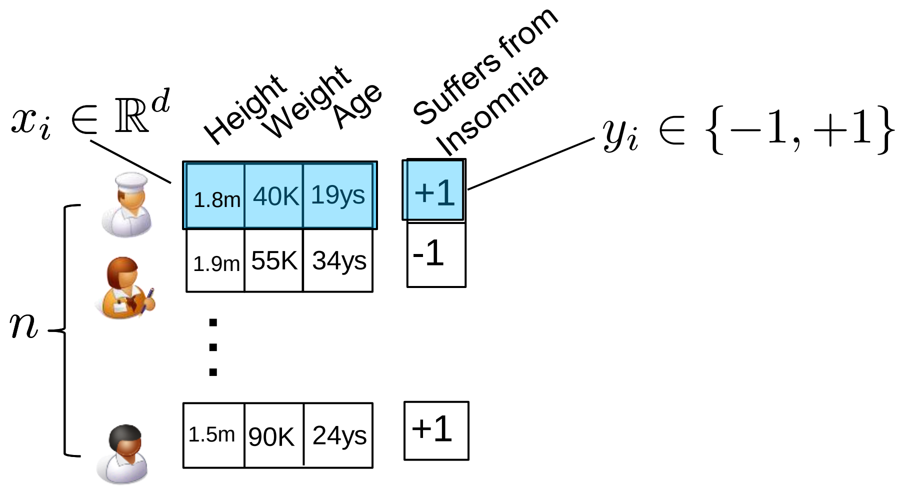
EECE5698
Parallel Processing for Data Analytics

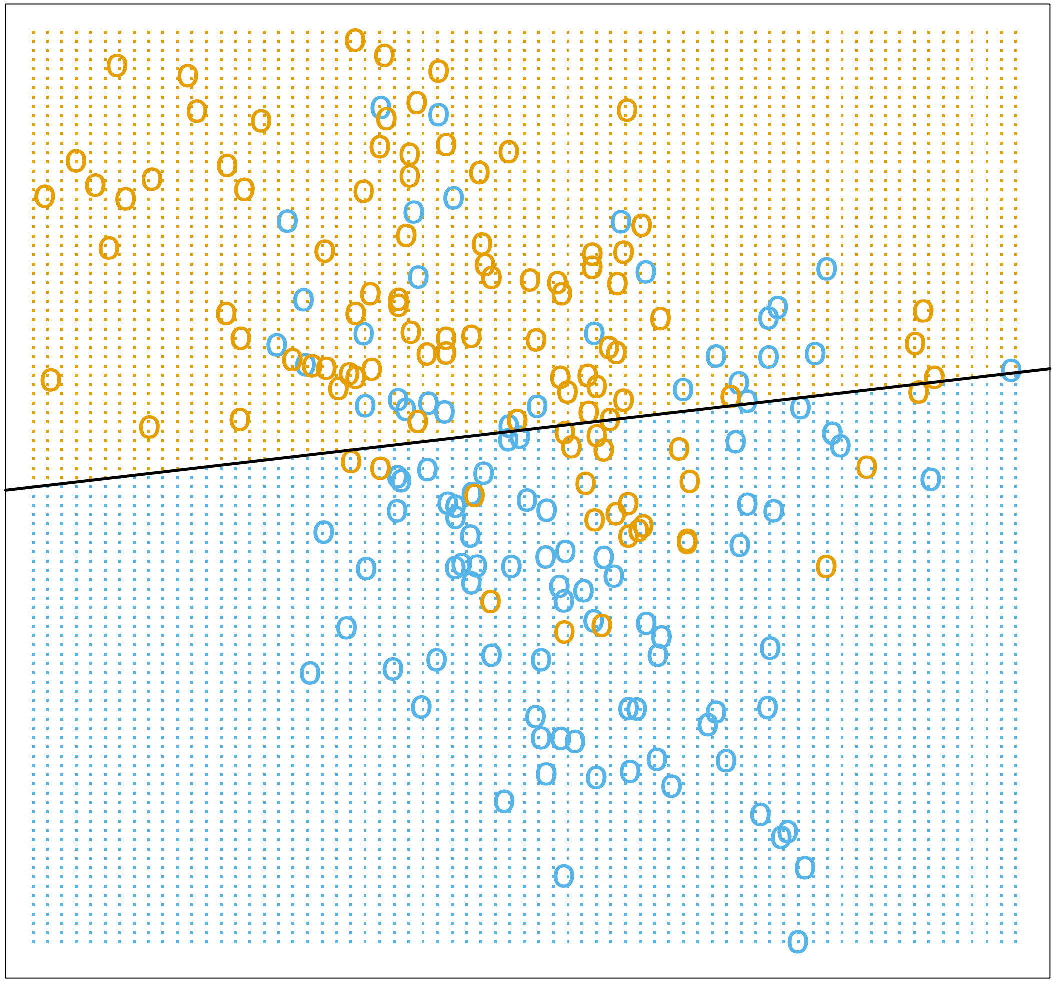
Lecture 12: Classification

Regression vs. Classification



- Standard regression: $y_i \in \mathbb{R}$
- Classification: y_i are **discrete/categorical**, e.g.:
  - $y_i \in \{-1, +1\}$ (binary)
  - $y_i \in \{\texttt{red}, \texttt{blue}, \texttt{green}\}$

Linear Classifiers

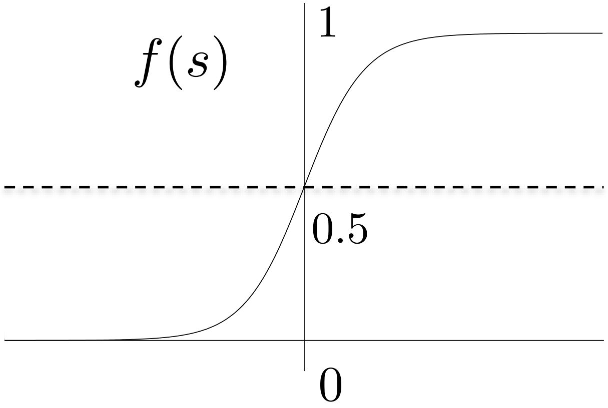


There exists a

$$\beta \in \mathbb{R}^d$$

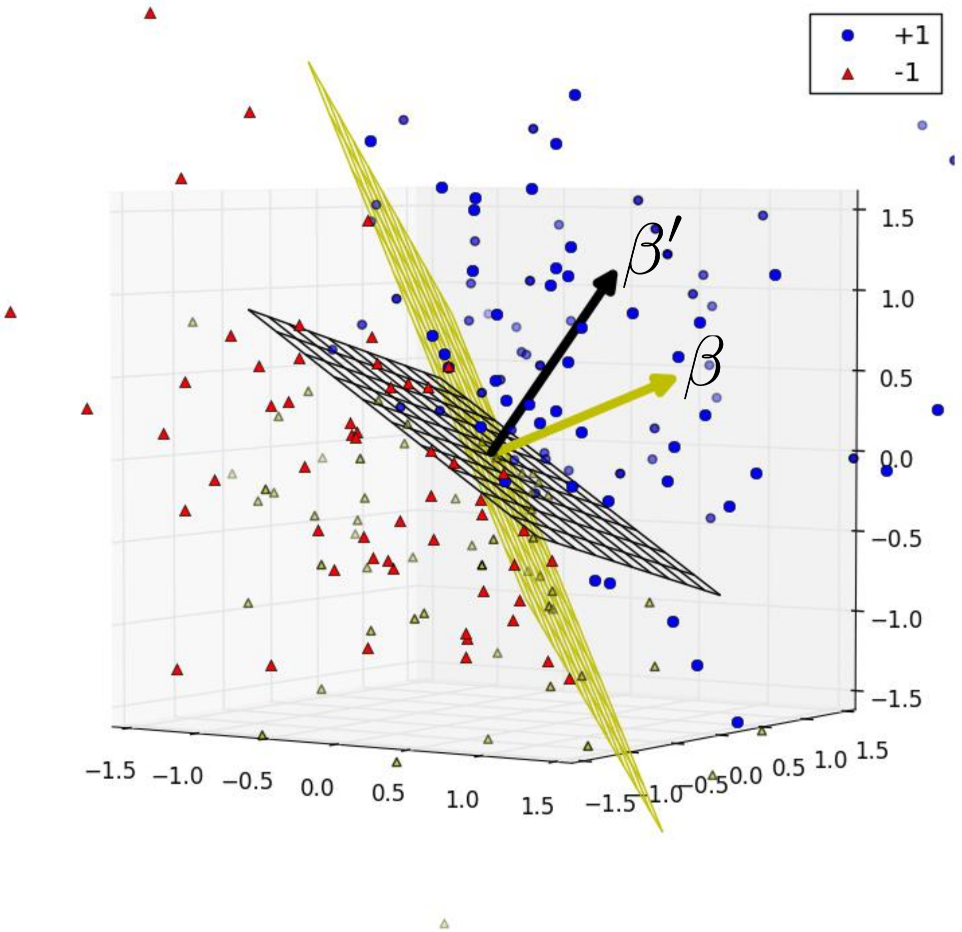
s.t.

