Visual Computing Institute

RWTH AACHEN UNIVERSITY

Elements of Machine Learning & Data Science

Winter semester 2025/26

Lecture 22 – Evaluation II

26.01.2026

Prof. Bastian Leibe

slides by Prof. Holger Hoos and Prof. Wil van der Aalst

Announcement

Lecture Evaluation

- Please fill out the lecture evaluation form
  - *The evaluation will be open until 27.01.2026*

- We are very interested in your feedback!
  - Tell us what you liked,
    but also what could still be improved.

Empirical Analysis and Performance Evaluation Topics

15. Data Quality and Preprocessing
16. Responsible Data Science
17. **Evaluation**
18. Performance Optimization

Key Questions

- **How good is an ML model?**
  - *Is it "fit for use" (i.e., good enough for deployment)?*
  - *What are its strengths and weaknesses?*
  - *Might anything have gone wrong during training?*

Empirical Analysis and Performance Evaluation Topics

15. Data Quality and Preprocessing
16. Responsible Data Science
17. **Evaluation**
18. Performance Optimization

Key Questions

- **How good is an ML model?**
  - *How do we assess whether it is "fit for use" (i.e., good enough for deployment)?*
  - *How do we assess its strengths and weaknesses?*
  - *How do we detect if anything has gone wrong during training?*

Empirical Analysis and Performance Evaluation Topics

15. Data Quality and Preprocessing
16. Responsible Data Science
17. **Evaluation**
18. Performance Optimization

Key Questions

- **How good could an ML model be?**
  - *Are we using the best possible ML method / model?*
  - *Have we configured and trained it in the best possible way?*
  - *Can we further improve performance?*

Empirical Analysis and Performance Evaluation Topics

15. Data Quality and Preprocessing
16. Responsible Data Science
17. **Evaluation**
18. Performance Optimization

Key Questions

- **How good could an ML model be?**
  - *How can we ensure we are using a good ML method / model?*
  - *How can we configure and train it for optimized performance?*
  - *How can we further improve performance?*

Key Questions for Evaluation

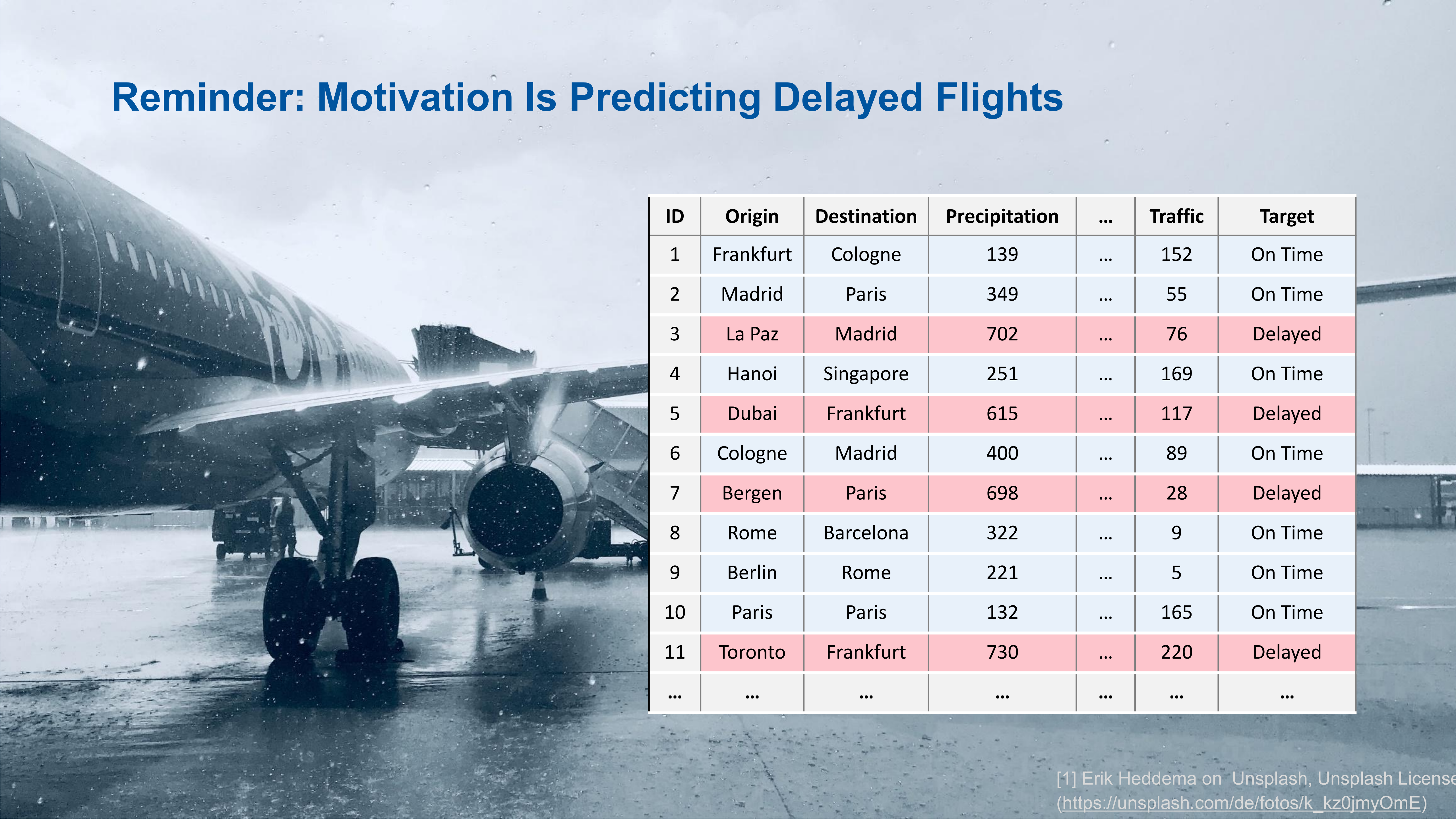
1. **How good is an ML model?**

2. How good could an ML model be?

Key Concepts Covered Last Week

- 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* ⇐ *We still need to cover that one…*

Reminder: Motivation Is Predicting Delayed Flights



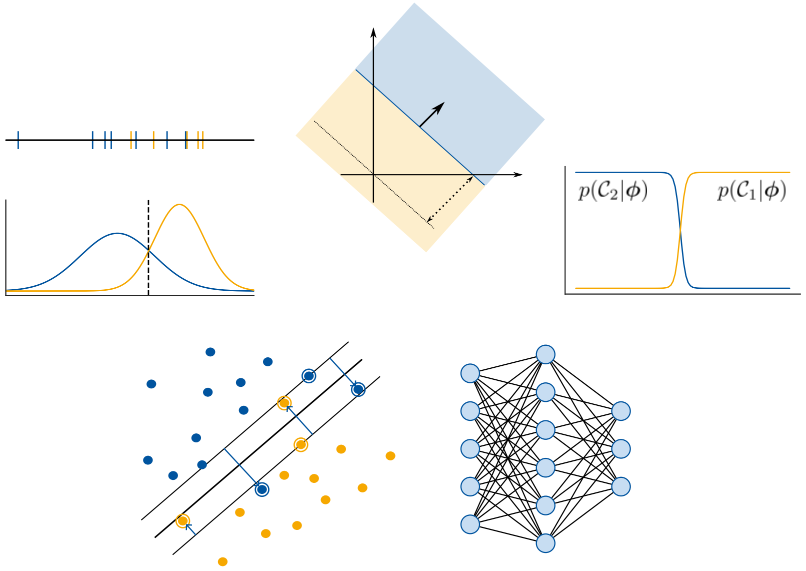
ID	Origin	Destination	Precipitation	…	Traffic	Target
1	Frankfurt	Cologne	139	…	152	On Time
2	Madrid	Paris	349	…	55	On Time
3	La Paz	Madrid	702	…	76	Delayed
4	Hanoi	Singapore	251	…	169	On Time
5	Dubai	Frankfurt	615	…	117	Delayed
6	Cologne	Madrid	400	…	89	On Time
7	Bergen	Paris	698	…	28	Delayed
8	Rome	Barcelona	322	…	9	On Time
9	Berlin	Rome	221	…	5	On Time
10	Paris	Paris	132	…	165	On Time
11	Toronto	Frankfurt	730	…	220	Delayed
…	…	…	…	…	…	…

[1] Erik Heddema on Unsplash, Unsplash License
(https://unsplash.com/de/fotos/k_kz0jmyOmE)

Practical Aspects

- You have used supervised learning to train a predictive model
  - And you have computed a confusion matrix based on the predictions on a given set of data

- Question: *What is worse – Predicting a flight to be delayed and having it arrive on time, or predicting it to be on time and find it to be delayed?*



	Positive Prediction	Negative Prediction
Positive Target Label	**TP=6**	**FN=3**
Negative Target Label	**FP=2**	**TN=9**

Similar problems occur in many real scenarios…

- Does the self-driving car need to stop?
- Should the patient be tested for a severe disease?

➜ **FP**s and **FN**s can have (very) different cost!

Image source: [2] Matt C on Unsplash, Unsplash License, (https://unsplash.com/de/fotos/ubHRHM37ddE)

Profit (Utility) Matrix

Example Flight Classification

- Correctly inform customers about a delay:
  - Customers can plan to arrive later
  - ***A little*** ‘profit’ from less unhappy customers
- Incorrectly inform customers about a delay:
  - Customers arrive too late
  - ***Huge*** loss of ‘profit’ by unnecessarily delayed flight

- Incorrectly predicting ‘Delayed’ (FN) costs more than incorrectly predicting ‘On Time’ (FP)

Profit Matrix		Prediction	
		On Time	Delay
Target Label	On Time	**0**	**-80**
	Delay	**-10**	**20**

Profit (Utility) Matrix

M_1		Prediction	
		On Time	Delay
Target Label	On Time	6	3
	Delay	2	9

M_2		Prediction	
		On Time	Delay
Target Label	On Time	5	0
	Delay	9	6

Profit Matrix		Prediction	
		On Time	Delay
Target Label	On Time	0	-80
	Delay	-10	20

Profit (Utility) Matrix

M_1		Prediction	
		On Time	Delay
Target Label	On Time	**6**	**3**
	Delay	**2**	**9**

M_2		Prediction	
		On Time	Delay
Target Label	On Time	**5**	**0**
	Delay	**9**	**6**

Profit Matrix		Prediction	
		On Time	Delay
Target Label	On Time	**0**	**-80**
	Delay	**-10**	**20**

M_1		Prediction	
		On Time	Delay
Target Label	On Time	**0**	**-240**
	Delay	**-20**	**180**
	Profit	**-80**	

M_2		Prediction	
		On Time	Delay
Target Label	On Time	**0**	**0**
	Delay	**-90**	**120**
	Profit	**30**	

Profit (Utility) Matrix

M_1		Prediction	
		On Time	Delay
Target Label	On Time	6	3
	Delay	2	9

M_2		Prediction	
		On Time	Delay
Target Label	On Time	5	0
	Delay	9	6

Profit Matrix		Prediction	
		On Time	Delay
Target Label	On Time	0	-80
	Delay	-10	20

$$\begin{aligned} \mathit{profit} &= \mathbf{FP} \cdot \mathbf{FP}_{\text{profit}} + \mathbf{TP} \cdot \mathbf{TP}_{\text{profit}} \\ &+ \mathbf{FN} \cdot \mathbf{FN}_{\text{profit}} + \mathbf{TN} \cdot \mathbf{TN}_{\text{profit}} \end{aligned}$$

