INFO251 – Applied Machine Learning

Lab 8 – Metrics & Fairness

Satej Soman

based heavily on material by Suraj Nair, Joshua Blumenstock and Simón Ramirez Amaya

Announcements

- PS5 due April 3

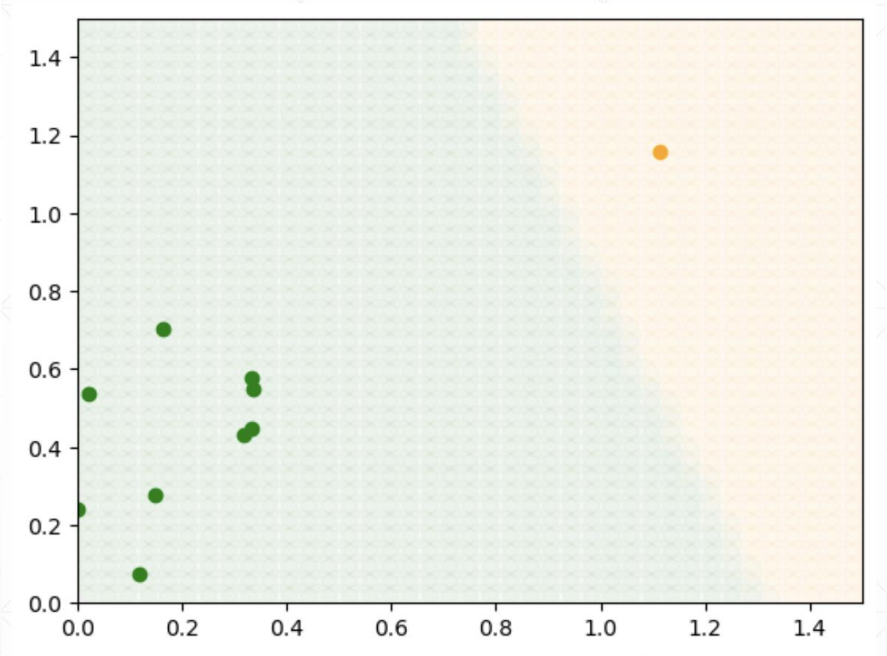
Today’s topics

- Review of prediction metrics
- Fairness:
  - Statistical non-discrimination
  - Fairness, datasets, benchmarks, and `folktables`
  - Useful components in `scikit-learn`

