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: $\mathbb{P}[Y = 1|R = r, S = \text{'Black'}] \geq \mathbb{P}[Y = 1|R = r, S \neq \text{'Black'}]$, 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

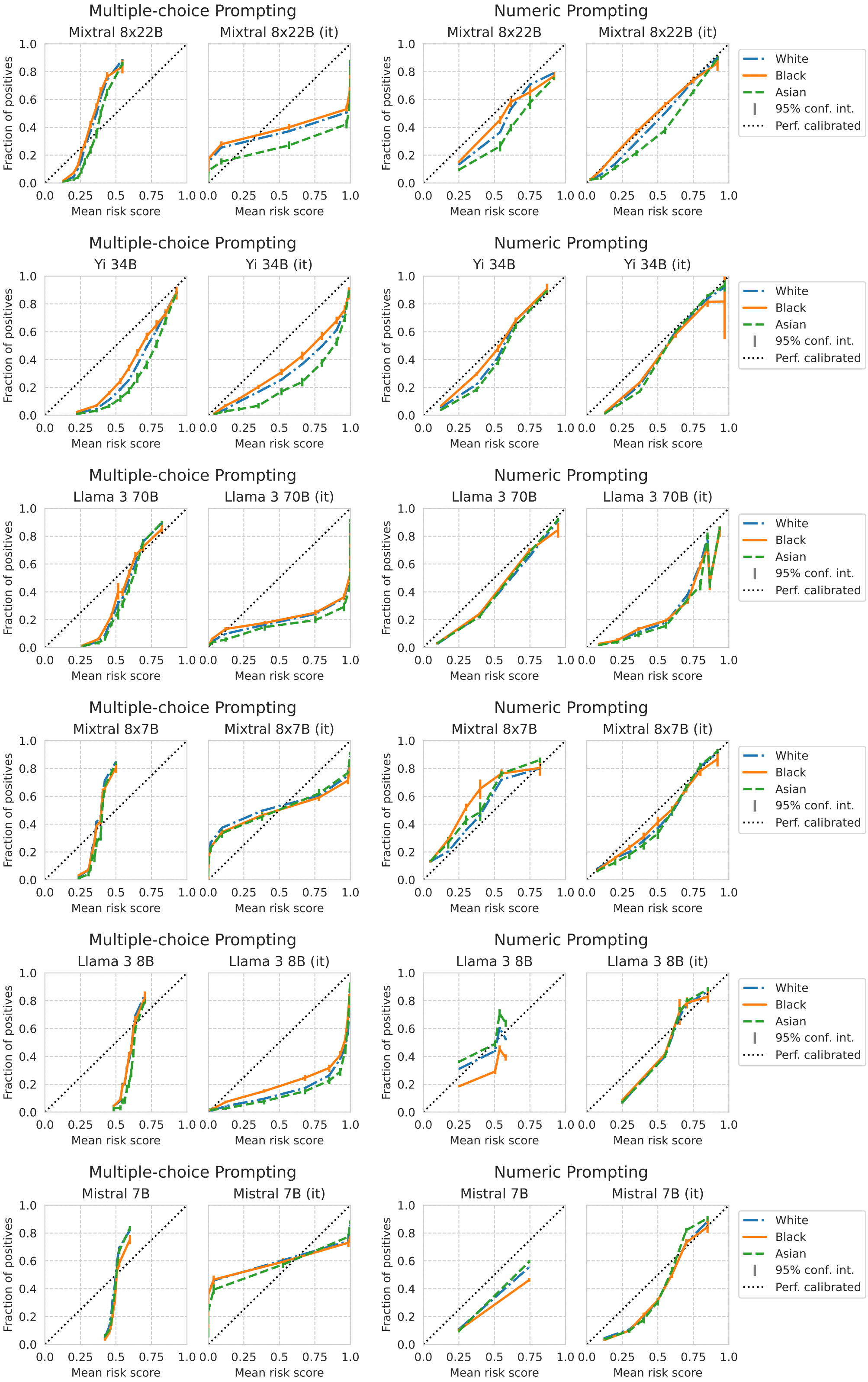
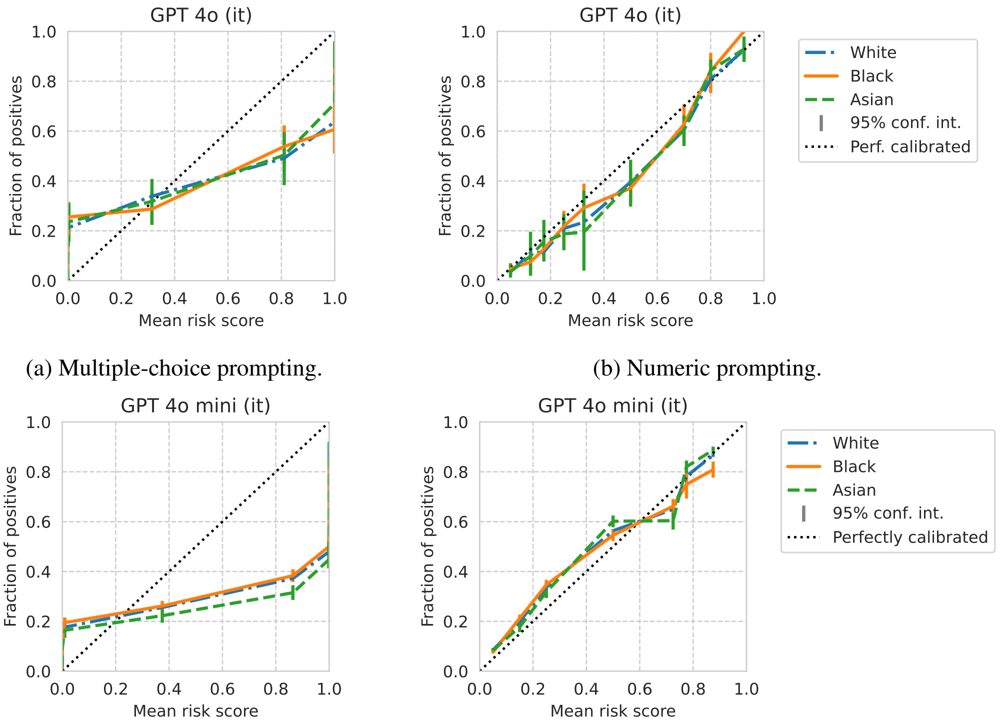


