Chair for
Artificial Intelligence
Methodology
RWTH AACHEN UNIVERSITY

Elements of Machine Learning & Data Science

Winter semester 2023/24

Evaluation of Supervised Learning (2)

Prof. Holger Hoos (partially based on material from Wil van der Aalst)

Key questions:

- How good is an ML model?
- How good could an ML model be?

You have used supervised ML to train a predictive model.

Question: How do you assess the quality of the model?

NB: So far, focus on binary classification problems.

Key concepts covered last class:

- confusion matrix
- performance measures for binary classification
- training, testing and validation sets
- *k*-fold cross validation
- leave-one-out cross validation (jackknife)
- bootstrap sampling validation
- imbalanced data, average class accuracy
- profit (utility) matrix

Preparation for today:

Investigate the following questions:

- **How to assess predictive models for multi-class classification?**
  (> 2 target classes, *e.g.*, on time, mildly delayed, severely delayed)

- **How to assess predictive models for regression tasks?**
  (predictions = numbers, *e.g.*, minutes of delay)

(We will use this for TPS exercises with the T part done before class.)

TPS Exercise (T part = done as homework)

Question:

How to assess predictive models for <u>multi-class classification</u>?
(> 2 target classes, e.g., on time, mildly delayed, severely delayed)

Multinomial Targets

ID	Target Label	Prediction
1	On Time	Delayed
2	On Time	Delayed
3	Delayed	Canceled
4	Canceled	On Time
5	Delayed	Delayed
6	On Time	On Time
7	Delayed	Delayed
8	Canceled	Canceled
9	On Time	On Time
10	On Time	**On Time**

- More than two possible values for the target feature
- How to compute confusion matrix-based performance measures?

Multinomial Targets

ID	Target Label	Prediction
1	On Time	Delayed
2	On Time	Delayed
3	Delayed	Canceled
4	Canceled	On Time
5	Delayed	Delayed
6	On Time	On Time
7	Delayed	Delayed
8	Canceled	Canceled
9	On Time	On Time
10	On Time	**On Time**

How to define TP, FP, TN, FN?

		Prediction		
		On Time	Delayed	Canceled
Target	On Time	3	2	0
	Delayed	0	2	1
	Canceled	1	0	1

Multinomial Targets

For each possible target label value:

- Consider this label as positive, all others as negative
- Compute TP, TN, FP, FN as before
- Compute performance measures as before

		Prediction		
		On Time	Delayed	Canceled
Target	On Time	3	2	0
	Delayed	0	2	1
	Canceled	1	0	1

Multinomial Targets

For each possible target label value:

- Consider this label as positive, all others as negative
- Compute TP, TN, FP, FN as before
- Compute performance measures as before

On Time → Positive

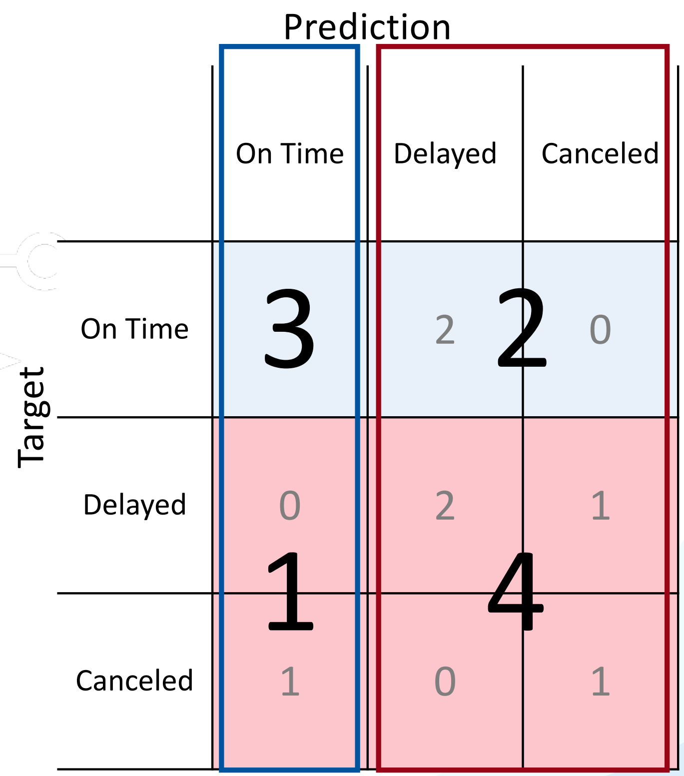
Delayed, Canceled → Negative

		Prediction		
		On Time	Delayed	Canceled
Target	On Time	3	2	0
	Delayed	0	2	1
	Canceled	1	0	1

Multinomial Targets

For each possible target label value:

- Consider this label as positive, all others as negative
- Compute TP, TN, FP, FN as before
- Compute performance measures as before



On Time → Positive

Delayed, Canceled → Negative

TP=3, FN=2+0=2, FP=0+1=1, TN=2+1+0+1=4

Multinomial Targets

For each possible target label value:

- Consider this label as positive, all others as negative
- Compute TP, TN, FP, FN as before
- Compute performance measures as before

On Time → Positive

Delayed, Canceled → Negative

$$precision_{\texttt{on time}} = \frac{TP_{\texttt{on time}}}{TP_{\texttt{on time}} + FP_{\texttt{on time}}} = \frac{3}{3+(0+1)} = \frac{3}{4}$$

$$recall_{\texttt{on time}} = \frac{TP_{\texttt{on time}}}{TP_{\texttt{on time}} + FN_{\texttt{on time}}} = \frac{3}{3+(2+0)} = \frac{3}{5}$$

Target \ Prediction	On Time	Delayed	Canceled
On Time	3	2	0
Delayed	0	2	1
Canceled	1	0	1

Multinomial Targets

For each possible target label value:

- Consider this label as positive, all others as negative
- Compute TP, TN, FP, FN as before
- Compute performance measures as before

Delayed→ Positive

On Time, Canceled → Negative

$$precision_{\texttt{delayed}} = \frac{TP_{\texttt{delayed}}}{TP_{\texttt{delayed}} + FP_{\texttt{delayed}}} = \frac{2}{2+(2+0)} = \frac{1}{2}$$

$$recall_{\texttt{delayed}} = \frac{TP_{\texttt{delayed}}}{TP_{\texttt{delayed}} + FN_{\texttt{delayed}}} = \frac{2}{2+(0+1)} = \frac{2}{3}$$

		Prediction		
		On Time	Delayed	Canceled
Target	On Time	3	2	0
	Delayed	0	2	1
	Canceled	1	0	1

Multinomial Targets

For each possible target label value:

- Consider this label as positive, all others as negative
- Compute TP, TN, FP, FN as before
- Compute performance measures as before

Canceled→ Positive

On Time, Delayed → Negative

$$precision_{\texttt{canceled}} = \frac{TP_{\texttt{canceled}}}{TP_{\texttt{canceled}} + FP_{\texttt{canceled}}} = \frac{1}{1+(0+1)} = \frac{1}{2}$$

$$recall_{\texttt{canceled}} = \frac{TP_{\texttt{canceled}}}{TP_{\texttt{canceled}} + FN_{\texttt{canceled}}} = \frac{1}{1+(1+0)} = \frac{1}{2}$$

	Prediction		
Target	On Time	Delayed	Canceled
On Time	3	2	0
Delayed	0	2	1
Canceled	1	0	1

Multinomial Targets

Individual recalls can be combined using **average class accuracy** (harmonic mean):

$$recall_{\texttt{on time}} = \frac{3}{5}$$

$$recall_{\texttt{delayed}} = \frac{2}{3}$$

$$recall_{\texttt{canceled}} = \frac{1}{2}$$

K is the number of label values

recall of the *k*th label value

$$\frac{1}{\frac{1}{K}\cdot\left(\sum_{k=1}^{K}\left(\frac{1}{recall_k}\right)\right)}$$

$$\Rightarrow \frac{1}{\frac{1}{3}\cdot\left(\frac{1}{recall_{\texttt{on time}}}+\frac{1}{recall_{\texttt{delayed}}}+\frac{1}{recall_{\texttt{canceled}}}\right)}$$

$$= \frac{18}{31} \approx 0.58$$

Target \ Prediction	On Time	Delayed	Canceled
On Time	3	2	0
Delayed	0	2	1
Canceled	1	0	1

TPS Exercise (T part = done as homework)

Question:

How to assess predictive models for regression tasks?
(predictions = numbers, *e.g.*, minutes of delay)

Reminder: Error Functions

Sum of squared errors	$\frac{1}{2}\sum_{i=1}^{N}((t_i - \mathbb{M}(\mathbf{x_i}))^2)$
Mean squared error	$\frac{1}{N}\sum_{i=1}^{N}((t_i - \mathbb{M}(\mathbf{x_i}))^2)$
Root mean squared error	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}((t_i - \mathbb{M}(\mathbf{x_i}))^2)}$
Mean absolute error	$\frac{1}{N}\sum_{i=1}^{N}\lvert t_i - \mathbb{M}(\mathbf{x_i})\rvert$

For the ith instance,
t_i is the true target value and
$\mathbb{M}(\mathbf{x_i})$ is the predicted value.