$$P(y_i = +1) = f(\beta^\top x_i)$$



Northeastern

Linear Classifiers

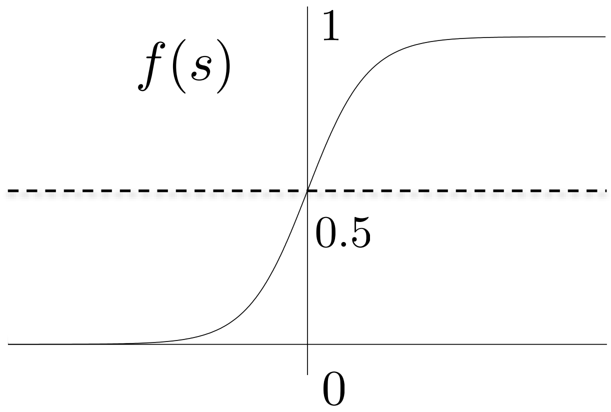


There exists a

$$\beta \in \mathbb{R}^d$$

s.t.

$$P(y_i = +1) = f(\beta^\top x_i)$$



Northeastern

Lifting

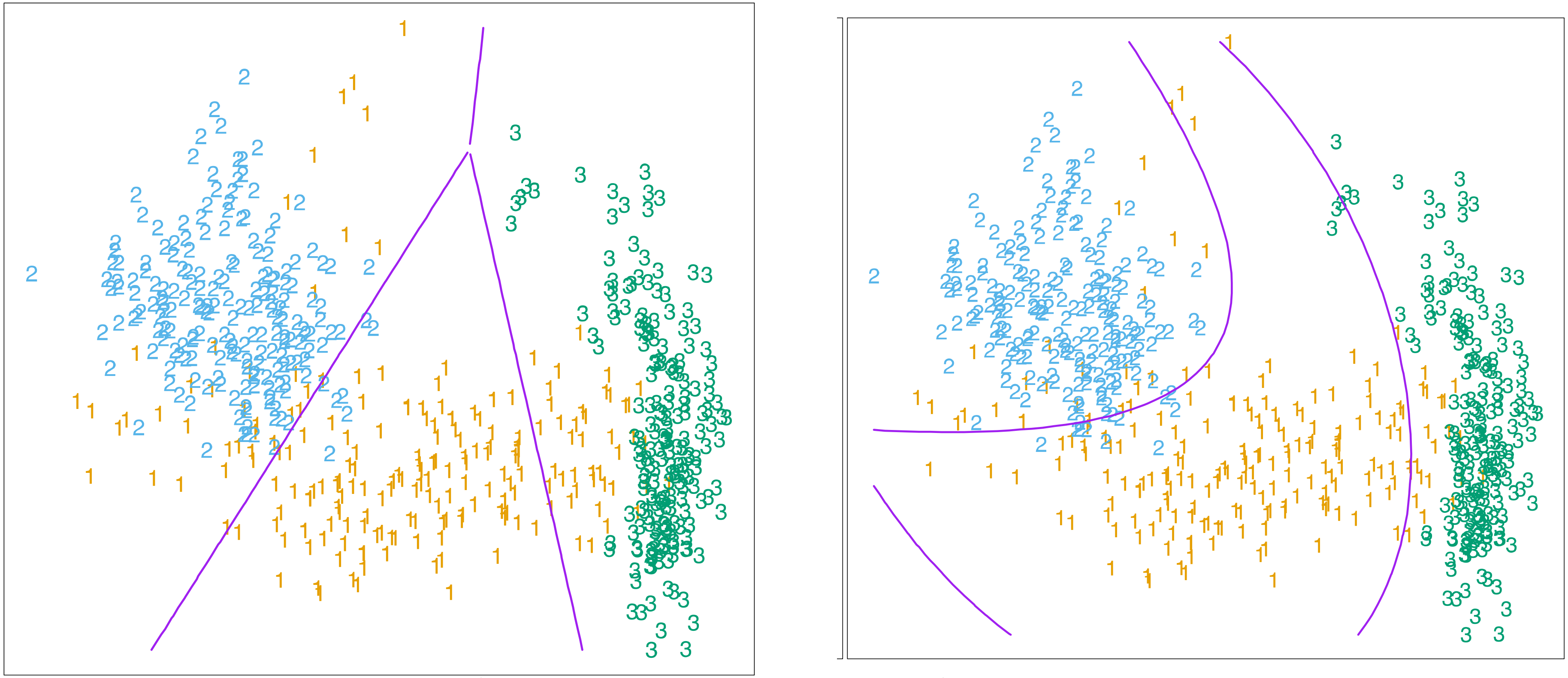
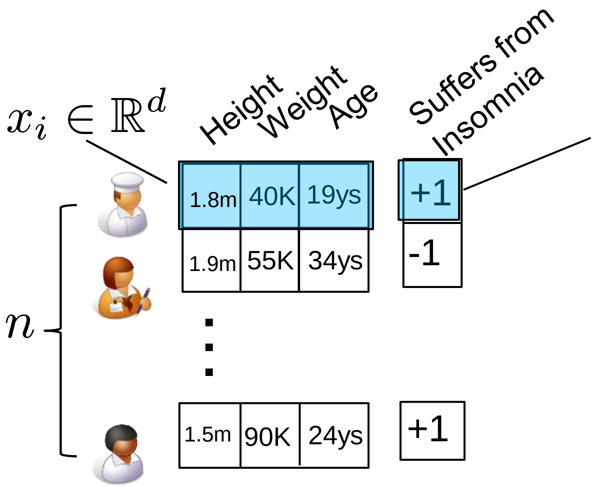


FIGURE 4.1. *The left plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis. The right plot shows quadratic decision boundaries. These were obtained by finding linear boundaries in the five-dimensional space $X_1, X_2, X_1X_2, X_1^2, X_2^2$. Linear inequalities in this space are quadratic inequalities in the original space.*