Review: metrics for assessing prediction quality

- Assume that green is the “positive” class here

		Predicted	
		Green	**Orange**
Actual	**Green**	TP	FN
	Orange	FP	TN

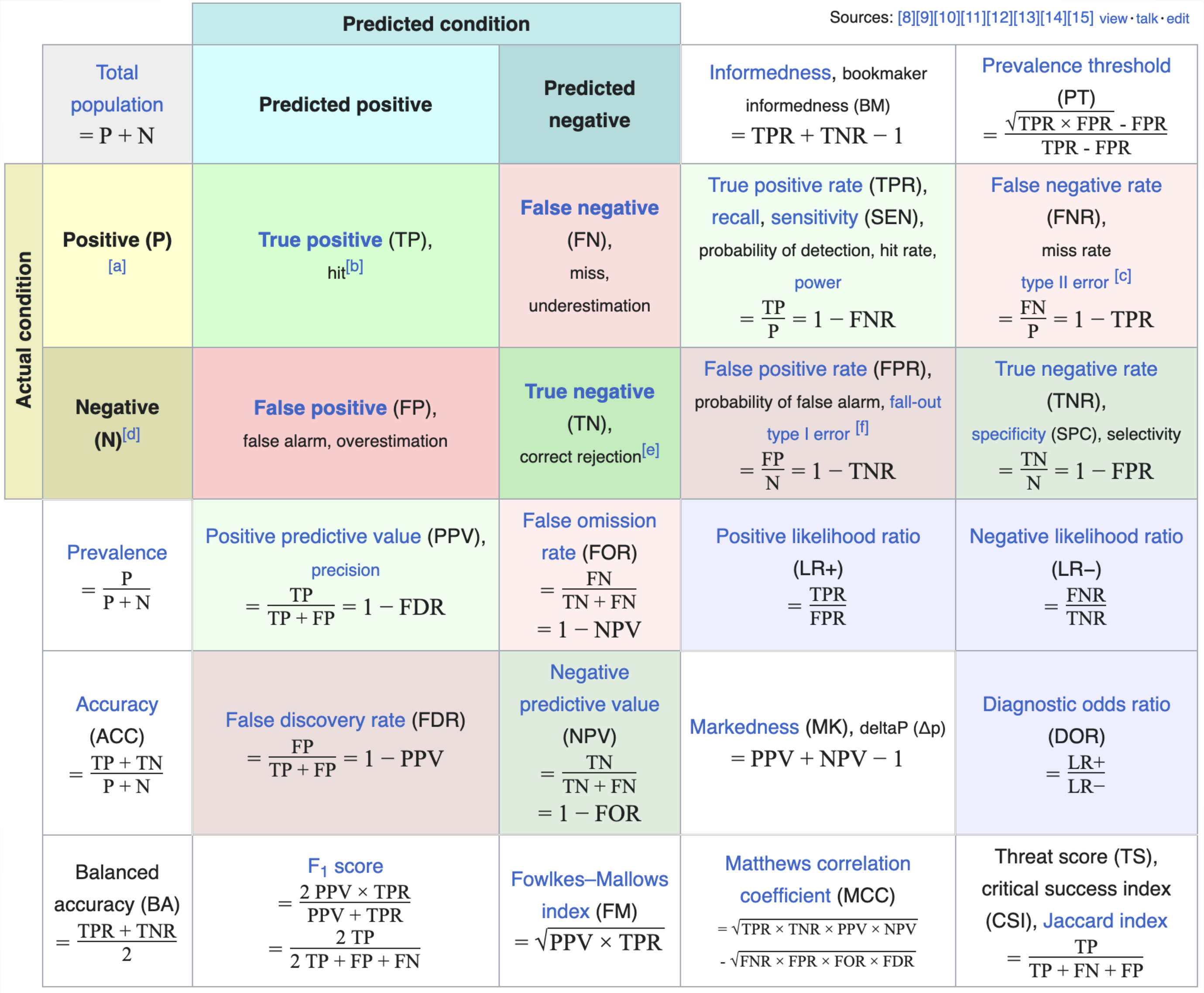


- Accuracy = (TP + TN)/(TP + FP + FN + TN)
- TPR = TP/(TP + FN)
- FPR = FP/(FP + TN)
- Precision = TP/(TP + FP)

Probabilistic interpretation

- Assume that green is the “positive” class here (Green = 1, Orange = 0)
- $\hat{y}$ is prediction, y is the true value (both take on values 0 or 1)

- TPR = TP/(TP + FN) $= P(\hat{y} = 1 \mid y = 1)$
- FPR = FP/(FP + TN) $= P(\hat{y} = 1 \mid y = 0)$
- Precision = TP/(TP + FP) $= P(y = 1 \mid \hat{y} = 1)$