Coefficient of Determination (R^2)

- Compare model performance with the model that always guesses the average (baseline)
- Close to 0 → no better than guessing the average
- Close to 1 → all predictions are perfect
- Cross validation as before

$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}}$$

$$\text{sum of squared errors} = \sum_{i=1}^{N} ((t_i - \mathbb{M}(\mathbf{x_i}))^2)$$

$$\text{total sum of squares} = \sum_{i=1}^{N} (t_i - \bar{t})^2$$

$\bar{t}$ is the mean of all target values: $\frac{1}{N} \sum_{j=1}^{N} t_j$

Coefficient of Determination (R^2) – Example

$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}}$$

$$\text{sum of squared errors} = \sum_{i=1}^{N}((t_i - \mathbb{M}(\mathbf{x_i}))^2)$$

$$\text{total sum of squares} = \sum_{i=1}^{N}(t_i - \bar{t})^2$$

ID	Delay [min]	Predicted Delay [min]	$t_i - \mathbb{M}(\mathbf{x_i})$	$(t_i - \mathbb{M}(\mathbf{x_i}))^2$	$t_i - \bar{t}$	$(t_i - \bar{t})^2$
1	34	15				
2	-6	-9				
3	3	2				
4	9	8				

Coefficient of Determination (R^2) – Example

$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}}$$

$$\text{sum of squared errors} = \sum_{i=1}^{N} \left((t_i - \mathbb{M}(\mathbf{x_i}))^2 \right)$$

$$\text{total sum of squares} = \sum_{i=1}^{N} (t_i - \bar{t})^2$$

ID	Delay [min]	Predicted Delay [min]	$t_i - \mathbb{M}(\mathbf{x_i})$	$(t_i - \mathbb{M}(\mathbf{x_i}))^2$	$t_i - \bar{t}$	$(t_i - \bar{t})^2$
1	34	15	19	361	24	576
2	-6	-9	3	9	-16	256
3	3	2	1	1	-7	49
4	9	8	1	1	-1	1
Mean:	10		**Sum:**	372	**Sum:**	**882**

Coefficient of Determination (R^2) – Example

$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}}$$

$$\text{sum of squared errors} = \sum_{i=1}^{N}((t_i - \mathbb{M}(\mathbf{x_i}))^2) = \frac{1}{2} \cdot 372 = 186$$

$$\text{total sum of squares} = \sum_{i=1}^{N}(t_i - \bar{t})^2 = \frac{1}{2} \cdot 882 = 441$$

ID	Delay [min]	Predicted Delay [min]	$t_i - \mathbb{M}(\mathbf{x_i})$	$(t_i - \mathbb{M}(\mathbf{x_i}))^2$	$t_i - \bar{t}$	$(t_i - \bar{t})^2$
1	34	15	19	361	24	576
2	-6	-9	3	9	-16	256
3	3	2	1	1	-7	49
4	9	8	1	1	-1	1
Mean:	10		**Sum:**	372	**Sum:**	**882**

Coefficient of Determination (R^2) – Example

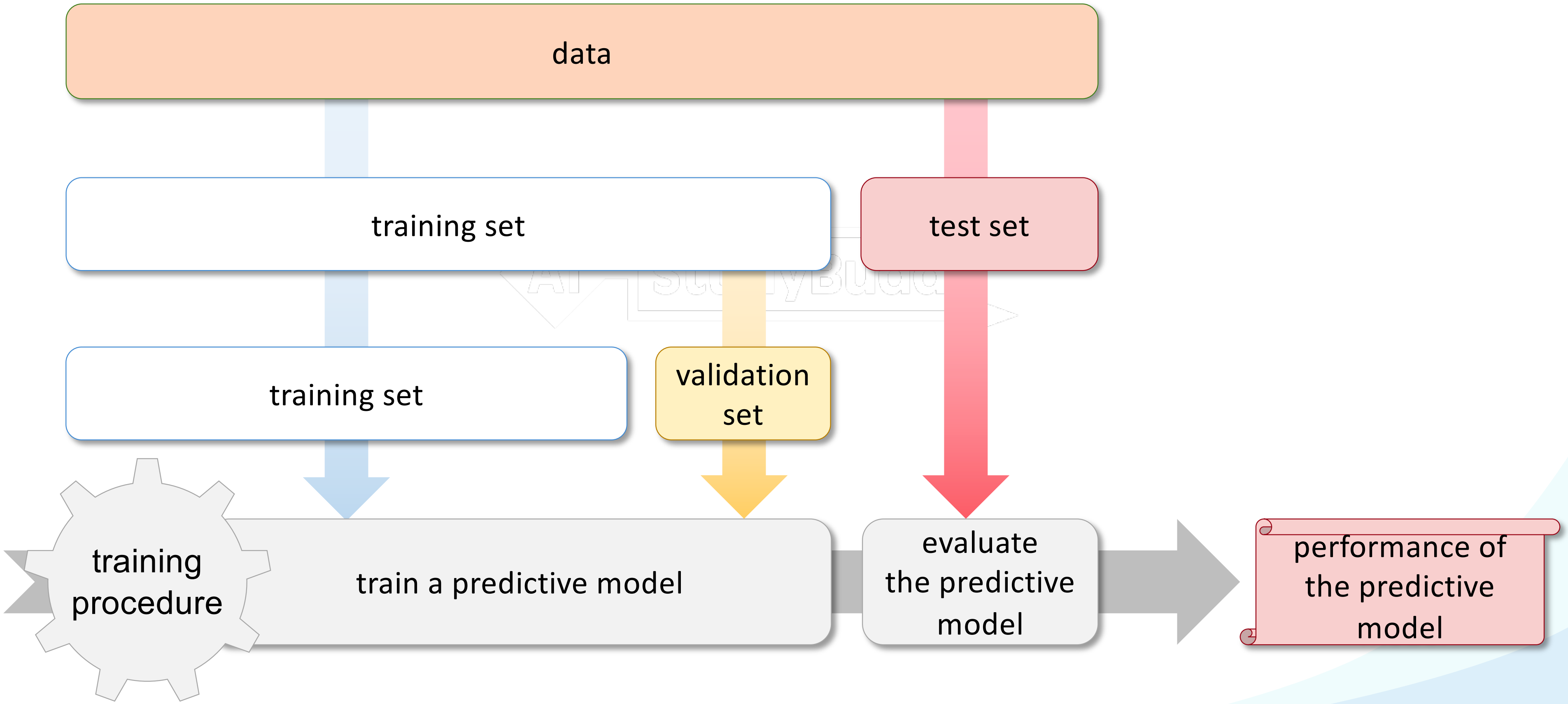
$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}} = 1 - \frac{186}{441} \approx 0.42$$

$$\text{sum of squared errors} = \sum_{i=1}^{N}((t_i - \mathbb{M}(\mathbf{x_i}))^2) = \frac{1}{2} \cdot 372 = 186$$

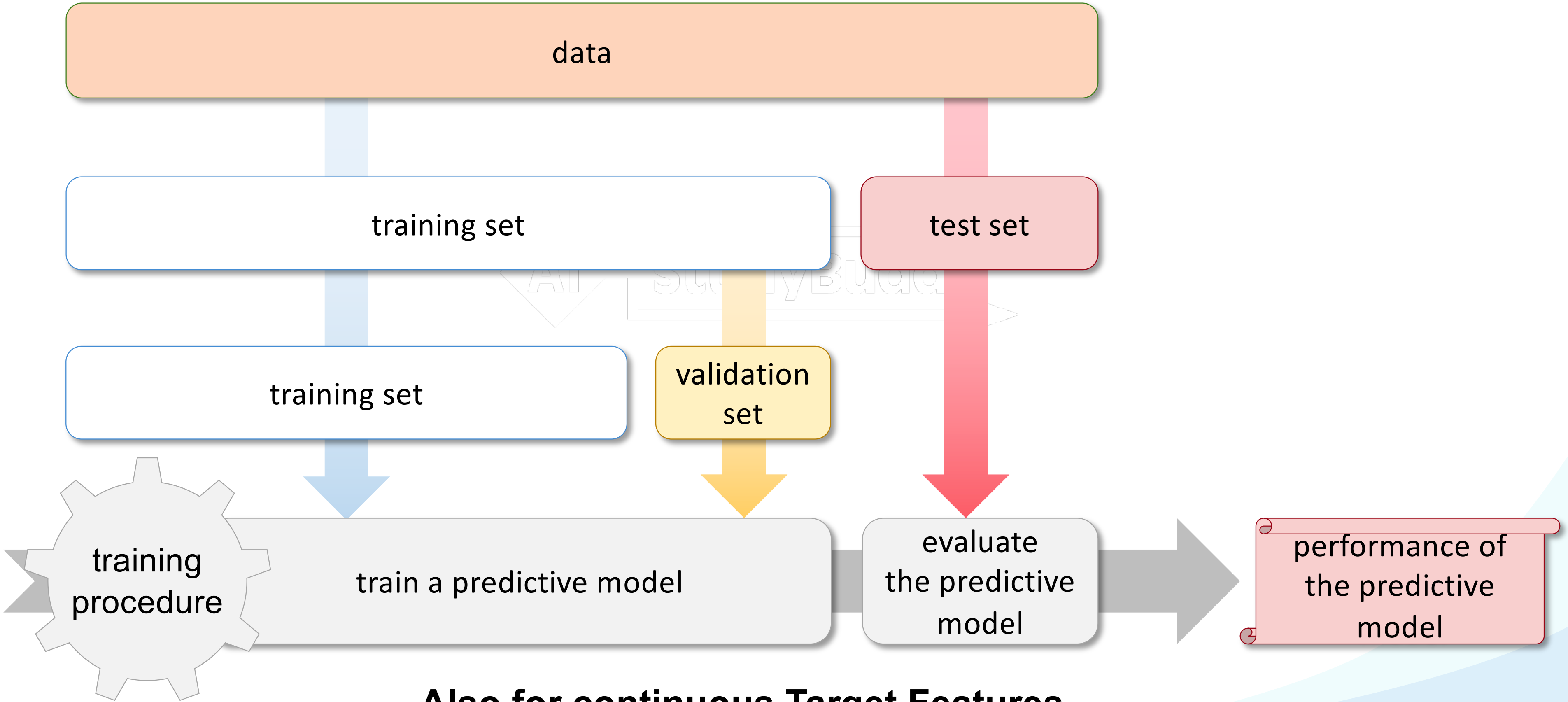
$$\text{total sum of squares} = \sum_{i=1}^{N}(t_i - \bar{t})^2 = \frac{1}{2} \cdot 882 = 441$$

ID	Delay [min]	Predicted Delay [min]	$t_i - \mathbb{M}(\mathbf{x_i})$	$(t_i - \mathbb{M}(\mathbf{x_i}))^2$	$t_i - \bar{t}$	$(t_i - \bar{t})^2$
1	34	15	19	361	24	576
2	-6	-9	3	9	-16	256
3	3	2	1	1	-7	49
4	9	8	1	1	-1	1
Mean:	10		**Sum:**	372	**Sum:**	**882**