Figure A1: [**ACSIncome**] Calibration curves across different census race sub-populations, computed using 10 quantile-based score bins, with 95% confidence intervals. The ‘Mixtral 8x22B’ and ‘Yi 34B’ models are the worst offenders in terms of group-conditional miscalibration; i.e., they violate the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



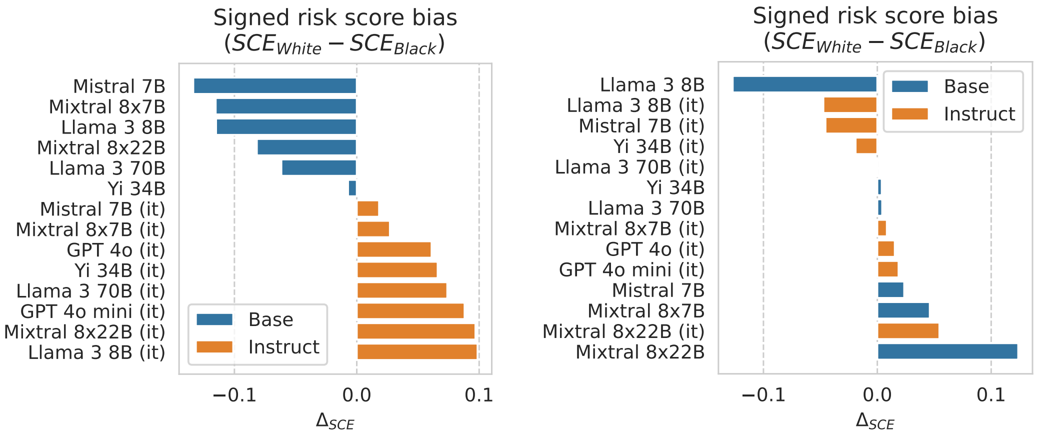
(a) Multiple-choice prompting.