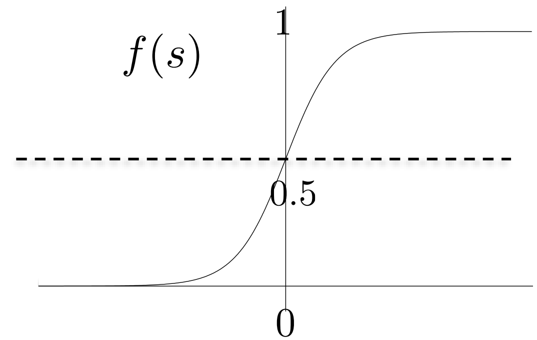
Northeastern

Examples of Linear Models

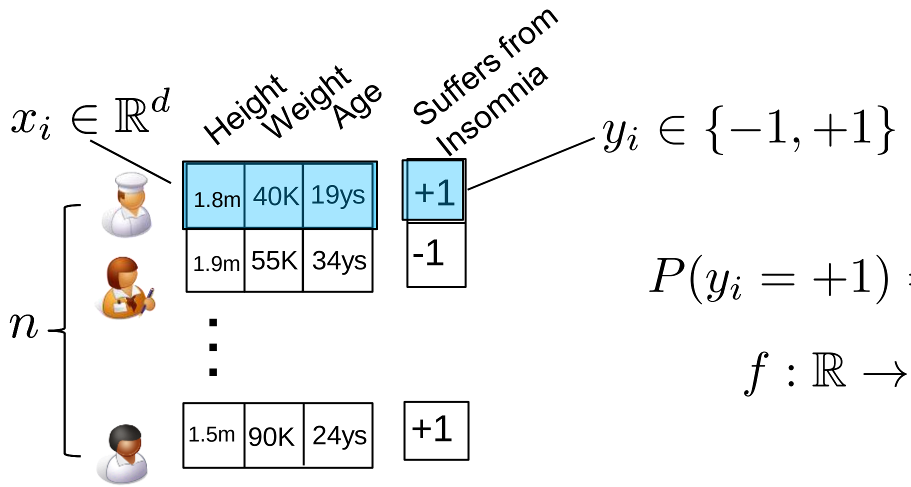


$$P(y_i = +1) = f(\beta^\top x_i)$$

$$f : \mathbb{R} \to [0, 1]$$

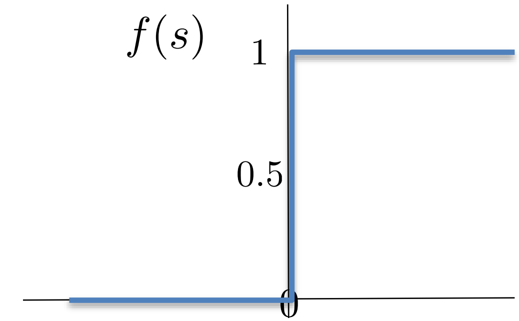


Examples of Linear Models: Step Function



$$P(y_i = +1) = f(\beta^\top x_i)$$

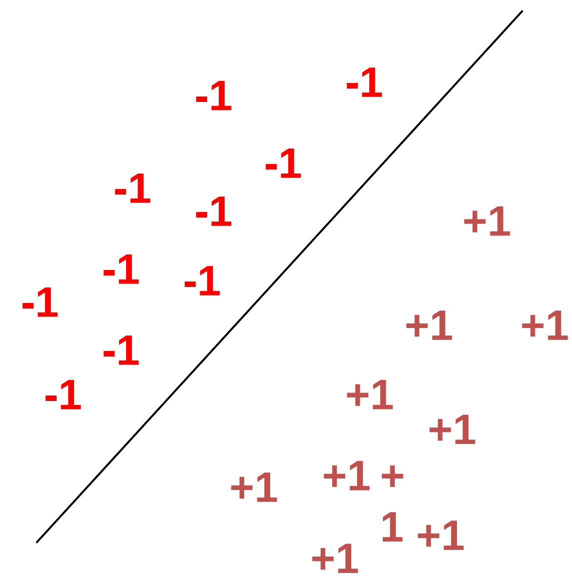
$$f : \mathbb{R} \to [0,1]$$



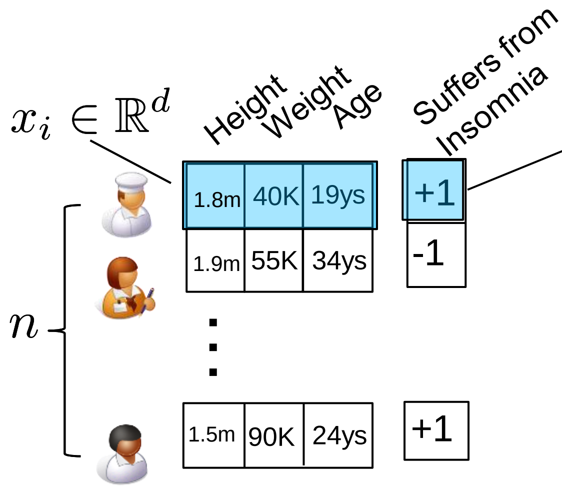
- ❑ Separating hyperplane

- ❑ Deterministic
- ❑ May not exist
- ❑ May not be unique

- ❑ SVMs, perceptron

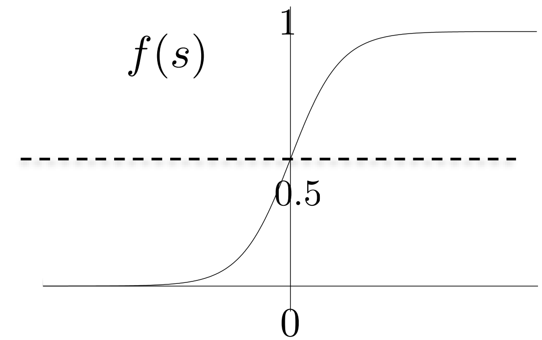


Examples of Linear Models: Logistic Regression



$y_i \in \{-1, +1\}$

$$P(y_i = +1) = f(\beta^\top x_i)$$



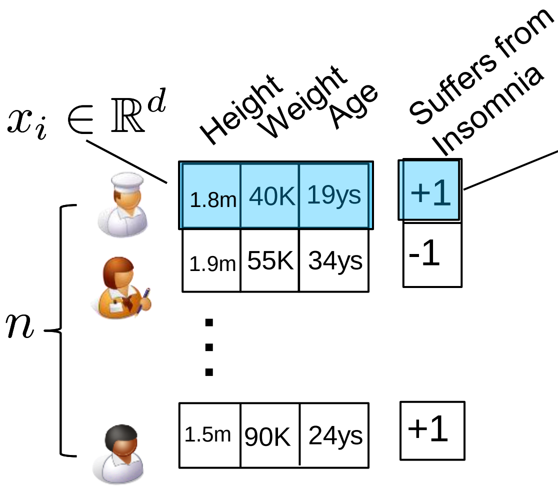
$$f(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$$

- ❑ MLE is a convex optimization problem!
- ❑ Negative log likelihood is:

$$L(\beta) = -\log P(y) = \sum_{i=1}^{n} \log(1 + e^{-y_i \beta^\top x_i})$$

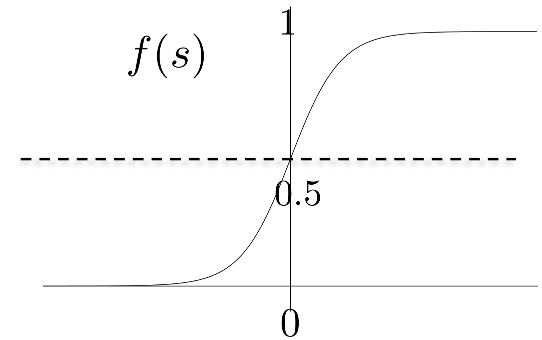
Northeastern

Regularized Logistic Regression



$$P(y_i = +1) = f(\beta^\top x_i)$$

$$f : \mathbb{R} \to [0, 1]$$



$$f(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}}$$

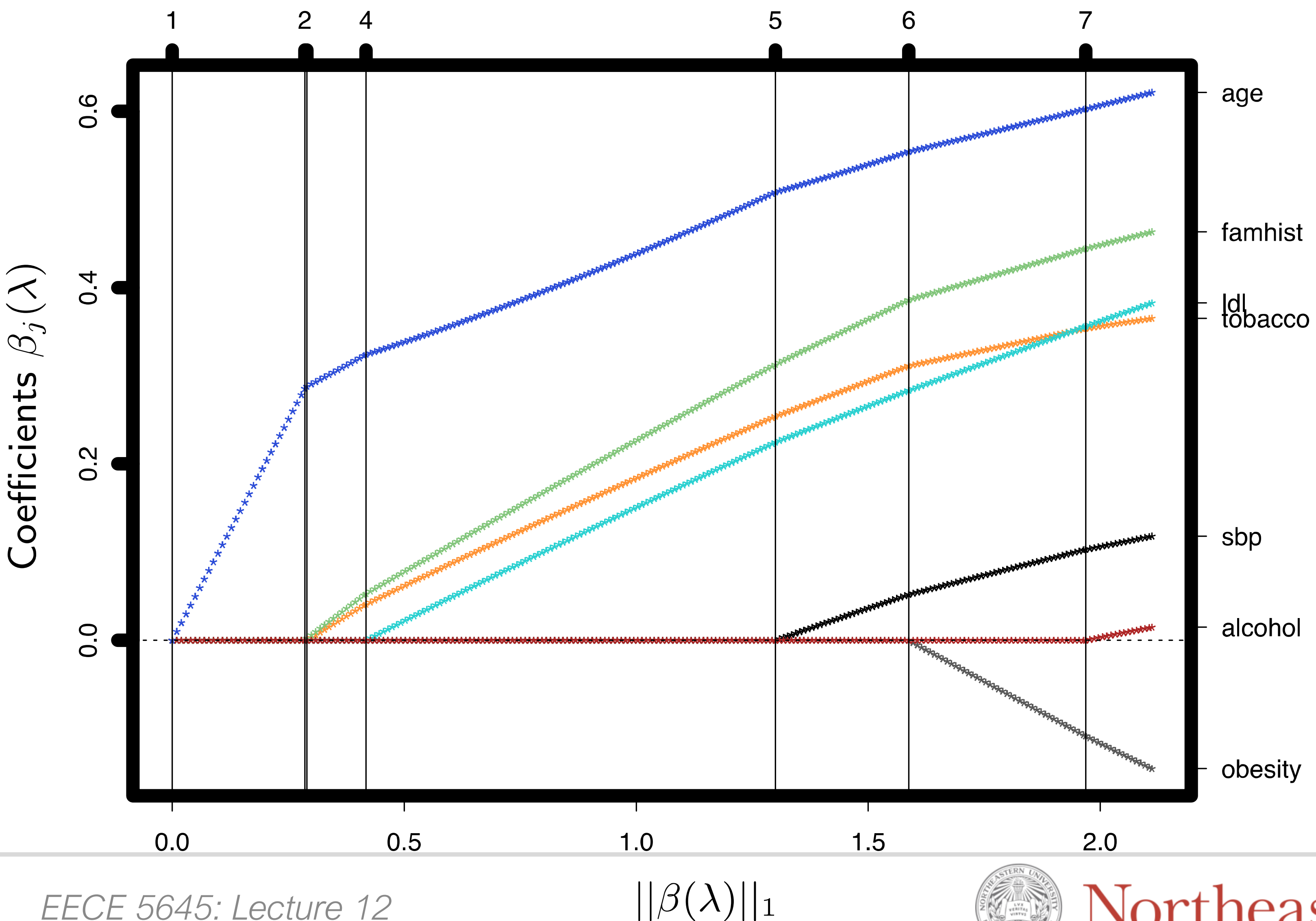
Lasso or ridge penalty

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \log(1 + e^{-y_i \beta^\top x_i}) + \lambda \|\beta\|$$

Selected through CV

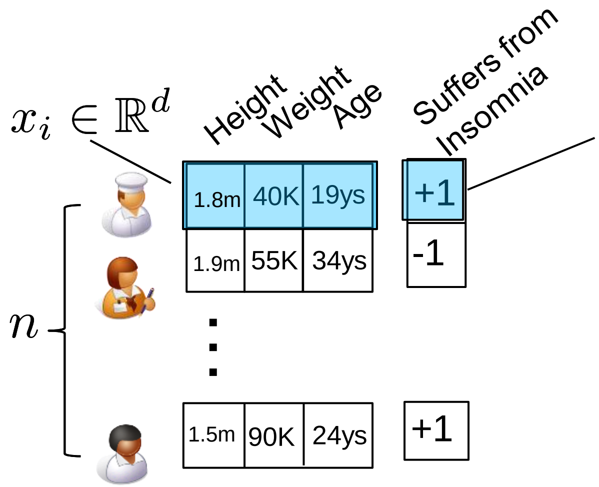
Northeastern

Lasso Logistic Regression



Northeastern

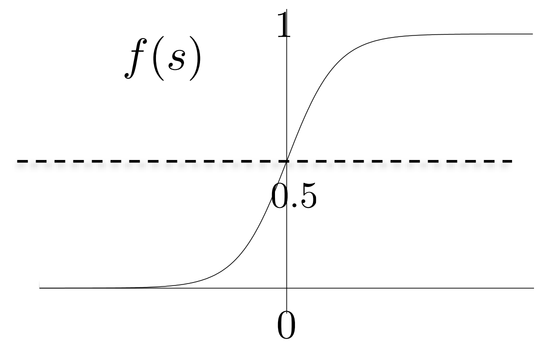
Prediction Using a Linear Classifier



$y_i \in \{-1, +1\}$

$$P(y_i = +1) = f(\beta^\top x_i)$$

$$f : \mathbb{R} \to [0, 1]$$

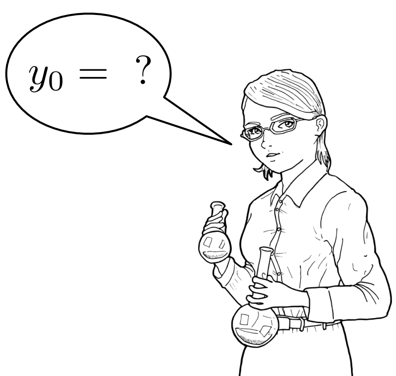


Predict most likely outcome:

If $f(\hat{\beta}^\top x_0) > 0.5$ predict +1, else predict -1

$x_0 =$

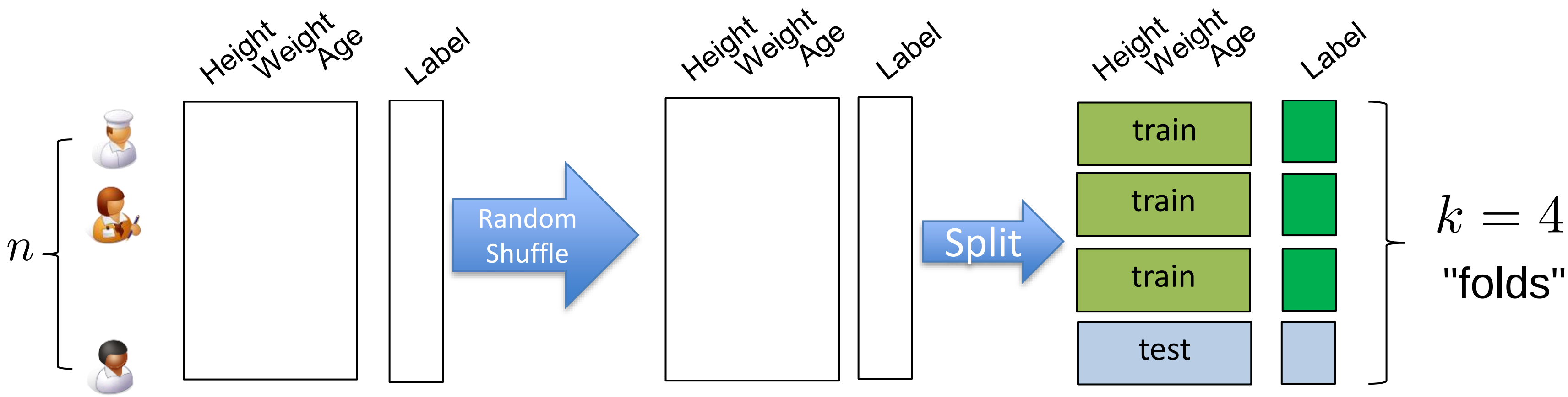
♀	90K	24ys



In other words:

$$\hat{y}_0 = \text{sign}(\beta^\top x_0)$$

Prediction Quality

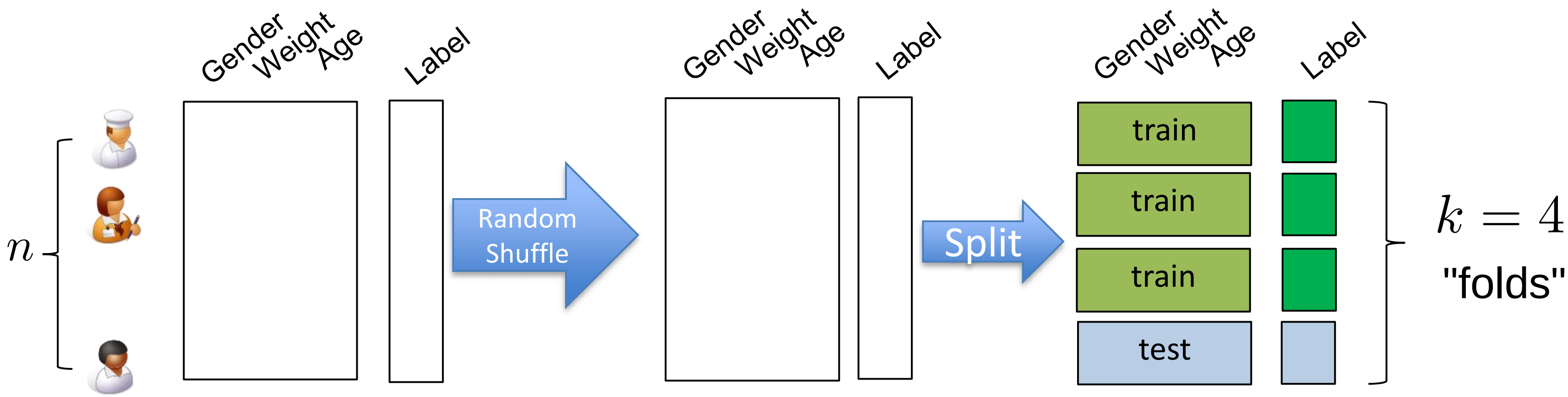


- How to test quality of trained model $\hat{\beta}$?

- Loss: $$L_{\text{test}}(\hat{\beta}) = \sum_{i=1}^{n} \ell(\hat{\beta}; y_i, x_i)$$

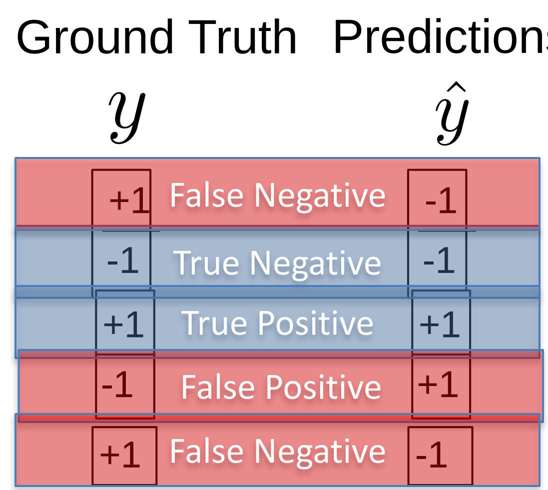
non-intuitive

Accuracy

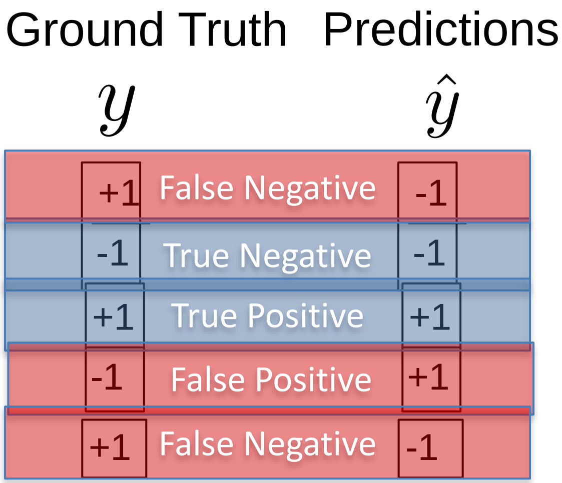


- How to test quality of trained model $\hat{\beta}$?

- Accuracy: $\mathtt{ACC} = \dfrac{TP + TN}{P + N}$



Accuracy



- Accuracy: $\mathtt{ACC} = \dfrac{TP + TN}{P + N}$

Q: Is a classifier with 0.99999 accuracy good?

Depends on how imbalanced dataset is (e.g., -1 is 0.00000001%)

Depends on relative value/cost of TP,TN,FP,FN

Additional Metrics

- Accuracy: $\texttt{ACC} = \dfrac{TP + TN}{P + N}$

- True Positive Rate (Sensitivity):

$$\texttt{TPR} = \frac{TP}{P}$$ (close to 1 is good)

- False Positive Rate (Fallout):

$$\texttt{FPR} = \frac{FP}{N}$$ (close to 0 is good)

- True Negative Rate (Specificity): $\texttt{TNR} = \dfrac{TN}{N} = 1 - \texttt{FPR}$

- False Negative Rate (Miss Rate): $\texttt{FNR} = \dfrac{FN}{P} = 1 - \texttt{TPR}$

Ground Truth y		Predictions $\hat{y}$
+1	False Negative	-1
-1	True Negative	-1
+1	True Positive	+1
-1	False Positive	+1
+1	False Negative	-1

Tradeoff Between TPR and FPR

- True Positive Rate (Sensitivity):

$$\texttt{TPR} = \frac{TP}{P} \quad \text{(close to 1 is good)}$$

- False Positive Rate (Fallout):

$$\texttt{FPR} = \frac{FP}{N} \quad \text{(close to 0 is good)}$$

Ground Truth y		Predictions $\hat{y}$
+1	False Negative	-1
-1	True Negative	-1
+1	True Positive	+1
-1	False Positive	+1
+1	False Negative	-1

Predict **all** $i \in \texttt{test}$ to be **positive**: TPR =1 (good) FPR= 1 (bad)