Reminder

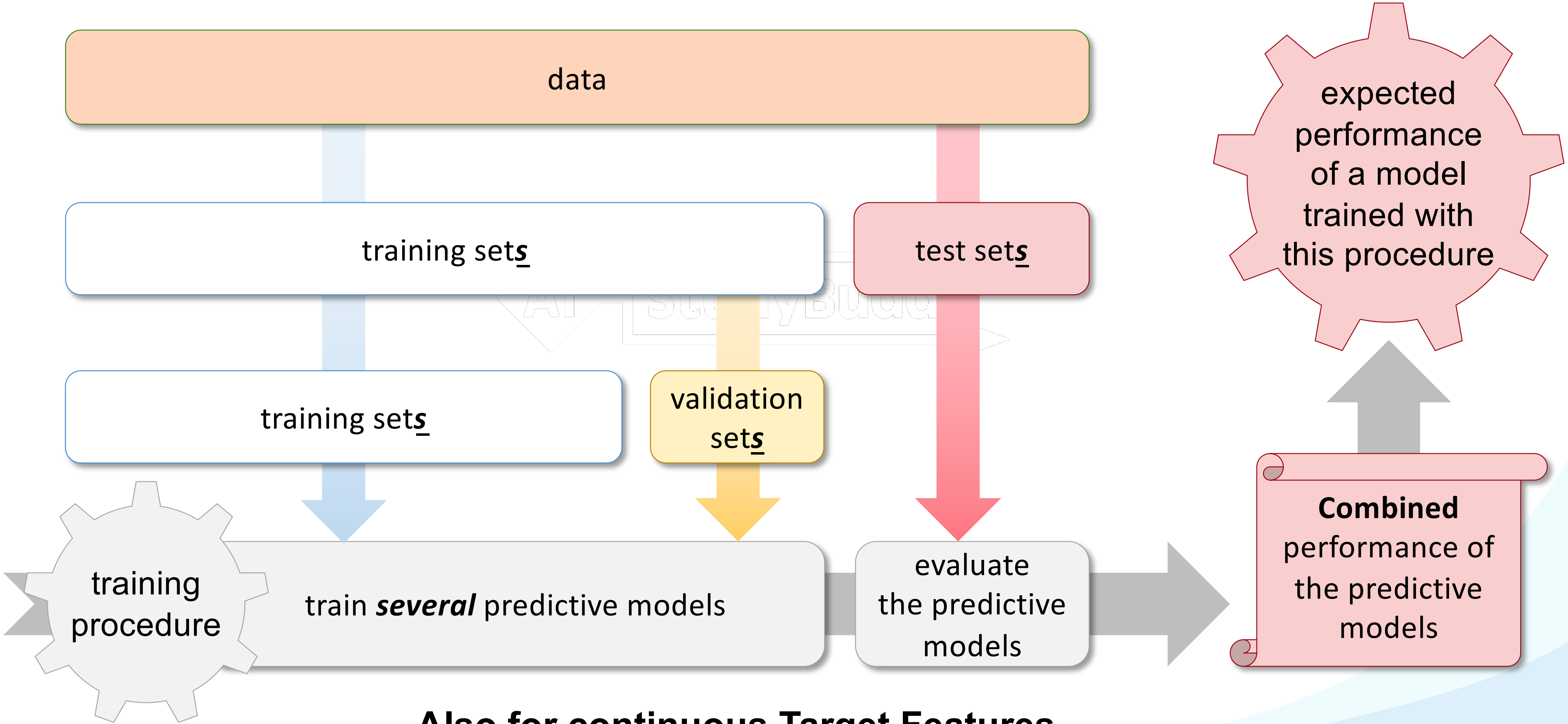

data
training set
test set
training set
validation set
training procedure
train a predictive model
evaluate the predictive model
performance of the predictive model

Reminder



Also for continuous Target Features

Reminder (2)



Also for continuous Target Features

TPS Exercise

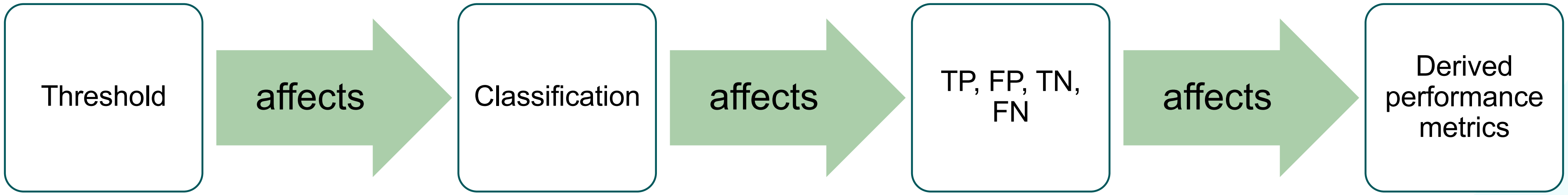
You have used supervised ML to train a predictive model for a binary classification problem. The model gives you a numerical <u>prediction score</u> between 0 and 1.

Question:

How to assess the quality of the model?

Motivation

- Models often return **prediction score** representing how ‘sure’ they are about the target feature (e.g., logistic regression, decision trees, Bayes, NNs)
- Assume prediction score ∈ [0,1]
- Prediction score is mapped to class based on **threshold**
  – often implicitly assume 0.5, but other values possible!



Changing the Threshold - Example

Prediction

	0.25	On Time	Delayed
Target	On Time	5	0
	Delayed	4	1
	Misclassification Rate:	0.4	

TPR = 1

FPR = 0.8

TNR = 1 - FPR

FNR = 1 - TPR

ID	Target Label	Prediction Score	Prediction for various thresholds		
			0.25	**0.5**	**0.75**
1	Delayed	0.12	Delayed		
2	Delayed	0.28	On Time		
3	Delayed	0.30	On Time		
4	Delayed	0.29	On Time		
5	On Time	0.43	On Time		
6	Delayed	0.54	On Time		
7	On Time	0.63	On Time		
8	On Time	0.72	On Time		
9	On Time	0.84	On Time		
10	On Time	0.99	On Time		

Changing the Threshold - Example

Prediction (Target)

0.25	On Time	Delayed
On Time	5	0
Delayed	4	1
Misclassification Rate:	0.4	

TPR = 1

FPR = 0.8

TNR = 1 - FPR

FNR = 1 - TPR

Prediction (Target)

0.5	On Time	Delayed
On Time	4	1
Delayed	1	4
Misclassification Rate:	0.2	

TPR = 0.8

FPR = 0.2

ID	**Target Label**	**Prediction Score**	**Prediction for various thresholds**		
			0.25	**0.5**	**0.75**
1	Delayed	0.12	Delayed	Delayed	
2	Delayed	0.28	On Time	Delayed	
3	Delayed	0.30	On Time	Delayed	
4	Delayed	0.29	On Time	Delayed	
5	On Time	0.43	On Time	Delayed	
6	Delayed	0.54	On Time	On Time	
7	On Time	0.63	On Time	On Time	
8	On Time	0.72	On Time	On Time	
9	On Time	0.84	On Time	On Time	
10	On Time	0.99	On Time	On Time	

Changing the Threshold - Example

Target	0.25	Prediction: On Time	Prediction: Delayed
	On Time	5	0
	Delayed	4	1
	Misclassification Rate:	0.4	

TPR = 1

FPR = 0.8

TNR = 1 - FPR

FNR = 1 - TPR

Target	0.5	Prediction: On Time	Prediction: Delayed
	On Time	4	1
	Delayed	1	4
	Misclassification Rate:	0.2	

TPR = 0.8

FPR = 0.2

Target	0.75	Prediction: On Time	Prediction: Delayed
	On Time	2	3
	Delayed	0	5
	Misclassification Rate:	0.3	

TPR = 0.4

FPR = 0

ID	Target Label	Prediction Score	Prediction for various thresholds		
			0.25	0.5	0.75
1	Delayed	0.12	Delayed	Delayed	Delayed
2	Delayed	0.28	On Time	Delayed	Delayed
3	Delayed	0.30	On Time	Delayed	Delayed
4	Delayed	0.29	On Time	Delayed	Delayed
5	On Time	0.43	On Time	Delayed	Delayed
6	Delayed	0.54	On Time	On Time	Delayed
7	On Time	0.63	On Time	On Time	Delayed
8	On Time	0.72	On Time	On Time	Delayed
9	On Time	0.84	On Time	On Time	On Time
10	On Time	0.99	On Time	On Time	**On Time**

Receiver Operating Characteristic (ROC) Curve – Example

Prediction

Target

0.25	On Time	Delayed
On Time	5	0
Delayed	4	1
Misclassification Rate:	0.4	

TPR = 1

FPR = 0.8

TNR = 1 - FPR

FNR = 1 - TPR

Prediction

Target

0.5	On Time	Delayed
On Time	4	1
Delayed	1	4
Misclassification Rate:	0.2	

TPR = 0.8

FPR = 0.2

Prediction

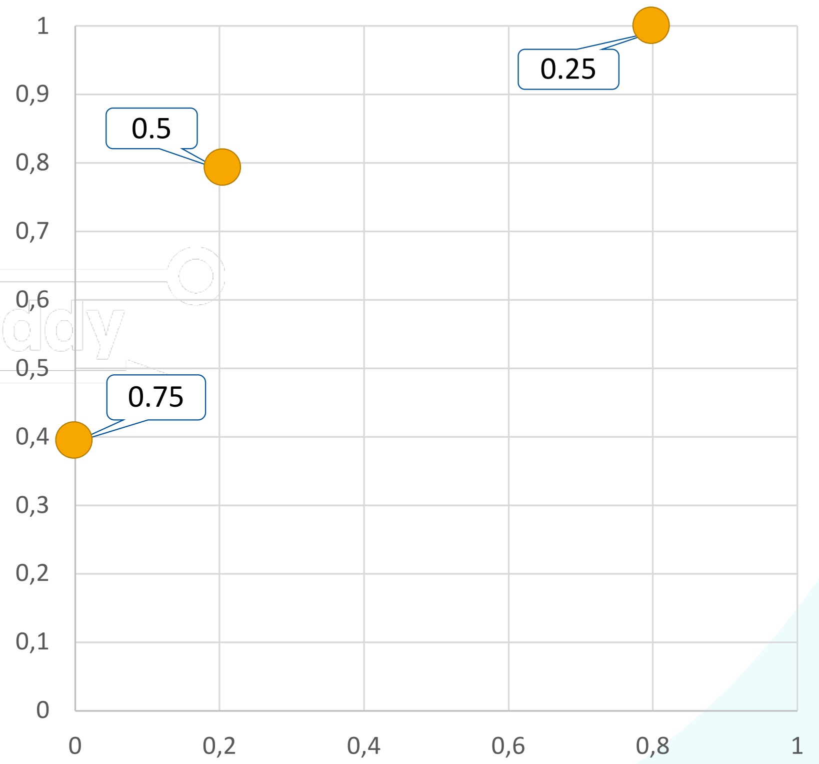
Target

0.75	On Time	Delayed
On Time	2	3
Delayed	0	5
Misclassification Rate:	0.3	

TPR = 0.4

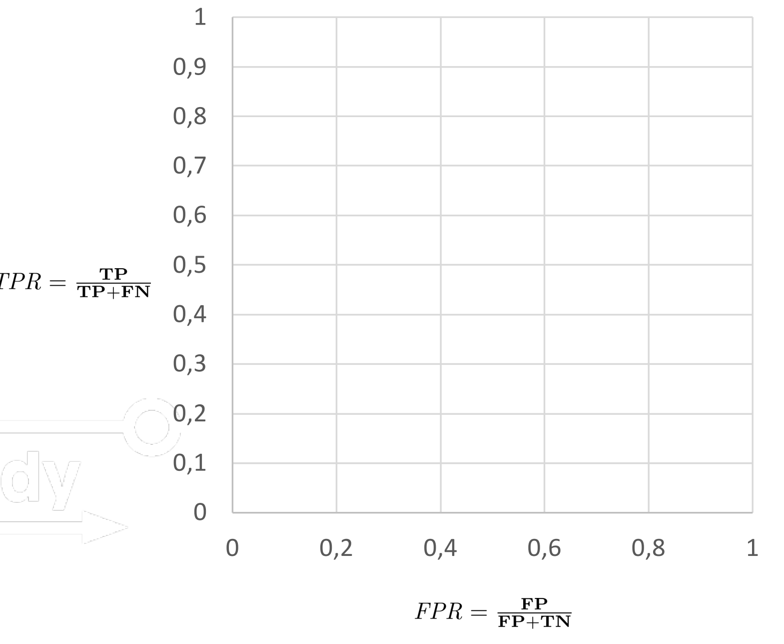
FPR = 0

$$TPR = \frac{\mathbf{TP}}{\mathbf{TP+FN}}$$



$$FPR = \frac{\mathbf{FP}}{\mathbf{FP+TN}}$$

TPS Exercise



Questions:

1) What does an ideal ROC Curve look like?
2) What about worst-case ROC Curve?