M_1		Prediction	
		On Time	Delay
Target Label	On Time	0	-240
	Delay	-20	180
	Profit	-80	

M_2		Prediction	
		On Time	Delay
Target Label	On Time	0	0
	Delay	-90	120
	Profit	30	

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)

Preparation for Today

Let’s address the first question:

- **How to assess predictive models for multi-class classification?**
  (> 2 target classes, *e.g.*, on time, mildly delayed, severely delayed)
  - *Let’s collect your ideas here…*
  - *What makes this problem different? What would still work, what would require changes?*

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

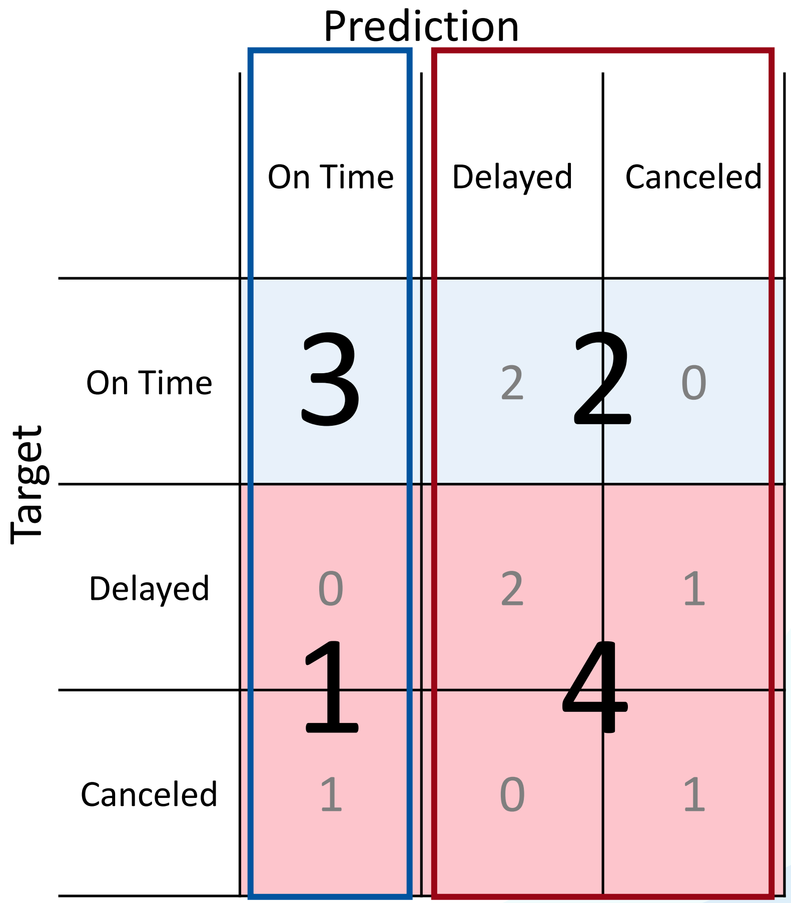
For each possible target label value:

- 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



		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

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

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

		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

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		
		On Time	Delayed	Canceled
Target	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

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

recall of the *k*th label value

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

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

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

Preparation for Today

Now let’s move on to the second question:

- **How to assess predictive models for regression tasks?**
  (predictions = numbers, *e.g.*, minutes of delay)
  - *Let’s collect your ideas here…*
  - *What makes **this** problem different? What would still work, what would **now** require changes?*

Reminder: Error Functions

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

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)

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

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

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

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

- 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

Coefficient of Determination (R^2) – Example

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

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

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

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

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

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

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

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

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

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

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

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

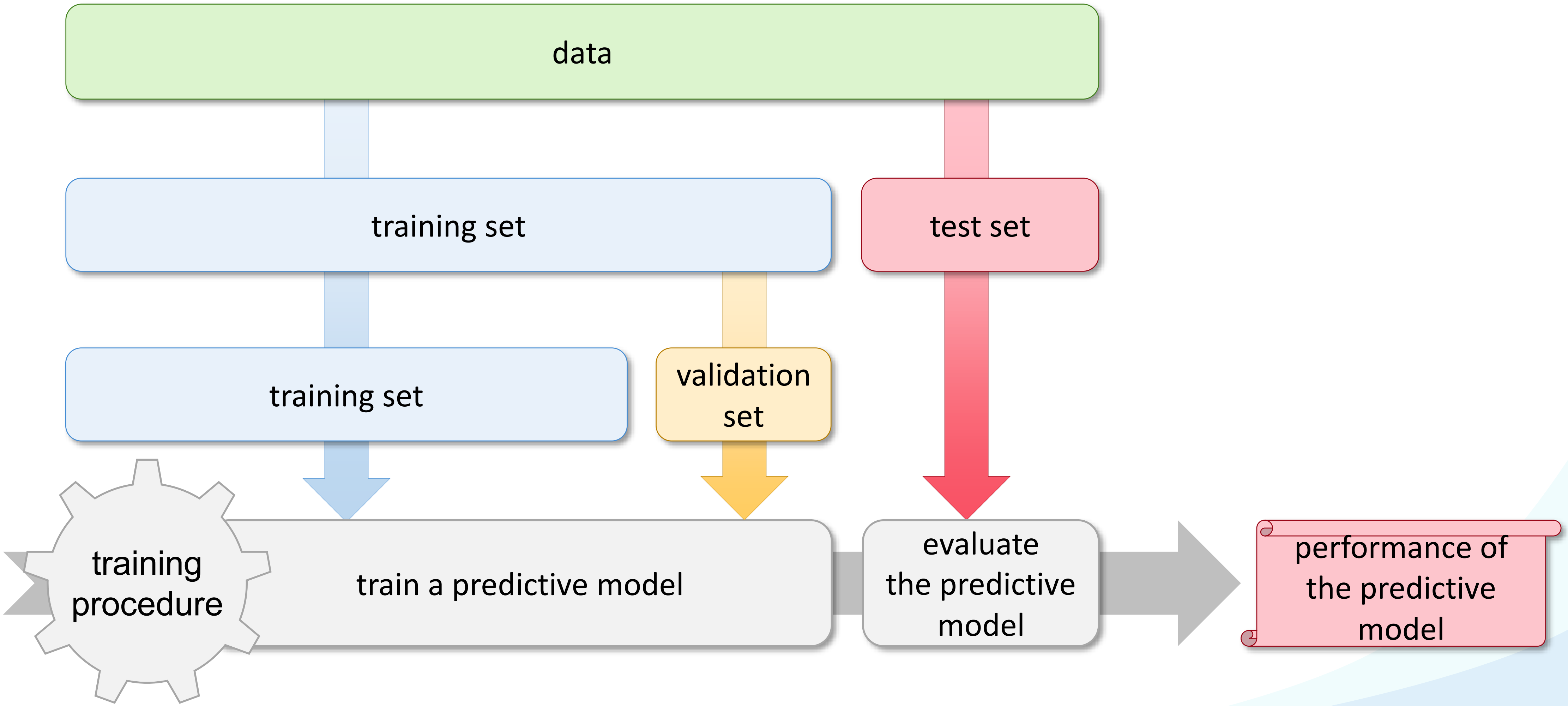
$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}} = 1 - \frac{186}{441} \approx 0.42$$

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

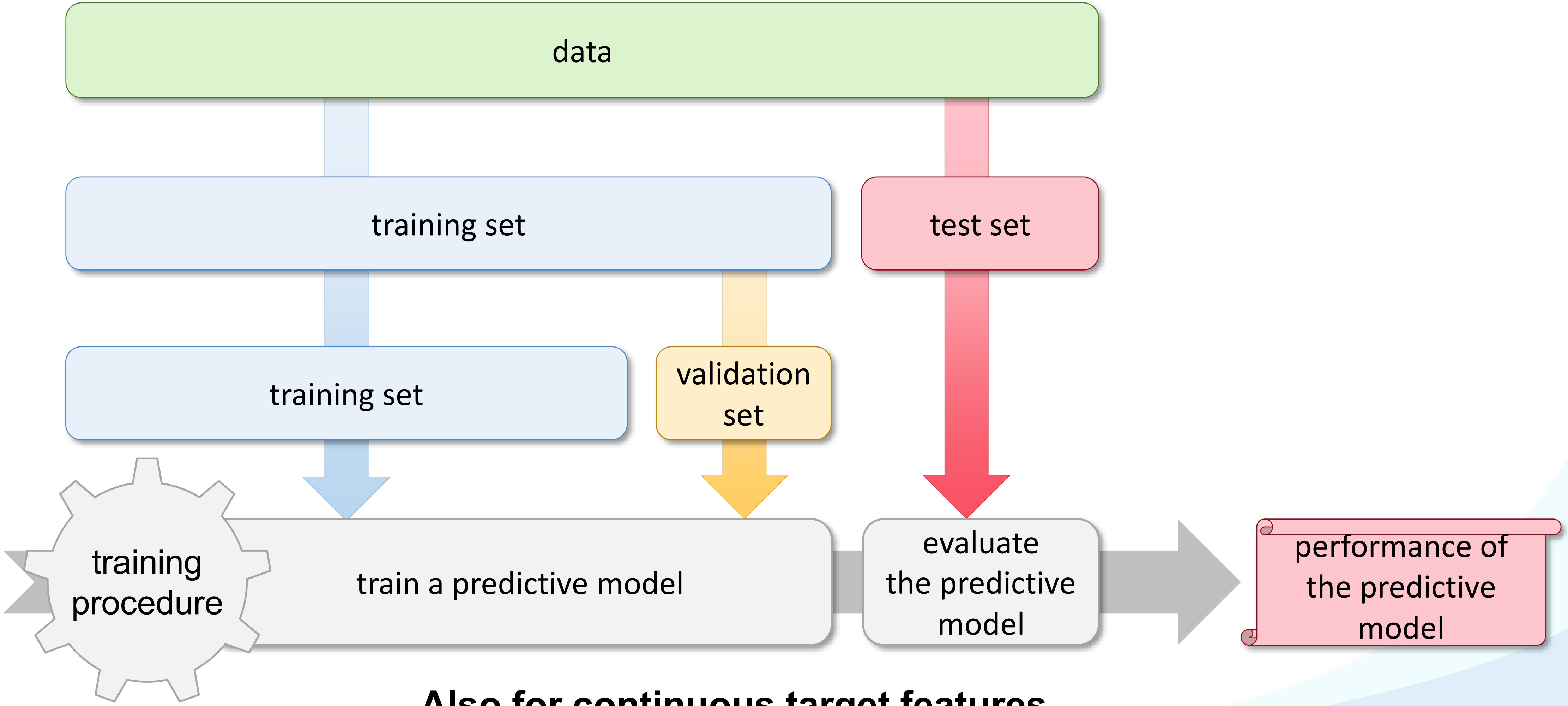
$$\text{total sum of squares} = \frac{1}{2}\sum_{i=1}^{N}(t_i - \bar{t})^2 = \frac{1}{2}\cdot 882 = 441$$