(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.

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.

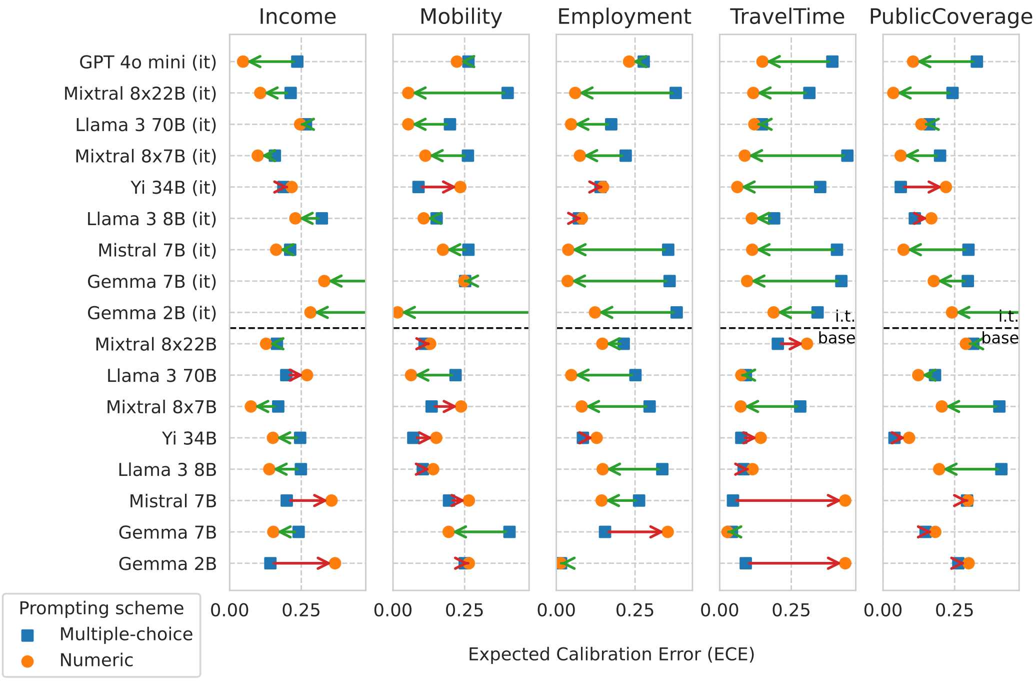


Figure A4: Change in calibration error (ECE) when using numeric risk prompting (●) versus multiple-choice prompting (■). Instruction-tuned models (*top rows*) show substantial calibration improvements, while base models (*bottom rows*) show mixed results. Green/red arrows signal ECE improvement/degradation.