ROC Curve – Example

Prediction

0.25	On Time	Delayed
On Time	5	0
Delayed	4	1
Misclassification Rate:	0.4	

Target

TPR = 1

FPR = 0.8

TNR = 1 - FPR

FNR = 1 - TPR

Prediction

0.5	On Time	Delayed
On Time	4	1
Delayed	1	4
Misclassification Rate:	0.2	

Target

TPR = 0.8

FPR = 0.2

$$TPR = \frac{\mathbf{TP}}{\mathbf{TP+FN}}$$

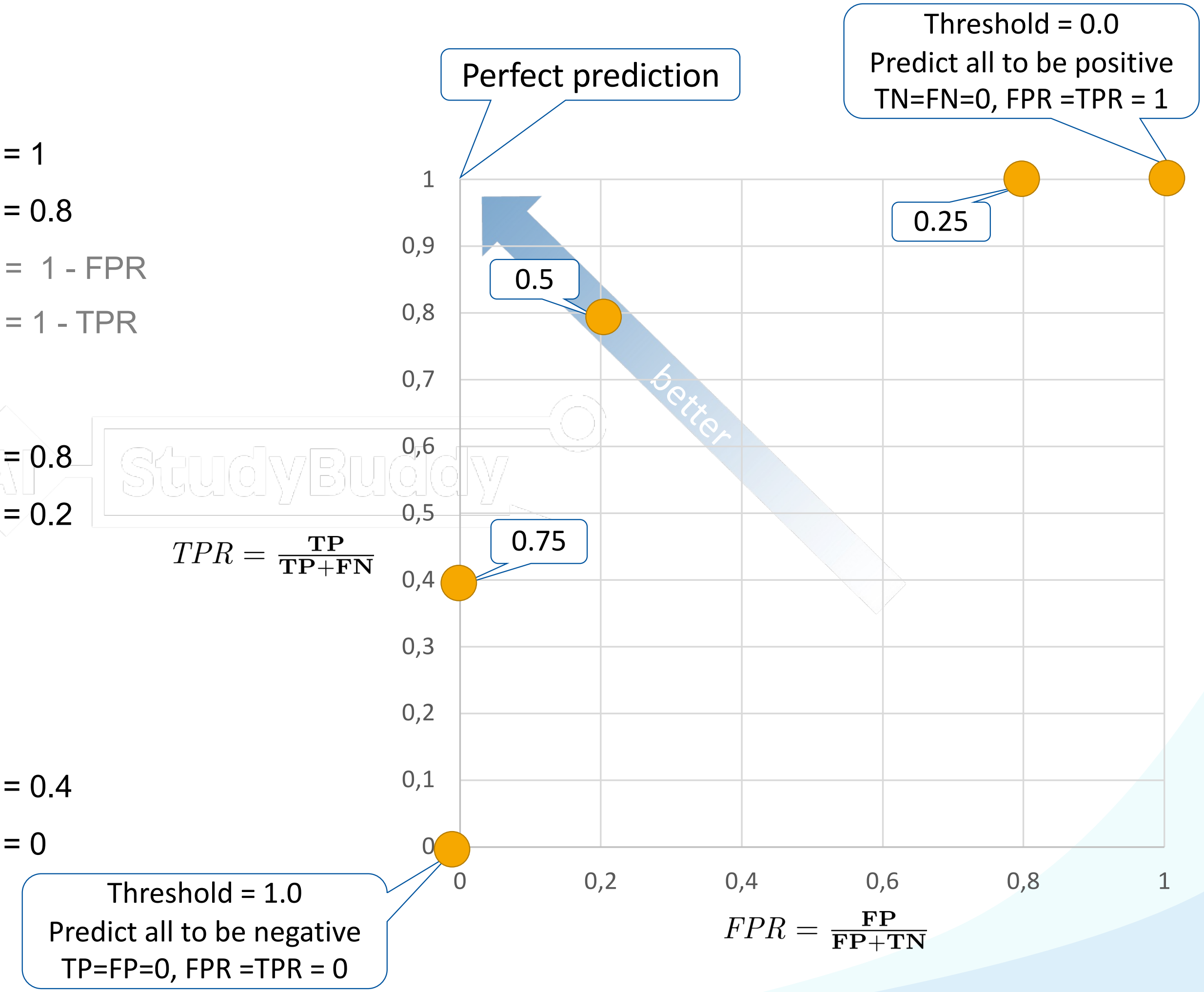
Prediction

0.75	On Time	Delayed
On Time	2	3
Delayed	0	5
Misclassification Rate:	0.3	

Target

TPR = 0.4

FPR = 0



$$FPR = \frac{\mathbf{FP}}{\mathbf{FP+TN}}$$

ROC Curve – Example

Prediction (Target rows)

0.25	On Time	Delayed
On Time	5	0
Delayed	4	1
Misclassification Rate:	0.4	

TPR = 1

FPR = 0.8

TNR = 1 - FPR

FNR = 1 - TPR

Prediction (Target rows)

0.5	On Time	Delayed
On Time	4	1
Delayed	1	4
Misclassification Rate:	0.2	

TPR = 0.8

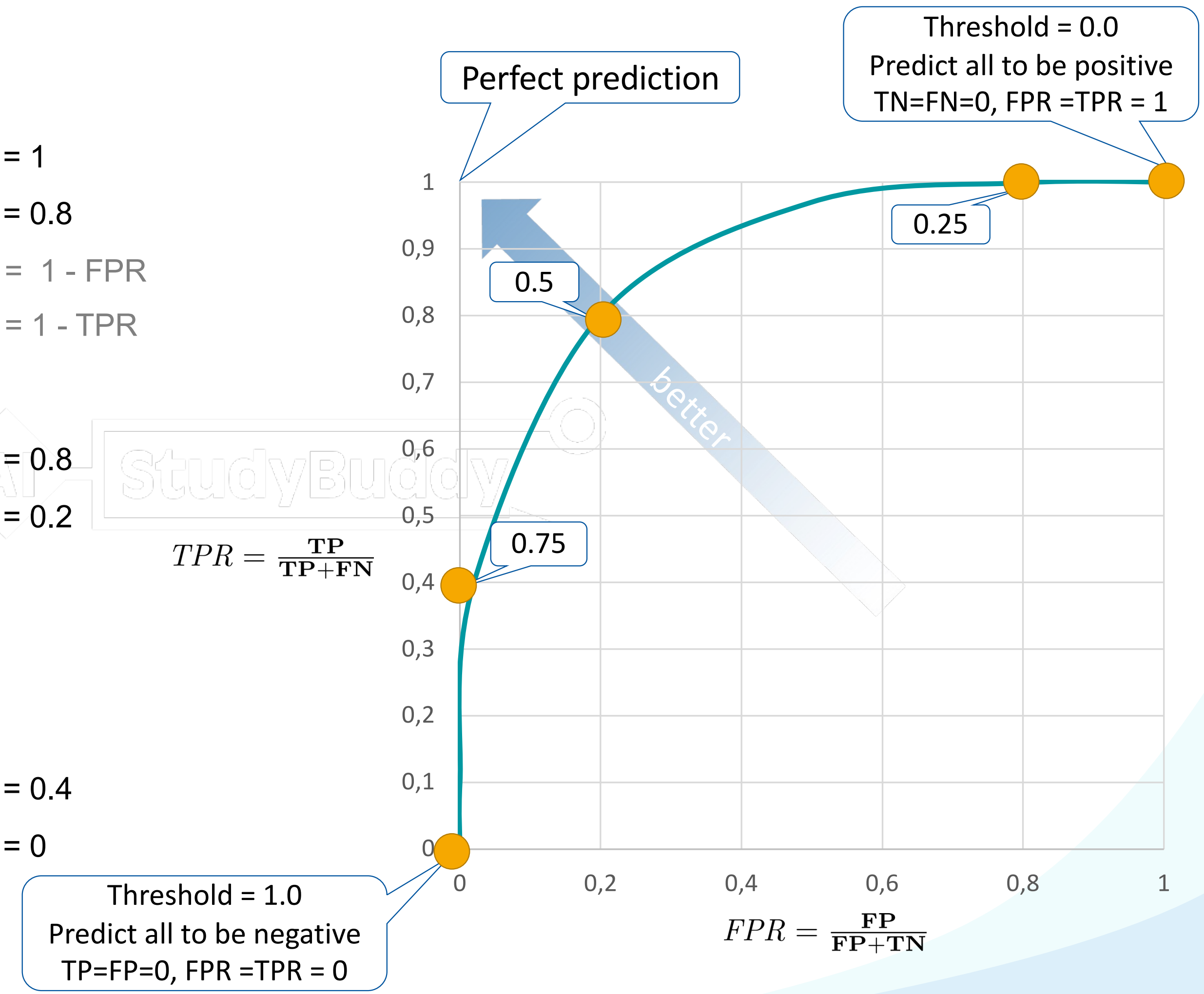
FPR = 0.2

Prediction (Target rows)