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



data

training set

test set

training set

validation set

training procedure

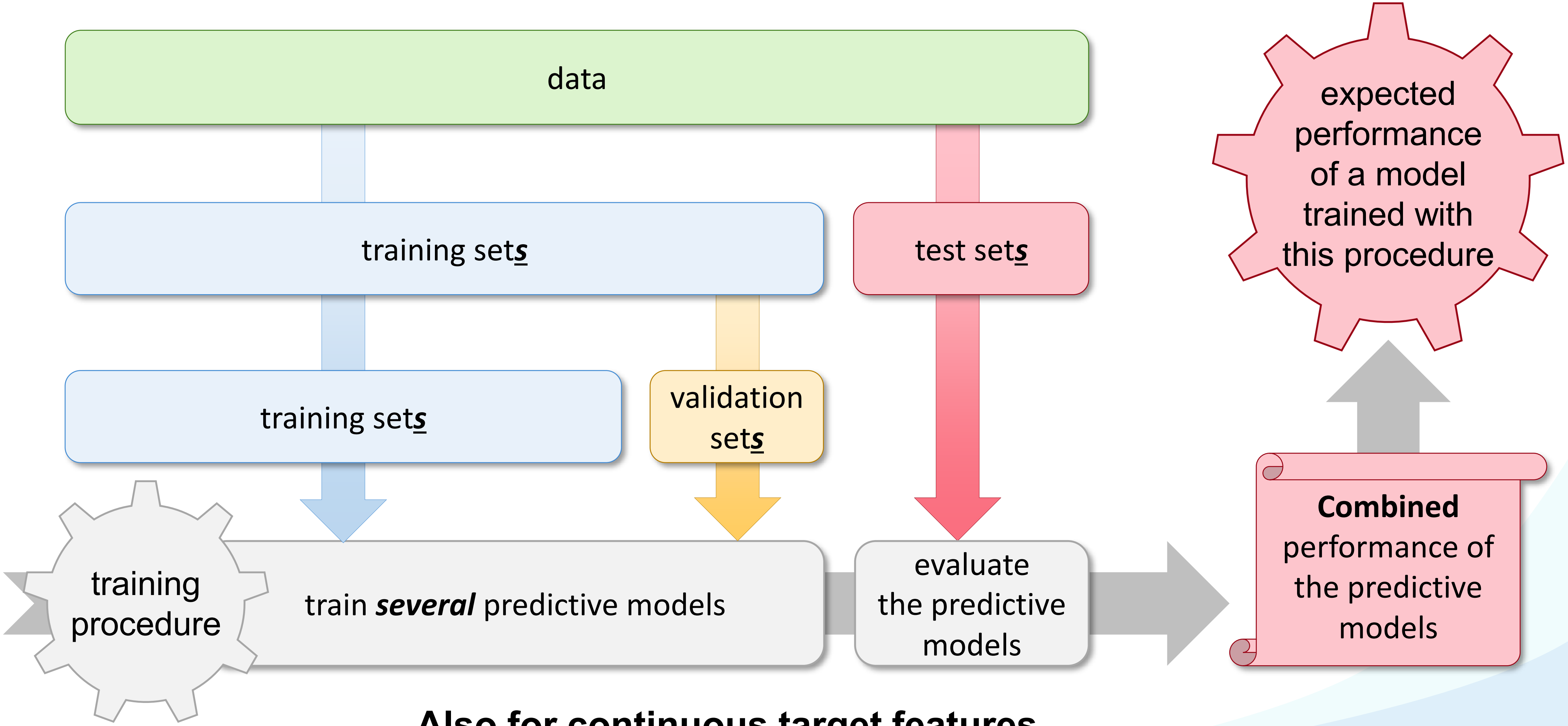
train a predictive model

evaluate the predictive model

performance of the predictive model

Also for continuous target features

Reminder (2)



data

training sets

test sets

expected performance of a model trained with this procedure

training sets

validation sets

training procedure

train *several* predictive models

evaluate the predictive models

Combined performance of the predictive models

Also for continuous target features

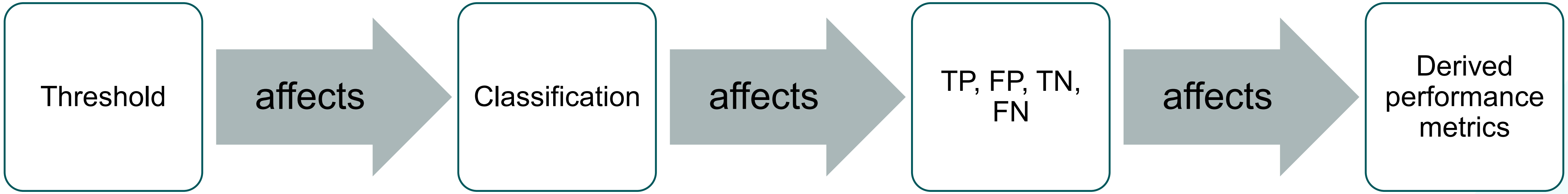
Assessing Model Quality

Let’s consider a variant of the assessment problem:

- **You have used supervised ML to train a predictive model for a binary classification problem. The model gives you a numerical prediction score between 0 and 1.**
- **Question: How to assess the quality of the model?**
  - *Let’s again collect your ideas here…*
  - *Why would it be useful to have such a model?*
  - *What is the added complexity here? What changes do we need to consider?*

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

	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

Prediction

	0.5	On Time	Delayed
Target	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

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

Prediction

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

TPR = 0.8

FPR = 0.2

Prediction

	0.75	On Time	Delayed
Target	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

	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

Prediction

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

TPR = 0.8

FPR = 0.2

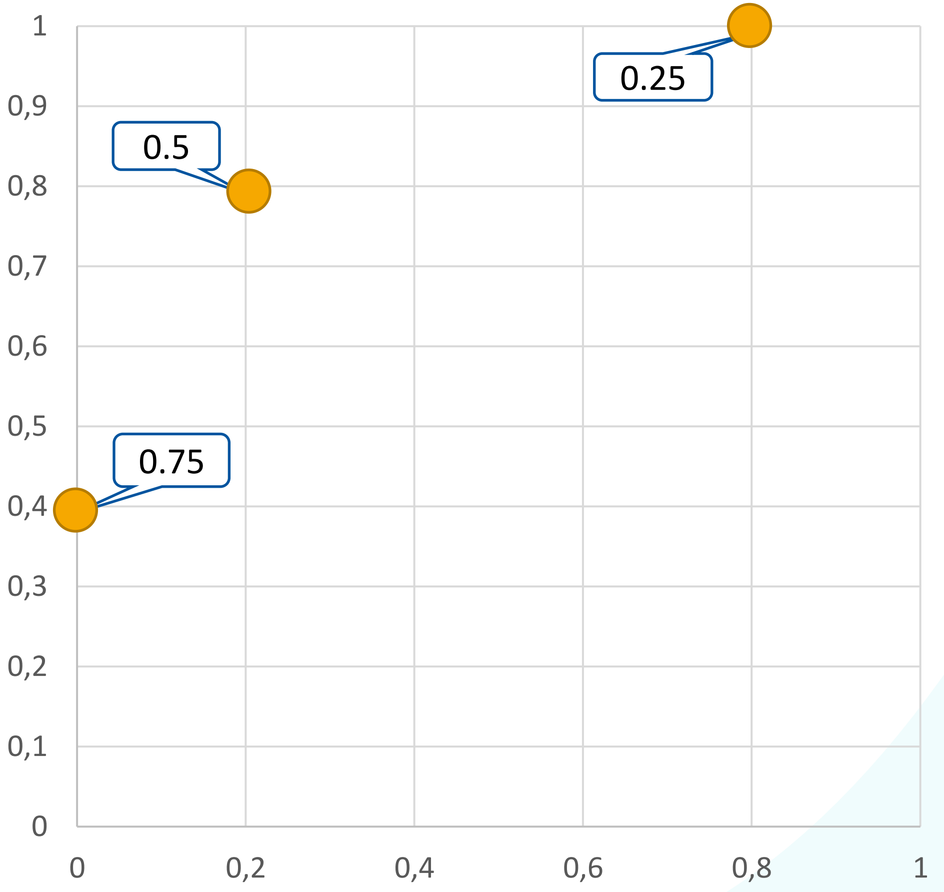
Prediction

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

TPR = 0.4

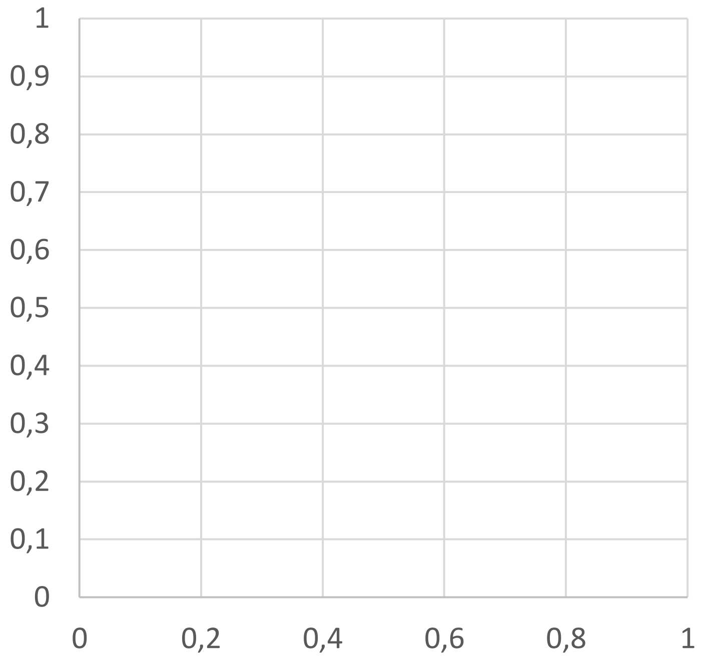
FPR = 0

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



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

Understanding ROC Curves



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

- **Questions:**
  1. **What does an ideal ROC curve look like?**
  2. **What about the worst-case ROC curve?**