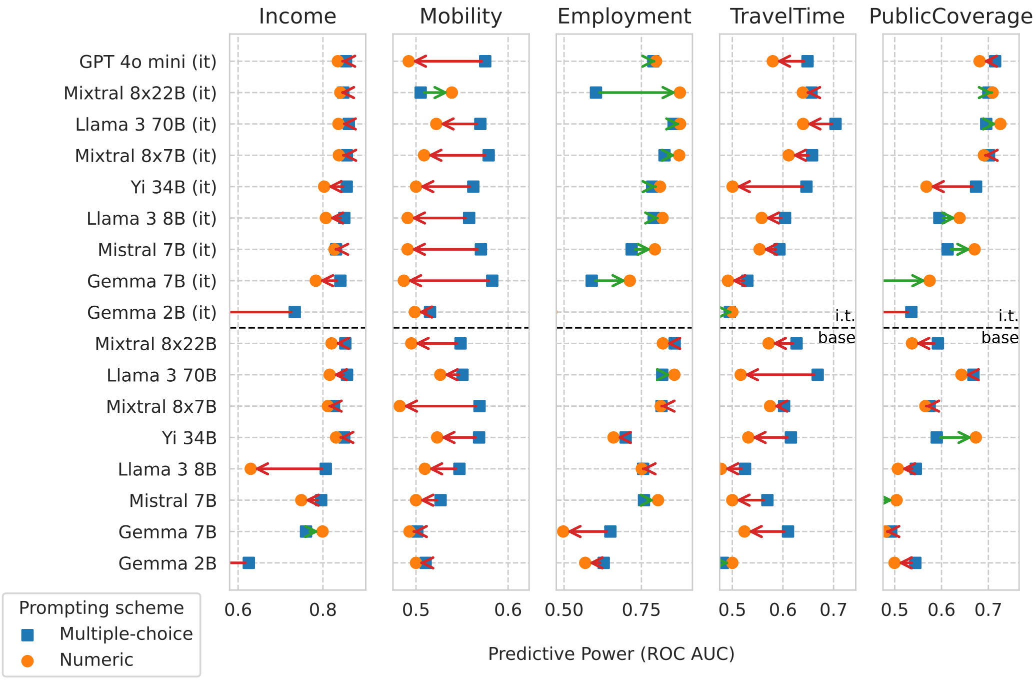


Figure A5: Change in predictive power (AUC) when using numeric risk prompting (●) versus multiple-choice prompting (■). Both instruction-tuned models (*top rows*) and base models (*bottom rows*) generally achieve worse AUC with numeric prompting. Green/red arrows signal AUC improvement/degradation, respectively.

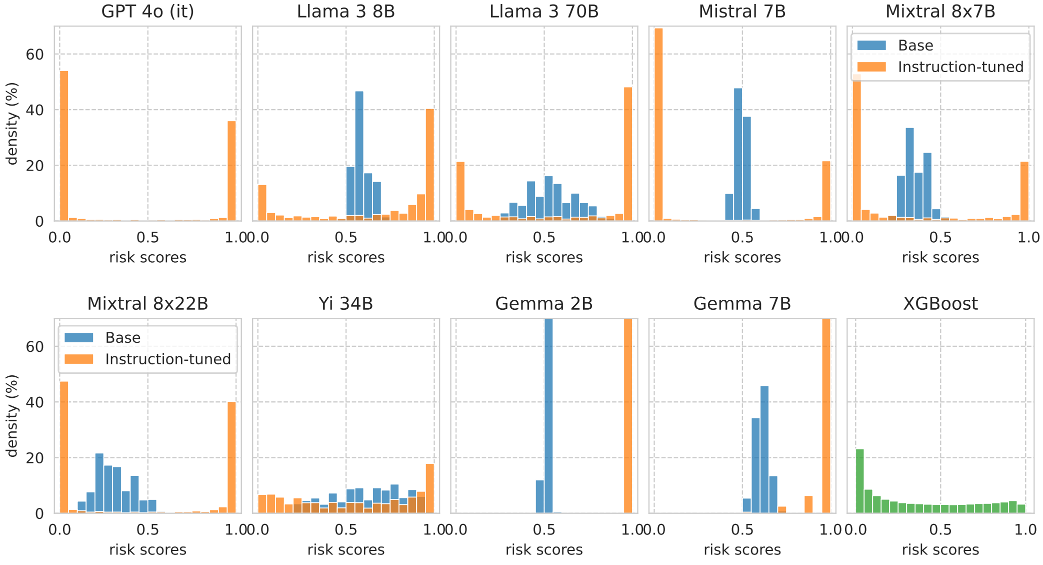


Figure A6: Risk score distribution for base and instruction-tuned model pairs on the ACSIncome task, using *multiple-choice* prompting. After instruction-tuning, models exhibit high confidence, but worse calibration in general.