Predict **all** $i \in \texttt{test}$ to be **negative**: TPR =0 (bad) FPR= 0 (good)

True irrespective of label imbalance in test set

Tradeoff Between TPR and FPR

- True Positive Rate (Sensitivity):

$$\texttt{TPR} = \frac{TP}{P} \quad \text{(close to 1 is good)}$$

- False Positive Rate (Fallout):

$$\texttt{FPR} = \frac{FP}{N} \quad \text{(close to 0 is good)}$$

Ground Truth y		Predictions $\hat{y}$
+1	False Negative	-1
-1	True Negative	-1
+1	True Positive	+1
-1	False Positive	+1
+1	False Negative	-1

Predict $i \in \texttt{test}$ to be **positive** if $s_i \equiv f(\beta^\top x_i) > 0.5$

Tradeoff Between TPR and FPR

- True Positive Rate (Sensitivity):

$$\mathtt{TPR} = \frac{TP}{P}$$ (close to 1 is good)

- False Positive Rate (Fallout):

$$\mathtt{FPR} = \frac{FP}{N}$$ (close to 0 is good)

Ground Truth y		Predictions $\hat{y}$
+1	False Negative	-1
-1	True Negative	-1
+1	True Positive	+1
-1	False Positive	+1
+1	False Negative	-1

Predict $i \in \mathtt{test}$ to be **positive** if $s_i \equiv f(\beta^\top x_i) > \tau$

- $\tau = 0.5$: standard classifier
- $\tau = 0.0$: label all positive (good TPR, bad FPR)
- $\tau = 1.0$: label all negative (bad TPR, good FPR)

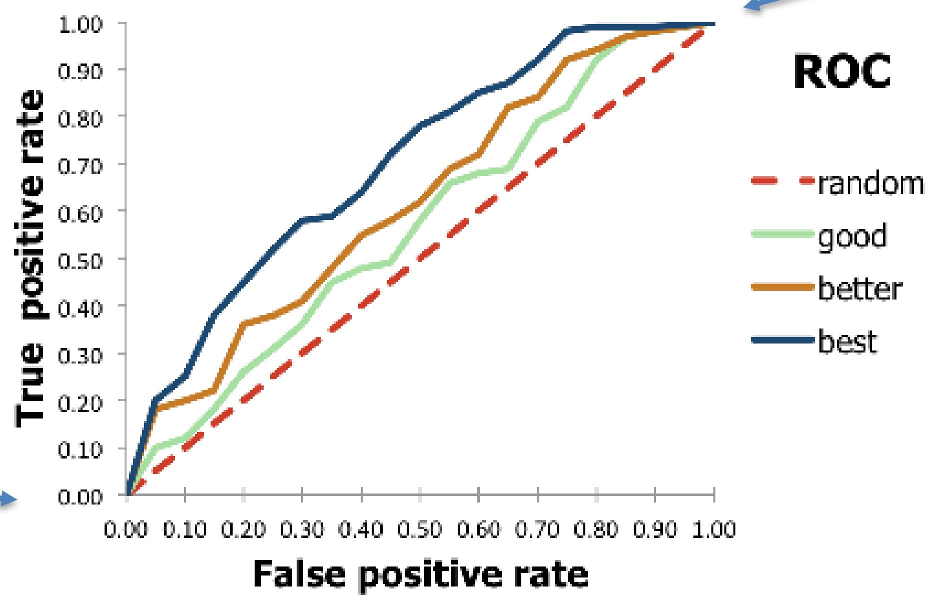
Receiver Operating Characteristic (ROC) Curve

- True Positive Rate (Sensitivity): $\mathtt{TPR} = \frac{TP}{P}$ (close to 1 is good)
- False Positive Rate (Fallout): $\mathtt{FPR} = \frac{FP}{N}$ (close to 0 is good)

Predict $i \in \mathtt{test}$ to be **positive** if $s_i \equiv f(\beta^\top x_i) > \tau$

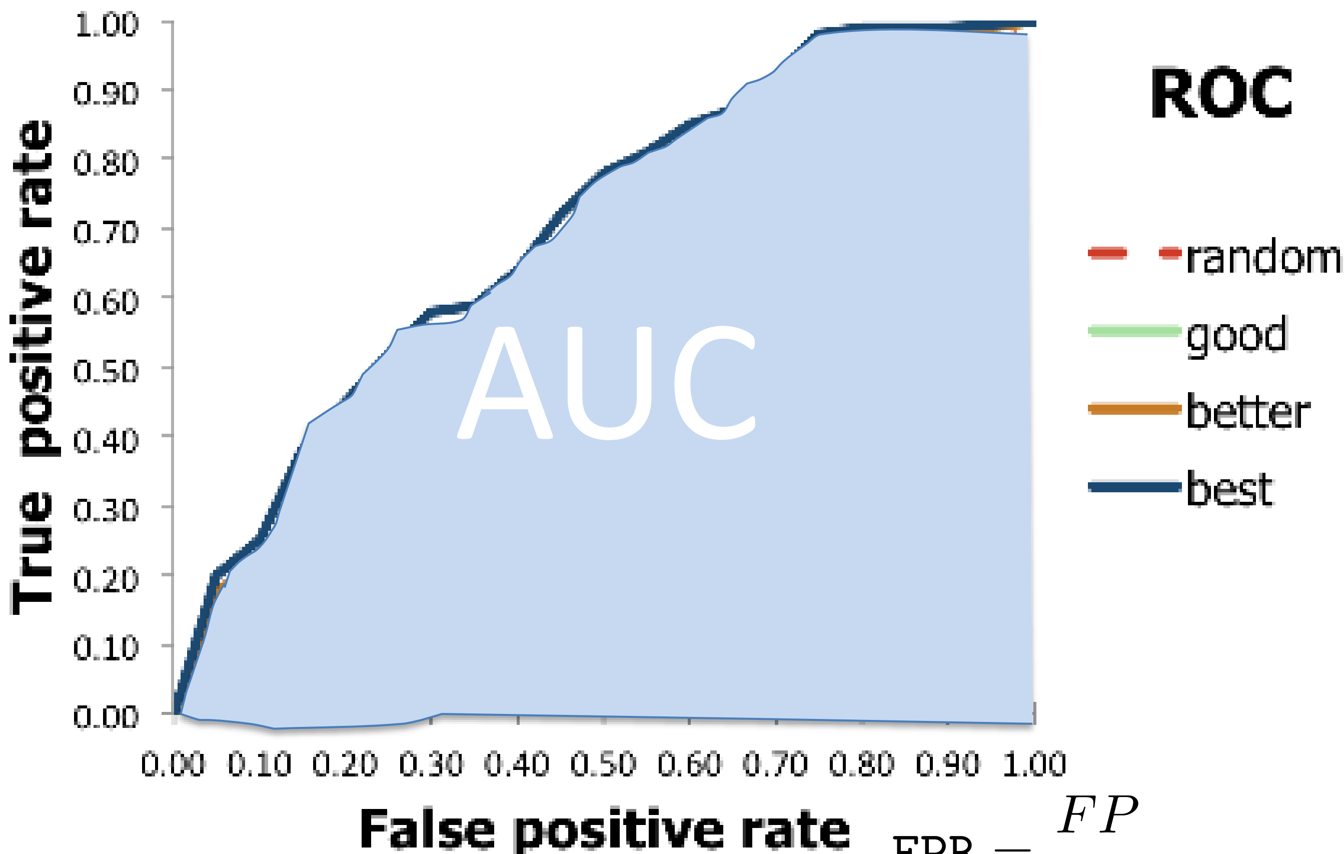
$\tau = 0.0$

$\tau = 1.0$



Area Under the Curve (AUC)

$$\mathrm{TPR} = \frac{TP}{P}$$



$$\mathrm{FPR} = \frac{FP}{N}$$

Area Under the Curve (AUC): Random Prediction

$$\mathrm{TPR} = \frac{TP}{P} \qquad \mathrm{FPR} = \frac{FP}{N}$$

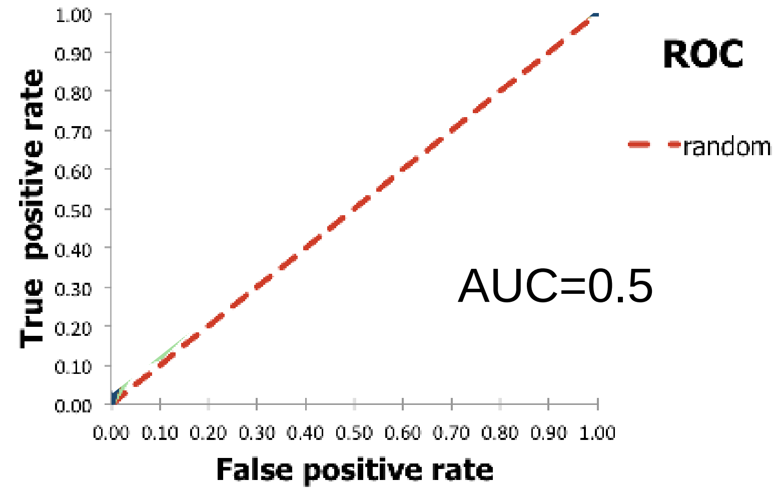
Suppose **a fraction x** of test set is **positive** and (1-x) is negative.