- *Let's again collect your ideas here…*

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

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

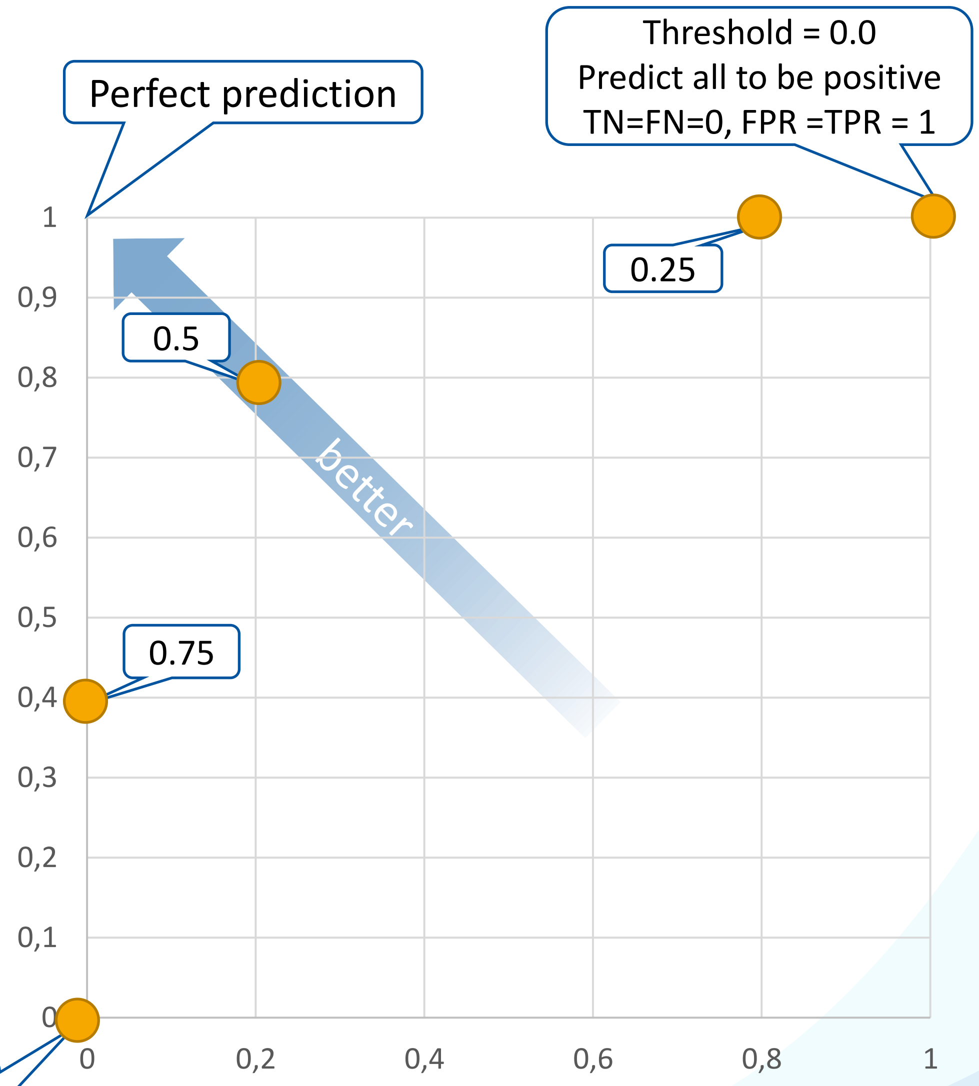
Prediction

Target

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

TPR = 0.4

FPR = 0



Threshold = 1.0
Predict all to be negative
TP=FP=0, FPR =TPR = 0

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

ROC Curve – 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

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

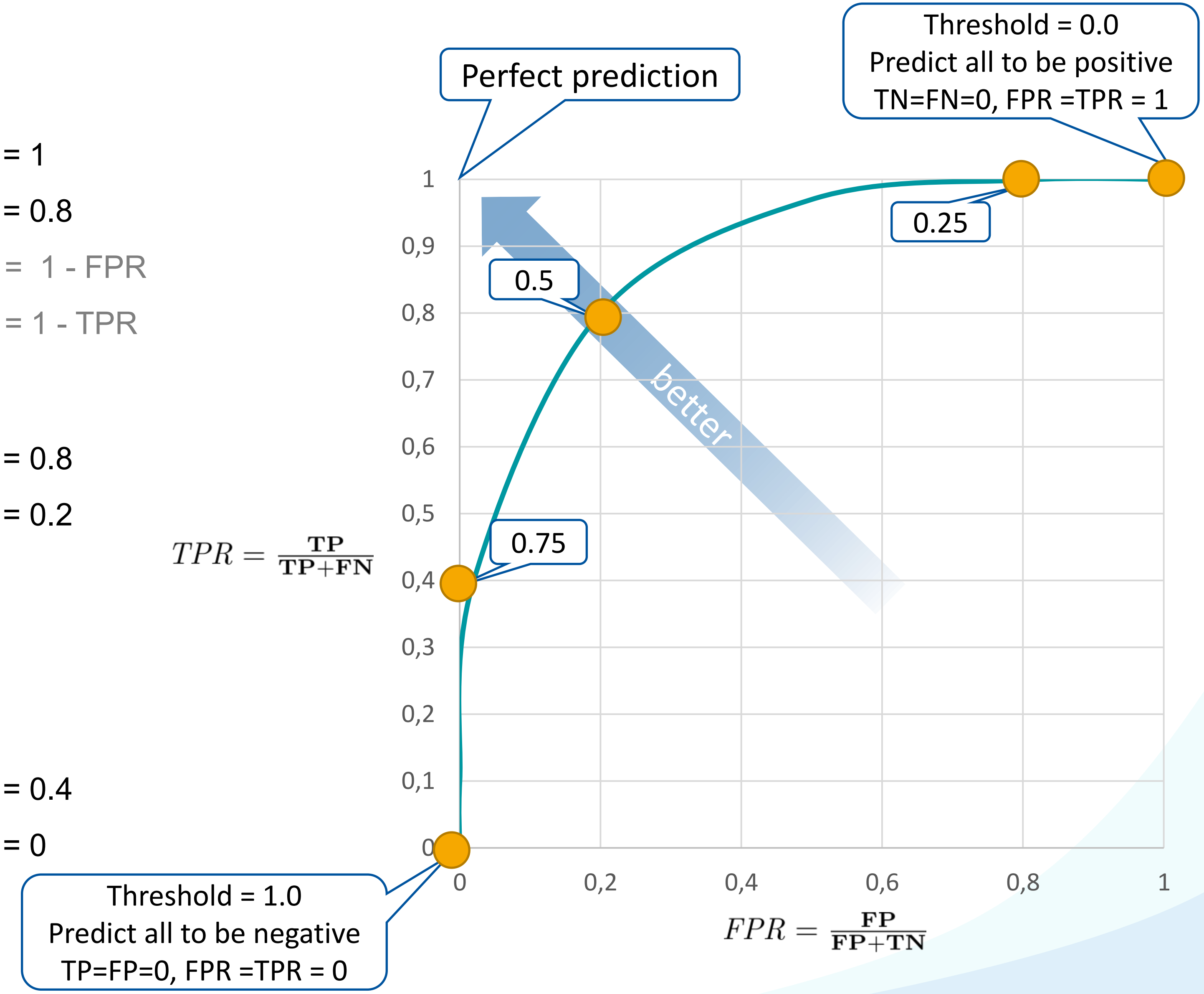
TPR = 0.8

FPR = 0.2

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

TPR = 0.4

FPR = 0



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

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

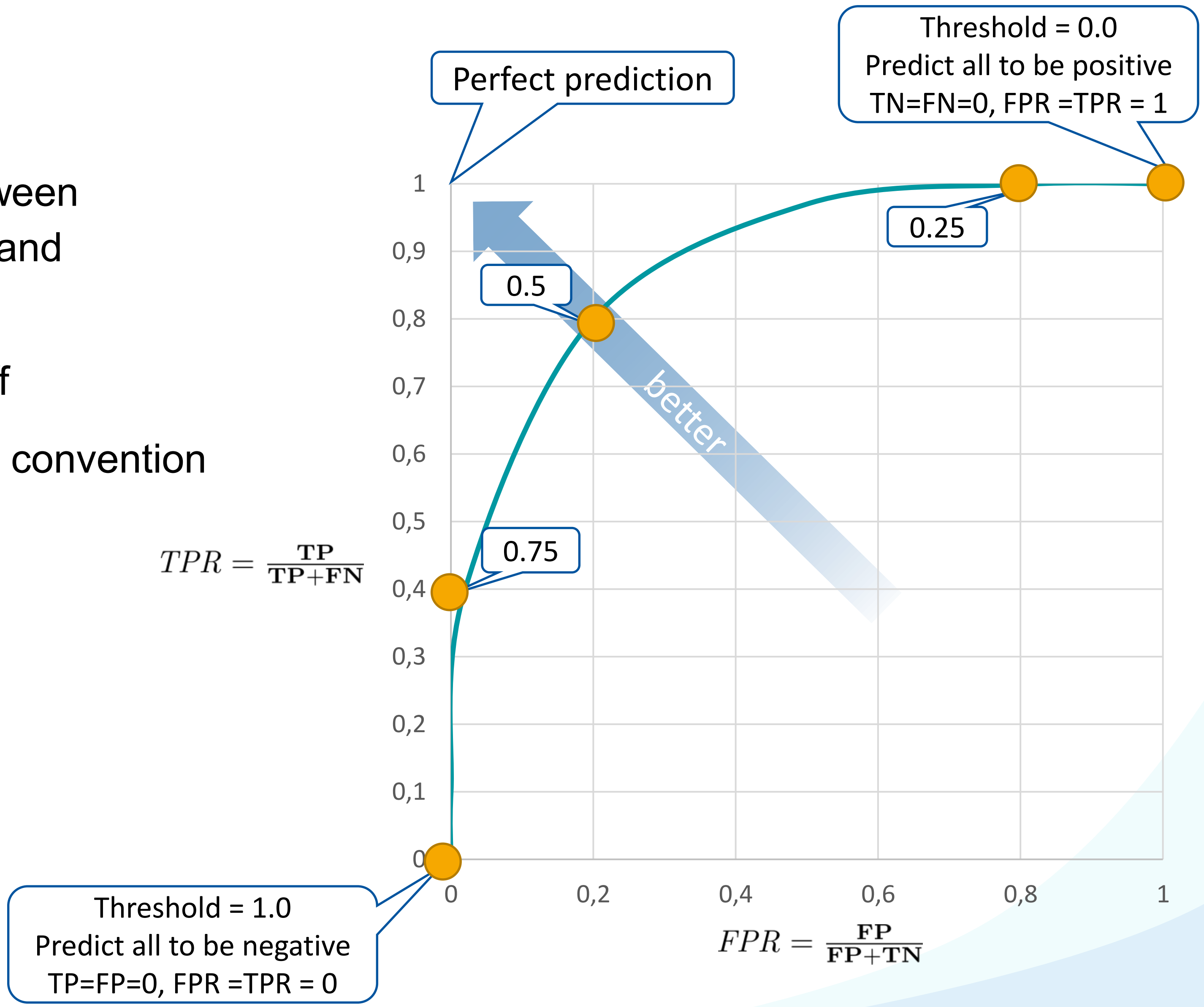
Perfect prediction

Threshold = 0.0
Predict all to be positive
TN=FN=0, FPR =TPR = 1

Threshold = 1.0
Predict all to be negative
TP=FP=0, FPR =TPR = 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



Perfect prediction

Threshold = 0.0
Predict all to be positive
TN=FN=0, FPR =TPR = 1

Threshold = 1.0
Predict all to be negative
TP=FP=0, FPR =TPR = 0

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

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

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***,
negative with probability ***1-p***

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***,
negative with probability ***1-p***

Expected Performance:

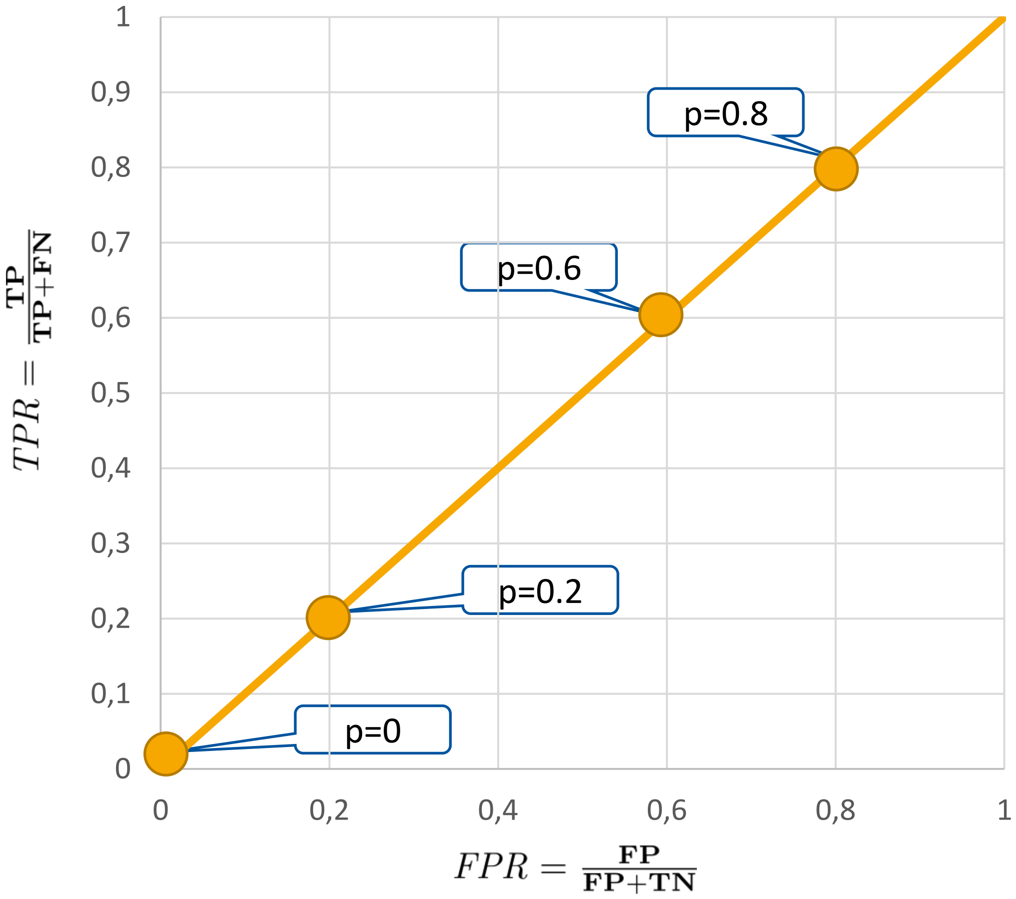
$$\mathbf{TP} = p \cdot q \cdot N$$
$$\mathbf{TN} = (1-p) \cdot (1-q) \cdot N$$
$$\mathbf{FP} = p \cdot (1-q) \cdot N$$
$$\mathbf{FN} = (1-p) \cdot q \cdot N$$

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

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

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

ROC Curve – Beating Random Guessing



Expected Performance:

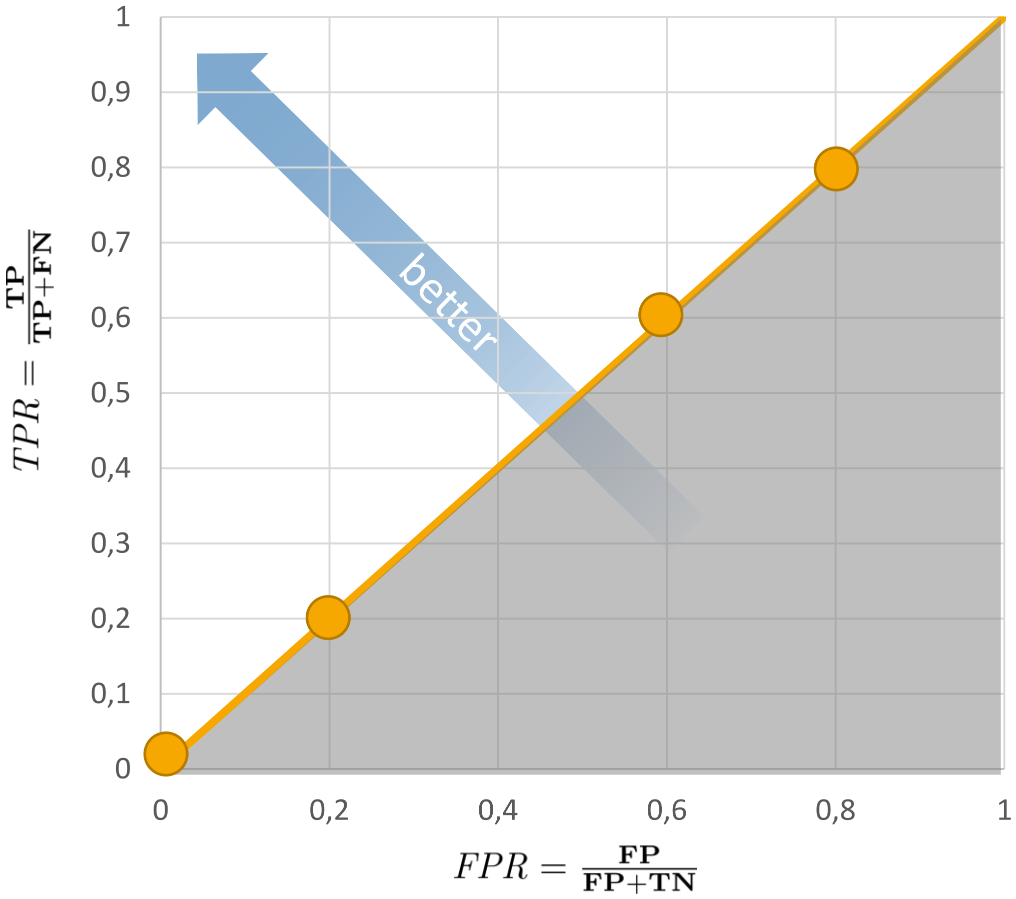
$$\mathbf{TP} = p \cdot q \cdot N$$
$$\mathbf{TN} = (1-p) \cdot (1-q) \cdot N$$
$$\mathbf{FP} = p \cdot (1-q) \cdot N$$
$$\mathbf{FN} = (1-p) \cdot q \cdot N$$

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

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

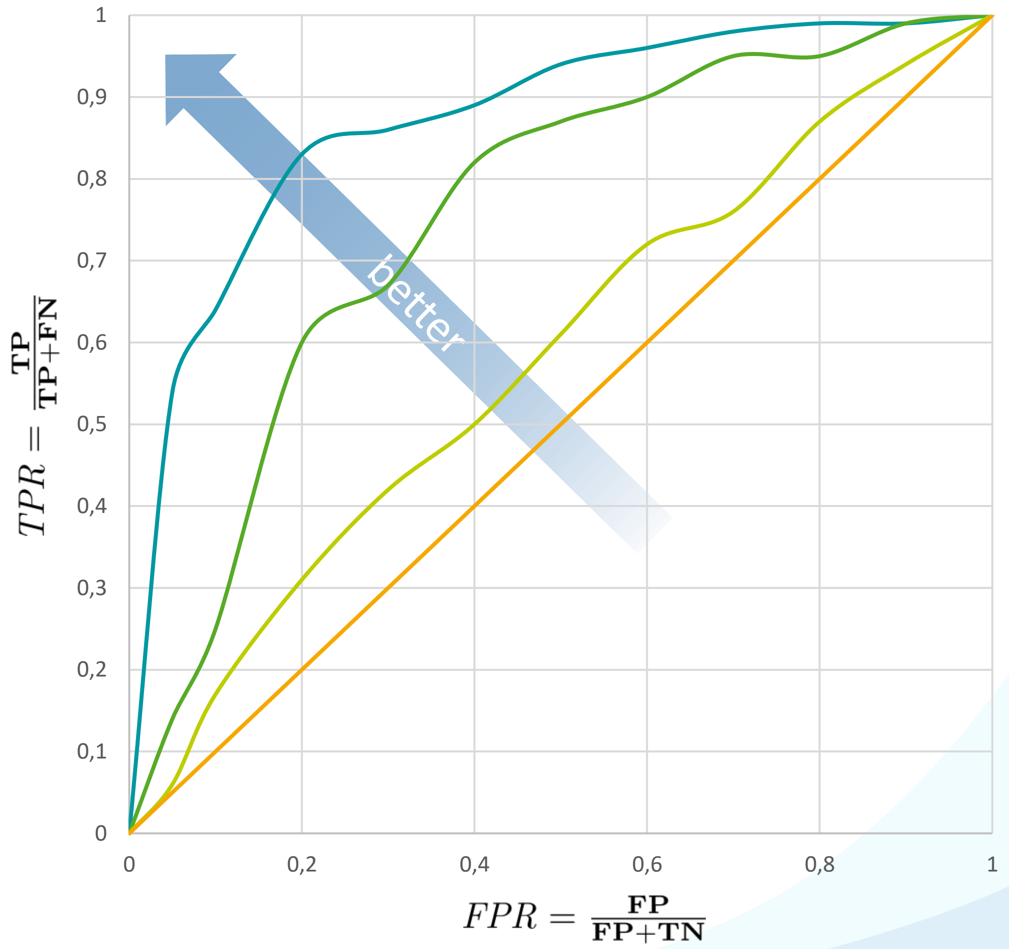
→ Performance is independent of ***q, N***!

