Evaluating language models as risk scores

André F. Cruz[1,2] **Moritz Hardt**[1,2] **Celestine Mendler-Dünner**[1,2,3]
[1]Max Planck Institute for Intelligent Systems, Tübingen
[2]Tübingen AI Center [3]ELLIS Institute Tübingen

Abstract

Current question-answering benchmarks predominantly focus on accuracy in realizable prediction tasks. Conditioned on a question and answer-key, does the most likely token match the ground truth? Such benchmarks necessarily fail to evaluate LLMs' ability to quantify ground-truth outcome uncertainty. In this work, we focus on the use of LLMs as risk scores for unrealizable prediction tasks. We introduce `folktexts`, a software package to systematically generate risk scores using LLMs, and evaluate them against US Census data products. A flexible API enables the use of different prompting schemes, local or web-hosted models, and diverse census columns that can be used to compose custom prediction tasks. We evaluate 17 recent LLMs across five proposed benchmark tasks. We find that zero-shot risk scores produced by multiple-choice question-answering have high predictive signal but are wildly miscalibrated. Base models consistently overestimate outcome uncertainty, while instruction-tuned models underestimate uncertainty and produce over-confident risk scores. In fact, instruction-tuning polarizes answer distribution regardless of true underlying data uncertainty. This reveals a general inability of instruction-tuned models to express data uncertainty using multiple-choice answers. A separate experiment using verbalized chat-style risk queries yields substantially improved calibration across instruction-tuned models. These differences in ability to quantify data uncertainty cannot be revealed in realizable settings, and highlight a blind-spot in the current evaluation ecosystem that `folktexts` covers.

1 Introduction

Fueled by the success of large language models (LLMs), it is increasingly tempting for practitioners to use such models for risk assessment and decision making in consequential domains [1–4]. Given the CV of a job applicant, for example, some might prompt a model, what are the chances that the employee will perform well on the job? The true answer is likely uncertain. Some applicants of the same features will do well, others won't. A good statistical model should faithfully reflect such outcome uncertainty.

Calibration is perhaps the most basic kind of uncertainty quantification to ask for. A calibrated model, on average, reflects the true frequency of outcomes in a population. Calibrated models must therefore give at least some indication of uncertainty. Fundamental to statistical practice across the board, calibration has also been a central component in the debate around the ethics and fairness of consequential risk scoring in recent years [5–8].

The evaluation of LLMs to date, however, has predominantly focused on accuracy metrics, often in realizable tasks where there is a unique correct label for each data point. Such benchmarks necessarily cannot speak to the use of language models as risk score estimators. A model can have high utility in well-defined question-answering tasks while being wildly miscalibrated. In fact, while accuracy corresponds to knowledge of the expected answer, proper uncertainty quantification corresponds to knowledge of the variance over answers.

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

 https://doi.org/10.52202/079017-3089

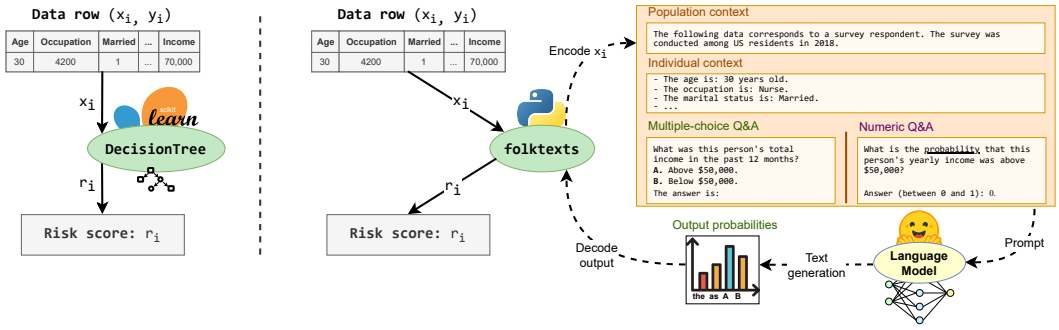


Figure 1: Information flow from tabular data to risk scores, using a supervised classifier (*left*) or a language model (*right*). The `folktexts` package maps language models to the traditional machine learning workflow.

1.1 Our contributions

We contribute an open-source software package, called `folktexts`,[1] that provides datasets[2] and tools to evaluate statistical properties of LLMs as risk scorers. We show-case its functionalities with a sweep of new empirical insights on the risk scores produced by 17 recently proposed LLMs.

The `folktexts` package offers a systematic way to translate between the natural language interface and the standard machine learning type signature. It translates prediction tasks defined by numeric features X and labels Y into natural text prompts and extracts risk scores R from LLMs. This opens up a rich repertoire of open-source libraries, benchmarks and evaluation tools to study their statistical properties. Figure 1 illustrates the workflow for producing risk scores using LLMs.

For benchmarking risk scores, we need ground truth samples from a known probability distribution. Inspired by the popular folktables package [9], `folktexts` builds on US Census data products, specifically, the American Community Survey (ACS) [10], collecting information about more than 3.2 million individuals representative of the US population. `Folktexts` systematically constructs prediction tasks and prompts from the individual survey responses using the US Census codebook, and the ACS questionnaire as a reference. Risk scores are extracted from the language model's output token probabilities using a standard question-answering interface. The package offers five pre-specified question-answering benchmark tasks that are ready-to-use with any language model. A flexible API allows for a variety of natural language tasks to be constructed out of 28 census features whose values are mapped to prompt-completion pairs (features detailed in Table A5). Furthermore, evaluations can easily be performed over subgroups of the population to conduct algorithmic fairness audits.

Empirical insights. We contribute a sweep of empirical findings based on our package. We evaluate 17 recently proposed LLMs, with sizes ranging from 2B parameters to 141B parameters. Our study demonstrates how inspecting risk scores of LLMs on underspecified prediction tasks reveals new insights that cannot be deduced from inspecting accuracy alone. The main findings are summarized as follows:

- Models' output token probabilities have strong predictive signal[3] but are wildly miscalibrated.
- The failure modes of models are different: Multiple-choice answer probabilities generated by base models consistently overestimate outcome uncertainty, while instruction-tuned models underestimate uncertainty and produce over-confident risk scores.
- Instruction-tuning polarizes multiple-choice answer probabilities, regardless of ground-truth outcome uncertainty — leading to a general inability to express data uncertainty.
- Using a chat-style prompt that verbally queries for probabilities results in materially different answer distributions, with significantly improved calibration for instruction-tuned models, accompanied by a small but consistent decrease in predictive power.

[1] Package and results available at: `https://github.com/socialfoundations/folktexts`
[2] Ready-to-use Q&A datasets available at: `https://huggingface.co/datasets/acruz/folktexts`
[3] We measure predictive signal using the area under the receiver operating characteristic curve (AUC) [11].

We hope our package facilitates future investigations into statistical properties of LLMs, not only as a way to faithfully reflect real-world uncertainties, but also as a required stepping stone to trustworthy model responses.

Outline. In Section 2 we provide necessary background on risk scores and calibration in statistical machine learning. In Section 3 we extend this background to the application to language models, providing various design choices around prompting templates and ways to extract risk scores from language models. In Section 4 we evaluate 17 recent LLMs on 5 proposed benchmark tasks and summarize empirical findings.

1.2 Limitations

Predictive modeling and statistical risk scoring in consequential settings is a matter of active debate. Numerous scholars have cautioned us about the dangers of statistical risk scoring and, in particular, the potential of risk scores to harm marginalized and vulnerable populations [5, 12–14]. Our evaluation suite is intended to help in identifying one potential problem with language models for risk assessment, specifically, their inability to faithfully represent outcome uncertainty. However, our metrics are not intended to be sufficient criteria for the use of LLMs in consequential risk assessment applications. The fact that a model is calibrated says little about the potential impact it might have when used as a risk score. Numerous works in the algorithmic fairness literature, for example, have discussed the limitations of calibration as a fairness metric, see, e.g., [7, 8, 15]. There is also significant work on the limitations of statistical tools for predicting future outcomes and making decisions based on these predictions [16–18]. Calibration cannot and does not address these limitations.

1.3 Related work

The use of LLMs for decision-making has seen increasing interest as of late. Hegselmann et al. [19] show that a `Bigscience T0 11B` model [20] surpasses the predictive performance (AUC) of supervised learning baselines in the very-few-samples regime. The authors find that fine-tuning an LLM outperforms fitting a standard statistical model on a variety of tasks up to training set sizes in the 10s to 100s of samples. Related work by Slack and Singh [21] shows that providing task-specific expert knowledge via instructions in context can lead to significant improvements in model predictive power. Tamkin et al. [1] generate hypothetical individual information for a variety of decision scenarios, and analyze how language models' outputs change when provided different demographic data. Some attributes are found to positively affect the model's decision (e.g., higher chance of approving a small business loan for minorities) while others affect it negatively (e.g., lower chance for older aged individuals).

A separate research thread considers how to leverage LLMs to model human population statistics. Argyle et al. [22] evaluate whether GPT-3 can faithfully reproduce political party preferences for different US subpopulations, and conclude that model outputs can accurately reflect a variety of correlations between demographics and political preferences. Other works use a similar methodology to model the distribution of human opinions on different domains [23–25]. Aher et al. [26], Dillion et al. [27] use LLMs to reproduce popular psychology experiments on human moral judgments, and confirm there's good alignment between human answers and LLM outputs. Literature on modeling human population statistics generally focuses on using LLMs to obtain accurate survey completions. That is, given demographic information on an individual, what was their response to a specific survey question? This methodology often ignores the fact that individuals described by the same set of demographic features will realistically give different answers. An accurate model would not only provide the highest likelihood answer, but also a measure of uncertainty corresponding to the expected variability within a sub-population. Our work tackles this arguably neglected research avenue: Analyzing LLM risk scores instead of discrete token answers, and whether they are accurate and calibrated to human populations.

Calibration. Calibration is a widely studied concept in the literature on forecasting in statistics and econometrics with a venerable history [28–32]. Recent years have seen a surge in interest in calibration in the context of deep learning [33]. Calibration of LLMs has been studied on diverse question-answering benchmarks, ranging from sentiment classification, knowledge testing, and mathematical reasoning to multi-task benchmarks [e.g., 34–43]. What differentiates our work is that

 https://doi.org/10.52202/079017-3089

we study calibration in naturally underspecified prediction tasks. This requires even the most accurate models to accurately reflect non-trivial probabilities over outcomes to be calibrated.

To systematically construct such prediction tasks we resort to survey data. Surveys have a long tradition in social science research as a tool for gathering statistical information about the characteristics and opinions of human populations [44]. Survey data comes with carefully curated questionnaires, as well as ground truth data. The value of this rich data source for model evaluation has not remained unrecognized. Surveys have recently gained attention to study bias and alignment of LLMs [22, 45–48], and inspecting systematic biases in multiple-choice responses [49]. Instead of using surveys to get insights about a models natural inclinations, we use them to test model calibration with respect to a given population.

Beyond task-calibration, calibration at the word and token-level has been explored in the context of language generation [e.g., 50]. Others have focused on connections to hallucination [51], and expressing uncertainty in natural language [52–54]. Related to our work, Lin et al. [55] emphasize the inherent outcome uncertainty in language generation. Focusing on generation at the word or token level poses the challenge of measuring a high-dimensional probability distribution. Considering binary classification tasks has the practical advantage of circumventing this problem.

Calibration has also played a major role in the algorithmic fairness literature. Group-wise calibration has been proposed as a fairness criterion since the 1960s [56, 57]. In particular, it's been a notion central to an active debate about the fairness of risk scores in consequential decision making [5–8]. A recent line of work originating in algorithmic fairness studies *multicalibration* as a strengthening of calibration [58].

2 Preliminaries

This section provides necessary background on risk scores and the statistical evaluation of binary predictors. Throughout, we assume a joint distribution $\mathcal{P}$ given by a pair of random variables (X, Y) where X is a set of features and Y is the outcome to be predicted. In the applications we study, the features X typically form a text sequence and the outcome Y is a discrete random variable that we would like to predict from the sequence. A parametric model $f_\theta(y|x)$ assigns a probability to each possible outcome y given a feature vector x. The goal of a generative model is generally to approximate the conditional distribution $\mathcal{P}(y|x)$, where parameters are fit to a huge corpus of training data.

We will focus on binary prediction throughout this work. Let $Y \in \{0, 1\}$ be a random variable indicating the outcome of an event the learner wishes to predict. We use the shorthand notation $f_\theta(x)$ to denote the model's estimate of the probability that $Y = 1$, given context $X = x$. Following standard terminology we will refer to $f_\theta : \mathcal{X} \to [0, 1]$ as *score function*, and its output $f_\theta(x)$ as *risk score*. There are various dimensions along which to evaluate risk scores by comparing them against samples of the reference population $\mathcal{P}$.