Consider **random** predictor: with **probability p predicts positive**, with (1-p) predicts negative

Then:

$$\mathrm{TPR} = \frac{TP}{P} \approx \frac{x \cdot p}{x} = p$$

$$\mathrm{FPR} = \frac{FP}{N} \approx \frac{(1-x) \cdot p}{1-x} = p$$



Northeastern

Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

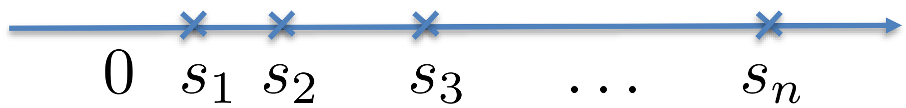
Suppose we sort test set so that

$$0 \leq s_1 \leq s_2 \leq s_3 \leq \ldots \leq s_n$$

Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

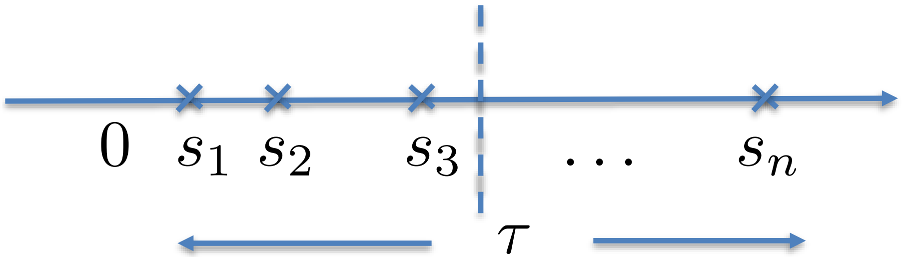
Suppose we sort test set so that



Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that



predict negative

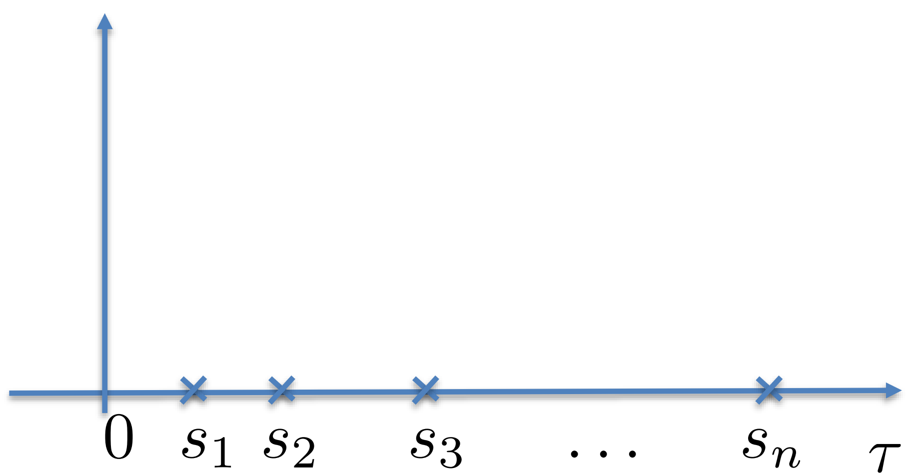
predict positive

Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

$$\mathrm{TPR} = \frac{TP}{P}$$



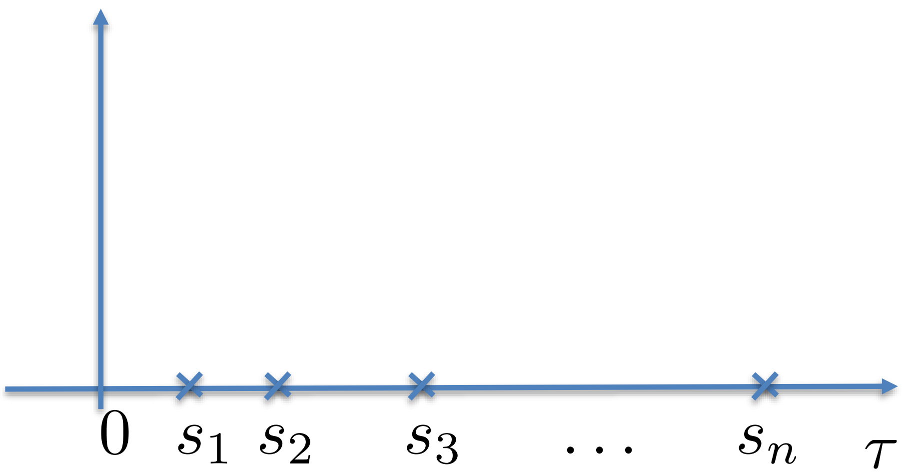
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

Ground Truth: +1 -1 -1 $\cdots$ +1

$$\text{TPR} = \frac{TP}{P}$$



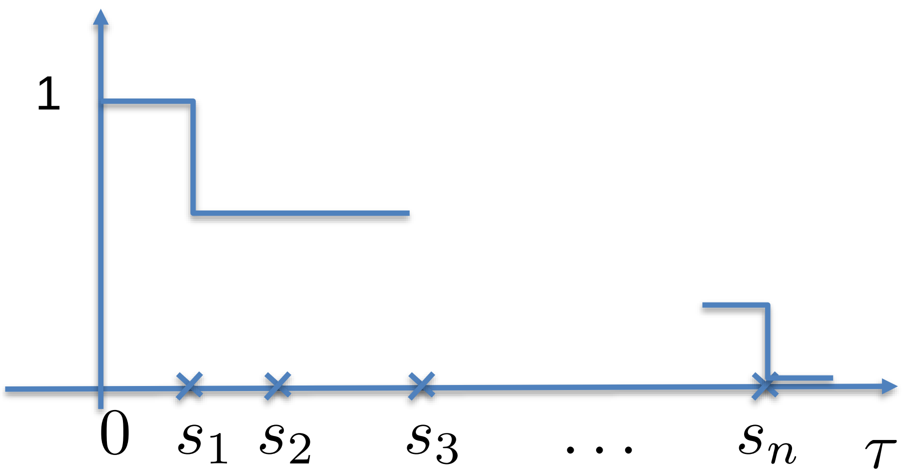
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

Ground Truth: +1 -1 -1 ... +1

$$\text{TPR} = \frac{TP}{P}$$



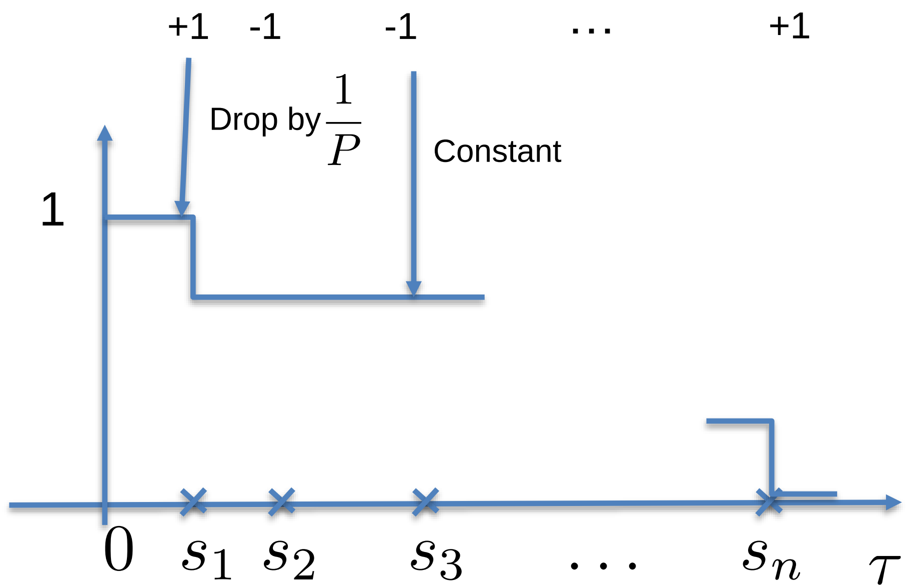
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

Ground Truth:

$\text{TPR} = \frac{TP}{P}$



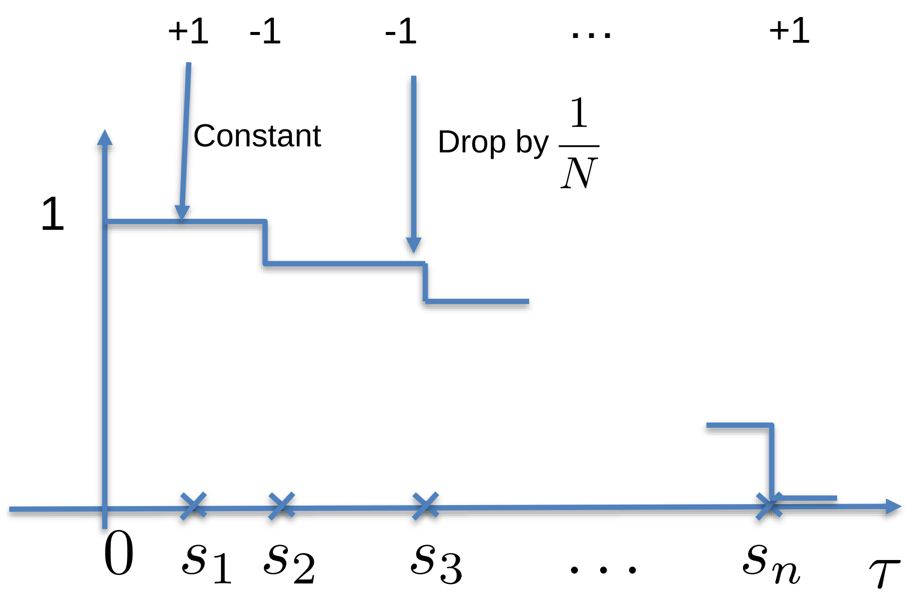
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