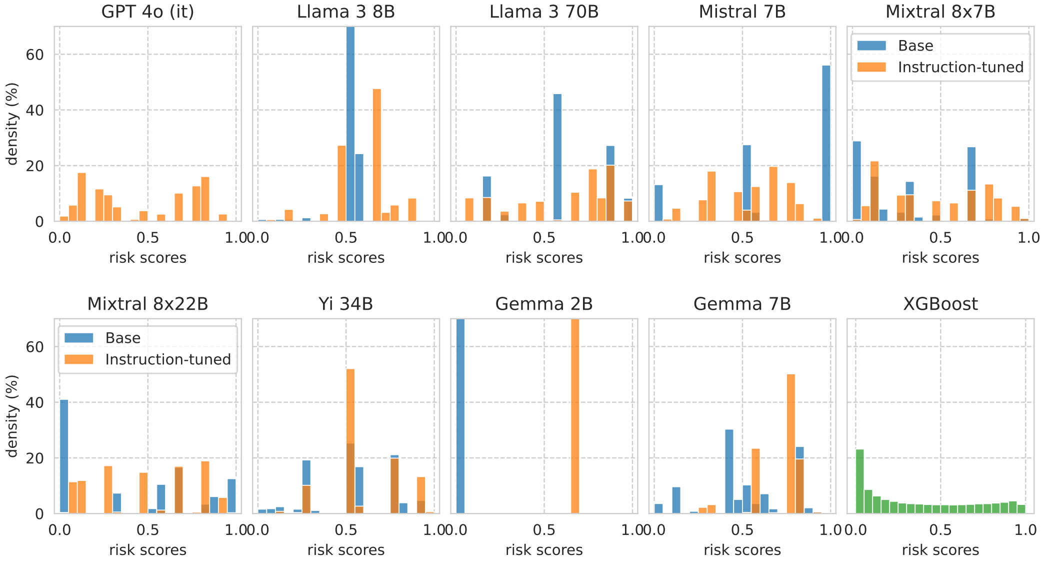


Figure A7: Distribution of risk scores produced using *numeric prompting* on the ACSIncome benchmark task. A baseline score distribution that achieves 0.00 calibration error is shown in green (XGBoost model). For each model, risk scores produced in this manner fall into only a few different possible 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.