2.1 Calibration

We say a score function is *calibrated* over a population $\mathcal{P}$ if and only if for all values $r \in [0, 1]$ with $\mathbb{P}\{f_\theta(X) = r\} > 0$, we have

$$\mathbb{P}[Y = 1 \mid f_\theta(x) = r] = r. \tag{1}$$

This condition asks that, over the set of all instances x with score value r, an r fraction of those instances must indeed have a positive label. Importantly, calibration is defined with respect to a population, it does not measure a model's ability to discriminate between instances. A model that outputs the constant value $\mu = \mathbb{E}[Y]$ on all instances is calibrated, by definition. In particular, we can always achieve calibration by setting $f_\theta(x)$ with the average value of y among all instances x' in a given partition such that $f_\theta(x') = f_\theta(x)$.

We use Expected Calibration Error (ECE) as the primary metric to empirically evaluate calibration. It is defined as the expected absolute difference between a classifier's confidence in its predictions and the accuracy on the same predictions. More formally, given n triplets (x_i, y_i, r_i) where r_i denotes

the model's score value for the corresponding data point (x_i, y_i). The ECE is defined as

$$\mathrm{ECE} \coloneqq \frac{1}{n} \sum_{m} \left| \sum_{i \in B_m} y_i - \sum_{i \in B_m} r_i \right|, \tag{2}$$

where data points are grouped into M equally spaced bins B_m according to their score values. We use $M = 10$ in our evaluation, which is a commonly used value [59]. We also provide a measure of ECE over quantile-based score bins, as well as Brier score [28] to allow for a more complete picture. Furthermore, we use reliability diagrams [30] to aid a visual interpretation of calibration. These diagrams plot expected sample accuracy as a function of confidence. Any deviation from a perfect diagonal represents miscalibration.

We can strengthen the calibration condition in (1) by requiring it in multiple subgroups of the population. Specifically, letting G denote any discrete random variable, we can require the conditional calibration condition $\mathbb{P}[Y = 1 \mid G = g\,, f_\theta(x) = r] = r$ for every setting g of the random variable G. This is often used to define fairness.

2.2 Predictive performance

When solving classification problems it's common practice to threshold risk scores to obtain a classifier. In our notation this corresponds to thresholding the risk scores $f_\theta(x)$:

$$c(x) = \mathbb{1}\{f_\theta(x) > \tau\}.$$

The classifier c that minimizes the misclassification error $\mathbb{E}\,\mathbb{1}\{c(X) \neq Y\} = \mathbb{P}\{c(X) \neq Y\}$ is given by $c^*(x) = \mathbb{1}\{f^*(x) > 0.5\}$, where f^* is the Bayes optimal scoring function. In the following, when we use *accuracy* we refer to the fraction of correct predictions after thresholding. If not specified otherwise we use $\tau = 0.5$. This corresponds to an argmax operator applied to the class probabilities. It is important to note that a classifier can achieve perfect accuracy even when derived from a suboptimal scoring function. Thus, accuracy alone provides an incomplete picture of a model's ability to express uncertainty.

Instance ranking. The area under the receiver operating characteristic curve (AUC) is a rank-based measure of predictive model performance. It measures the probability that a randomly chosen positive observation ($Y = 1$) will have a higher score than a randomly chosen negative observation ($Y = 0$). A high AUC value in no way reflects accurate or calibrated probability estimates, it relates only to the signal-to-noise ratio in the risk scoring function [11]. Using AUC allows us to neatly separate risk score calibration from their predictive signal, although both are crucial for accurate class predictions.

3 Evaluating language models as risk scores

We are interested in the ability of LLMs to express natural uncertainty in outcomes. Therefore, we construct unrealizable binary prediction tasks, and test the model's ability to reflect natural variations in underspecified *individual* outcomes. Specifically, we prompt models with feature values x to elicit risk scores r and then evaluate these scores against ground truth labels y (see Figure 1).

3.1 Prediction tasks

We construct natural language prediction tasks from the American Community Survey (ACS) Public Use Microdata Sample (PUMS).[4] The data contains survey responses of about 3.2 million anonymous individuals, carefully curated to offer statistical insights into the population of the United States. We refer to the data as Census data. The Census data contains demographic attributes, as well as information related to income, employment, health, transportation, and housing. Prediction tasks are defined by selecting a subset of attributes to define the features and one attribute to be the label. We threshold continuous target variables and bin multi-class predictions to obtain a binary classification task. Specifically, we test models on their ability to reflect natural variations in the outcome across the benchmark population. To enable straightforward comparison with existing tabular benchmarks, we consider natural text analogues to the tasks in the popular folktables benchmark package [9]. Appendix B describes each task in further detail.

[4] https://www.census.gov/programs-surveys/acs/microdata.html

 https://doi.org/10.52202/079017-3089

Natural uncertainty in risk scoring. Prediction tasks on human populations typically come with natural outcome uncertainty, meaning that the target label is not uniquely determined by the input features (also known as *aleatoric uncertainty* [60]). The fewer features are provided the higher the uncertainty in the outcome. We take this to our advantage to systematically evaluate calibration. Namely, to be calibrated, a risk score has to reflect both model uncertainty and uncertainty inherent to the prediction task. In fact, the optimal predictor would often output low-confidence answers. In contrast, prevailing question-answering benchmarks have no data uncertainty, and thus require high confidence for an accurate model to be calibrated. Underspecified, non-realizable prediction tasks allow us to circumvent such potential confounding between calibration and accuracy.

3.2 Extracting risk scores

To extract risk scores from LLMs, we map each inference problem to a natural text prompt. For a given data point, we specify the classification task in the prompt and extract class probabilities from the model's next token probabilities, similar to traditional question-answering interfaces. The prompt consists of three components (as shown in Figure 1):

- **Instantiating population:** We first instantiate the population $\mathcal{P}$ in the prompt context. This corresponds to the population represented by our reference data. We use third-person prompting: "`The following data describes a survey respondent. The survey was conducted among US residents in 2018. Please answer the question based on the information provided.`" This step is typically not needed in supervised classification tasks, because the training data implicitly defines the population. However, for LLMs this is particularly important for evaluating calibration outside the realizable setting, as risk scores cannot in general simultaneously be calibrated to different populations. Skipping this step could provide insights related to alignment [22, 45, 46] rather than calibration.
- **Instantiating features:** Next we instantiate an individual. Each individual in the population corresponds to a row in our tabular dataset. We use the US Census codebook to construct a template for transforming attribute/value pairs from the dataset into meaningful natural text representations. Consider the values $x_i = \{\text{SEX} : \text{male}, \text{AGEP} : 50\}$ which would correspond to "`Information about this person:\n - Gender is: Male.\n - Age is: 50 years old.`" We use a bulleted list of short sentences to encode features. Related literature has found this simple approach to yield the best results [1, 19, 61–63].
- **Querying outcome:** We use a standard *multiple-choice* prompting format to elicit outcome predictions from LLMs. The framing of the question for an individual outcome is taken from the original multiple-choice Census questionnaire based on which the data was collected. All answers are presented as binary choices. Querying about an individual's income would be: "`Question: What was this person's total income during the past 12 months?\n A: Below $50,000.\n B: Above $50,000.\n Answer:`" The model's confidence on a given answer is given by the next token probabilities for `A` and `B`. Additionally, we conduct experiments using a separate chat-style prompt that verbally queries for a numeric probability estimate (dubbed *numeric* prompting). The above income query would be: "`Question: What is the probability that this person's yearly income is above $50,000?\n Answer (between 0 and 1): `" This more closely matches how real-world users interact with LLMs, and has been reported to improve uncertainty quantification [64].

When using the constructed multiple-choice prompts, we query the models and extract scores from the next token probabilities for the choice labels `A,B` as $r_i = \mathbb{P}(\texttt{A})/(\mathbb{P}(\texttt{A}) + \mathbb{P}(\texttt{B}))$, following the methodology of standard question-answering benchmarks [39, 45]. As LLMs are known to have ordering biases in multiple-choice question-answering [47, 65], we evaluate responses on all choice orderings and average the resulting scores. When using numeric prompting, we prefix the answer with '`0.`' to improve the likelihood of a direct numeric response, and run two forward passes, selecting the highest likelihood numeric token at each iteration. We refer to Xiong et al. [37] for an overview on alternative design choices on how to elicit confidence scores from LLMs.

3.3 The `folktexts` package

The `folktexts` package is designed to offer a flexible interface between tabular prediction tasks and natural language question-answering tasks in order to extract risk scores from LLMs. The

package makes available the ACSIncome, ACSPublicCoverage, ACSMobility, ACSEmployment, and ACSTravelTime prediction tasks [9] as natural language benchmarks, together with various functionalities to customize the task definitions (e.g., use a different set of features to predict income) and subsample the reference data (e.g., predict income only among college graduates in California). The set of attributes available to define the features and label can be found in the ACS PUMS data dictionary.[4] Additionally, `folktexts` is compatible with open-source models running locally, as well as with closed-source models hosted through a web API (e.g., GPT 4o).

In addition to providing a reproducible way to extract risk scores from LLMs, `folktexts` also offers pre-implemented evaluation metrics to benchmark and compare the calibration and accuracy of LLMs, as well as easy plotting of group-conditional calibration curves for a cursory view of potential biases. The package is easy to use within a python notebook, as a dependency, or directly from the command line. Further details on usage and design choices are available in Appendix C.

4 Empirical findings

We use `folktexts` to evaluate several recently released models together with their instruction-tuned counterparts: the Llama 3 models [66], including the 8B and the 70B versions, the Mistral 7B [67], the Mixtral 8x7B and 8x22B variants [68], the Yi 34B [69], and the Gemma [70] 2B and 7B variants. We also evaluate GPT 4o and GPT 4o mini [71] through the OpenAI API (note that no base model version is available). Instruction-tuned models are marked '(i.t.)'. For comparison, results are also shown for a logistic regression (LR) model, and a gradient boosted decision trees model (XGBoost). The XGBoost model [72] is generally regarded as the state-of-the-art in tabular data tasks [73].

In this section we focus on the ACSIncome prediction task, which is the default `folktexts` benchmarking task. It consists in predicting whether a person's income is above or below $50K from 10 demographic features, and closely emulates the popular UCI Adult prediction task [74]. The evaluation test set consists of 160K randomly selected samples from the 2018 Census data. A separate set of 1.5M samples is used to train the supervised LR and XGBoost models. LLMs are used as zero-shot classifiers without fine-tuning. Both multiple-choice prompting and numeric prompting were used to obtain two separate risk score distributions for each model (as described in Section 3.2). We focus on analyzing multiple-choice prompting results, as it is arguably the standard in LLM benchmarking [19–24, 39, 45–48]. Appendix A presents additional results for the experiments analyzed in this section, including a more in-depth look into numeric prompting results, as well as results on alternative prediction tasks. Experiments consumed approximately 500 A100 GPU hours.

4.1 Benchmark results

We perform a comprehensive evaluation of LLM-generated risk scores, summarized in Table 1.

Multiple-choice prompting. We observe that a majority of LLMs (all of size 8B or larger) outperform the linear model baseline (LR) in terms of predictive power (AUC). However, LLMs are clearly far from matching the supervised baselines with respect to calibration: all language models achieve very high calibration error (ECE), while baselines achieve near-perfect calibration. Due to this high miscalibration, models struggle to translate scores with high predictive signal into high accuracy. In fact, while most models achieve high AUC, they struggle to surpass the supervised linear baseline in terms of accuracy. We recall that AUC is agnostic to calibration, while accuracy on the maximum likelihood answer is not. All of the Gemma models have worse than random accuracy, despite having clearly above random AUC (random would be 0.5). Only the instruction-tuned Mistral models (7B, 8x7B, and 8x22B) outperform the linear model in terms of accuracy. Interestingly, while larger models achieve higher AUC, calibration is not reliably improved by model size — differences across model families are more pronounced than across model sizes.