ROC Curve – Beating Random Guessing



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

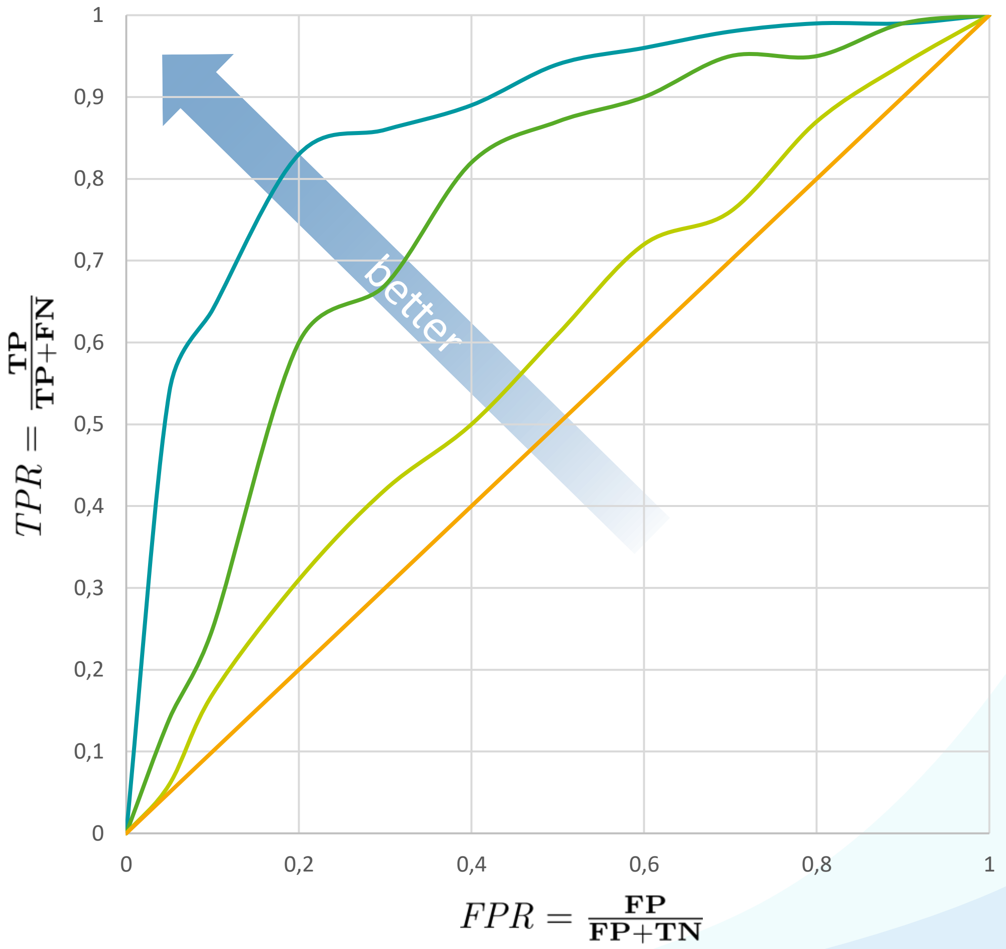
Example ROC Curves



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?

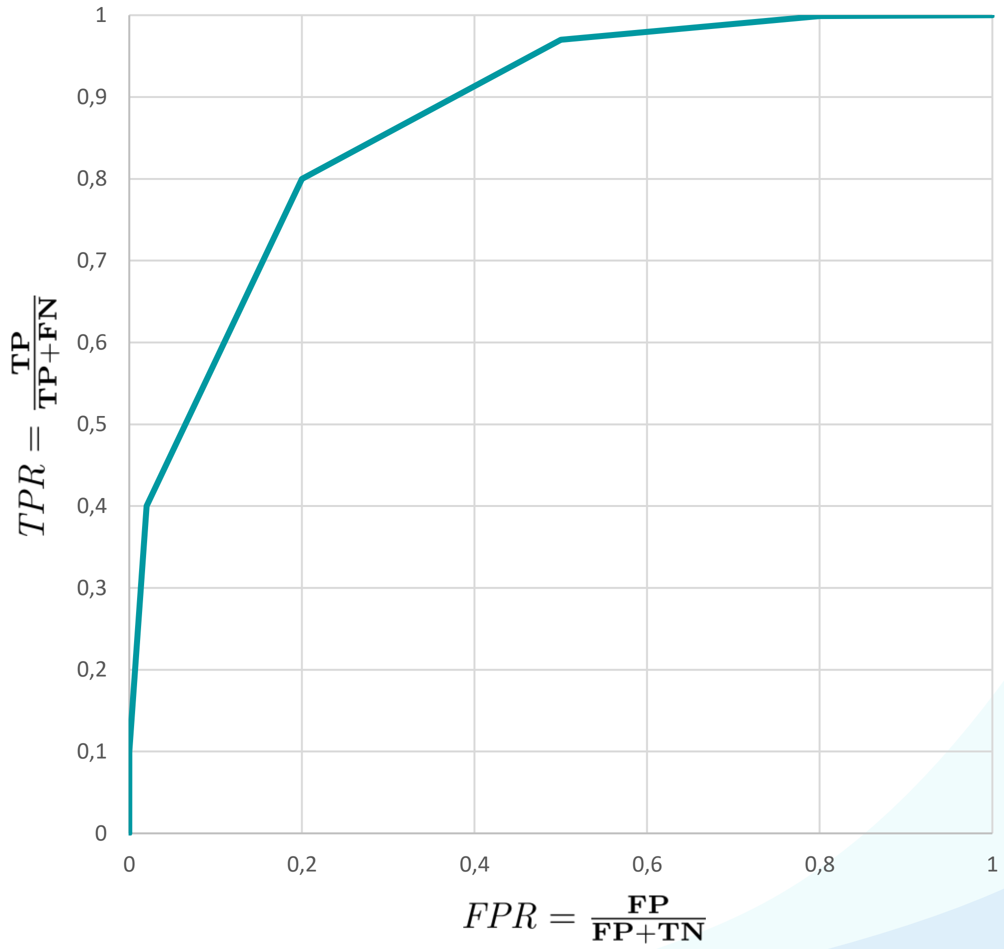
- ROC Index / AUC (Area Under the Curve)
- Larger area → closer to optimum
- Computable as integral of curve

T is the set of thresholds

FPR for the *i*th threshold

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

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



ROC Index / AUC (Area Under the Curve)

Example

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

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

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

1.0 to 0.75

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

0.75 to 0.5

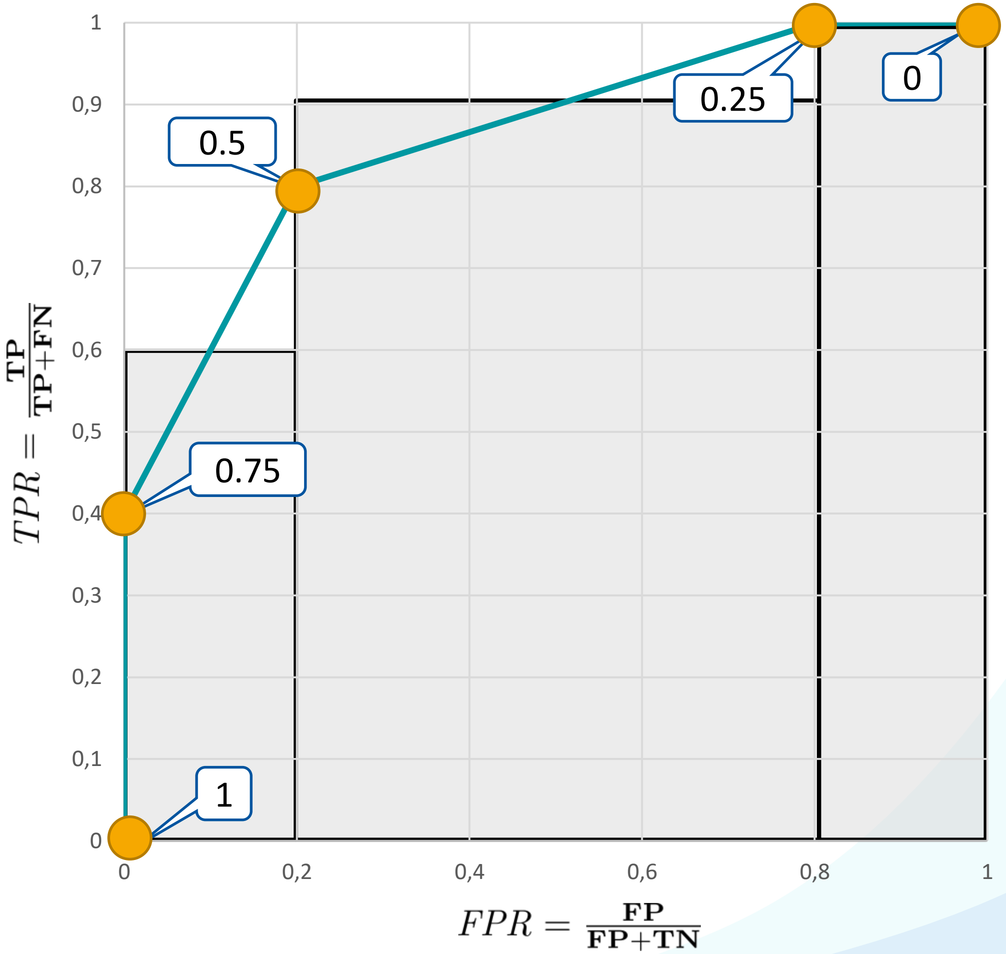
$$+(0.8 - 0.2) \cdot \frac{(1.0 + 0.8)}{2}$$

0.5 to 0.25

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

0.25 to 0.0

$$= 0.0 + 0.12 + 0.54 + 0.2 = 0.86$$



Assessing Model Quality

- **Now suppose you are comparing two predictive models (e.g., obtained from two different supervised learning methods).**
- **Question:**
  1. **How to assess performance differences?**
  2. **What could go wrong?**

- *Let’s again collect your ideas here…*
- *When can we make a statement about which model is best?*

Which is better?

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

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

Which is better?

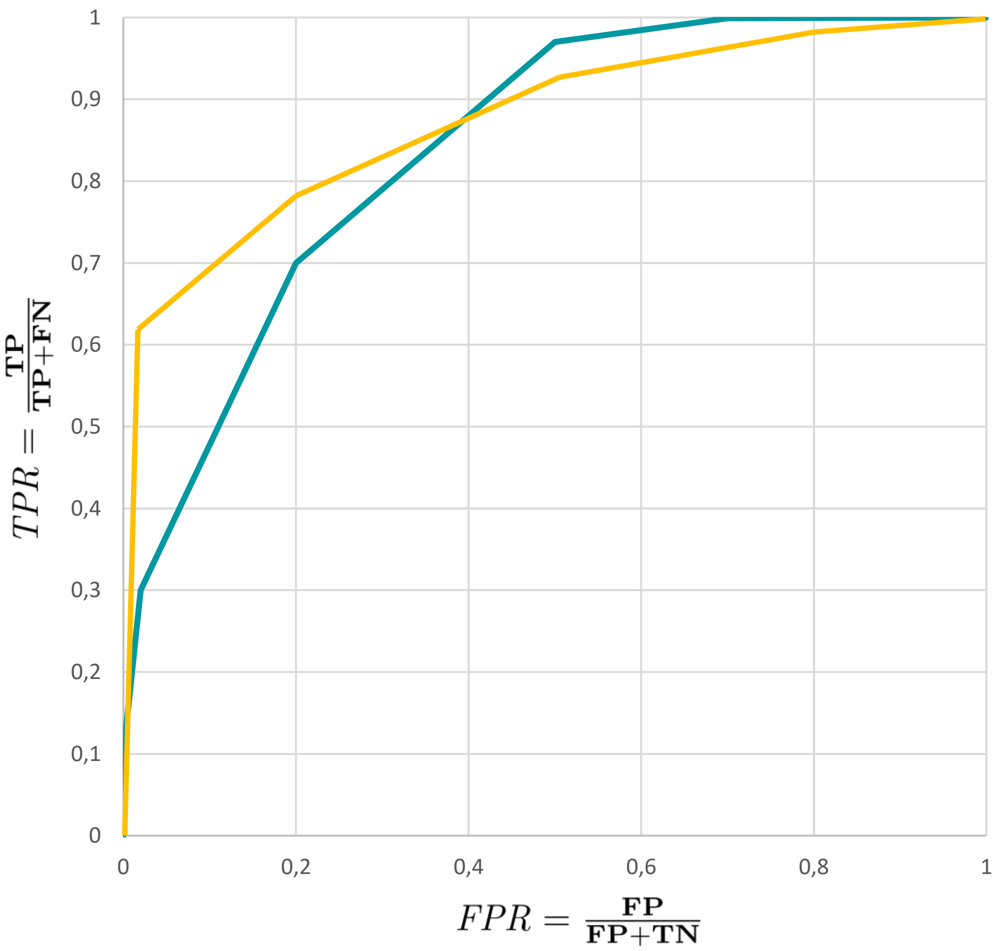
M_1		Prediction	
		On Time	Delay
Target Label	On Time	5	**5**
	Delay	**4**	6

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

Which is better?

