amounts of electricity and results in several environmental impacts, including increased water usage for cooling data centers. If LLMs are used to create synthetic data at scale, it is crucial to balance these environmental costs. However, it's worth considering that generating synthetic data might be more efficient than collecting, cleaning, and preprocessing raw real-world data.

**Privacy and Data Security:** When generating synthetic data, especially from sensitive datasets, there is a risk of inadvertently including identifiable information or patterns that could compromise privacy. Even though the data is synthetic, it might still be possible to reverse-engineer or infer information about individuals in the original dataset, or in other datasets that the LLM was trained on. It is essential to implement robust privacy-preserving techniques and thoroughly evaluate the synthetic data to ensure it does not pose any privacy risks.

**Bias and Fairness in Synthetic Data:** While the goal of generating synthetic data is to mitigate bias and improve fairness, there is a risk that the synthetic data itself could introduce new biases or fail to adequately address existing ones. In this paper, we focused on TPR and FPR, but it is important to use a variety of fairness metrics and evaluation techniques.

**Transparency and Accountability:** The use of LLMs for synthetic data generation should be transparent, with clear documentation of the methods, datasets, and prompt strategies used. Researchers should be accountable for the ethical implications of their work, including any unintended consequences. Openly sharing findings, limitations, and ethical considerations can help foster trust and facilitate collaborative efforts to address ethical challenges in synthetic data generation.

## 8 GROUP MEMBER CONTRIBUTIONS

The group divided the tasks in the following way:

- Amber worked on the overall study design and made the poster for the presentation. In the report she wrote part of the Methods and Results sections and worked on feature visualization. Due to a minor head injury, part of her tasks were taken over by other team members.
- Simon: Together with Olivier, Simon focused on crafting the prompts and collecting the data generated by ChatGPT. He was also responsible for training the random forest model and evaluating the datasets using fairness and standard machine learning metrics.
- Max created the synthetic baseline datasets using Random Oversampling and CTGAN. He also performed feature visualizations for the original and synthetic datasets. For the report, he wrote the Abstract, Introduction, Related Work, Conclusion, and parts of the Results and Discussion sections.

- Olivier: Together with Simon, Olivier focused on crafting the prompts and collecting the data generated by ChatGPT. Therefore, he worked on the methodology section of the report. Olivier also wrote the ethical considerations section.

## REFERENCES

- [1] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XW20>.
- [2] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring Adult: New Datasets for Fair Machine Learning. *arXiv preprint arXiv:2108.04884* (2021). <https://arxiv.org/abs/2108.04884>
- [3] Alvaro Figueira and Bruno Vaz. 2022. Survey on Synthetic Data Generation, Evaluation Methods and GANs. *Mathematics* 10, 15 (2022). <https://doi.org/10.3390/math10152733>
- [4] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems* 33, 1 (2012), 1–33.
- [5] Jinhee Kim, Taesung Kim, and Jaegul Choo. 2024. Exploring Prompting Methods for Mitigating Class Imbalance through Synthetic Data Generation with Large Language Models. *arXiv e-prints*, Article arXiv:2404.12404 (April 2024), arXiv:2404.12404 pages. <https://doi.org/10.48550/arXiv.2404.12404> [cs.LG]
- [6] Marcus Maloof. 2003. Learning When Data Sets are Imbalanced and When Costs are Unequal and Unknown. *Analysis* 21 (07 2003).
- [7] Ninad Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. *Synthetic Data Vault (SDV)*. <https://sdv.dev/> Python library for generating synthetic data.
- [8] Hilde Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. 2023. Fairlearn: Assessing and Improving Fairness of AI Systems. , 8 pages. <http://jmlr.org/papers/v24/23-0389.html>
- [9] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. *Modeling tabular data using conditional GAN*. Curran Associates Inc., Red Hook, NY, USA.
- [10] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, New Orleans, Louisiana, 15–20. <https://doi.org/10.18653/v1/N18-2003>

## A PROMPTS

**4.1 Researcher:** *I've compiled your output into "prompt 3 - 410 samples.csv" and attached it. Compare the feature distributions of it with the original "ACS\_50k\_RACE\_3.csv".*

**4.2 Researcher:** *How would you suggest making the distributions even more similar to the original?*

ChatGPT: *Preserve correlations: use techniques like multivariate normal distributions or copulas to generate data that preserves the correlations between features. This ensures that not just individual feature distributions, but also the relationships between features, are similar.*

**4.3 Researcher:** *How should we preserve correlations?*

*Preserving correlations in synthetic data can be achieved through several techniques. Here are a few effective methods:*

(1) **Multivariate Normal Distribution:**

- *If the data is approximately normally distributed, you can use a multivariate normal distribution to generate synthetic data that preserves the covariance structure of the original data.*
- ... (omitted for conciseness and lack of relevance)

**4.4 Researcher:** *AGEP seems normally distributed, do the first option (Multivariate Normal Distribution)*

*ChatGPT's output:*

*Returns a table with 410 instances*

*The synthetic data has been adjusted to ensure realistic values. Here are the adjustments made:*

**Capped and Floored Values:**

- *WKHP (Work Hours per Week) is capped between 1 and 99.*
- *RELP (Relationship) values are floored at 0.*
- *PINCP (Income) values are floored at 0 to avoid negative incomes.*