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

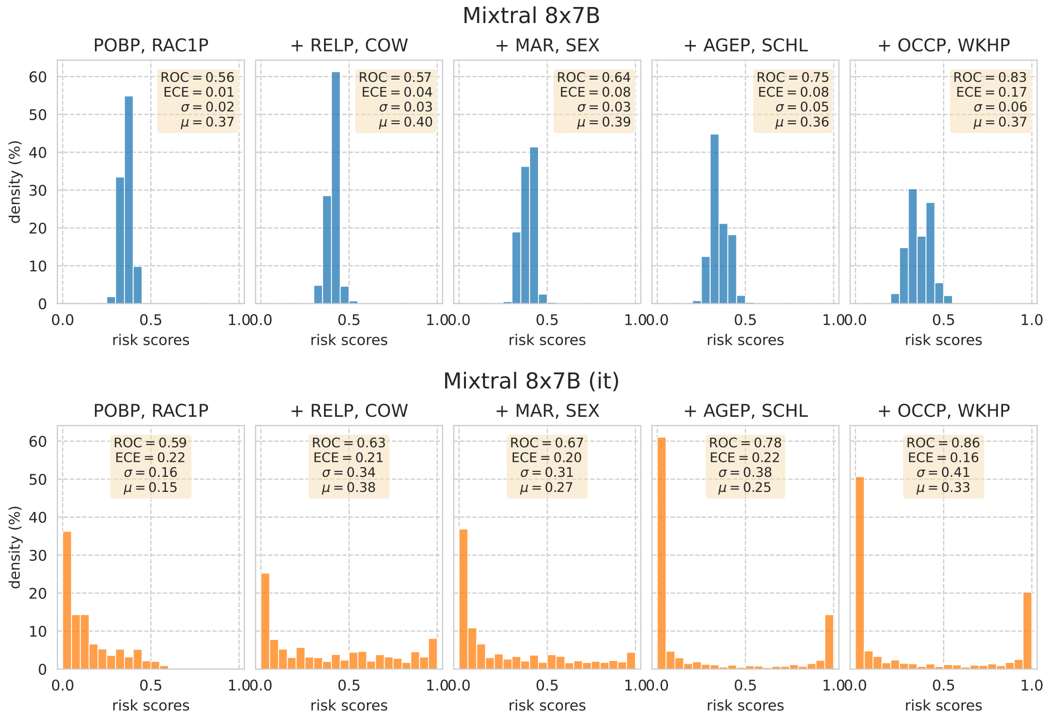


Figure A8: [**ACSIncome**; **Multiple-choice**] Shift of score distribution with increasing evidence for the Mixtral 8x7B model (which achieved the best Brier score), using multiple-choice Q&A on the ACSIncome task. Features are described in Table A5. Score distribution gets more discriminative as more 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`

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.

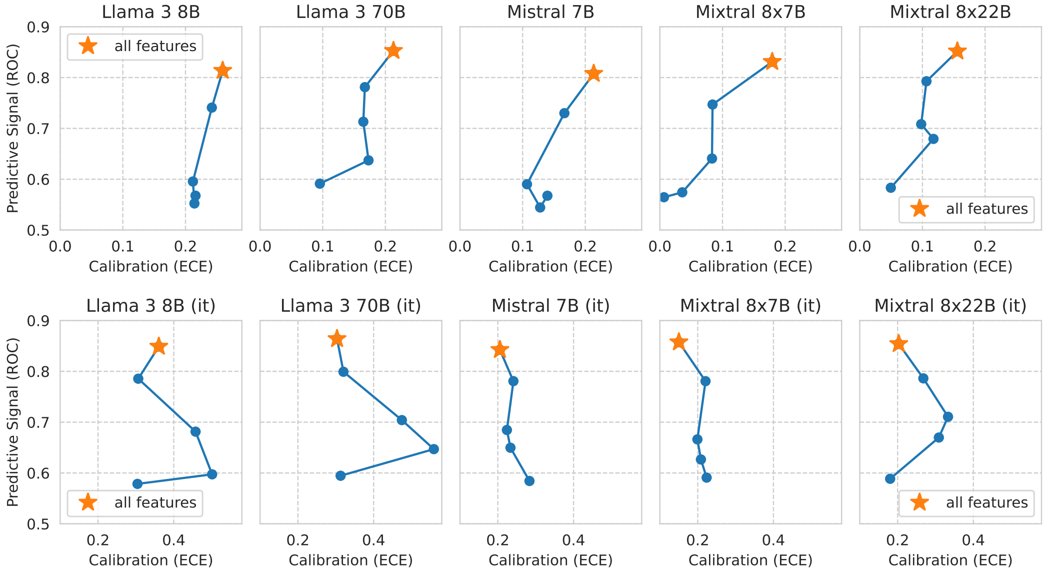


Figure A9: [**ACSIncome**; **Multiple-choice**] Evaluation of calibration (ECE) and predictive performance (AUC) on Llama and Mistral models, with an increasing number of features provided. For each dot along the line we add two features, up to all 10 features being used in the point marked with a star. *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