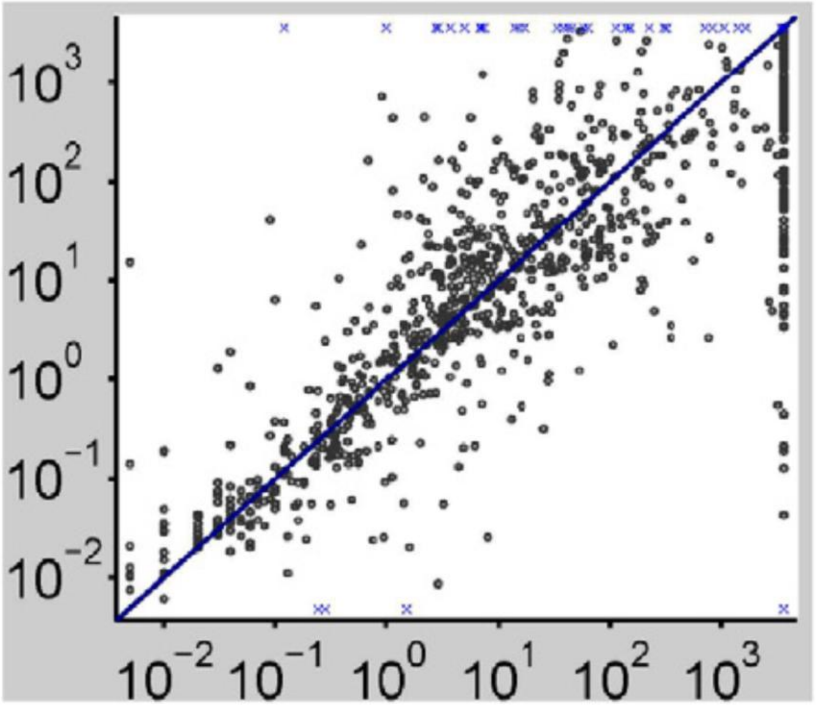
M_1

M_2

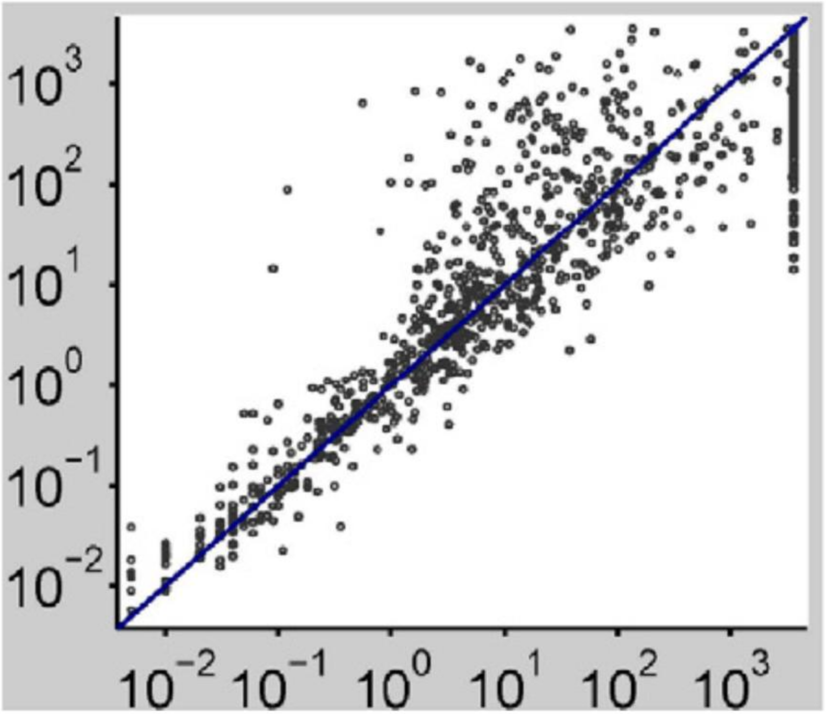


Which is better?

M_1 (Neural Network)



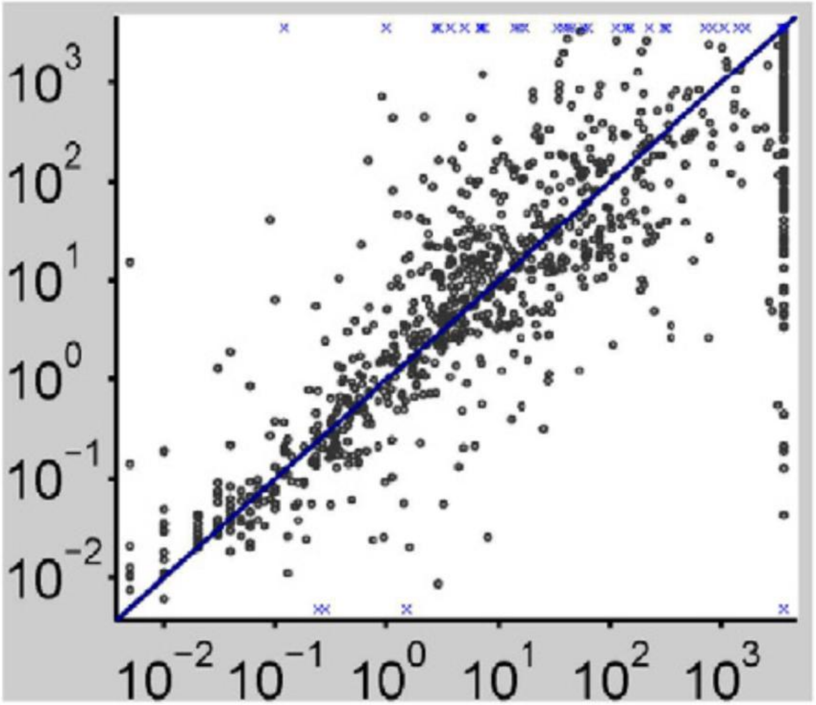
M_2 (Random Forest)



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

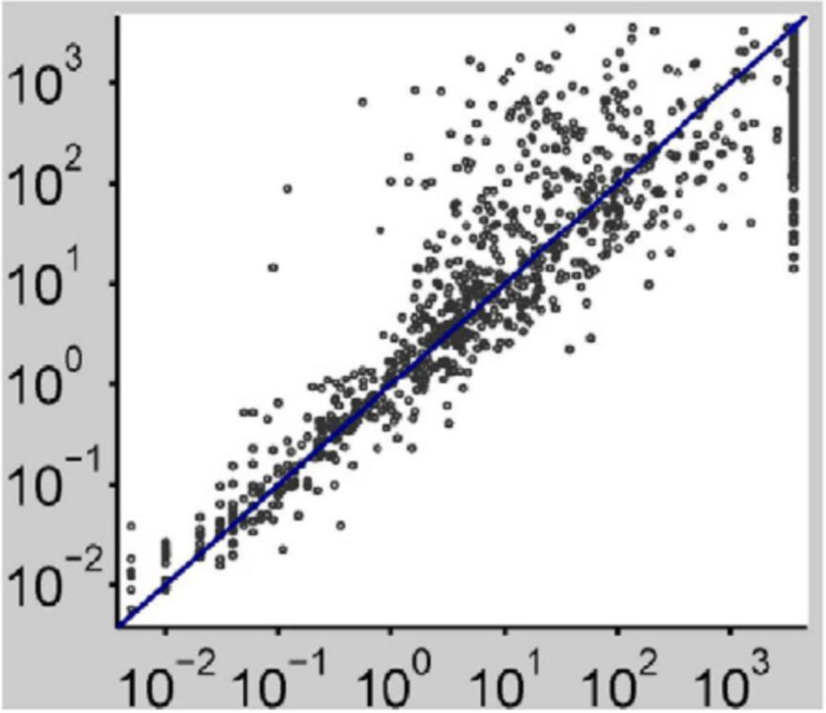
Which is better?

M_1 (Neural Network)



RMSE = 1.1

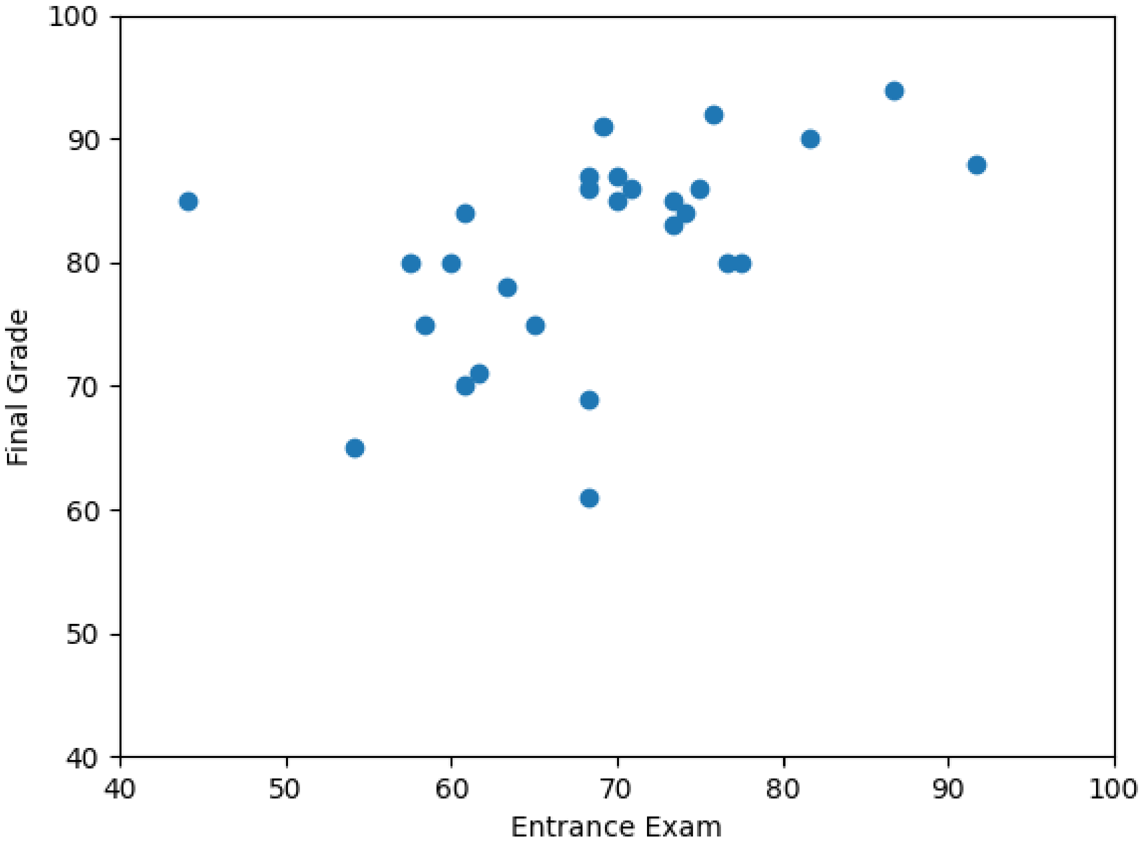
M_2 (Random Forest)