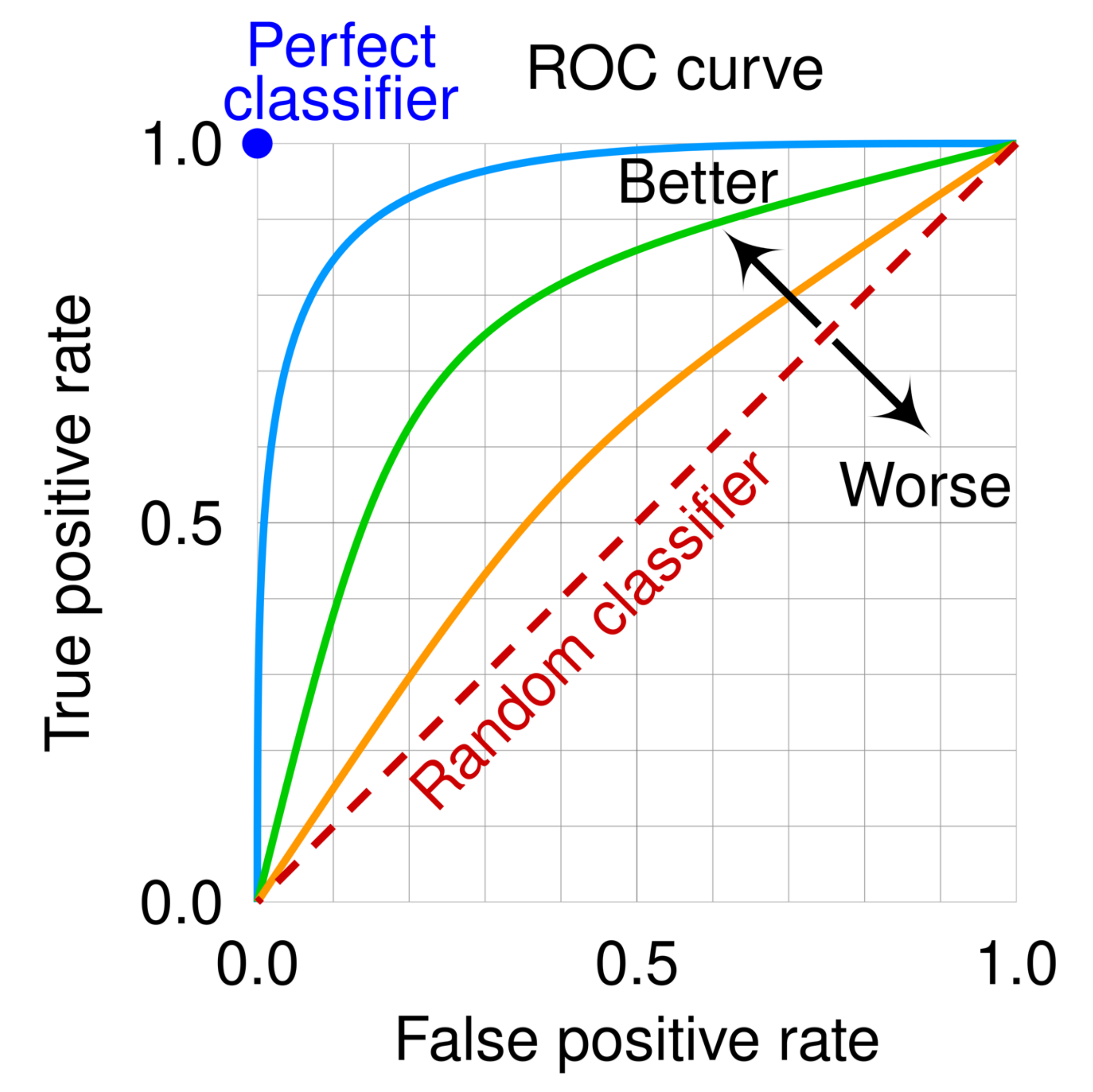
		Predicted condition		Sources: [8][9][10][11][12][13][14][15] view · talk · edit	
	Total population $= \mathrm{P} + \mathrm{N}$	**Predicted positive**	**Predicted negative**	Informedness, bookmaker informedness (BM) $= \mathrm{TPR} + \mathrm{TNR} - 1$	Prevalence threshold (PT) $= \frac{\sqrt{\mathrm{TPR} \times \mathrm{FPR}} - \mathrm{FPR}}{\mathrm{TPR} - \mathrm{FPR}}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{\mathrm{TP}}{\mathrm{P}} = 1 - \mathrm{FNR}$	False negative rate (FNR), miss rate type II error [c] $= \frac{\mathrm{FN}}{\mathrm{P}} = 1 - \mathrm{TPR}$
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{\mathrm{FP}}{\mathrm{N}} = 1 - \mathrm{TNR}$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{\mathrm{TN}}{\mathrm{N}} = 1 - \mathrm{FPR}$
	Prevalence $= \frac{\mathrm{P}}{\mathrm{P} + \mathrm{N}}$	Positive predictive value (PPV), precision $= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR}$	False omission rate (FOR) $= \frac{\mathrm{FN}}{\mathrm{TN} + \mathrm{FN}} = 1 - \mathrm{NPV}$	Positive likelihood ratio (LR+) $= \frac{\mathrm{TPR}}{\mathrm{FPR}}$	Negative likelihood ratio (LR–) $= \frac{\mathrm{FNR}}{\mathrm{TNR}}$
	Accuracy (ACC) $= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}}$	False discovery rate (FDR) $= \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{PPV}$	Negative predictive value (NPV) $= \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} = 1 - \mathrm{FOR}$	Markedness (MK), deltaP (Δp) $= \mathrm{PPV} + \mathrm{NPV} - 1$	Diagnostic odds ratio (DOR) $= \frac{\mathrm{LR+}}{\mathrm{LR-}}$
	Balanced accuracy (BA) $= \frac{\mathrm{TPR} + \mathrm{TNR}}{2}$	F_1 score $= \frac{2\,\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$	Fowlkes–Mallows index (FM) $= \sqrt{\mathrm{PPV} \times \mathrm{TPR}}$	Matthews correlation coefficient (MCC) $= \sqrt{\mathrm{TPR} \times \mathrm{TNR} \times \mathrm{PPV} \times \mathrm{NPV}} - \sqrt{\mathrm{FNR} \times \mathrm{FPR} \times \mathrm{FOR} \times \mathrm{FDR}}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$