Finally, we focus on comparing base models to their instruction-tuned counterparts, marked with '(it)'. A striking trend is visible across the board: instruction-tuning generally worsens calibration (higher ECE) when using multiple-choice prompting. At the same time, we generally see improvements in AUC and accuracy after instruction-tuning. Appendix A.3 presents results on the four additional prediction tasks. The same trend of instruction-tuning leading to worse calibration and higher AUC is broadly replicated. However, performance across different tasks is somewhat inconsistent: LLMs

 https://doi.org/10.52202/079017-3089

Model	Multiple-choice prompting				Numeric risk prompting			
	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑
GPT 4o (it)	0.18	0.19	0.87	0.80	0.08	0.16	0.85	0.78
GPT 4o mini (it)	0.24	0.24	0.85	0.74	0.05	0.16	0.83	0.78
Mixtral 8x22B (it)	0.21	0.22	0.85	0.76	0.11	0.17	0.84	0.77
Mixtral 8x22B	0.17	0.19	0.85	0.68	0.13	0.18	0.82	0.74
Llama 3 70B (it)	0.27	0.27	0.86	0.69	0.25	0.23	0.84	0.67
Llama 3 70B	0.20	0.20	0.86	0.70	0.27	0.24	0.82	0.54
Mixtral 8x7B (it)	0.16	0.18	0.86	0.78	0.10	0.17	0.84	0.76
Mixtral 8x7B	0.17	0.21	0.83	0.65	0.07	0.17	0.81	0.78
Yi 34B (it)	0.19	0.19	0.86	0.72	0.22	0.21	0.80	0.48
Yi 34B	0.25	0.22	0.85	0.62	0.15	0.19	0.83	0.61
Llama 3 8B (it)	0.32	0.30	0.85	0.62	0.23	0.23	0.81	0.67
Llama 3 8B	0.25	0.26	0.81	0.38	0.14	0.24	0.63	0.40
Mistral 7B (it)	0.21	0.22	0.83	0.77	0.16	0.19	0.83	0.70
Gemma 7B (it)	0.61	0.59	0.84	0.37	0.33	0.30	0.78	0.42
Mistral 7B	0.20	0.23	0.80	0.73	0.36	0.32	0.75	0.49
Gemma 7B	0.24	0.27	0.76	0.37	0.15	0.20	0.80	0.73
Gemma 2B (it)	0.63	0.63	0.73	0.37	0.28	0.31	0.50	0.37
Gemma 2B	0.14	0.25	0.62	0.45	0.37	0.37	0.50	0.63
LR	0.03	0.18	0.79	0.74	0.03	0.18	0.79	0.74
XGBoost	0.00	0.13	0.90	0.82	0.00	0.13	0.90	0.82

Table 1: Zero-shot LLM results on the **ACSIncome** benchmark task, colored from worst (in orange) to best (in cyan). LR and XGBoost baselines were trained on 1.5M samples. Models generally achieve high predictive signal (high AUC) but poor calibration (high ECE). Numeric prompting leads to improved ECE but worse AUC.

generally have stronger predictive signal than a supervised linear model on the income prediction and travel-time prediction tasks, but consistently underperform the linear baseline on ACSMobility.

Numeric prompting. We observe broad improvements in calibration (lower ECE) and Brier score loss across instruction-tuned models when using numeric prompting. Figures A4–A5 of Appendix A.2 show the change between prompting schemes in ECE and AUC, respectively. The ECE improvement trend is clear across most instruct models across all five benchmark tasks. The Yi 34B model is the exception, showing outlier results throughout the different experiments we conduct. Base models show mixed results (small ECE improvements or degradation). At the same time, numeric prompting leads to small but consistent drops in predictive power of risk scores (AUC) on 4 out of 5 benchmark tasks, including ACSIncome. Finally, we point out that the GPT 4o and GPT 4o mini produce a surprisingly well-calibrated risk score distribution for the income prediction task (ECE $= 0.05$). In fact, for each benchmark task, at least one LLM is able to produce a remarkably well-calibrated score distribution (although a different model for different tasks). This promising result points to the fact that some small amount of data will likely always be needed to properly evaluate LLMs capabilities to model human population statistics, but such modeling can often be done with high degree of confidence and a good understanding on which outputs might be wrong.

4.2 Score distribution

To get additional insights into the difference between base models and their instruction-tuned counterparts, we inspect their risk score distributions more closely.

Figure 2 shows calibration curves of the largest LLMs studied, for both base and instruct variants and for both prompting schemes. When using multiple-choice prompting (left-most plots of Fig. 2), both base and instruction-tuned models have poor score calibration, but failure modes are entirely different: Base models output under-confident scores, while instruction-tuned models output over-confident scores. To quantify this result, we introduce a measure of risk score *confidence bias*

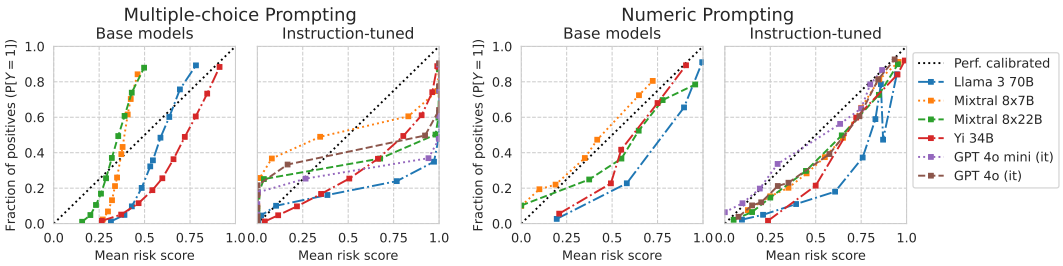


Figure 2: Calibration curves for base and instruction-tuned versions of the largest models studied, on the ACSIncome task. Curves are computed using 10 quantile-based score bins. Risk scores were generated using multiple-choice-style prompting (*left plots*) or numeric chat-style prompting (*right plots*).

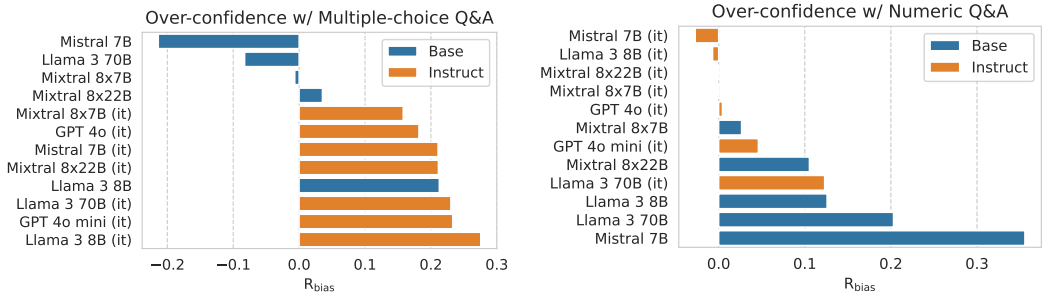


Figure 3: Risk score confidence bias for all LLMs on the ACSIncome task. *Left:* Multiple-choice prompting. *Right:* Numeric prompting. Negative values indicate under-confident risk scores (overestimating uncertainty), while positive values indicate over-confident risk scores (underestimating uncertainty). Instruction-tuned models are generally over-confident when using multiple-choice prompting (*left plot*), but this bias is substantially diminished when using numeric prompting (*right plot*).

similar to the ECE metric: $\mathrm{R}_{\text{bias}} := \sum_{m=1}^{M} \frac{|B_m|}{n} \left[\mathrm{Conf}(B_m) - \mathrm{Acc}(B_m)\right]$, where M is the number of score bins, B_m is the set of samples in score bin m, $\mathrm{Acc}(.)$ is the accuracy on a given set of samples, and $\mathrm{Conf}(.)$ is the confidence on a given set of samples measured as the mean risk score for the highest likelihood class. Figure 3-*left* shows the risk score confidence bias results. The two miscalibration modes are evident: confidence bias is higher for instruction-tuned models and lower (or even negative) for base models. That is, instruction-tuned models are generally over-confident in their predictions, outputting higher scores than their accuracy would warrant, while no such trend is visible for base models. On the other hand, when using numeric prompting, the two aforementioned failure modes are no longer evident, and the differences between base and instruction-tuned models are blurred (right-most plots of Fig. 2). In fact, Figure 3-*right* shows a trend reversal when using numeric prompting: Base models now show a higher over-confidence in their risk scores, while instruction-tuned models show approximately neutral score bias.

Figure 4 shows the risk score distribution for a variety of model pairs, produced using multiple-choice prompting. The score distributions for base and instruct model variants are immediately distinguishable: base models consistently produce low-variance distributions centered around 0.5, while instruction-tuned variants often output scores near 0 or 1. The same trend is visible on all model pairs, with Yi 34B showing the smallest difference between base and instruct variants. The score distributions produced by base/i.t. model pairs are markedly different, even among base/i.t. pairs achieving the exact same AUC; e.g., Llama 3 70B and Mixtral 8x22B. For the largest Llama (70B) and largest Mistral (8x22B) models, no predictive performance is gained by instruction tuning, but answers of the instruction-tuned models have significantly higher (over-)confidence and worse calibration. In fact, while the base Llama 3 70B has an average of 0.07 under-confidence bias, the instruct variant produces risk scores on average 0.22 over-confident (see Figure 3, left). Appendix A.1 investigates whether this under-/over-confidence bias disproportionately affects protected societal sub-groups. We raise concerns regarding subgroup miscalibration, which should caution practitioners against using such scores in consequential domains without a comprehensive fairness audit.

https://doi.org/10.52202/079017-3089

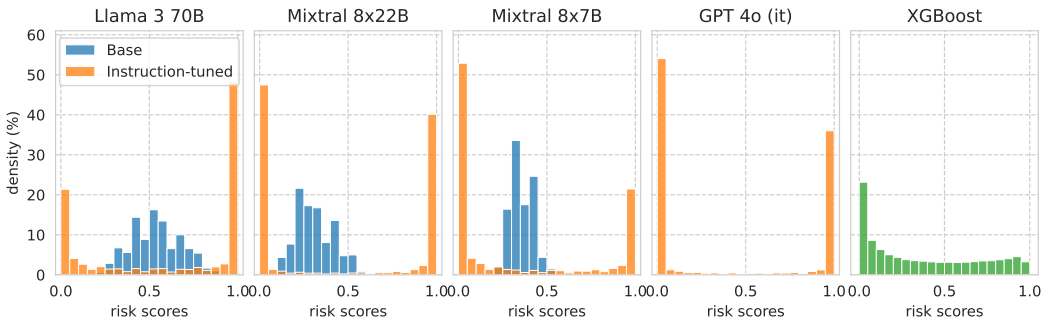


Figure 4: Risk score distribution for base and instruction-tuned model pairs on the ACSIncome task, using *multiple-choice* prompting. After instruction-tuning, models exhibit high confidence, but worse calibration in general. The XGBoost scores showcase a perfectly calibrated distribution (ECE ≈ 0.00). Fig. A6 shows all models.

Crucially, our evaluation reveals a previously unreported shortcoming of using multiple-choice prompting with instruction-tuned models: instruction-tuning polarizes score distributions, even if the true outcome has high entropy. Standard realizable knowledge testing benchmarks can easily disguise this polarization phenomenon as improper quantification of *model* uncertainty. In fact, it seems evident that it is improper quantification of uncertainty in general, regardless of underlying uncertainty in the modeled distribution. The following subsection goes further in-depth on the influence of data uncertainty in score distribution. Appendix A.2 analyzes numeric prompting results.

4.3 Varying degree of uncertainty

Next, we consider the dependence of multiple-choice risk scores on the available evidence. For this study we use the Mixtral 8x7B model, which achieves the best (lowest) Brier score among evaluated models (reflecting both high accuracy and high calibration). We compute income prediction risk scores with increasing evidence: starting with only 2 features, and iteratively adding 2 features at a time (see results in Figure A8). This sequence demonstrates how unrealizable, underspecified prediction tasks differ from realizable prediction tasks. Predicting income based on an individual's place of birth (POBP) and race (RAC1P) is naturally not possible to a high degree of accuracy, forcing any calibrated model to output lower-confidence risk scores. Indeed, both base and instruct variants correctly output lower confidence scores for the smaller feature sets when compared with the larger feature sets (compare left-most to right-most plots of Fig. A8). However, instruction-tuning still leads to a clear polarization of risk score distribution, regardless of true data uncertainty: Score variance for base models is in range $\sigma \in [0.02, 0.06]$, while for i.t. models it's in range $\sigma \in [0.16, 0.41]$. Appendix A.4 goes further in-depth on how score distributions change with varying data uncertainty. Varying individual features also enables us to study LLM feature importance and its main differences to traditional supervised models — analyzed in Appendix A.5.

5 Discussion

We introduced `folktexts`, a software package that provides datasets and tools to evaluate risk scores produced by language models. Unlike most existing LLM benchmarks, the datasets we introduced have inherent outcome uncertainty. While uncertainty on *realizable* tasks reflects only model uncertainty (i.e., whether the model is aware of its lack of knowledge), uncertainty on *unrealizable* tasks is itself a type of knowledge over the underlying data distribution.