0.75	On Time	Delayed
On Time	2	3
Delayed	0	5
Misclassification Rate:	0.3	

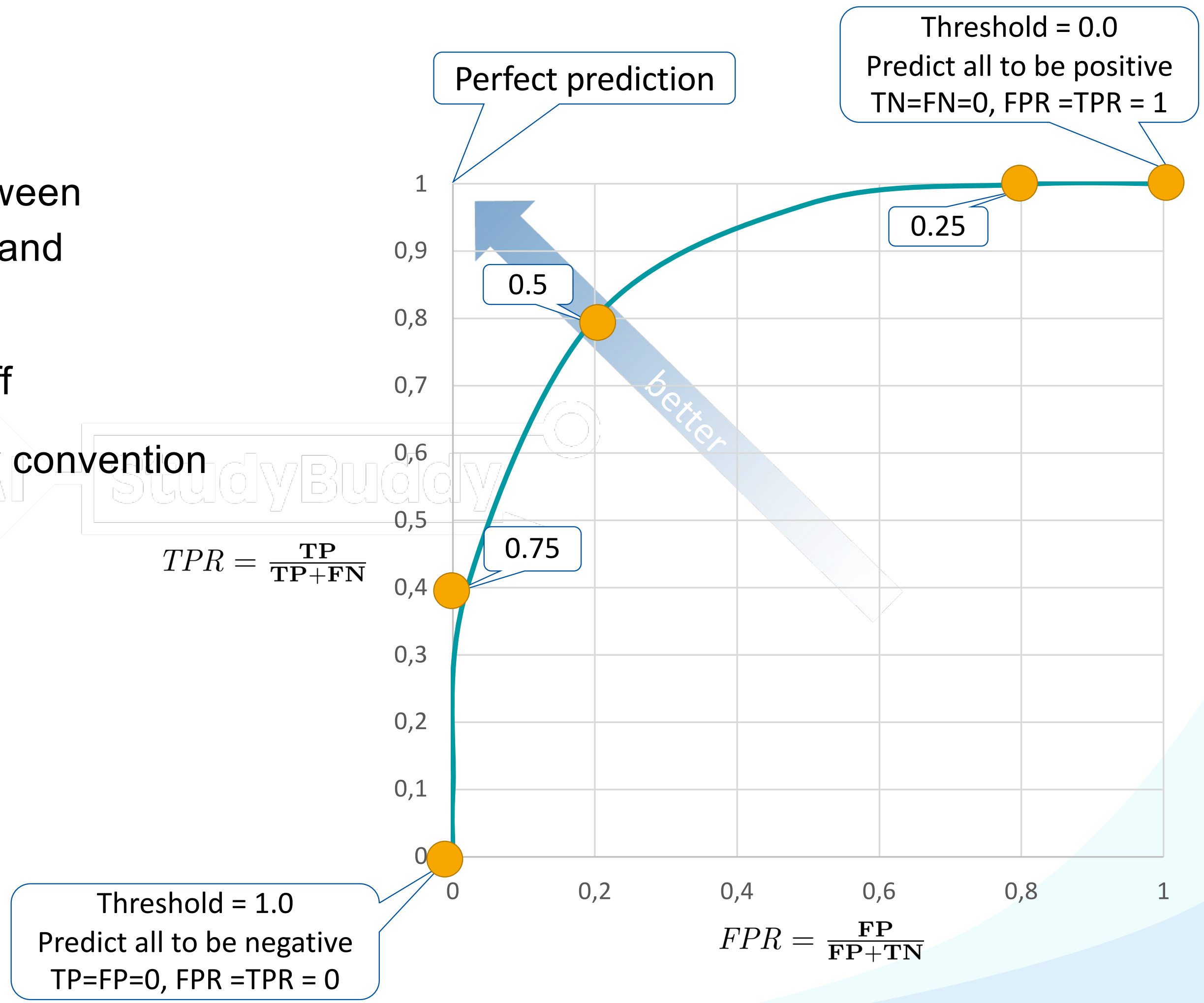
TPR = 0.4

FPR = 0



ROC Curve – Example

- Threshold controls **trade-off** between accuracy for positive predictions and accuracy for negative predictions
- ROC curve captures this trade-off
- Focus on positive (TPR, FPR) by convention



ROC Curve – Beating Random Guessing

Data set with *N* instances:
Fraction of ***q*** positive instances,
fraction of ***1-q*** negative instances

Prediction Model:
Guess positive with probability ***p*** and
negative with probability ***1-p***

ROC Curve – Beating Random Guessing

Data set with *N* instances:
Fraction of ***q*** instances is positive,
fraction of ***1-q*** instances is negative

Prediction Model:
Guess positive with probability ***p*** and
negative with probability ***1-p***

Expected Performance:

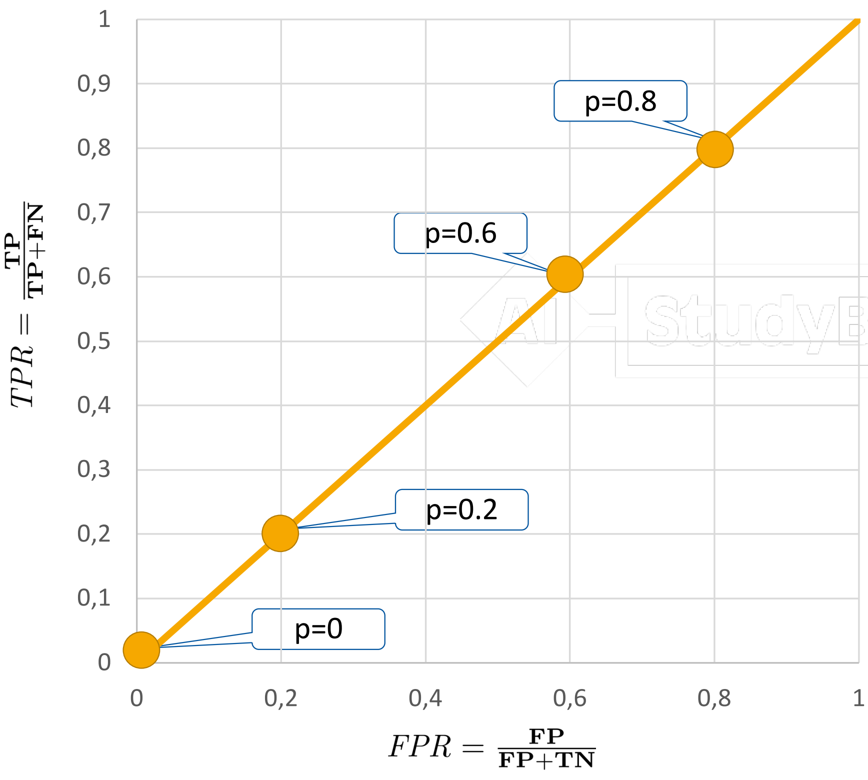
$$\mathbf{TP} = p \cdot q \cdot N$$
$$\mathbf{TN} = (1-p) \cdot (1-q) \cdot N$$
$$\mathbf{FP} = p \cdot (1-q) \cdot N$$
$$\mathbf{FN} = (1-p) \cdot q \cdot N$$

$$\mathbf{TPR} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}} = \frac{p \cdot q \cdot N}{p \cdot q \cdot N + (1-p) \cdot q \cdot N} = p$$

$$\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{TN}+\mathbf{FP}} = \frac{p \cdot (1-q) \cdot N}{(1-p) \cdot (1-q) \cdot N + p \cdot (1-q) \cdot N} = p$$

→ Performance is independent of ***q***, ***N***!

ROC Curve – Beating Random Guessing



Expected Performance:

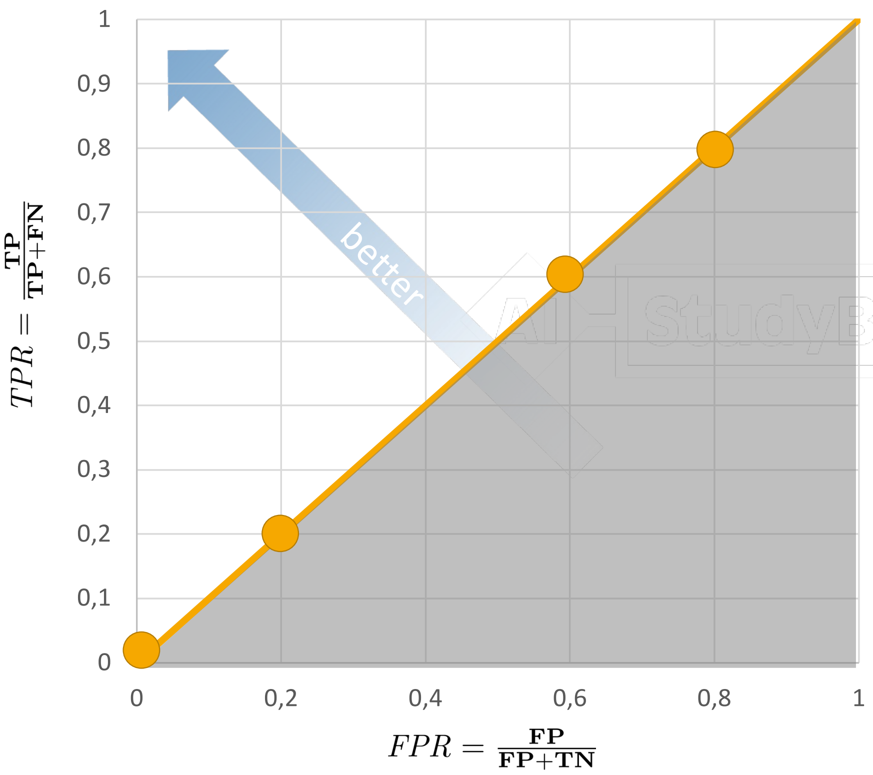
$$\mathbf{TP} = p \cdot q \cdot N$$
$$\mathbf{TN} = (1-p) \cdot (1-q) \cdot N$$
$$\mathbf{FP} = p \cdot (1-q) \cdot N$$
$$\mathbf{FN} = (1-p) \cdot q \cdot N$$

$$\mathbf{TPR} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} = \frac{p \cdot q \cdot N}{p \cdot q \cdot N + (1-p) \cdot q \cdot N} = p$$

$$\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{TN+FP}} = \frac{p \cdot (1-q) \cdot N}{(1-p) \cdot (1-q) \cdot N + p \cdot (1-q) \cdot N} = p$$