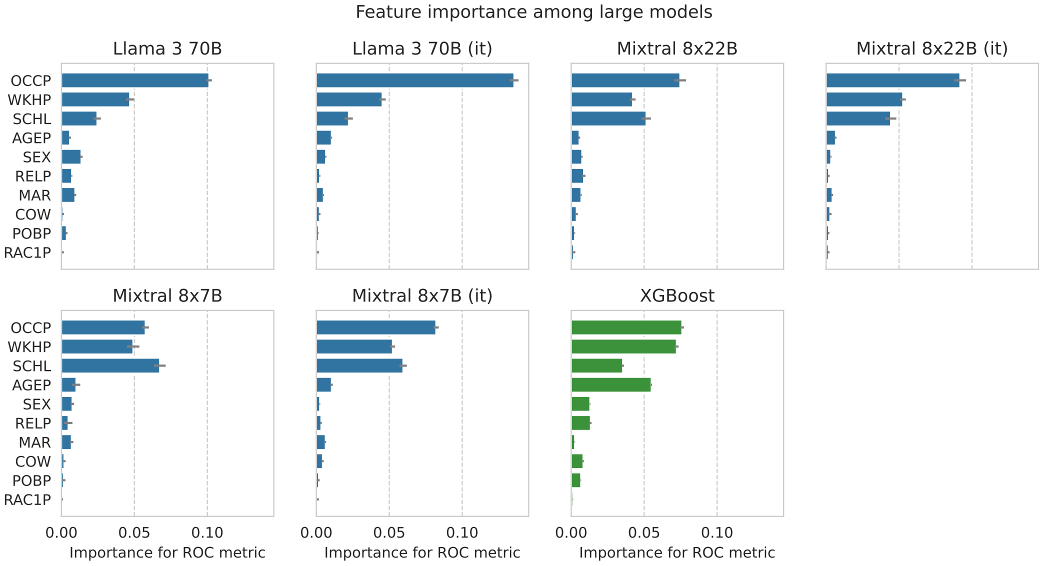


Figure A10: [**ACSIncome**; **Multiple-choice**] Feature importance among the largest language models tested, plus results for the XGBoost baseline. Feature importance values are calculated as the loss in AUC when the values of a given column are randomly permuted [75].

`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, 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/

Column	***Description***	***Example***
AGEP	Age	The individual’s age is: `42 years old`.
COW	Class of worker	The individual’s current employment status is: `Working for a non-profit organization`.
SCHL	Educational attainment	The individual’s highest grade completed is: `12th grade`.
MAR	Marital status	The individual’s marital status is: `Married`.
OCCP	Occupation	The individual’s occupation is: `Human Resources Manager`.
POBP	Place of birth	The individual’s place of birth is: `New Zealand`.
RELP	Relationship	The individual’s relationship to the reference survey respondent in the household is: `Brother or sister`.
WKHP	Work-hours per week	The individual’s usual number of hours worked per week is: `40 hours`.
SEX	Sex	The individual’s sex is: `Female`.
RAC1P	Race	The individual’s race is: `Black or African American`.
PINCP	Total yearly income	The individual’s total yearly income is: `$75,000`.
CIT	Citizenship status	The individual’s citizenship status is: `Naturalized US citizen`.
DIS	Disability status	The individual `has a disability`.
ESP	Employment status of parents	The individual is `living with two parents: both parents in labor force`.
MIG	Mobility (lived here 1 year ago)	The individual `lived in the same house 1 year ago.`
MIL	Military service	The individual `was on active duty in the past, but not currently`.
PUBCOV	Public health coverage	The individual `is covered by public health insurance`.
ANC	Ancestry	The individual has `single ancestry`.
NATIVITY	Nativity	The individual is `foreign born`.
DEAR	Hearing	The individual `has hearing difficulty`.
DEYE	Vision	The individual `does not have vision difficulty`.
DREM	Cognition	The individual `does not have cognitive difficulties`.
ESR	Employment status #2	The individual is `not in the labor force`.
ST	State	The individual lives in `California`.
FER	Parenthood (1 year)	The individual `gave birth to a child within the past 12 months`.
JWMNP	Commute time	The individual takes `45 minutes travelling to work every day`.
JWTR	Means of transport	The individual’s means of transport to work is `a bicycle`.
POVPIP	Income-to-poverty ratio	The individual’s income to poverty ratio is `150%`.

Table A5: Description of all column-to-text 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 feature are available in the ACS PUMS data dictionary.[5]