Our empirical findings show that LLM risk scores produced using standard multiple-choice Q&A generally have strong predictive signal, but are wildly miscalibrated. Such models may be good for knowledge testing, but lack adequate indicators of uncertainty, making them unsuitable for synthetic data generation. We further reveal that instruction-tuning leads to marked polarization of multiple-choice answer distribution, regardless of ground-truth data uncertainty. This reveals a general inability of instruction-tuned LLMs to quantify uncertainty using multiple-choice Q&A. Conversely, verbally querying models for numeric probability estimates considerably improves calibration of instruction-tuned models, at a small but consistent cost in AUC.

Acknowledgments

We thank Florian Dorner, Mila Gorecki, and Ricardo Dominguez-Olmedo for invaluable feedback on an earlier version of this paper. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting André F. Cruz.

References

[1] Alex Tamkin, Amanda Askell, Liane Lovitt, Esin Durmus, Nicholas Joseph, Shauna Kravec, Karina Nguyen, Jared Kaplan, and Deep Ganguli. Evaluating and mitigating discrimination in language model decisions. *arXiv preprint arXiv:2312.03689*, 2023.

[2] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

[3] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[4] Johann D Gaebler, Sharad Goel, Aziz Huq, and Prasanna Tambe. Auditing the use of language models to guide hiring decisions. *arXiv preprint arXiv:2404.03086*, 2024.

[5] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[6] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

[7] Sam Corbett-Davies, Johann D Gaebler, Hamed Nilforoshan, Ravi Shroff, and Sharad Goel. The measure and mismeasure of fairness. *The Journal of Machine Learning Research*, 24(1): 14730–14846, 2023.

[8] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

[9] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Sarah Flood, Miriam King, Renae Rodgers, Steven Ruggles, and J Robert Warren. *Integrated public use microdata series, Current Population Survey: Version 6.0*. Minneapolis, MN: IPUMS, 2018.

[11] WWTG Peterson, T Birdsall, and We Fox. The theory of signal detectability. *Transactions of the IRE professional group on information theory*, 4(4):171–212, 1954.

[12] Frank Pasquale. *The black box society: The secret algorithms that control money and information*. Harvard University Press, 2015.

[13] Virginia Eubanks. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press, 2018.

[14] Ruha Benjamin. *Race after technology: Abolitionist tools for the new Jim code*. John Wiley & Sons, 2019.

[15] Maximilian Kasy and Rediet Abebe. Fairness, equality, and power in algorithmic decision-making. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 576–586, 2021.

[16] Juan Carlos Perdomo, Tolani Britton, Moritz Hardt, and Rediet Abebe. Difficult lessons on social prediction from wisconsin public schools. *arXiv preprint arXiv:2304.06205*, 2023.

 https://doi.org/10.52202/079017-3089

[17] Juan Carlos Perdomo. The relative value of prediction in algorithmic decision making. In *International Conference on Machine Learning*, 2024.

[18] Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy. *ACM Journal on Responsible Computing*, 1(1):1–45, 2024.

[19] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 5549–5581, 2023.

[20] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=9Vrb9D0WI4`.

[21] Dylan Slack and Sameer Singh. TABLET: Learning from instructions for tabular data. *arXiv*, 2023.

[22] Lisa P. Argyle, Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, 2023.

[23] Nathan E Sanders, Alex Ulinich, and Bruce Schneier. Demonstrations of the potential of ai-based political issue polling. *arXiv preprint arXiv:2307.04781*, 2023.

[24] John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

[25] James Brand, Ayelet Israeli, and Donald Ngwe. Using gpt for market research. *Harvard Business School Marketing Unit Working Paper*, (23-062), 2023.

[26] Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR, 2023.

[27] Danica Dillion, Niket Tandon, Yuling Gu, and Kurt Gray. Can ai language models replace human participants? *Trends in Cognitive Sciences*, 27(7):597–600, 2023.

[28] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950.

[29] Allan H Murphy and Robert L Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 26(1):41–47, 1977.

[30] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[31] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[32] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, page 609–616, 2001.

[33] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[34] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[35] Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7273–7284, 2022.

[36] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

[37] Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024.

[38] Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. A close look into the calibration of pre-trained language models. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 1343–1367, 2023.

[39] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[40] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[41] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[42] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[43] Hanlin Zhang, Yi-Fan Zhang, Yaodong Yu, Dhruv Madeka, Dean Foster, Eric Xing, Himabindu Lakkaraju, and Sham Kakade. A study on the calibration of in-context learning. *Arxiv preprint arxiv:2312.04021*, 2024.

[44] R.M. Groves, F.J. Fowler, M.P. Couper, J.M. Lepkowski, E. Singer, and R. Tourangeau. *Survey Methodology*. Wiley, 2009.

[45] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[46] Esin Durmus, Karina Nyugen, Thomas I Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, et al. Towards measuring the representation of subjective global opinions in language models. *arXiv preprint arXiv:2306.16388*, 2023.

[47] Ricardo Dominguez-Olmedo, Moritz Hardt, and Celestine Mendler-Dünner. Questioning the survey responses of large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024.

[48] Nino Scherrer, Claudia Shi, Amir Feder, and David Blei. Evaluating the moral beliefs encoded in LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 51778–51809, 2023.

[49] Lindia Tjuatja, Valerie Chen, Sherry Tongshuang Wu, Ameet Talwalkar, and Graham Neubig. Do LLMs exhibit human-like response biases? a case study in survey design. *Arxiv preprint arxiv:2311.04076*, 2024.

 https://doi.org/10.52202/079017-3089

[50] Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR, 2018.

[51] Adam Tauman Kalai and Santosh S. Vempala. Calibrated language models must hallucinate. *Arxiv preprint arxiv:2311.14648*, 2024.

[52] Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[53] Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022.

[54] Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of language models. *arXiv preprint arXiv:2404.00474*, 2024.

[55] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models, 2024.

[56] T Anne Cleary. Test bias: Prediction of grades of negro and white students in integrated colleges. *Journal of Educational Measurement*, 5(2):115–124, 1968.

[57] Ben Hutchinson and Margaret Mitchell. 50 years of test (un) fairness: Lessons for machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 49–58, 2019.

[58] Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1939–1948, 2018.

[59] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), 2015.

[60] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[61] Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models(LLMs) on tabular data: Prediction, generation, and understanding – a survey. *Arxiv preprint arxiv:2402.17944*, 2024.

[62] Bowen Yu, Cheng Fu, Haiyang Yu, Fei Huang, and Yongbin Li. Unified language representation for question answering over text, tables, and images. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4756–4765, 2023.

[63] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. TableGPT: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, 2020.

[64] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL `https://openreview.net/forum?id=g3faCfrwm7`.

[65] Joshua Robinson and David Wingate. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*, 2023.

[66] AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[67] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *Arxiv preprint arxiv:2310.06825*, 2023.

[68] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[69] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

[70] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology. *Arxiv preprint arxiv:2403.08295*, 2024.

[71] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[72] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

[73] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024. doi: 10.1109/TNNLS.2022.3229161.

[74] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[75] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

 https://doi.org/10.52202/079017-3089

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 1.2.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 1.2.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code and results available at `https://github.com/socialfoundations/folktexts`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Error bars are computed and plotted whenever relative comparisons are made within the same plot (e.g., group-conditional calibration plots).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Experiments were ran on an internal cluster with Nvidia-A100-80GB GPUs, consuming an approximate total of 500 GPU hours.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] All assets use licenses that allow for free non-commercial use, as listed in the software package `README.md` file.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The Public Use Microdata Sample (PUMS) was collected and made publicly available by the US Census Bureau. Its use in scientific works is well established. Please consult the data documentation[4] for details on data gathering consent.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data we use is governed by the US Census Bureau. PUMS data does not contain personally identifiable information or offensive content. Please consult the data documentation[4] for further details.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Additional results

This appendix section shows additional results and corresponding plots to support the insights presented in Section 4. Section A.1 presents an investigation and discussion into group-conditional miscalibration of LLM risk scores. Section A.2 shows results using a chat-style verbalized numeric prompting scheme. Section A.3 shows results on four extra benchmark tasks made available with `folktexts`. Section A.4 goes further in-depth on how to use `folktexts` to control data uncertainty in the benchmark prediction tasks. Finally, Section A.5 presents and discusses results on feature importance for LLM predictions.

A.1 Subgroup calibration results

In this section, we evaluate risk score calibration on the income prediction task across different subpopulations, such as typically done as part of a fairness audit. For random variables (R, S, Y), the *sufficiency* fairness criterion [8] dictates that the label Y should be independent of the sensitive attribute S conditioned on the score R; i.e., $Y \perp S|R$. Simply put, model score miscalibration should not disproportionately affect a particular societal sub-group.

Figures A1–A2 show group-conditional calibration curves for all models on the ACSIncome task, evaluated on three subgroups specified by the race attribute in the ACS data. We show the three race categories with largest representation. Note that a positive prediction of $Y = 1$ is arguably the advantageous outcome, as it corresponds to the high-income category ("Earns above $50,000 per year"). The 'Mixtral 8x22B' and 'Yi 34B' models shown are the worst offenders, where samples belonging to the 'Black' population see consistently lower scores for the same positive label probability when compared to the 'Asian' or 'White' populations. In other words, for the same score $R = r$, the probability of a positive label $Y = 1$ is higher for $S =$ 'Black' individuals: $\mathbb{P}[Y = 1|R = r, S =$ 'Black'$] \geq \mathbb{P}[Y = 1|R = r, S \neq$ 'Black'$]$, where S denotes the census encoding of *race*. On average, the 'Mixtral 8x22B (it)' model classifies a Black individual with a 0.17 lower score than an Asian individual with the same true probability of high-income, $\mathbb{P}\{Y = 1\}$. This bias is 0.13 for the 'Yi 34B (it)' model. This poses a higher bar for Black individuals to get a "high-income" prediction. Note that the remaining models studied show much smaller differences in group-conditional calibration. In fact, this score bias can be reversed for some *base* models, overestimating scores from Black individuals compared with other subgroups. The mechanism behind this phenomenon is analyzed in the following paragraphs.

Group-conditional signed calibration error We can quantify a model's score bias by evaluating the *signed calibration error* (SCE):

$$\text{SCE} \coloneqq \frac{1}{n} \sum_{m=1}^{M} \sum_{i \in B_m} (r_i - y_i), \tag{3}$$

where M is the number of score buckets, B_m is the set of sample indices belonging to bucket m, $r_i = f_\theta(x_i)$ is the risk score given to sample x_i and y_i is its label. This metric does not evaluate overall calibration, as a value of 0 does not indicate a calibrated classifier. Instead, negative/positive values indicate a bias towards lower/higher risk scores, respectively. If lower scores are related to unfavorable real-world outcomes (e.g., low chance of loan repayment), then a bias towards lower scores on samples of specific protected subgroups would lead to unfair real-world outcomes. Conversely, if lower scores are associated with a favorable outcome (e.g., low chance of recidivism), then a bias towards lower scores would be beneficial for the affected group.

Figure A3 shows the difference between the signed calibration error (SCE) on different group pairings, on the ACSIncome task. Positive predictions ($Y = 1$) correspond to the advantaged high-income class. Negative differences, $\Delta_{SCE} < 0$, indicates advantaged scores for Black individuals, while positive differences, $\Delta_{SCE} > 0$, indicate disadvantaged scores for Black individuals. Interestingly, while subgroup calibration curves appear similar for base models and disadvantage Black individuals for instruction-tuned models (see Figures A1–A2), this is not entirely reflected on the SCE metric. Indeed, a clear-cut split is visible: base models benefit the score of Black individuals, and instruct