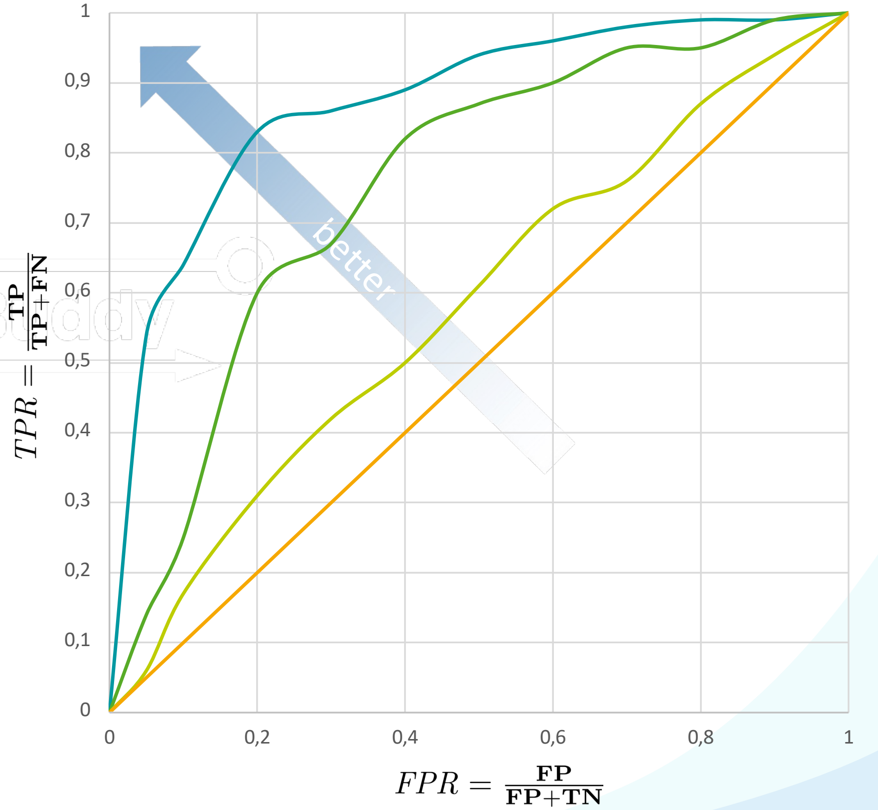
→ Performance is independent of ***q, N***!

ROC Curve – Beating Random Guessing



- Every prediction model is at least as good as random guessing
(if not, just invert the predictions)
- Thus, area under diagonal is uninteresting

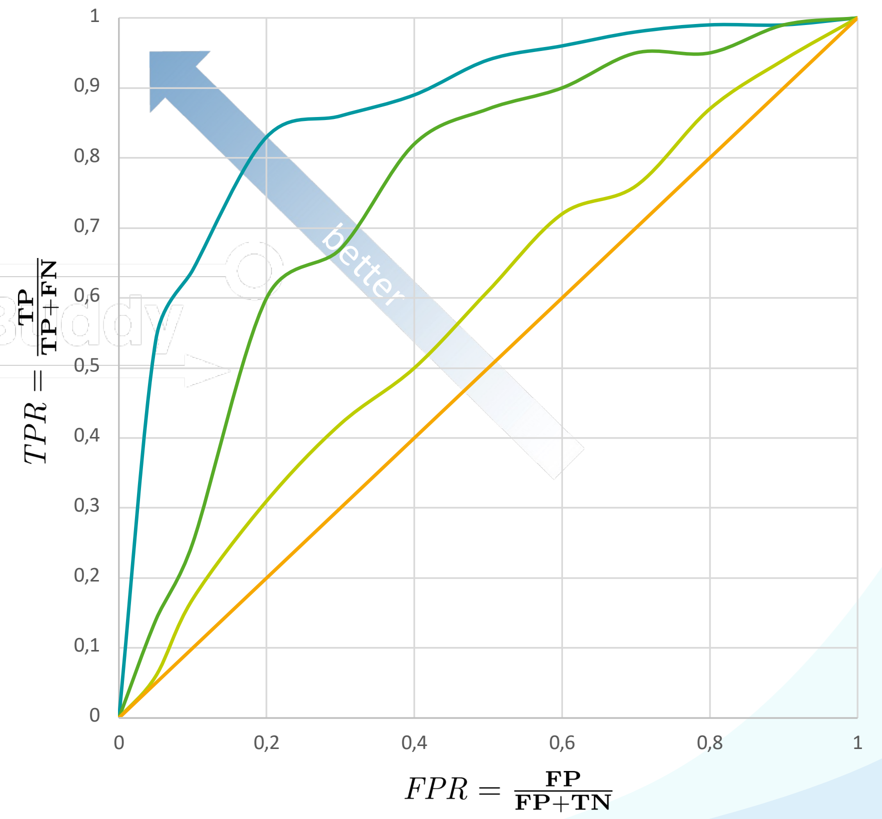
Example ROC Curves



ROC Index / AUC (Area Under the Curve)

Which model has best performance?

- ROC Index / AUC (Area Under the Curve)
- Larger area → closer to optimum
- Computable as integral of curve



ROC Index / AUC (Area Under the Curve)

Which model has best performance?

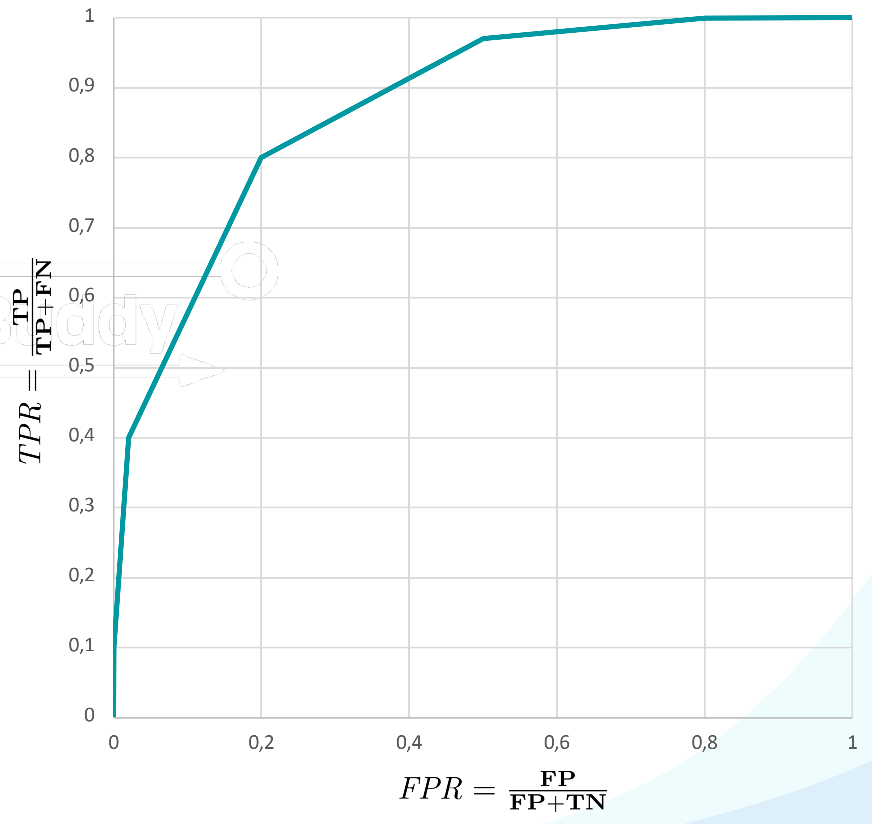
- ROC Index / AUC (Area Under the Curve)
- Larger area → closer to optimum
- Computable as integral of curve

T is the set of thresholds

FPR for the *i*th threshold

TPR for the (*i-1*)th threshold

$$\sum_{i=2}^{|T|}\left((\mathbf{FPR}_i - \mathbf{FPR}_{i-1}) \cdot \frac{(\mathbf{TPR}_i + \mathbf{TPR}_{i-1})}{2}\right)$$



ROC Index / AUC (Area Under the Curve)

Example

$$\sum_{i=2}^{|T|}\left((\mathbf{FPR}_i - \mathbf{FPR}_{i-1}) \cdot \frac{(\mathbf{TPR}_i + \mathbf{TPR}_{i-1})}{2}\right)$$

$$T = \{1.0, 0.75, 0.5, 0.25, 0.0\}$$

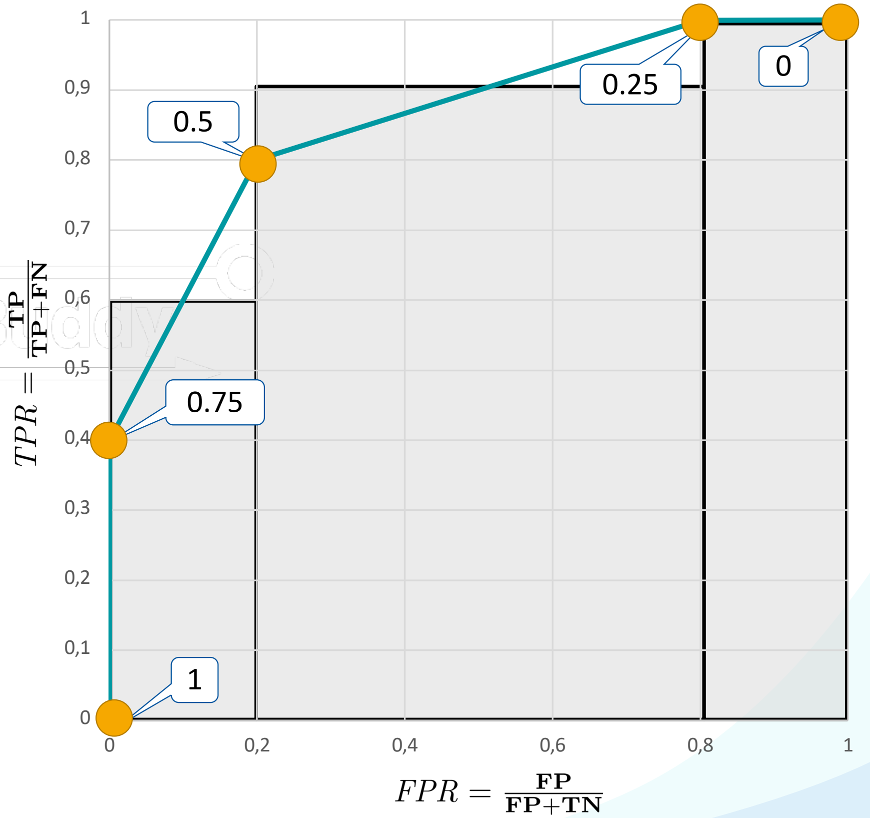
$$(0.0 - 0.0) \cdot \frac{(0.4 + 0.0)}{2}$$ 1.0 to 0.75

$$+(0.2 - 0.0) \cdot \frac{(0.8 + 0.4)}{2}$$ 0.75 to 0.5

$$+(0.8 - 0.2) \cdot \frac{(1.0 + 0.8)}{2}$$ 0.5 to 0.25

$$+(1.0 - 0.8) \cdot \frac{(1.0 + 1.0)}{2}$$ 0.25 to 0.0

$$= 0.0 + 0.04 + 0.54 + 0.2 = 0.78$$



TPS Exercise

You are <u>comparing two predictive models</u> (e.g., obtained from two different supervised learning methods).