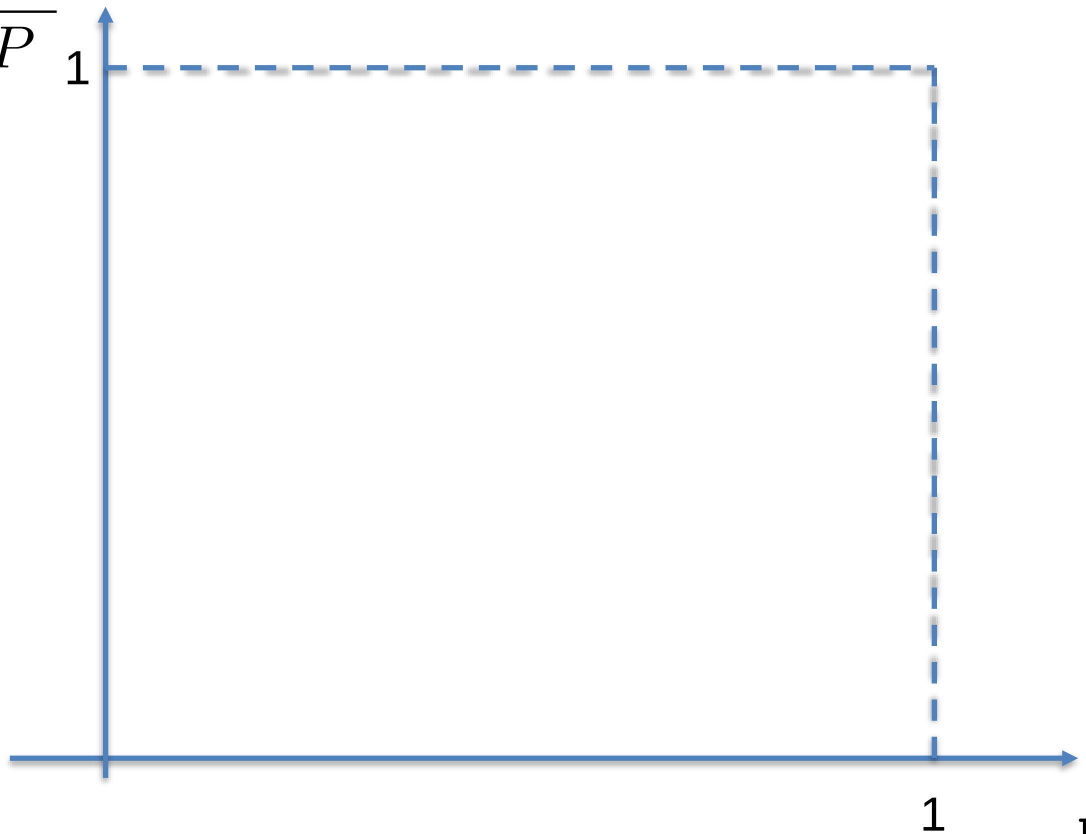
Ground Truth:

$\text{FPR} = \frac{FP}{N}$



Computing the AUC

$$\text{TPR} = \frac{TP}{P}$$



$$\text{FPR} = \frac{FP}{N}$$

Sorted Ground Truth

+1

-1

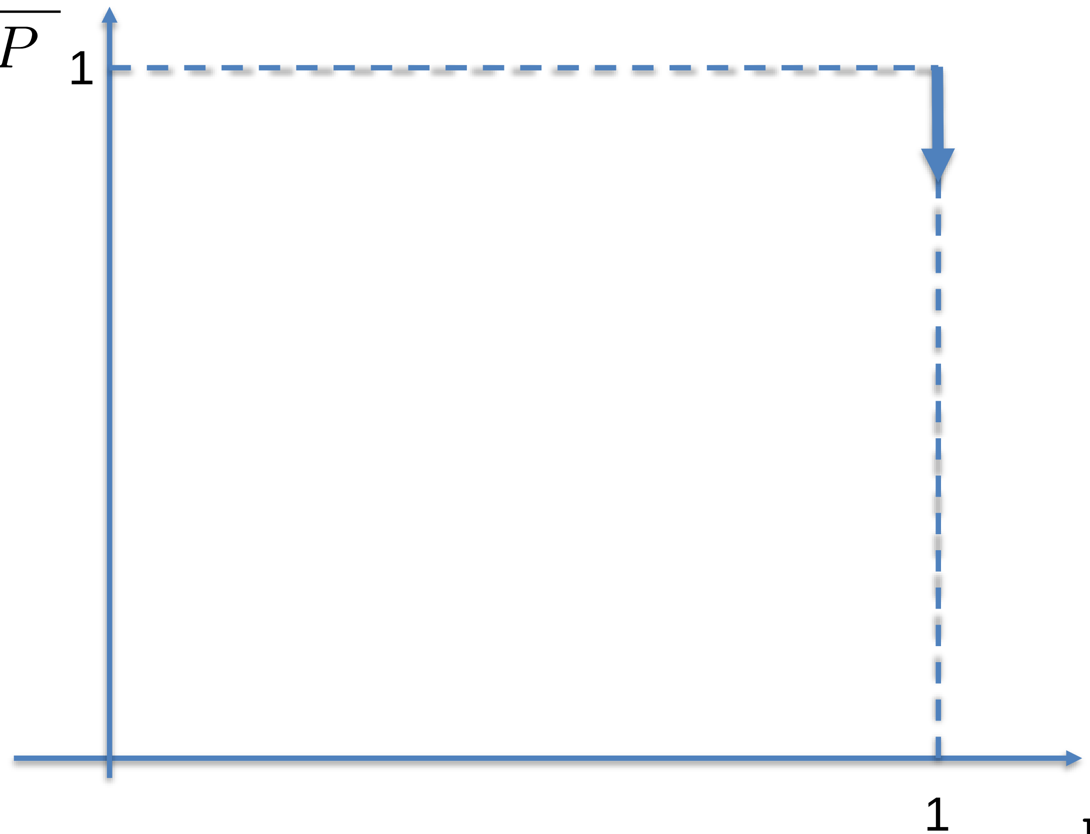
-1

+1

Computing the AUC

$$\text{TPR} = \frac{TP}{P}$$

Move down by $\frac{1}{P}$



$$\text{FPR} = \frac{FP}{N}$$

Sorted Ground Truth

+1

-1

-1

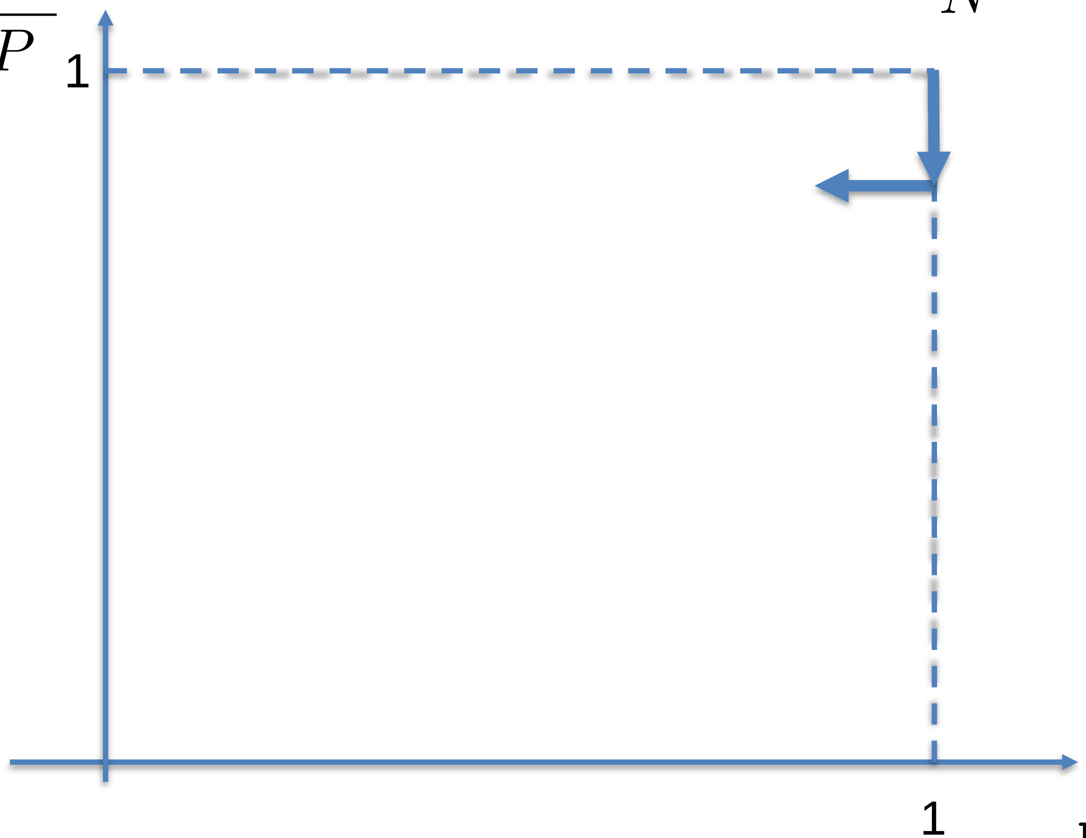
+1

Northeastern

Computing the AUC

Move left by $\frac{1}{N}$

$\text{TPR} = \frac{TP}{P}$



$\text{FPR} = \frac{FP}{N}$

Sorted Ground Truth

+1

-1

-1

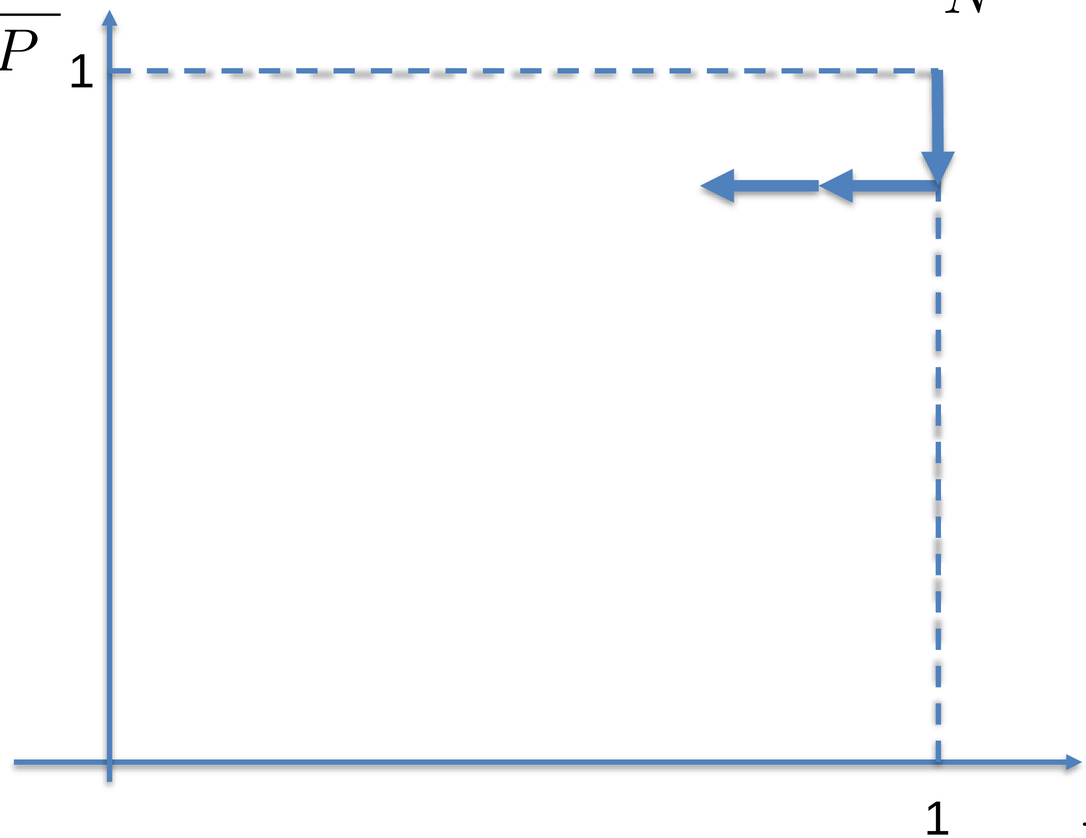
+1

Northeastern

Computing the AUC

Move left by $\frac{1}{N}$

$$\text{TPR} = \frac{TP}{P}$$



$$\text{FPR} = \frac{FP}{N}$$

Sorted Ground Truth

+1

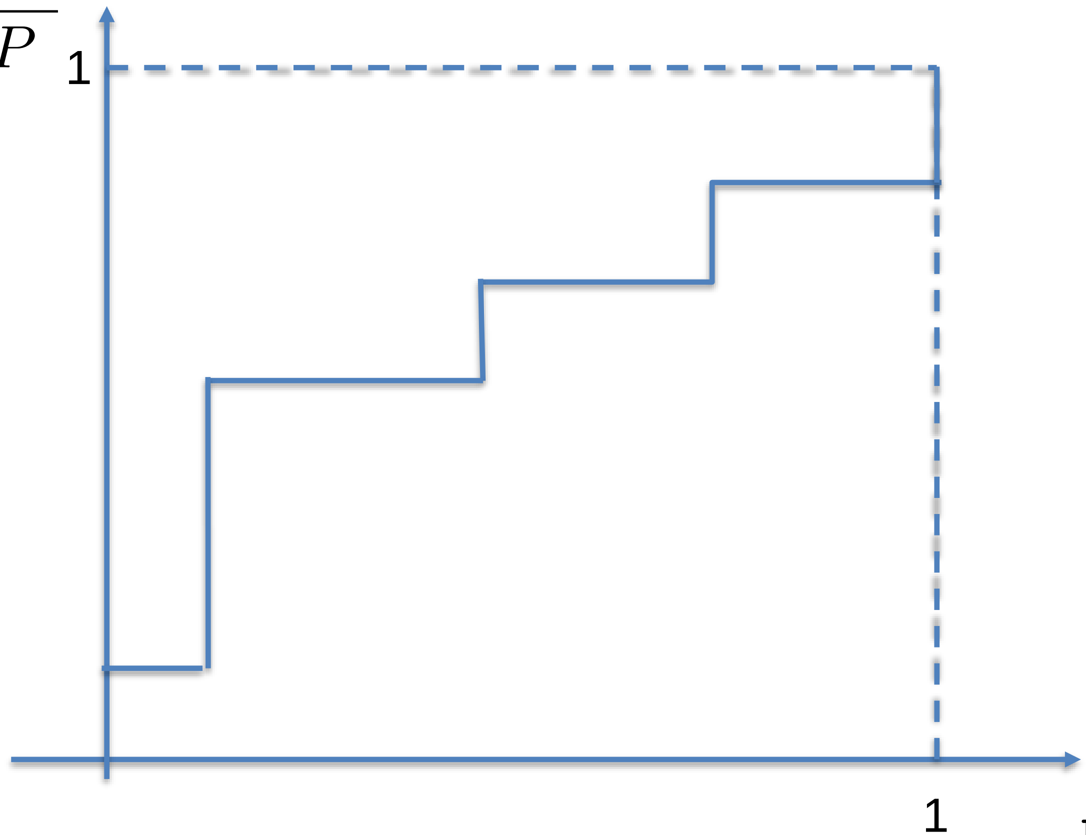
-1

-1

+1

Computing the AUC

$$\text{TPR} = \frac{TP}{P}$$



$$\text{FPR} = \frac{FP}{N}$$

Sorted Ground Truth

+1

-1

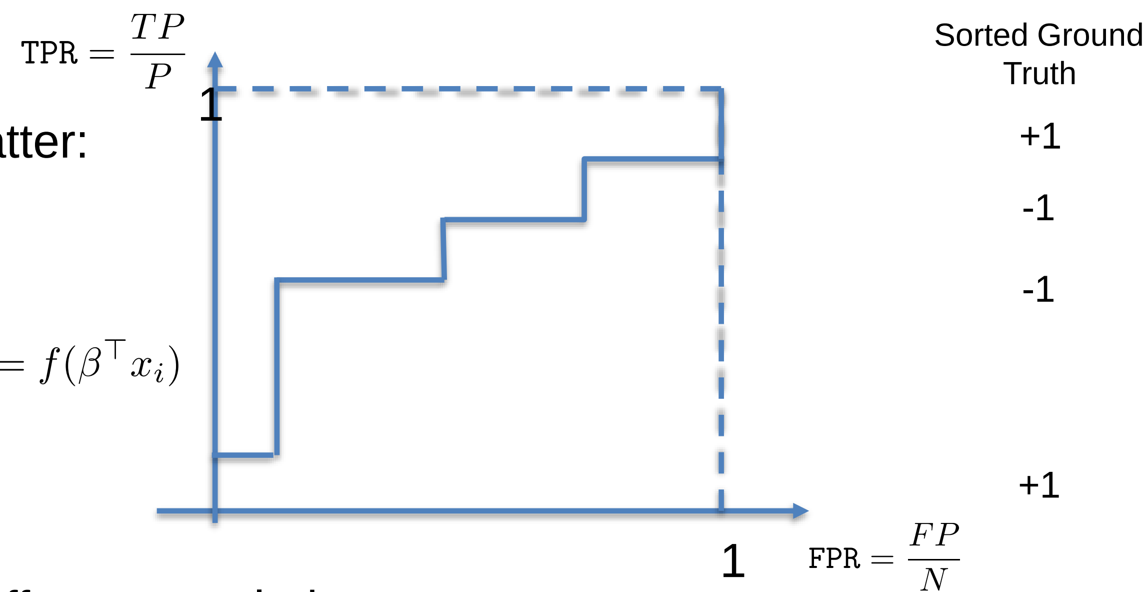
-1

+1

Computing the AUC

- ❑ Values of scores do not matter: only ordering does.

e.g. $s_i = \beta^\top x_i$ instead of $s_i = f(\beta^\top x_i)$



- ❑ Imbalance of P vs N only effects granularity

- ❑ You do not need to "integrate" over τ

Computing the AUC

```
def AUC(scores,labels):
    P = len( [x for x in labels if x >0  ])
    N = len( [x for x in labels if x <=0  ])
    zipped = sorted(zip(scores,labels))
    auc = 0.0
    count = 0.0
    for score,label in zipped:
        if label < 0:
            count += 1.0/N
        else:
            auc += count/P
    return auc

if __name__=="__main__":
    scores = [1,3,2,5,4]
    labels = [-1,-1,+1,+1,+1]
    print AUC(scores,labels)
```

0.833333333333