https://doi.org/10.52202/079017-3089

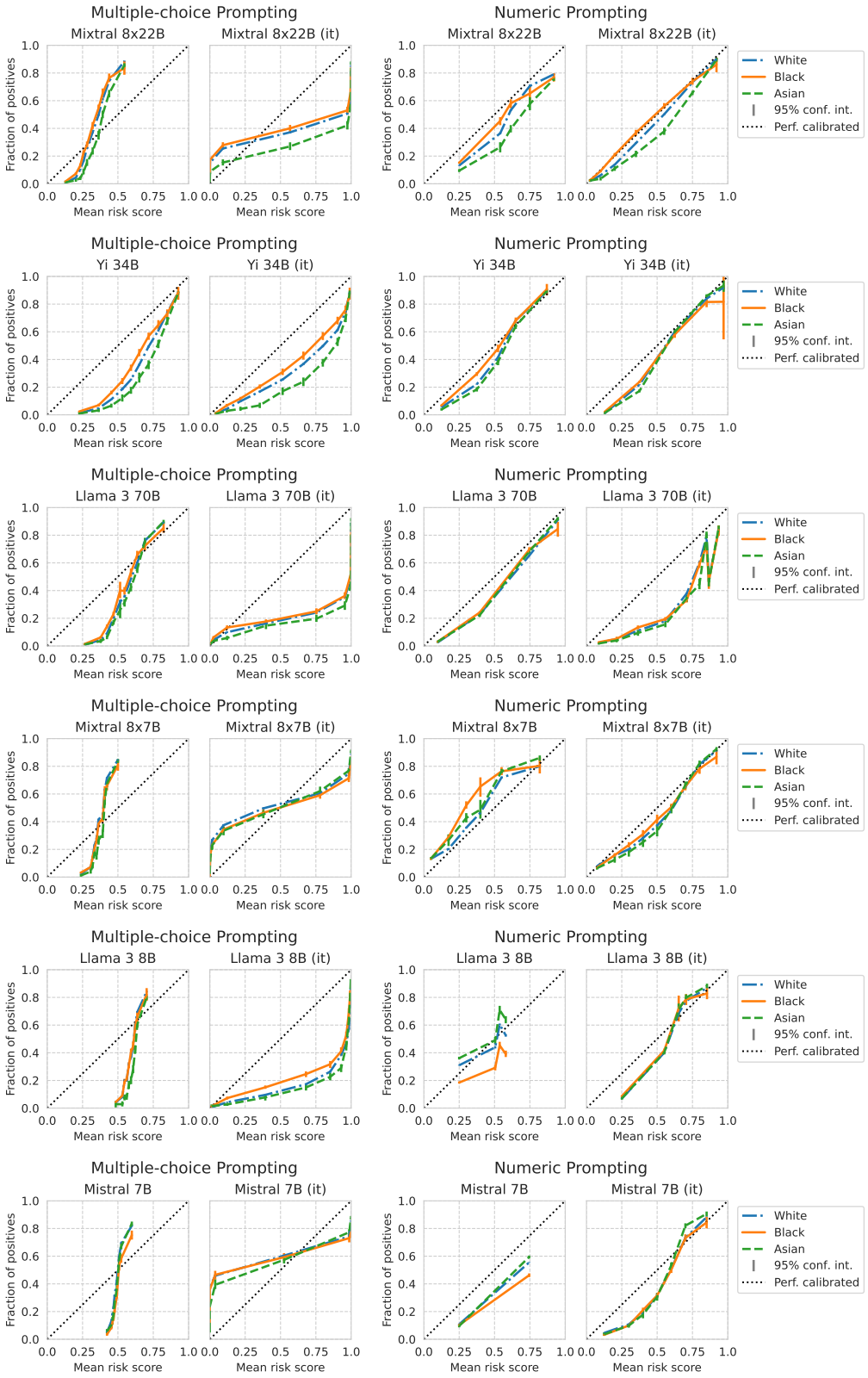
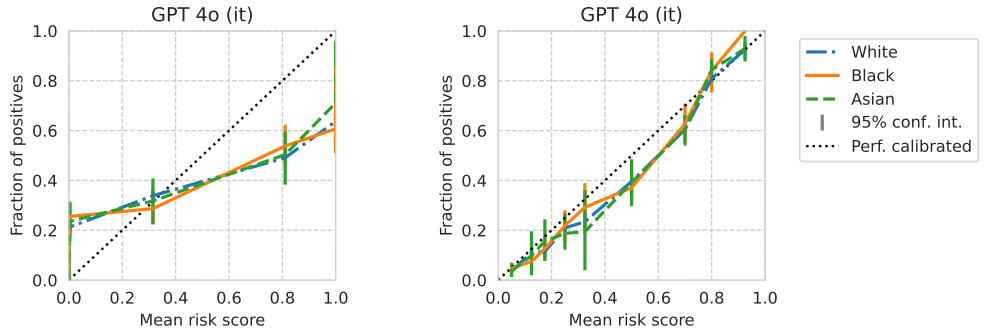
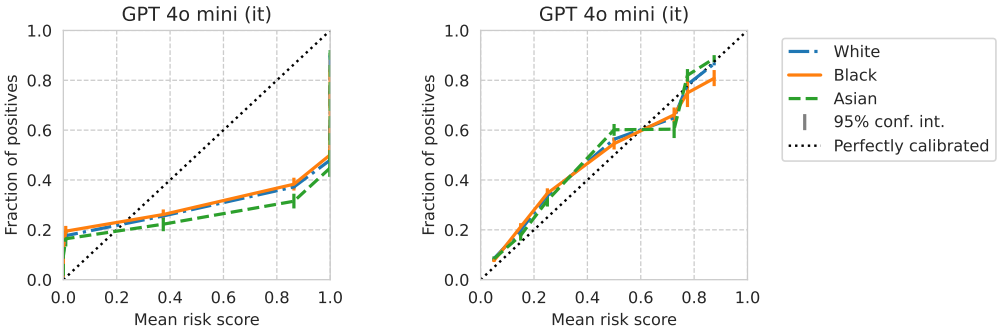


Figure A1: [**ACSIncome**] Calibration curves across different census race sub-populations, computed using 10 quantile-based score bins, with 95% confidence intervals. The 'Mixtral 8x22B' and 'Yi 34B' models are the worst offenders in terms of group-conditional miscalibration; i.e., they violate the *sufficiency* fairness criterion.



(a) Multiple-choice prompting.

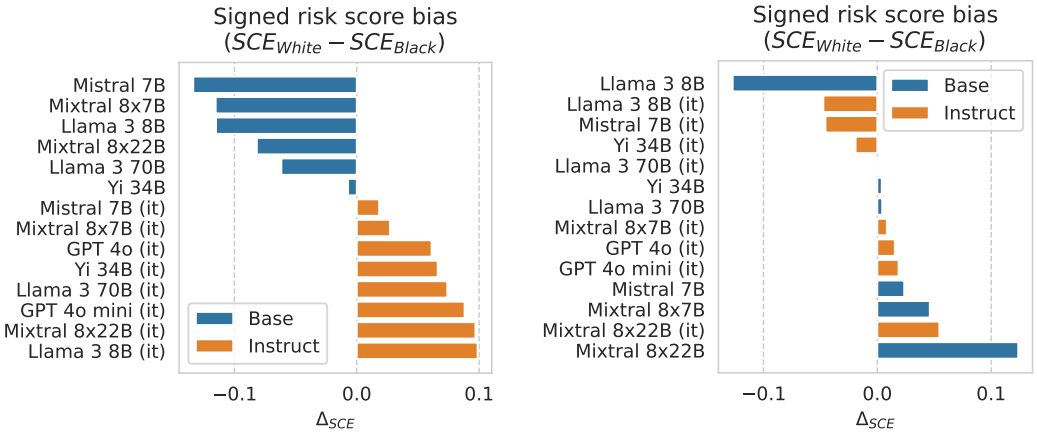
(b) Numeric prompting.



(c) Multiple-choice prompting.

(d) Numeric prompting.

Figure A2: [**ACSIncome**] Calibration curves for the 'GPT 4o' (top) and 'GPT 4o mini' (bottom) models, across different census race sub-populations, computed using 10 quantile-based bins. No base model variant exists for these models. Numeric prompting (Fig. A2b and A2d) leads to significant improvements in calibration, as well as reduced differences in group-conditional calibration for the 'GPT 4o mini' model.



(a) Multiple-choice prompting.

(b) Numeric prompting.

Figure A3: [**ACSIncome**] Racial group bias in risk score calibration error, between White and Black sub-groups, $\mathrm{SCE}_{\mathrm{White}} - \mathrm{SCE}_{\mathrm{Black}}$. When comparing score bias of group A with score bias of group B, $\Delta_{\mathrm{SCE}} = \mathrm{SCE}_A - \mathrm{SCE}_B$, positive values indicate an undue score advantage of group A, and negative values an undue score advantage of group B. *Left:* Using multiple-choice prompting. *Right:* Using numeric prompting. Note that the Gemma models were omitted, as their instruct versions degenerate into strictly predicting the same outcome for all samples. Consequently, the two instruct Gemma models are the only exceptions to the trend shown in these plots.

https://doi.org/10.52202/079017-3089

models disadvantage the score of Black individuals. This finding could be partially explained by the fact that base models produce score distributions with low variance, and instruct models produce high-variance polarized outcomes. Specifically, two conclusions can be drawn from the score distributions produced by base models: (1) models under-estimate the score of high-income earners (which are disproportionately Asian), and (2) models over-estimate the score of low-income earners (which are disproportionately Black). This simple statistical fact leads base models to over-estimate the earnings of the Black population disproportionately to other groups. Crucially, the opposite is true for instruction-tuned models: high-income earners see their score over-estimated, which benefits groups with a higher prevalence of high earners.

Such differences in score calibration arguably warrant a more in-depth analysis that escapes the scope of this paper. We raise concerns regarding subgroup miscalibration, which should caution practitioners against using such scores in consequential domains without a comprehensive fairness audit. This in no way serves as an exhaustive analysis of risk score fairness on LLMs, as it is bound to be highly task dependent and language model dependent. We simply surface the fact that, on this income prediction task, risk scores are not group-calibrated [8] and could lead to unfair outcomes. Crucially, even though race has the lowest mean feature importance among all features (see Appendix A.5), we report and explain how different trends in risk score distributions can effectively lead to unfair outcomes.

A.2 Additional results using verbalized numeric prompting

Figure A4 shows the change in calibration error (ECE) between using multiple-choice prompting and verbalized numeric prompting, on all five benchmark tasks. Instruction-tuned models (top rows) show ECE improvements on an overwhelming majority of model/task pairs, while base models (bottom rows) show less consistent results. However, using numeric prompting comes at a consistent cost of diminished predictive power (AUC) of the risk scores, shown in Figure A5. A majority of model/task pairs have worse AUC with numeric prompting, with the exception of the employment prediction task. One potential explanation for this generalized decrease in AUC lies in the fact that numeric prompting generates a large number of tied risk scores. Figure A7 shows the score distribution of all models using numeric prompting (compare with multiple-choice prompting shown in Figure 4). While multiple-choice prompting produces a smoother continuous score distribution, numeric prompting generally results in a small set of possible uncertainty estimates. This arguably makes intuitive sense, as numeric prompting produces uncertainty estimates in discrete token space, while multiple-choice prompting produces uncertainty estimates in the continuous token-probability space.

A.3 Results on additional benchmark tasks

The main body of the paper focuses on results on the ACSIncome prediction task. This task is arguably the most popular for benchmarking models on tabular data, as it closely mirrors the older but widely used UCI Adult dataset [9, 74]. The `folktexts` package makes available natural-language versions of four additional tabular data tasks: ACSEmployment, ACSMobility, ACSTravelTime, and ACSPublicCoverage. The GPT 4o model was only ran on the main ACSIncome task, due to the high API costs of querying it on millions of tabular data points. The GPT 4o mini variant was ran on all tasks.

Tables A1–A4 show results for the ACSEmployment, ACSMobility, ACSTravelTime, and ACSPublicCoverage tasks, respectively. Trends discussed in the main body of the paper are broadly confirmed. Models' moderate predictive performance is accompanied by substantial miscalibration of their risk scores. Additionally, base models output low-variance high-uncertainty score distributions, while instruction-tuned models output high-variance low-uncertainty score distributions. There is clear predictive signal for large models across all tasks (e.g., see the AUC of Llama 3 70B it, or Mixtral 8x22B it). However, the extent to which models' scores are predictable varies substantially across tasks. Most models surpass the AUC of the linear baseline (LR) on the ACSTravelTime task, as well as the main ACSIncome task; but consistently lag behind the linear baseline on the ACSMobility task. On the ACSEmployment and ACSPublicCoverage tasks, the best performing models manage to match the linear baseline AUC.