More general metrics derived from *confusion matrix*

Source: https://en.wikipedia.org/wiki/Confusion_matrix

Receiver Operator Characteristic Curve (ROC Curve)

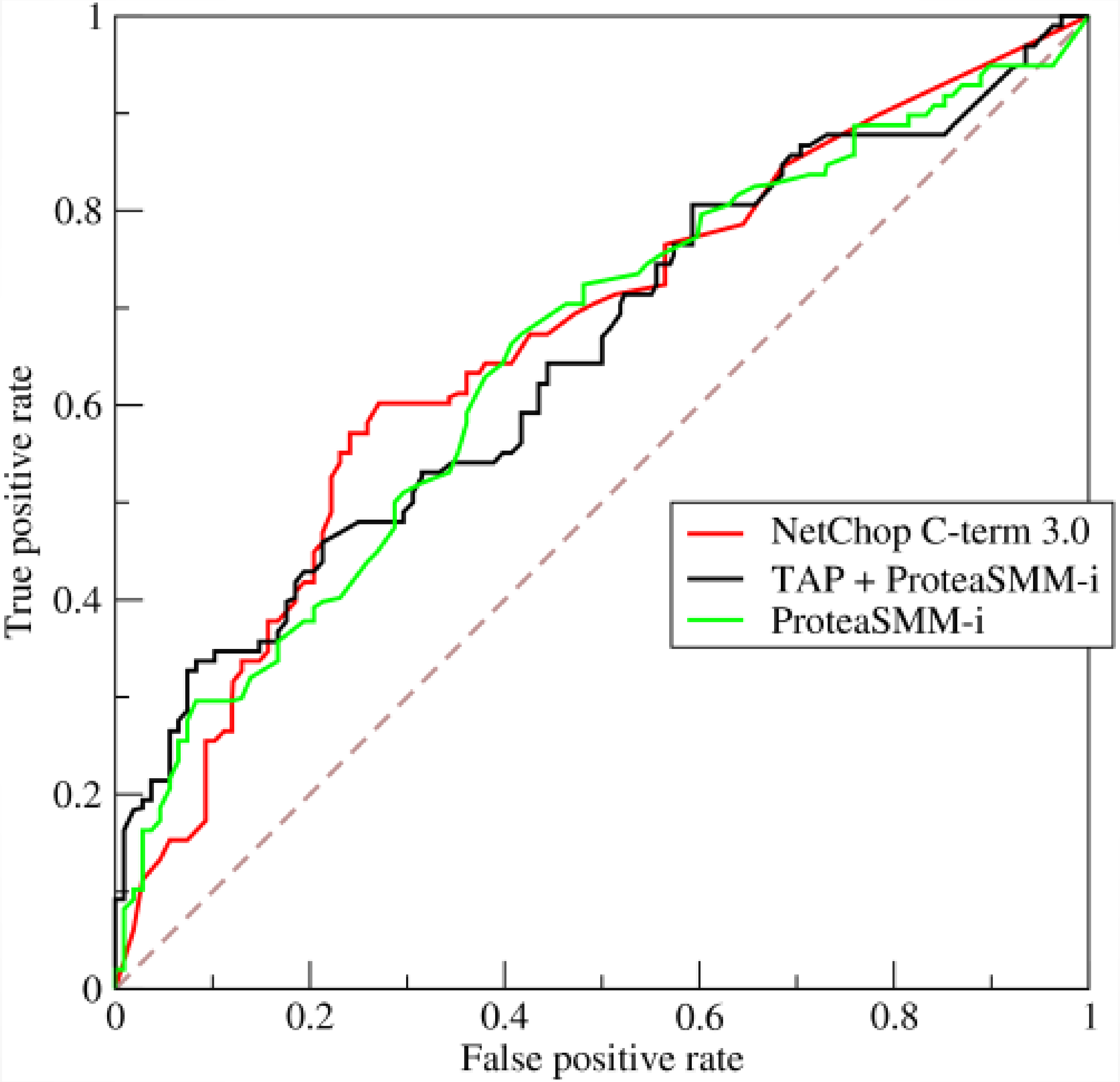
- In classification problems, we usually get a predictive score that we threshold:
  - e.g. sigmoid predictor outputs a “score” in the range [0, 1]
  - a classifier takes this score, and compares it to a threshold
  - see: `model.predict_proba` in scikit-learn classifiers
- ROC Curve
  - Comes from WW2 – radar-based sensors scored by how well they detected enemy tanks/guns/ships
  - For every threshold, calculate the TPR and FPR and plot it out – see tradeoff between TPR and FPR
  - Other option for quota problems: Set “acceptance rate” to the rate of positive observations in the training set