Question:

1) How to assess performance differences?

2) What could go wrong?

Which is better?

M_1		Prediction: On Time	Prediction: Delay
Target Label	On Time	**7**	**3**
	Delay	**4**	**6**

M_2		Prediction: On Time	Prediction: Delay
Target Label	On Time	**5**	**5**
	Delay	**4**	**6**

Which is better?

M_1

		Prediction: On Time	Prediction: Delay
Target Label	On Time	5	**5**
	Delay	**4**	6

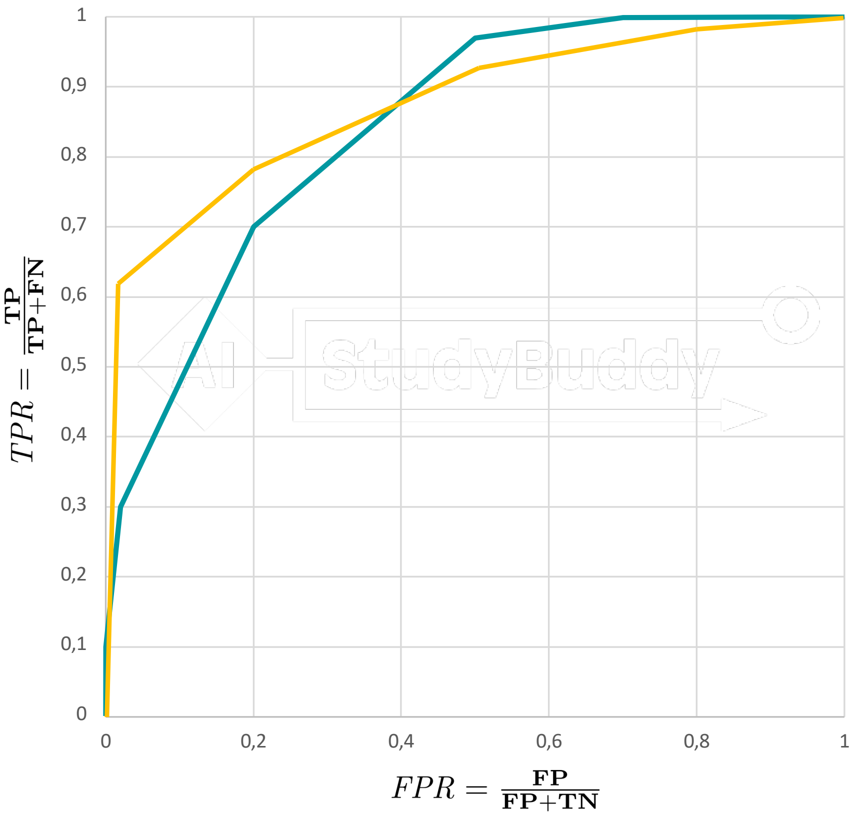
M_2

		Prediction: On Time	Prediction: Delay
Target Label	On Time	5	**4**
	Delay	**5**	6

Which is better?

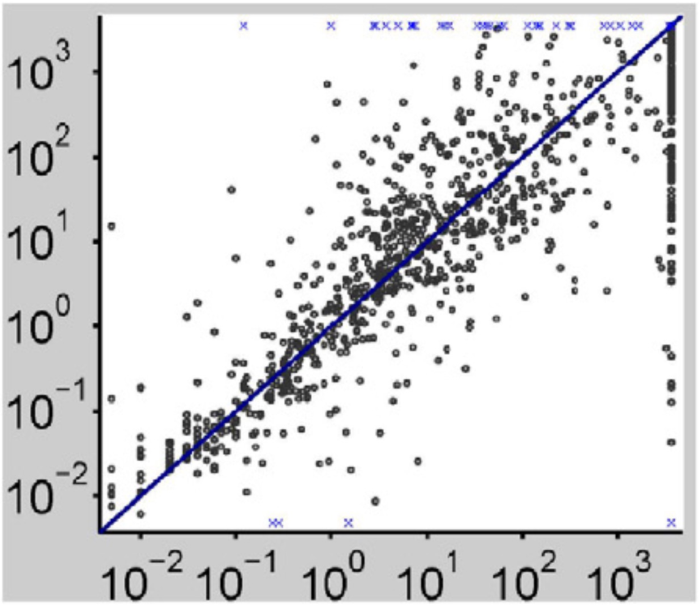
M$_1$

M$_2$

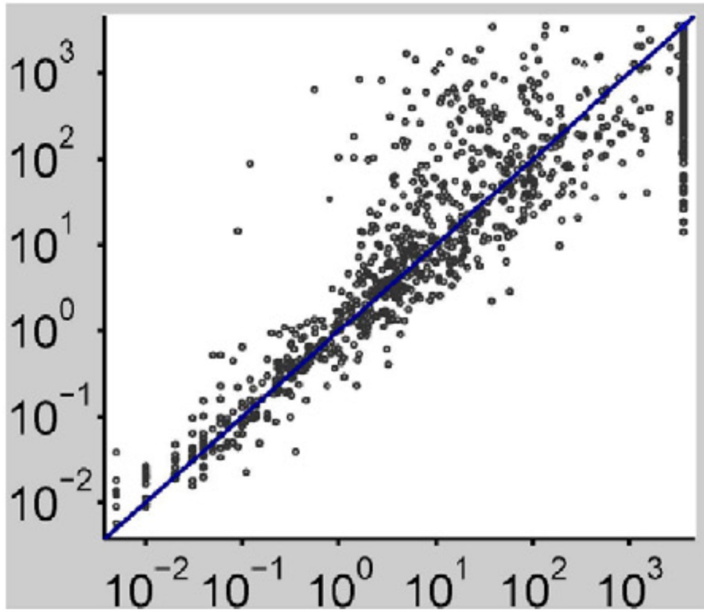


Which is better?

M_1 (Neural Network)



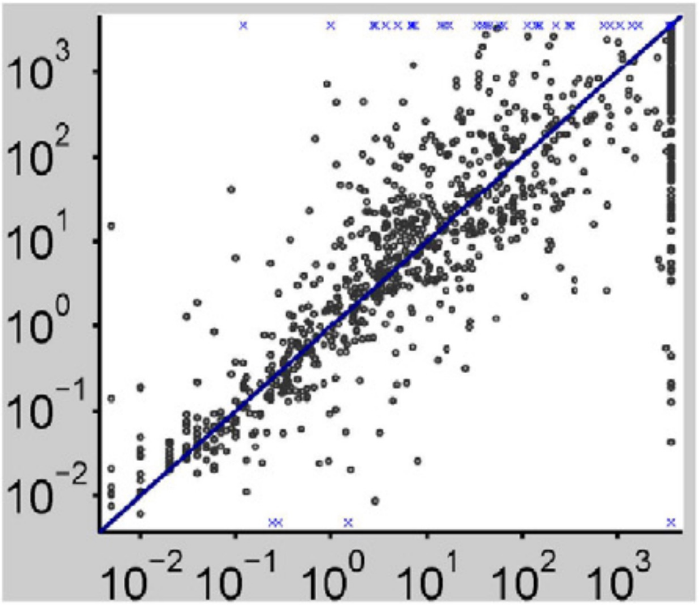
M_2 (Random Forest)



(Source: F. Hutter, L. Xu, H. Hoos, Kevin Leyton-Brown: Algorithm runtime prediction: Methods & evaluation, Artificial Intelligence 206 (2014) 79–111)

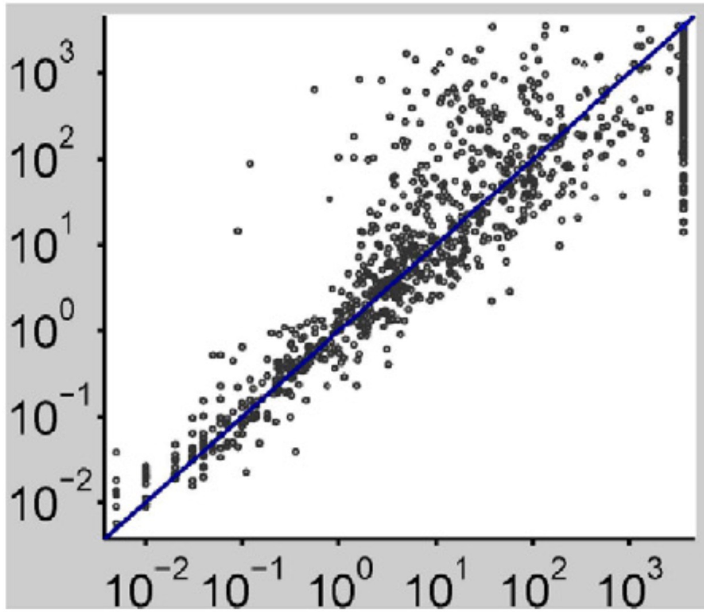
Which is better?

M_1 (Neural Network)



RMSE = 1.1

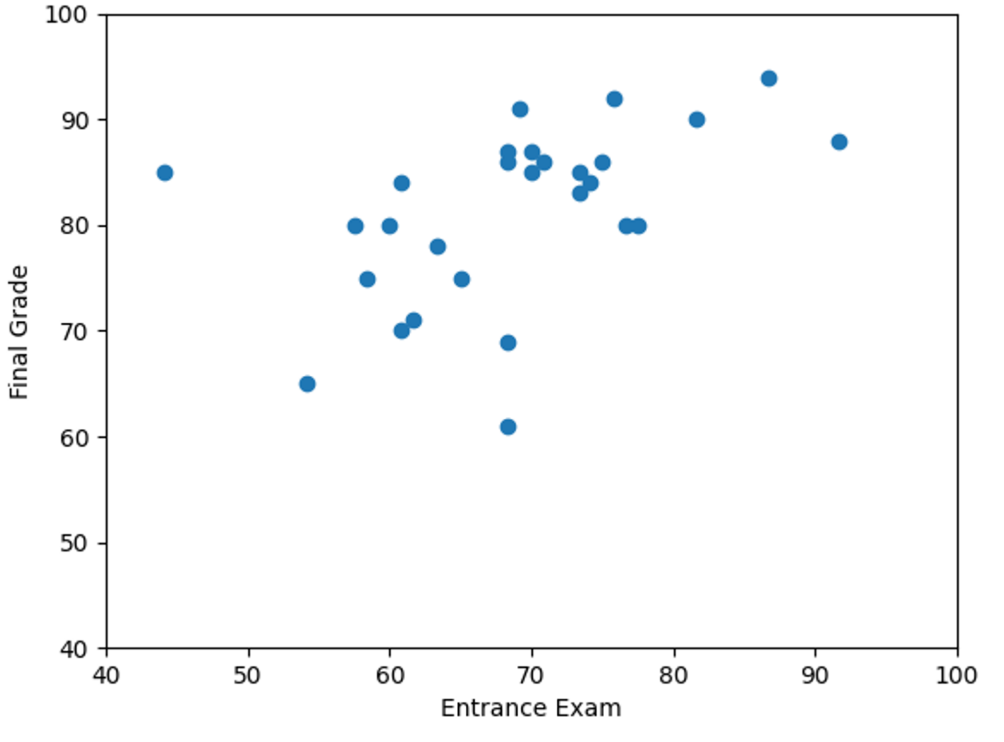
M_2 (Random Forest)