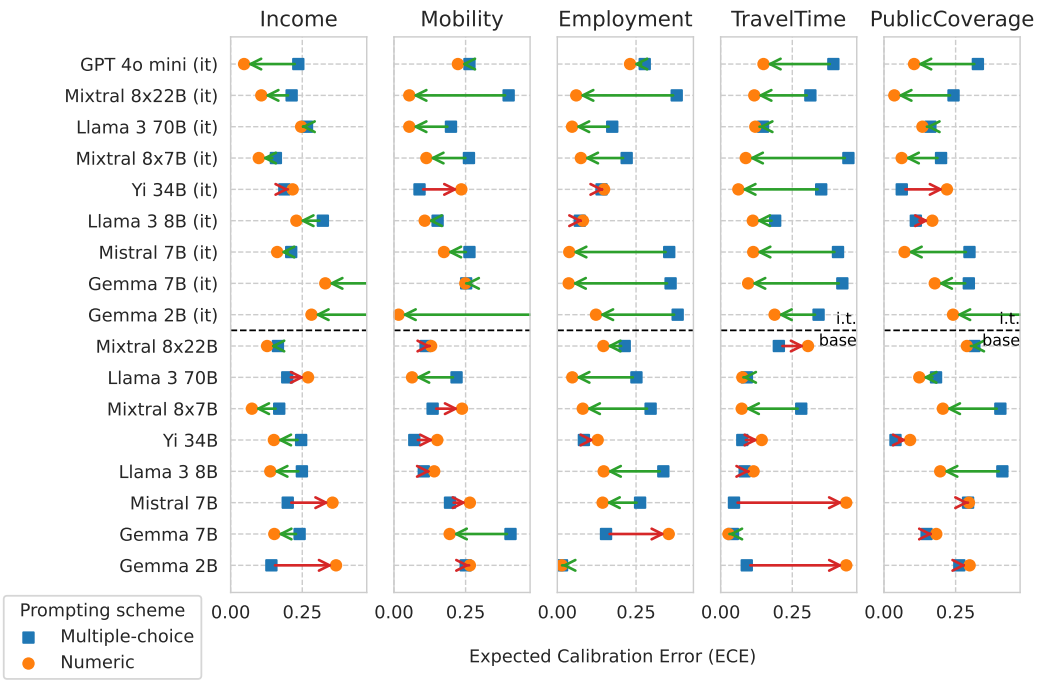


Figure A4: Change in calibration error (ECE) when using numeric risk prompting (●) versus multiple-choice prompting (■). Instruction-tuned models (*top rows*) show substantial calibration improvements, while base models (*bottom rows*) show mixed results. Green/red arrows signal ECE improvement/degradation.

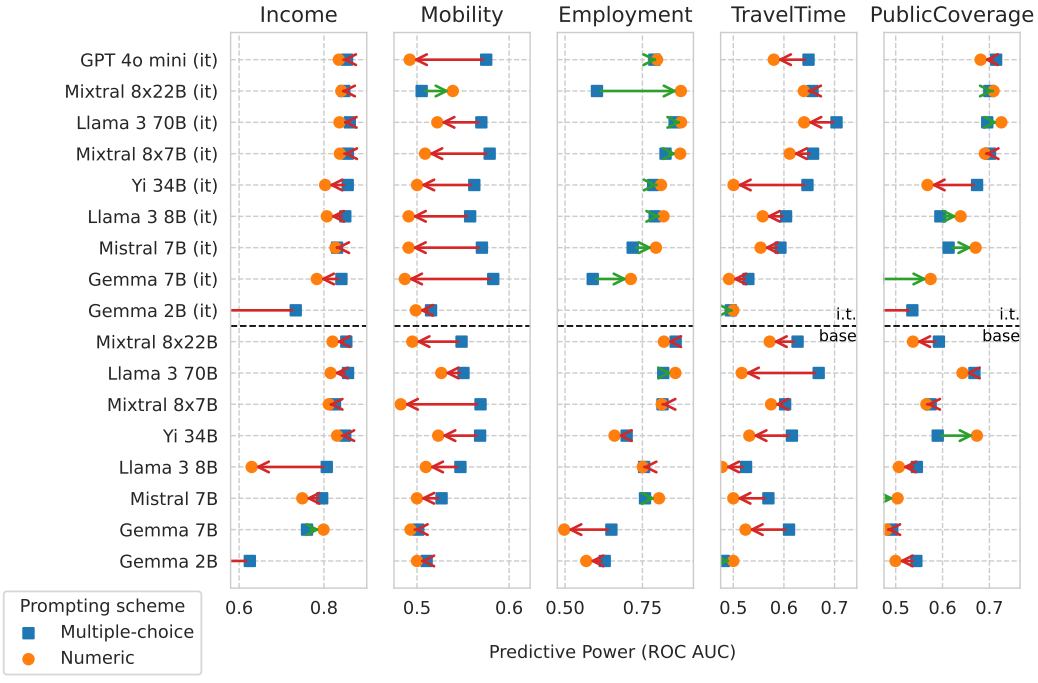


Figure A5: Change in predictive power (AUC) when using numeric risk prompting (●) versus multiple-choice prompting (■). Both instruction-tuned models (*top rows*) and base models (*bottom rows*) generally achieve worse AUC with numeric prompting. Green/red arrows signal AUC improvement/degradation, respectively.

https://doi.org/10.52202/079017-3089

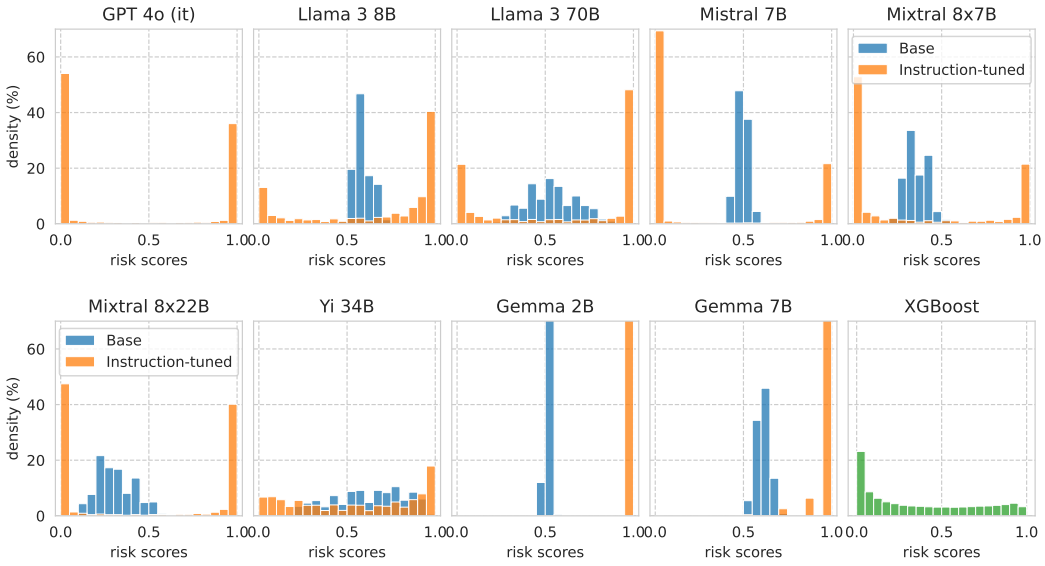


Figure A6: Risk score distribution for base and instruction-tuned model pairs on the ACSIncome task, using *multiple-choice* prompting. After instruction-tuning, models exhibit high confidence, but worse calibration in general.

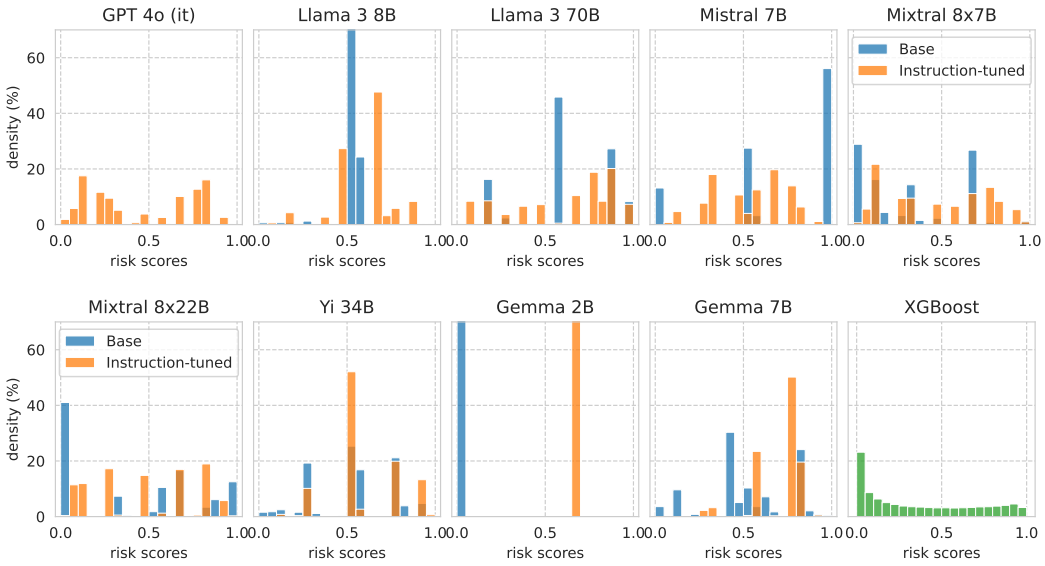


Figure A7: Distribution of risk scores produced using *numeric prompting* on the ACSIncome benchmark task. A baseline score distribution that achieves 0.00 calibration error is shown in green (XGBoost model). For each model, risk scores produced in this manner fall into only a few different possible values, contrasting with the neatly continuous distribution produced by multiple-choice prompting. This fact leads to numerous more ties among predicted risk scores, which can explain the reduced AUC performance with this prompting scheme. Nonetheless, calibration error is considerably smaller for instruction-tuned models.

https://doi.org/10.52202/079017-3089

These findings pose into question one of the main advantages of using LLMs for risk scoring: the fact that no labeled data is required. Given the inconsistency of model performance, some small amount of testing data may always be needed to assert reliability of results.

A.4 Varying uncertainty

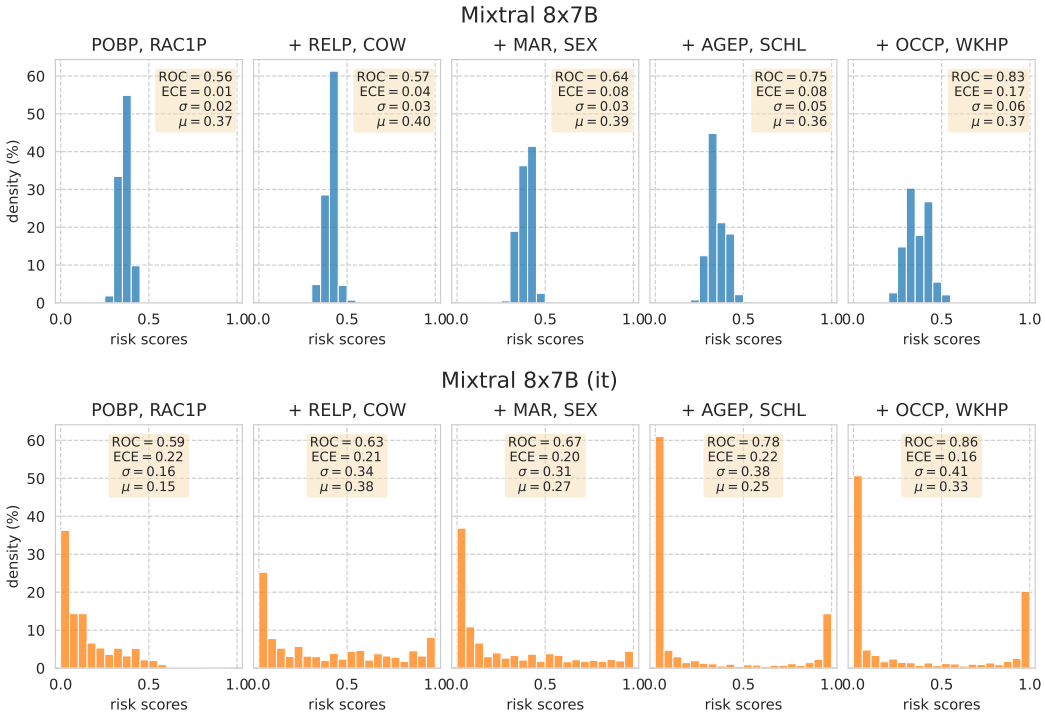


Figure A8: [**ACSIncome**; **Multiple-choice**] Shift of score distribution with increasing evidence for the Mixtral 8x7B model (which achieved the best Brier score), using multiple-choice Q&A on the ACSIncome task. Features are described in Table A5. Score distribution gets more discriminative as more evidence is added, successfully increasing scores' predictive signal (AUC). The true label prevalence is $\mathbb{P}[Y = 1] = 0.37$.

A simple API call in our package allows for selecting different subsets of attributes to include as features when using the LLM as a predictor. Figure A9 inspects the effect of increasing the feature on models' calibration and predictive power. Each dot along the line represents an increasing feature set used for LLM predictions, added in order of mean feature importance on all models. Appendix B details all features used in the ACSIncome task. We refer the reader to Ding et al. [9] and the ACS codebook[5] for an in-depth description of each categorical value a feature can take. Predictive signal (ROC AUC) reliably increases with each added feature, for all tested models except the Gemma 2B variants. This is expected with standard supervised learning algorithms trained and evaluated on i.i.d. data, but arguably somewhat unexpected of pre-trained models trained on a variety of datasets that are out-of-distribution relative to the evaluation set. On the other hand, there is no clear trend for calibration: for Mistral models, it seems that calibration actually worsens for larger feature sets, while for Llama models calibration is approximately stable across all points.