Hypothetical ROC Curves

Source:

https://en.wikipedia.org/wiki/Receiver_operating_characteristic



Example ROC Curves

Source:

https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Statistical definitions of fairness

- When we calculate TPR, FPR, etc – do we always calculate it for the entire dataset?
- The metrics we discussed (especially in their probabilistic formulation) are properties of the joint distribution of y and $\hat{y}$
- What if some of the aspects of each data point also mattered? Consider a discrete variable A (race, gender, location, etc) – it could be a feature, or it could be strongly correlated
- Now, the criteria are properties of the joint distribution of y, $\hat{y}$, and A
- **Error parity**: probability of error is the same for members and non-members
  - e.g. TPR parity: $P(\hat{y} = 1 \mid y = 1, A = 1) = P(\hat{y} = 1 \mid y = 1, A = 0)$

Exercise on your own:

- Does this classifier satisfy error parity for the TPR? For the FPR?

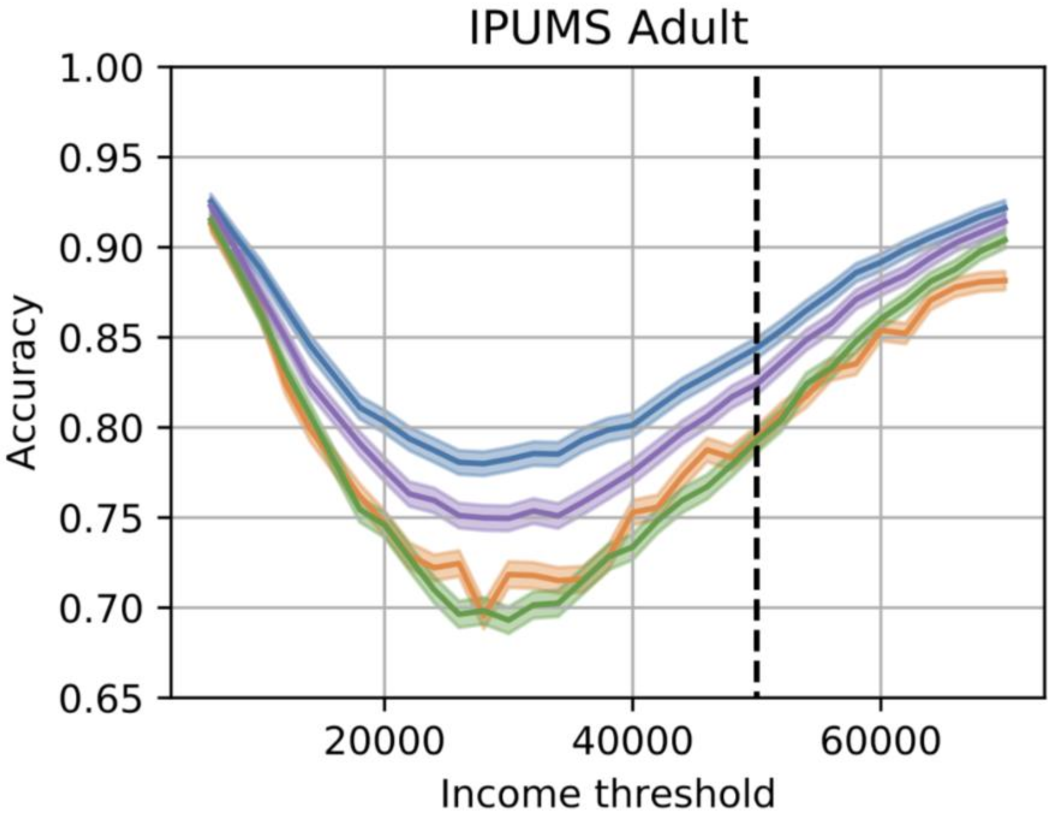
$A = 0$	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	6/32	6/32
$Y = 1$	1/32	3/32