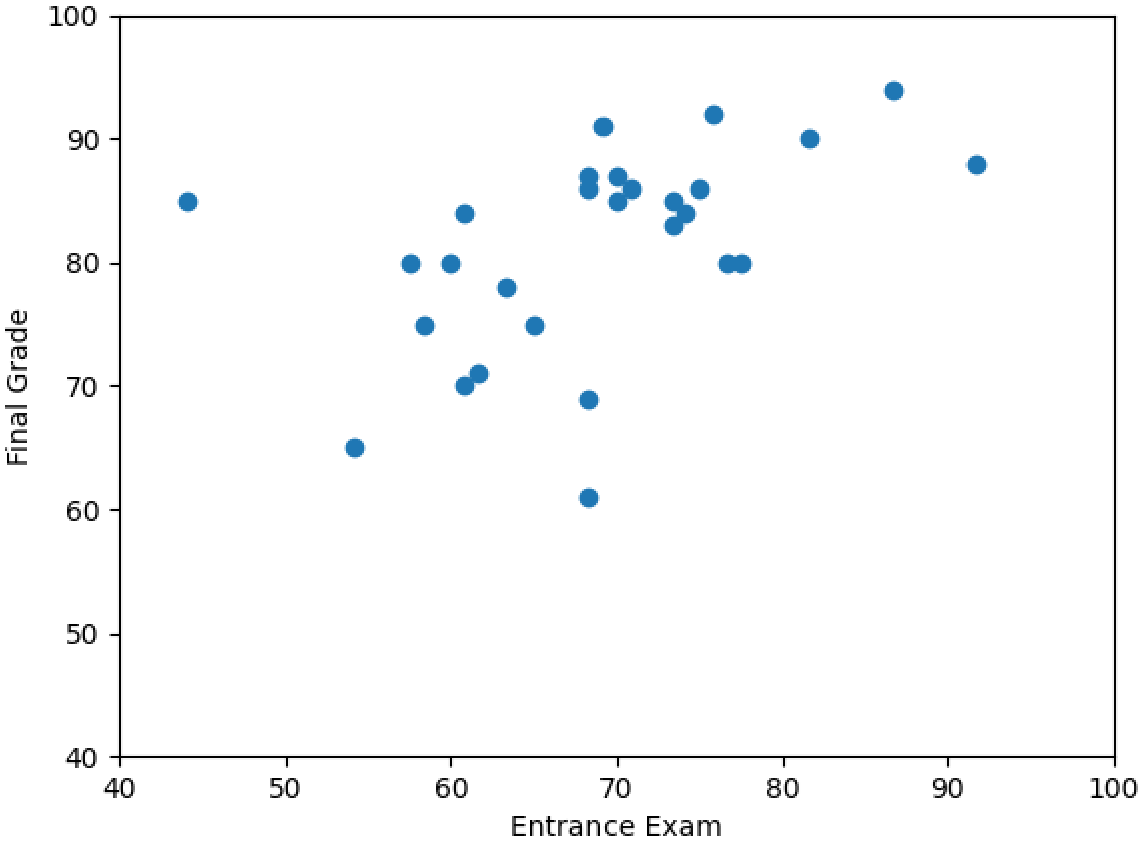
RMSE = 0.72

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

Assessing performance correlation

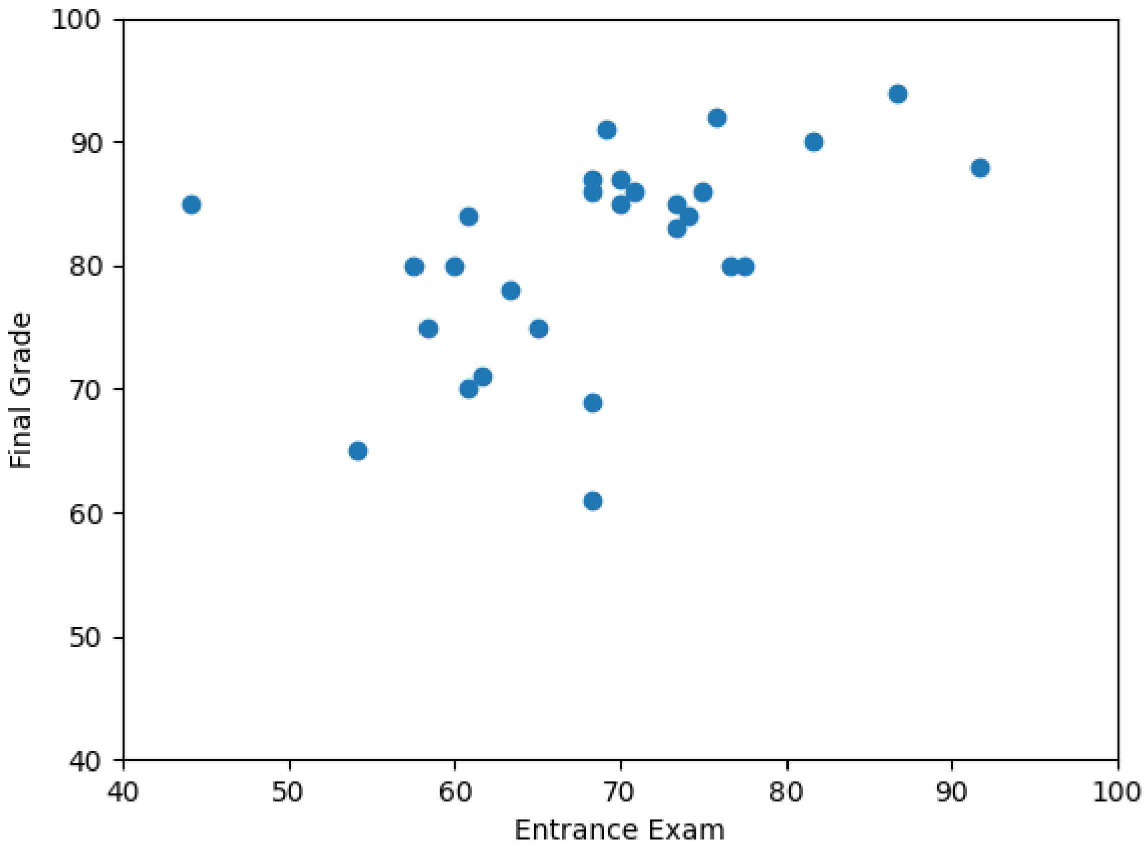


Assessing performance correlation



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)

Background: Measuring Correlation

- **Pearson correlation coefficient**
  - Measures linear relationship between two sets of data
  - Both sets of data follow normal distribution (no outliers)

$$\rho_{X,Y} = Corr(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- **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)

$$r_S = \rho_{R[X],R[Y]} = \frac{\mathrm{cov}(R[X],R[Y])}{\sigma_{R[X]} \sigma_{R[Y]}}$$

Practical Aspects of Assessing Model Quality

- **Which is better?**
  - M_1: accuracy from k-fold cross-validation = 0.712
  - M_2: accuracy from k-fold cross-validation = 0.721

- **Important realization**
  - Performance differences may be due to random effects

  $\Rightarrow$ Assess statistical significance using statistical hypothesis testing.

Refresher on Statistical Hypothesis Testing

- **Concepts**
  - 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)
  - Note: 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

$\Rightarrow$ Note: 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
  - $(m_{1,1},\ m_{1,2}, \dots, m_{1,k})$,
  - $(m_{2,1},\ m_{2,2}, \dots, m_{2,k})$,

- 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 $\alpha = 0.05$

Quick poll: *Who is already familiar with the paired t-test?*

Background: Student’s t-test

Multiple types of tests

- One-sample t-test:
  - Test whether the mean of a distribution has a value specified in the null hypothesis.

- Two-sample (paired) t-test:
  - Test of the null hypothesis that the means of two distributions are equal.
  - Dependent (related) samples:
    - For comparing the means of two conditions in which the same (or closely matched) participants participated
  - Independent (unrelated samples):
    - For comparing the means of two different groups of participants

Background: Student's t-test

Interpretation:

$$t = \frac{mean - comparison\ \ value}{Standard\ \ Error}$$

Multiple types of tests

- One-sample t-test:
    - Test whether the mean of a distribution has a value specified in the null hypothesis.
    - Procedure:
        - Compute the test statistic $t = \frac{\bar{X} - \mu_0}{SE}$ $SE = \sigma_X / \sqrt{N}$
        - Determine the degrees of freedom $df = N - 1$
        - Look up the p-value in a table of the Student t-distribution with df degrees of freedom

- Assumptions
    - Random and independent sampling
    - Data are from normally distributed populations (or $N \geq 30$)

Background: Student’s t-test

Interpretation:

$$t = \frac{mean\ \ difference - 0}{Standard\ \ Error_D}$$

Multiple types of tests

- Two-sample (paired) t-test with dependent (related) samples
  - Test of the null hypothesis that the means of two distributions are equal.
  - Compare the mean difference of the scores in the two conditions with $\mu_D = 0$
  - Normalize by the Standard Error SE_D of the differences (computed from the stddev SD_D of the differences)
  - Procedure:
    - Compute the test statistic $t = \frac{(\overline{X_1 - X_2}) - \mu_D}{SE_D}$ $SE_D = SD_D / \sqrt{N}$
    - Determine the degrees of freedom $df = N - 1$
    - Look up the p-value in a table of the Student t-distribution with df degrees of freedom

Testing for Significance of Performance Differences

- **Caution:** paired t-test requires a normality assumption!
- *How can we know whether performance data over folds follows a normal distribution?*

  ⇒ Check **QQ plot** or use a normality test (e.g., **Shapiro-Wilk**)
- *What to do if it doesn't?*

  ⇒ Use a non-parametric test, e.g., **Wilcoxon Signed-Ranks Test**

***Homework:** Look up what those terms mean.*

Comparing Two Predictive Models

- **Do…**
  - Assess performance of each model individually
  - Analyze 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)

Assessing Model Quality

- **Suppose you are using a randomized supervised ML procedure to train a predictive model.**
- **Question:**
  1. **How to assess the training procedure?**
  2. **What could go wrong?**

- *Let's again collect your ideas here…*
- *What makes randomized methods different? How can we adjust for that?*

Evaluating Randomized Supervised ML Procedures

- Adjustments to account for the randomness
  - Perform p independent runs ($p \geq 2$) → p models
  - Assess & compare performance of all p models
  - Inspect / analyze 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!)

Assessing Model Quality

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

- **Question:**
  - **What could happen to invalidate earlier performance assessments?**

- *Let's again collect your ideas here…*

- *What fundamental assumptions do we rely on?*

  ⇒ Performance degradation due to **concept drift**
  (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