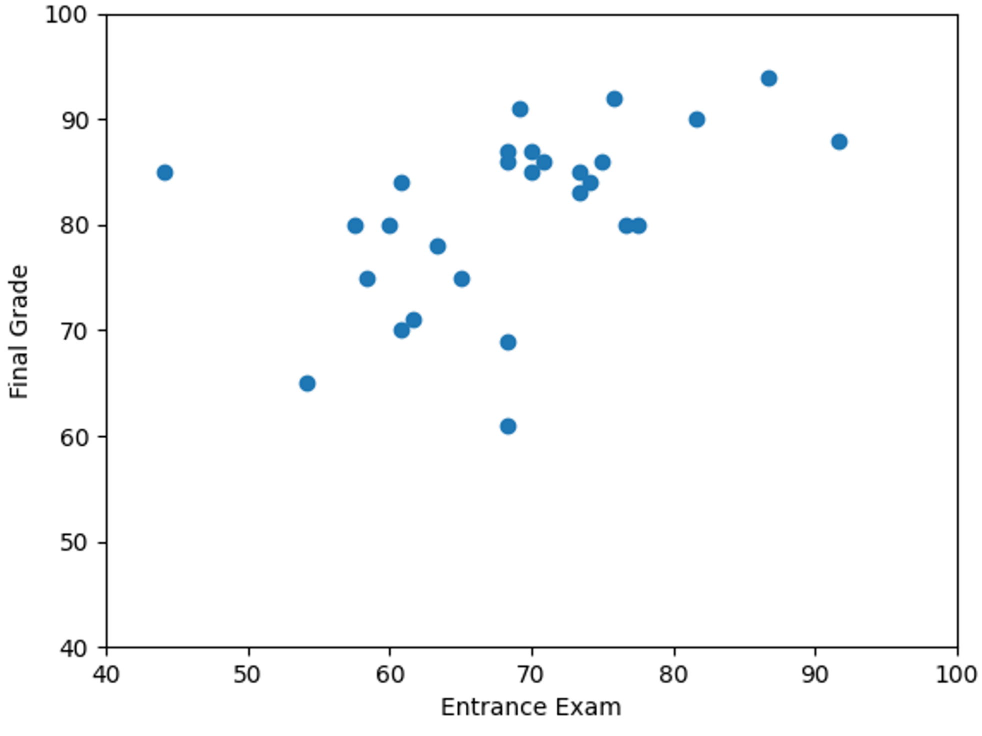
RMSE = 0.72

(Source: F. Hutter, L. Xu, H. Hoos, Kevin Leyton-Brown: Algorithm runtime prediction: Methods & evaluation, Artificial Intelligence 206 (2014) 79–111)

Assessing performance correlation

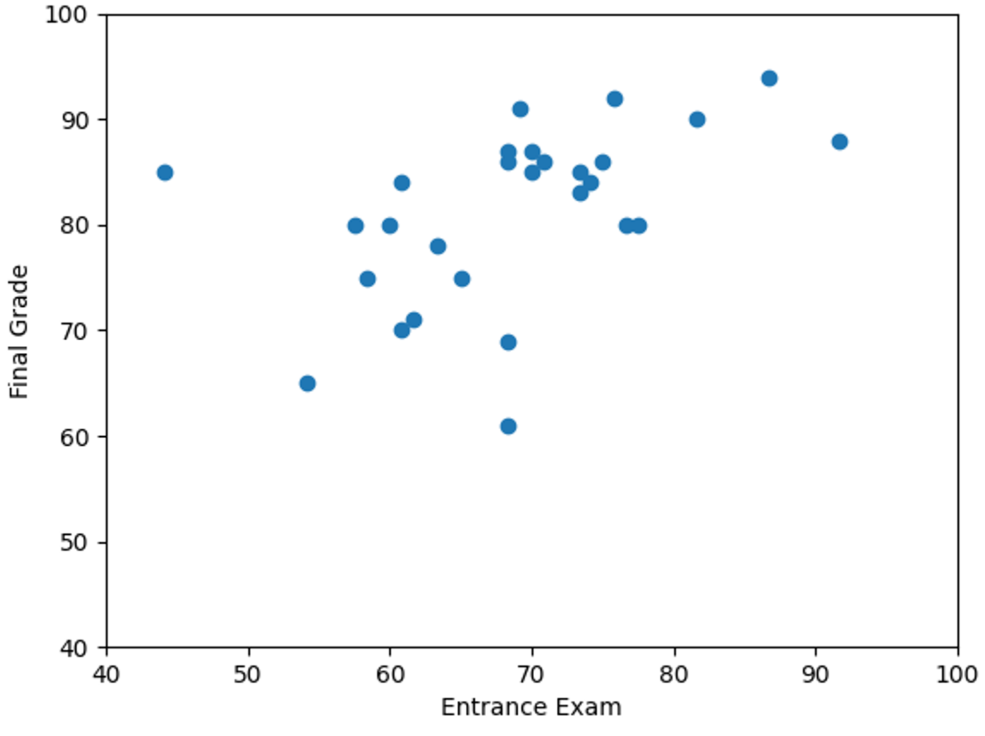


Assessing performance correlation



Pearson correlation coefficient = 0.41 (barely moderate association)

Assessing performance correlation



Pearson correlation coefficient = 0.41 (barely moderate association)
Spearman rank correlation coefficient = 0.58 (borderline strong association)

Pearson correlation coefficient:
- measures linear relationship between two sets of data
- both sets of data follow normal distribution (no outliers)

Pearson correlation coefficient:
- measures linear relationship between two sets of data
- both sets of data follow normal distribution (no outliers)

Spearman rank correlation coefficient:
- sort the data and assign ranks (1, 2, …) = rank transformation
- compute Pearson CC ➡ Spearman CC
- assumes monotonic relationship between two sets of data
- does not require normality assumption (non-parametric)

Which is better?

M_1: accuracy from *k*-fold cross-validation = 0.712

M_2 : accuracy from k-fold cross-validation = 0.721

Which is better?

M_1: accuracy from *k*-fold cross-validation = 0.712

M_2 : accuracy from k-fold cross-validation = 0.721

➡ **performance differences may be due to random effects**

➡ **assess statistical significance using statistical hypothesis testing**

Quick refresher on statistical hypothesis testing

H_0 : null-hypothesis, typically a statement of no significant effect
here: no significant performance difference between M_1, M_2

α : significance threshold = max. probability of incorrectly rejecting H_0
(incorrectly claiming significant differences = false positive = type I error)

NB: false negatives can also occur = failure to reject correct H_0
= type II error = incorrectly claiming 'equal' performance
(determined by power of the test)

p-value : (estimate) of the probability of committing a type I error

$p < \alpha$ ➡ reject H_0

➡ NB: tests rely on assumptions to work correctly

Testing for significance of performance differences

- consider performance values (e.g., accuracy) over folds (= empirical distribution) for $M_{1,}$ M_2

  ➡ $(m_{1,1}, m_{1,2}, \ldots m_{1,k})$,
  $(m_{2,1}, m_{2,2}, \ldots m_{2,k})$,

- consider pairs $(m_{1,i}, m_{2,i})$ for each fold (NB: these correspond to the points in a scatter plot, one point per fold)

- use a paired *t*-test to assess statistical significance of performance differences between $M_{1,}$ M_2 on the given test set based on the given fold, using standard significance level $\alpha = 0.05$

Caution: paired *t*-test requires normality assumption!

How can we know whether performance data over folds follows a normal distribution?

What to do if it doesn’t?

Caution: paired *t*-test requires normality assumption!

How can we know whether performance data over folds follows a normal distribution?

➡ check QQ plot or use normality test (e.g., Shapiro–Wilk)

What to do if it doesn't?

➡ use a non-parametric test, e.g., Wilcoxon Signed-Ranks Test

Comparing two predictive models

- assess performance of each model individually
- analyse performance correlation
  - classification: overlap/differences in FP, FN, misclassifications
  - regression: scatter plot, correlation coefficient
- use appropriate statistical tests

Don't…

- limit analysis to single performance metric
- limit correlation to single number
  (in particular: standard = Pearson correlation coefficient)

TPS Exercise

You are using a <u>randomised</u> supervised ML procedure to train a predictive model.

Questions:

1) **How to assess the training procedure?**

2) **What could go wrong?**

Evaluating randomised supervised ML procedures

- perform *p* independent runs ($p \geq 2$)
  ➡ *p* models
- assess & compare performance of all *p* models
- inspect / analyse distribution of performance metrics, multiple performance metrics

Don't…

- just aggregate performance over all *p* models
- limit analysis to single performance metric
- report only the best result! (No cherry picking!)

TPS Exercise

You have trained a predictive model using supervised ML, you've carefully assessed its performance and deployed it in practice.

Questions:

What could happen to invalidate earlier performance assessment?

TPS Exercise

You have trained a predictive model using supervised ML, you've carefully assessed its performance and deployed it in practice.

Questions:

What could happen to invalidate earlier performance assessment?

➡ **Performance degradation due to <u>concept drift</u> (violation of supervised learning assumption)**

Key concepts covered today:

- performance measures for multi-class classification (multinomial prediction targets)
- performance measures for regression models (numerical prediction targets)
- ROC curves, AUC
- randomness in the training procedure
- comparative performance analysis
- Spearman’s rank correlation coefficient
- statistical significance tests

Learning Goals

At the end of this module, students should be able to

- assess the quality of a model obtained from a supervised machine learning method using widely accepted methods, including standard performance metrics, confusion matrices, ROC curves
- demonstrate understanding and working knowledge of the problems that can occur when using supervised learning procedures and the models obtained from them
- explain when and why it is important to distinguish between training, validation and testing data
- explain standard validation techniques, including *k*-fold and leave-one-out cross-validation
- assess performance differences using appropriate statistical techniques
- explain the problems that can arise from unbalanced data sets and demonstrate understanding as well as working knowledge of methods for addressing these problems