This experiment show-cases one unique way of using LLMs with survey prediction tasks: while supervised learning models would have to be retrained every time a different feature set is used, LLMs can freely change the evidence they use to make a prediction. If a model were to exhibit properties of a joint distribution with the ability to marginalize over hidden features, then calibration with respect to evidence $\mathcal{X}$ implies that it is also calibrated with respect to restricted evidence $\mathcal{X}' \subset \mathcal{X}$. To what extent a model satisfies such properties is an interesting question for future work; we hope our package proves useful as an investigative tool.

[5] See the ACS PUMS data dictionary for the full list of available variables:
`https://www.census.gov/programs-surveys/acs/microdata/documentation.html`

https://doi.org/10.52202/079017-3089

Model	Multiple-choice prompting				Numeric risk prompting			
	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑
GPT 4o mini (it)	0.28	0.29	0.79	0.65	0.23	0.23	0.80	0.73
Mixtral 8x22B (it)	0.38	0.39	0.60	0.51	0.06	0.14	0.87	0.79
Mixtral 8x22B	0.21	0.24	0.86	0.52	0.15	0.18	0.82	0.80
Llama 3 70B (it)	0.17	0.19	0.85	0.73	0.05	0.14	0.88	0.81
Llama 3 70B	0.25	0.26	0.82	0.52	0.05	0.15	0.86	0.78
Mixtral 8x7B (it)	0.22	0.24	0.82	0.73	0.07	0.15	0.87	0.78
Mixtral 8x7B	0.30	0.31	0.81	0.45	0.08	0.17	0.81	0.73
Yi 34B (it)	0.14	0.21	0.79	0.69	0.15	0.21	0.81	0.51
Yi 34B	0.08	0.23	0.70	0.62	0.13	0.23	0.66	0.50
Llama 3 8B (it)	0.07	0.19	0.79	0.74	0.08	0.17	0.82	0.77
Llama 3 8B	0.34	0.34	0.76	0.45	0.15	0.23	0.75	0.46
Mistral 7B (it)	0.35	0.36	0.72	0.63	0.04	0.19	0.79	0.69
Mistral 7B	0.26	0.30	0.76	0.45	0.14	0.19	0.80	0.79
Gemma 7B (it)	0.36	0.38	0.59	0.58	0.04	0.22	0.71	0.60
Gemma 7B	0.15	0.25	0.65	0.48	0.35	0.38	0.50	0.51
Gemma 2B (it)	0.38	0.41	0.42	0.46	0.12	0.27	0.46	0.46
Gemma 2B	0.01	0.24	0.63	0.54	0.01	0.23	0.57	0.53
LR	0.02	0.15	0.86	0.78	0.02	0.15	0.86	0.78
XGBoost	0.00	0.12	0.91	0.83	0.00	0.12	0.91	0.83

Table A1: Zero-shot LLM results on the **ACSEmployment** benchmark task, together with supervised learning baselines fitted on 2.9M samples.

Model	Multiple-choice prompting				Numeric risk prompting			
	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑
GPT 4o mini (it)	0.26	0.26	0.57	0.73	0.22	0.25	0.49	0.73
Mixtral 8x22B (it)	0.40	0.40	0.51	0.39	0.05	0.20	0.54	0.73
Mixtral 8x22B	0.11	0.21	0.55	0.73	0.13	0.22	0.49	0.73
Llama 3 70B (it)	0.20	0.25	0.57	0.58	0.05	0.20	0.52	0.73
Llama 3 70B	0.22	0.24	0.55	0.53	0.06	0.20	0.53	0.73
Mixtral 8x7B (it)	0.26	0.26	0.58	0.73	0.11	0.21	0.51	0.73
Mixtral 8x7B	0.14	0.21	0.57	0.73	0.24	0.25	0.48	0.73
Yi 34B (it)	0.09	0.20	0.56	0.72	0.23	0.25	0.50	0.27
Yi 34B	0.07	0.20	0.57	0.73	0.15	0.23	0.52	0.44
Llama 3 8B (it)	0.15	0.22	0.56	0.70	0.11	0.21	0.49	0.73
Llama 3 8B	0.10	0.20	0.55	0.73	0.14	0.21	0.51	0.72
Mistral 7B (it)	0.26	0.26	0.57	0.73	0.17	0.23	0.49	0.73
Mistral 7B	0.20	0.23	0.53	0.73	0.27	0.27	0.50	0.73
Gemma 7B (it)	0.25	0.26	0.58	0.73	0.25	0.26	0.49	0.73
Gemma 7B	0.41	0.37	0.50	0.27	0.19	0.24	0.49	0.73
Gemma 2B (it)	0.73	0.73	0.52	0.27	0.02	0.20	0.50	0.73
Gemma 2B	0.25	0.26	0.51	0.34	0.27	0.27	0.50	0.73
LR	0.02	0.19	0.61	0.74	0.02	0.19	0.61	0.74
XGBoost	0.00	0.16	0.74	0.76	0.00	0.16	0.74	0.76

Table A2: Zero-shot LLM results on the **ACSMobility** benchmark task, together with supervised learning baselines fitted on 0.6M samples.

Model	Multiple-choice prompting				Numeric risk prompting			
	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑
GPT 4o mini (it)	0.39	0.40	0.65	0.55	0.15	0.27	0.58	0.57
Mixtral 8x22B (it)	0.31	0.33	0.66	0.59	0.12	0.24	0.64	0.59
Mixtral 8x22B	0.20	0.28	0.63	0.44	0.30	0.34	0.57	0.58
Llama 3 70B (it)	0.15	0.24	0.70	0.60	0.12	0.24	0.64	0.53
Llama 3 70B	0.09	0.24	0.67	0.55	0.08	0.25	0.52	0.46
Mixtral 8x7B (it)	0.45	0.45	0.66	0.52	0.09	0.24	0.61	0.57
Mixtral 8x7B	0.28	0.32	0.60	0.44	0.07	0.25	0.57	0.58
Yi 34B (it)	0.35	0.36	0.65	0.56	0.06	0.25	0.50	0.44
Yi 34B	0.08	0.24	0.62	0.56	0.14	0.27	0.53	0.44
Llama 3 8B (it)	0.19	0.28	0.60	0.57	0.11	0.25	0.56	0.56
Llama 3 8B	0.08	0.25	0.53	0.56	0.12	0.26	0.48	0.44
Mistral 7B (it)	0.41	0.42	0.59	0.57	0.11	0.25	0.55	0.56
Mistral 7B	0.05	0.25	0.57	0.56	0.44	0.44	0.50	0.56
Gemma 7B (it)	0.42	0.43	0.53	0.56	0.10	0.26	0.49	0.44
Gemma 7B	0.04	0.24	0.61	0.58	0.03	0.25	0.52	0.55
Gemma 2B (it)	0.34	0.36	0.49	0.56	0.19	0.28	0.50	0.56
Gemma 2B	0.09	0.26	0.48	0.44	0.44	0.44	0.50	0.56
LR	0.04	0.24	0.58	0.56	0.04	0.24	0.58	0.56
XGBoost	0.02	0.19	0.77	0.70	0.02	0.19	0.77	0.70

Table A3: Zero-shot LLM results on the **ACSTravelTime** benchmark task, together with supervised learning baselines fitted on 1.3M samples.

Model	Multiple-choice prompting				Numeric risk prompting			
	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑	ECE ↓	Brier score ↓	AUC ↑	Acc. ↑
GPT 4o mini (it)	0.33	0.34	0.71	0.60	0.10	0.20	0.68	0.73
Mixtral 8x22B (it)	0.24	0.25	0.70	0.72	0.04	0.18	0.71	0.75
Mixtral 8x22B	0.32	0.30	0.59	0.30	0.29	0.29	0.54	0.70
Llama 3 70B (it)	0.16	0.21	0.69	0.75	0.13	0.20	0.73	0.75
Llama 3 70B	0.18	0.22	0.67	0.63	0.12	0.21	0.64	0.53
Mixtral 8x7B (it)	0.20	0.23	0.70	0.74	0.06	0.19	0.69	0.74
Mixtral 8x7B	0.41	0.37	0.57	0.30	0.20	0.25	0.56	0.70
Yi 34B (it)	0.06	0.19	0.67	0.74	0.22	0.24	0.57	0.31
Yi 34B	0.04	0.21	0.59	0.70	0.09	0.20	0.67	0.64
Llama 3 8B (it)	0.11	0.21	0.59	0.71	0.17	0.22	0.64	0.68
Llama 3 8B	0.41	0.38	0.55	0.30	0.20	0.25	0.51	0.34
Mistral 7B (it)	0.30	0.30	0.61	0.70	0.07	0.20	0.67	0.65
Mistral 7B	0.29	0.30	0.45	0.30	0.30	0.30	0.50	0.70
Gemma 7B (it)	0.30	0.34	0.46	0.50	0.18	0.24	0.57	0.61
Gemma 7B	0.15	0.23	0.49	0.49	0.18	0.26	0.48	0.70
Gemma 2B (it)	0.70	0.70	0.54	0.30	0.24	0.29	0.42	0.42
Gemma 2B	0.26	0.28	0.54	0.30	0.30	0.30	0.50	0.70
LR	0.03	0.19	0.70	0.72	0.03	0.19	0.70	0.72
XGBoost	0.00	0.14	0.84	0.80	0.00	0.14	0.84	0.80

Table A4: Zero-shot LLM results on the **ACSPublicCoverage** benchmark task, together with supervised learning baselines fitted on 1.0M samples.

https://doi.org/10.52202/079017-3089

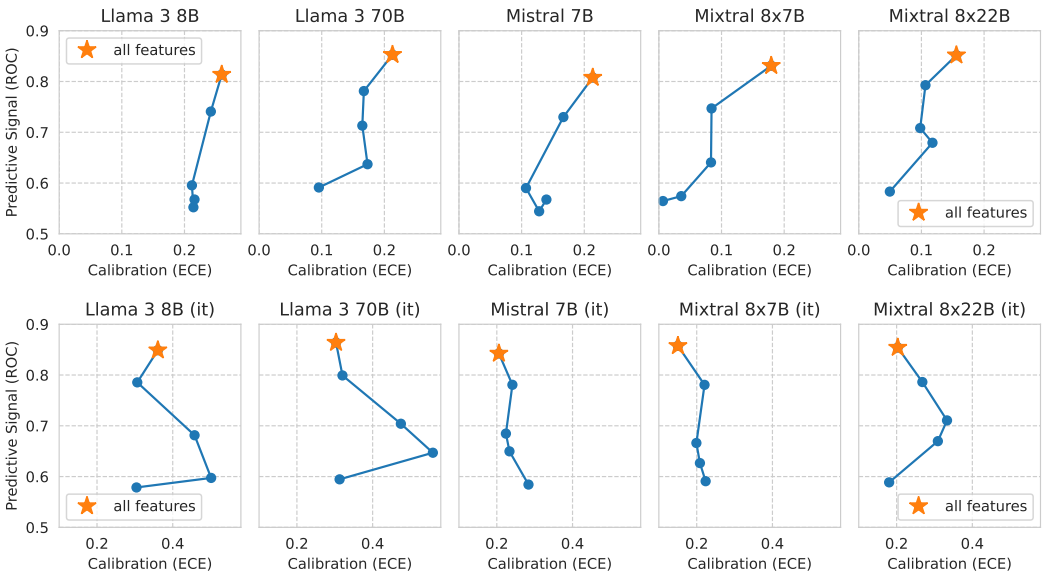


Figure A9: [**ACSIncome**; **Multiple-choice**] Evaluation of calibration (ECE) and predictive performance (AUC) on Llama and Mistral models, with an increasing number of features provided. For each dot along the line we add two features, up to all 10 features being used in the point marked with a star. *Top row*: base models. *Bottom row*: instruction-tuned models. Models can successfully use each extra feature to increase predictive signal. Calibration trends worse the more features are added for base models, while instruct models show no clear trend.

A.5 Feature importance

In this section, we present feature importance results for different LLMs on the ACSIncome prediction task. The importance value of feature j is computed as the drop in AUC after permuting all values of feature j across the dataset. That is, each sample x, sees its value for feature j randomly permuted with another sample. This is a common feature importance implementation [75], as it does not rely on any internal characteristics of the model.

Figure A10 shows feature importance values for the largest language models studied (above 40B active parameters). Results for the XGBoost model are also shown in green. Note that XGBoost achieves the best result on every single metric in Table 1. While for supervised models, a given categorical value is nothing more than a 1 or 0, LLMs have the potential to surface the real-world meaning of such values, benefiting from the rich embedding representations of each category. As such, we'd expect to see LLMs assigning higher importance to categorical features. Indeed, Llama 3 models assign considerably higher importance to the occupation feature (OCCP), which is a numerically encoded categorical feature with over 500 different possible values. Conversely, the XGBoost model assigns considerably higher importance than LLMs to 'work-hours per week' (WKHP) and 'age' (AGEP), both integer-encoded features. Lastly, feature importance results indicate that the studied LLMs do not explicitly use sensitive categories such as age (AGEP), sex (SEX), or race (RAC1P) for risk score estimation.