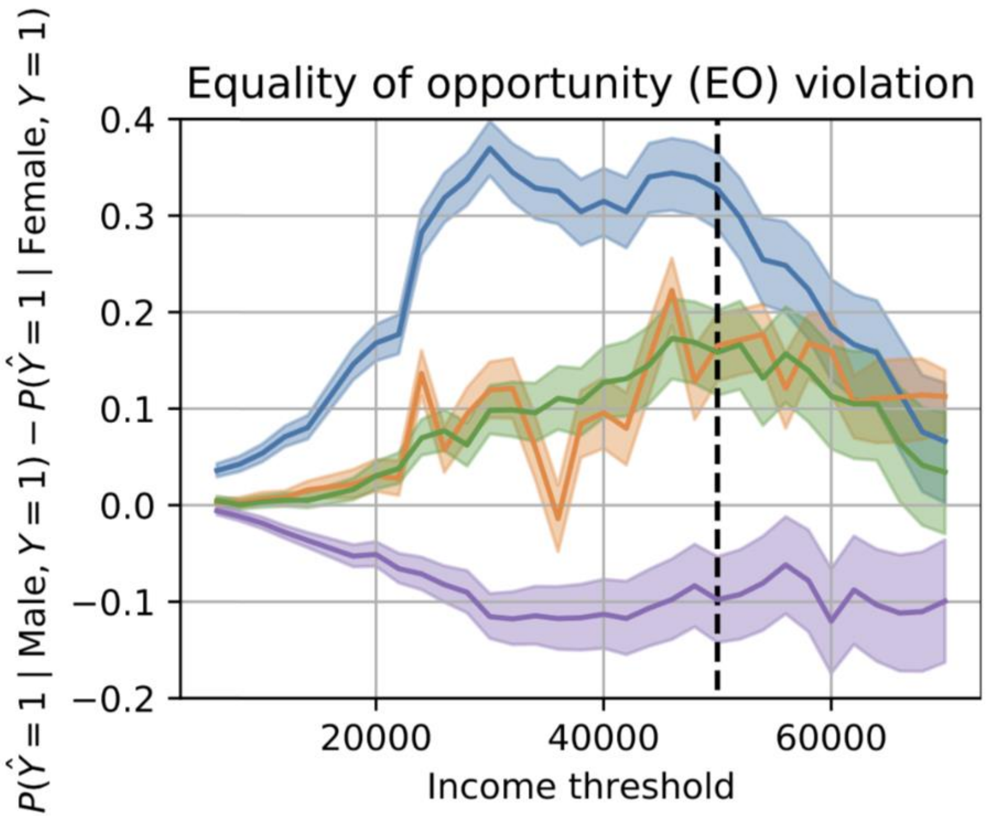
$A = 1$	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	4/32	4/32
$Y = 1$	2/32	6/32

Datasets and benchmarks

- Datasets are pretty key to ML practice!
- Performance on widely-used datasets (ImageNet, MNIST digits) becomes a benchmark to compare different methods' efficacy
- In ML-based fairness, the "UCI Adult" dataset was extremely popular – extract of 1994 US census data with a target binary variable based on whether income > $50k
- Hundreds if not thousands of ML fairness studies done using this dataset
- But… $50k was an arbitrary threshold! What if we used something else?

Overfitting to UCI Adult





folktables

- `folktables` is a Python package that offers a convenient and well documented interface to the ACS information!
- Using the resources in `folktables` you can generate data samples on the fly and get a better sense of whether a model is overfitting to a particular realization of the census data.
- For instance: you can train a model in CA data and used it to predict in AL. Or use data from two consecutive years to test how quickly performance degrades.
- Needless to say, you can include different covariates, set thresholds, etc...

https://github.com/socialfoundations/folktables

i’m not directly concerned with fairness – is this relevant to general ML practice?

- yes.

- Understanding differential performance by subgroup can be
    - A. key to boosting overall performance
    - B. key to operationalizing predictions

Useful components in sklearn

- Tree-based methods
  - sklearn.tree.DecisionTreeClassifier
  - sklearn.ensemble.RandomForestClassifier
- Pre-processing
  - sklearn.compose.ColumnTransformer
  - sklearn.preprocessing.OneHotEncoder
  - sklearn.preprocessing.StandardScaler
- Validation
  - sklearn.pipeline.Pipeline
  - sklearn.model selection.GridSearchCV
  - sklearn.model selection.cross validate