Interestingly, feature importance is similar for base and instruct variants of the same model. This contrasts with the score distribution and calibration curve results, where all base models followed a similar trend, distinct from their instruction-tuned versions.

B Details on provided benchmark tasks

The `folktexts` package defines natural-text mappings for a variety of columns in the ACS PUMS data files. Table A5 lists and describes each implemented column-to-text mapping. Any combination of column-to-text objects can be used to create a prediction task from ACS data, both as features and as the prediction target. To enable straightforward comparison with existing benchmarks, we mimic the feature set and population filters used by the prediction tasks available in the popular

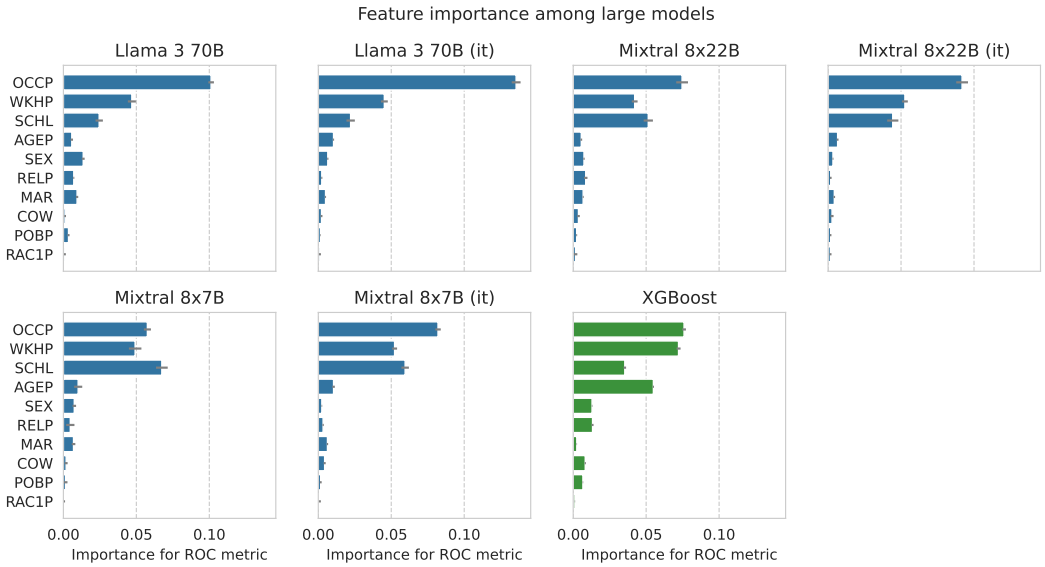


Figure A10: [**ACSIncome**; **Multiple-choice**] Feature importance among the largest language models tested, plus results for the XGBoost baseline. Feature importance values are calculated as the loss in AUC when the values of a given column are randomly permuted [75].

`folktables` benchmark package [9]. Specifically, we put forth natural-text variants of the ACSIncome, ACSPublicCoverage, ACSMobility, ACSEmployment, and ACSTravelTime tasks. These prediction tasks define a restricted set of columns from the ACS PUMS data files to be used as input features for machine learning models, as well as a binarized target column. As such, we extend the use of these ACS prediction tasks to benchmark language models, enabling direct comparison with a wide-ranging set of literature works. Although any ACS survey year could be used for benchmarking, we define the standard set of benchmark tasks as those using data from the 2018 1-year-horizon person-level survey (following Ding et al. [9]). Notwithstanding, we welcome the addition of new column-to-text mappings by new users of the package, both for ACS data and for new datasets. The following paragraphs detail each pre-implemented prediction task.

ACSIncome The goal of the ACSIncome task is to predict whether a person's yearly income is above $50,000, given by the PINCP column. The ACS columns used as features are: AGEP, COW, SCHL, MAR, OCCP, POBP, RELP, WKHP, SEX, and RAC1P. The sub-population over which the task is conducted is employed US residents with age greater than 16 years. The ACSIncome prediction task was put-forth as the successor to the popular UCI Adult dataset [74], used extensively in the algorithmic fairness literature. This task is the default task when running the `folktexts` benchmark.

ACSPublicCoverage The goal of the ACSPublicCoverage task is to predict whether an individual is covered by public health insurance, given by the PUBCOV column. The ACS columns used as features are: AGEP, SCHL, MAR, SEX, DIS, ESP, CIT, MIG, MIL, ANC, NATIVITY, DEAR, DEYE, DREM, PINCP, ESR, ST, FER, and RAC1P. The sub-population over which the task is conducted is US residents with age below 65 years old, and with personal income below $30,000.

ACSMobility The goal of the ACSMobility task is to predict whether an individual has changed their home address in the last year, given by the MIG column. The ACS columns used as features are: AGEP, SCHL, MAR, SEX, DIS, ESP, CIT, MIL, ANC, NATIVITY, RELP, DEAR, DEYE, DREM, RAC1P, COW, ESR, WKHP, JWMNP, and PINCP. The sub-population over which the task is conducted is US residents with age between 18 and 35.

ACSEmployment The goal of the ACSEmployment is to predict whether an individual is employed, given by the ESR column. The ACS columns used as features are: AGEP, SCHL, MAR, SEX, DIS,

https://doi.org/10.52202/079017-3089

ESP, MIG, CIT, MIL, ANC, NATIVITY, RELP, DEAR, DEYE, DREM, and RAC1P. The sub-population over which the task is conducted is US residents with age between 16 and 90.

ACSTravelTime The goal of the ACSTravelTime task is to predict whether a person's commute time to work is greater than 20 minutes, given by the JWMNP column. The ACS columns used as features are: AGEP, SCHL, MAR, SEX, DIS, ESP, MIG, RELP, RAC1P, ST, CIT, OCCP, JWTR, and POVPIP. The sub-population over which the task is conducted is employed US residents with age greater than 16 years.

C `folktexts` package usage

The `folktexts` package is made available to the public via its open-source code repository[1] and as a standalone package to be installed via the Python Package Index (PyPI). It is compatible with PyTorch models used locally, as well as with web-hosted models available through an API. The main user-facing classes are `Benchmark`, `BenchmarkConfig`, `LLMClassifier`, `TaskMetadata`, `ColumnToText`, and `Dataset`. The responsibilities of each class are ascribed as follows

- The `Benchmark` class is responsible for running a benchmark task, which consists in obtaining risk scores from a given LLM on a given dataset, and evaluating those predictions on a variety of benchmark metrics.
- The `BenchmarkConfig` class details all configurations of a benchmark (see Figure A11 for available options).
- The `LLMClassifier` class is comprised of a transformers model, a tokenizer, and a task; and is responsible for producing risk scores given some tabular rows for the provided task.
- The `TaskMetadata` class is responsible for defining a set of feature columns and target column, together with holding the corresponding column-to-text objects to map an entire tabular row to its natural-text representation. The benchmark ACS tasks instantiate a subclass named `ACSTaskMetadata`.
- The `ColumnToText` class is responsible for producing meaningful natural-text representations of each possible value of a numeric or categorical column.
- The `Dataset` class is responsible for holding tabular data and enabling reproducible manipulation of that data, such as splitting in train/test/validation, or filtering for a specified sub-population. The data used for the benchmark ACS tasks is provided by a subclass named `ACSDataset`.

Additionally, a command-line interface is provided to ease usability: The benchmark ACS tasks can be ran using the `run_acs_benchmark` executable. Figure A11 details each available flag. Further infromation and example notebooks can be found on github at: `https://github.com/socialfoundations/folktexts`.

```
usage:

run_acs_benchmark [-h] --model MODEL --results-dir RESULTS_DIR --data-dir DATA_DIR [--task
      TASK] [--few-shot FEW_SHOT] [--batch-size BATCH_SIZE] [--context-size CONTEXT_SIZE]
      [--fit-threshold FIT_THRESHOLD] [--subsampling SUBSAMPLING] [--seed SEED] [--use-web-
      api-model] [--dont-correct-order-bias] [--numeric-risk-prompting] [--reuse-few-shot-examples]
      [--use-feature-subset USE_FEATURE_SUBSET]
                [--use-population-filter USE_POPULATION_FILTER] [--logger-level {DEBUG,INFO,
      WARNING,ERROR,CRITICAL}]

Benchmark risk scores produced by a language model on ACS data.

options:
  -h, --help            show this help message and exit
  --model MODEL         [str] Model name or path to model saved on disk
  --results-dir RESULTS_DIR
                        [str] Directory under which this experiment's results will be saved
  --data-dir DATA_DIR   [str] Root folder to find datasets on
  --task TASK           [str] Name of the ACS task to run the experiment on
  --few-shot FEW_SHOT   [int] Use few-shot prompting with the given number of shots
  --batch-size BATCH_SIZE
                        [int] The batch size to use for inference
  --context-size CONTEXT_SIZE
                        [int] The maximum context size when prompting the LLM
  --fit-threshold FIT_THRESHOLD
                        [int] Whether to fit the prediction threshold, and on how many samples
  --subsampling SUBSAMPLING
                        [float] Which fraction of the dataset to use (if omitted will use all data)
  --seed SEED           [int] Random seed -- to set for reproducibility
  --use-web-api-model   [bool] Whether use a model hosted on a web API (instead of a local model)
  --dont-correct-order-bias
                        [bool] Whether to avoid correcting ordering bias, by default will correct it
  --numeric-risk-prompting
                        [bool] Whether to prompt for numeric risk-estimates instead of multiple-choice Q&A
  --reuse-few-shot-examples
                        [bool] Whether to reuse the same samples for few-shot prompting (or sample new ones every
      time)
  --use-feature-subset USE_FEATURE_SUBSET
                        [str] Optional subset of features to use for prediction, comma separated
  --use-population-filter USE_POPULATION_FILTER
                        [str] Optional population filter for this benchmark; must follow the format 'column_name=
      value' to filter the dataset by a specific value.
  --logger-level {DEBUG,INFO,WARNING,ERROR,CRITICAL}
                        [str] The logging level to use for the experiment
```

Figure A11: Documentation for using `folktexts` package through the command-line interface. An executable named `run_acs_benchmark` is made available to run the standard ACS benchmark tasks with a variety of available customization options. Detailed documentation available at socialfoundations.github.io/folktexts/

 https://doi.org/10.52202/079017-3089

Column	*Des*		*Example*
AGEP	Age		The individual's age is: `42 years old`.
COW	Clas		The individual's current employment status is: `Working for a non-profit organization`.
SCHL	Edu	ment	The individual's highest grade completed is: `12th grade`.
MAR	Mar		The individual's marital status is: `Married`.
OCCP	Occ		The individual's occupation is: `Human Resources Manager`.
POBP	Plac		The individual's place of birth is: `New Zealand`.
RELP	Rela		The individual's relationship to the reference survey respondent in the household is: `Brother or sister`.
WKHP	Wor	eek	The individual's usual number of hours worked per week is: `40 hours`.
SEX	Sex		The individual's sex is: `Female`.
RAC1P	Rac		The individual's race is: `Black or African American`.
PINCP	Tota	ıe	The individual's total yearly income is: `$75,000`.
CIT	Citi		The individual's citizenship status is: `Naturalized US citizen`.
DIS	Disa		The individual `has a disability`.
ESP	Emp	s of parents	The individual is `living with two parents: both parents in labor force`.
MIG	Mol	re 1 year ago)	The individual `lived in the same house 1 year ago.`
MIL	Mili		The individual `was on active duty in the past, but not currently`.
PUBCOV	Pub	rage	The individual `is covered by public health insurance`.
ANC	Anc		The individual has `single ancestry`.
NATIVITY	Nati		The individual is `foreign born`.
DEAR	Hea		The individual `has hearing difficulty`.
DEYE	Visi		The individual `does not have vision difficulty`.
DREM	Cog		The individual `does not have cognitive difficulties`.
ESR	Emp	s #2	The individual is `not in the labor force`.
ST	Stat		The individual lives in `California`.
FER	Pare	r)	The individual `gave birth to a child within the past 12 months`.
JWMNP	Con		The individual takes `45 minutes travelling to work every day`.
JWTR	Mea	t	The individual's means of transport to work is `a bicycle`.
POVPIP	Inco	/ ratio	The individual's income to poverty ratio is `150%`.

Table A5: Description of all c mappings implemented for ACS features. The variable part of each example is shown in `typeset grey font`. Details on each possible categorical value for each feat ole in the ACS PUMS data dictionary.[5]

97407