Introduction to Data Science (IDS) course

Evaluation of Supervised Learning Problems

Lecture 9

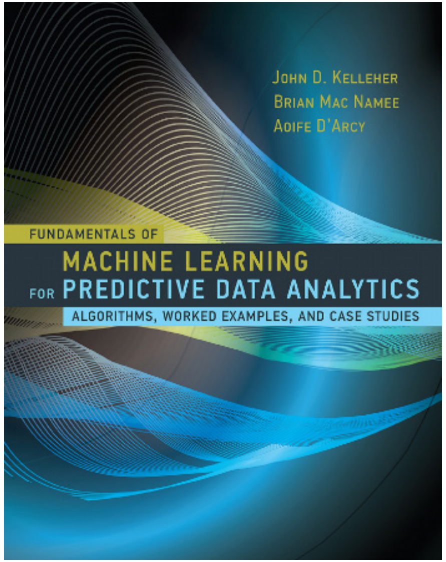
IDS-L9

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Outline of Today’s Lecture

- The need for evaluation
- Training, validation, test sets
- Cross validation
- Categorical target features
- ROC Curves
- Multinomial targets
- Continuous target features
- A/B Testing
- Concept drift



Based on Chapter 8 of Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Chair of Process and Data Science

The need for evaluation

Supervised learning

descriptive features

Naïve Bayes’ Classifier
(Logistic) Regression
Support Vector Machines
Neural Networks
…

target feature

training instances vs unseen instances

Example: Classifier for Study result

Name	Birth date	Hobby	Eye color	City	Lectures present	Study result
John	12-2-1966	Golf	Blue	Bonn	24	Pass
Mary	18-3-1966	Golf	Brown	Aachen	23	Pass
Sue	27-4-1966	Golf	Green	Bonn	7	Fail
Chris	24-5-1966	Tennis	Blue	Aachen	12	Fail
Peter	26-5-1966	Tennis	Brown	Bon	24	Pas
Karen	30-6-1966	Tennis	Green	Aachen	22	Pas
…	…	…	…	…	…	…

2 3 4 5 1

Possible problems: Overfitting & Underfitting

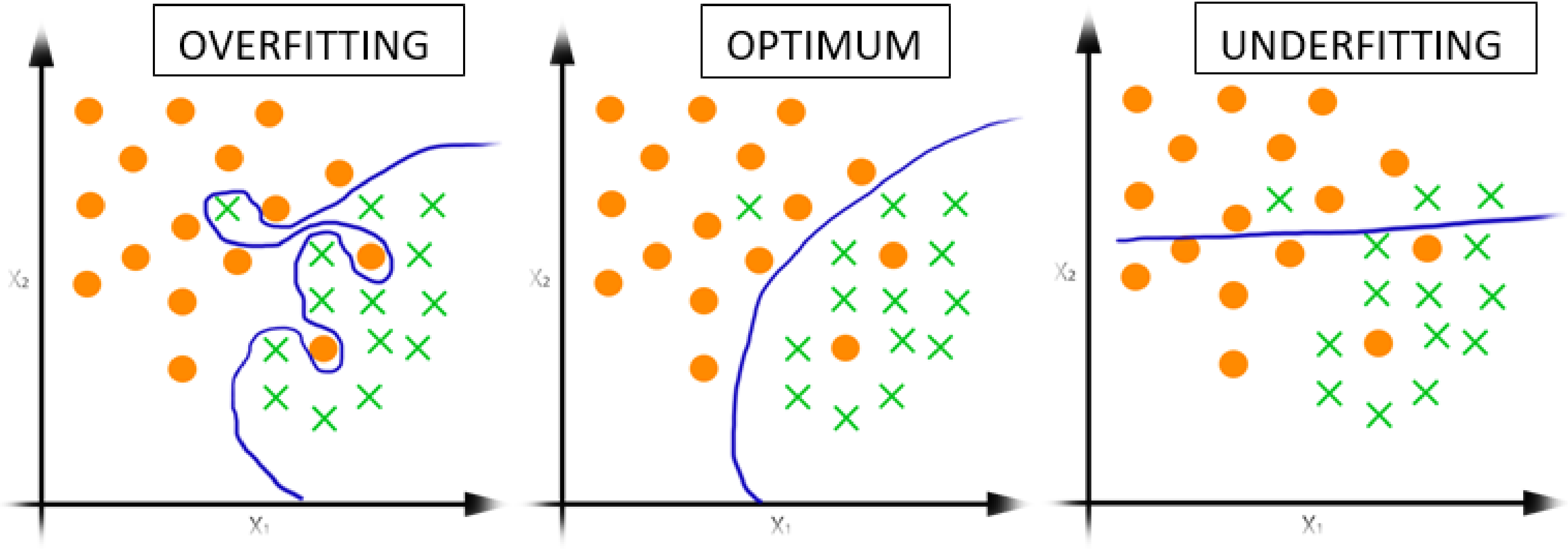
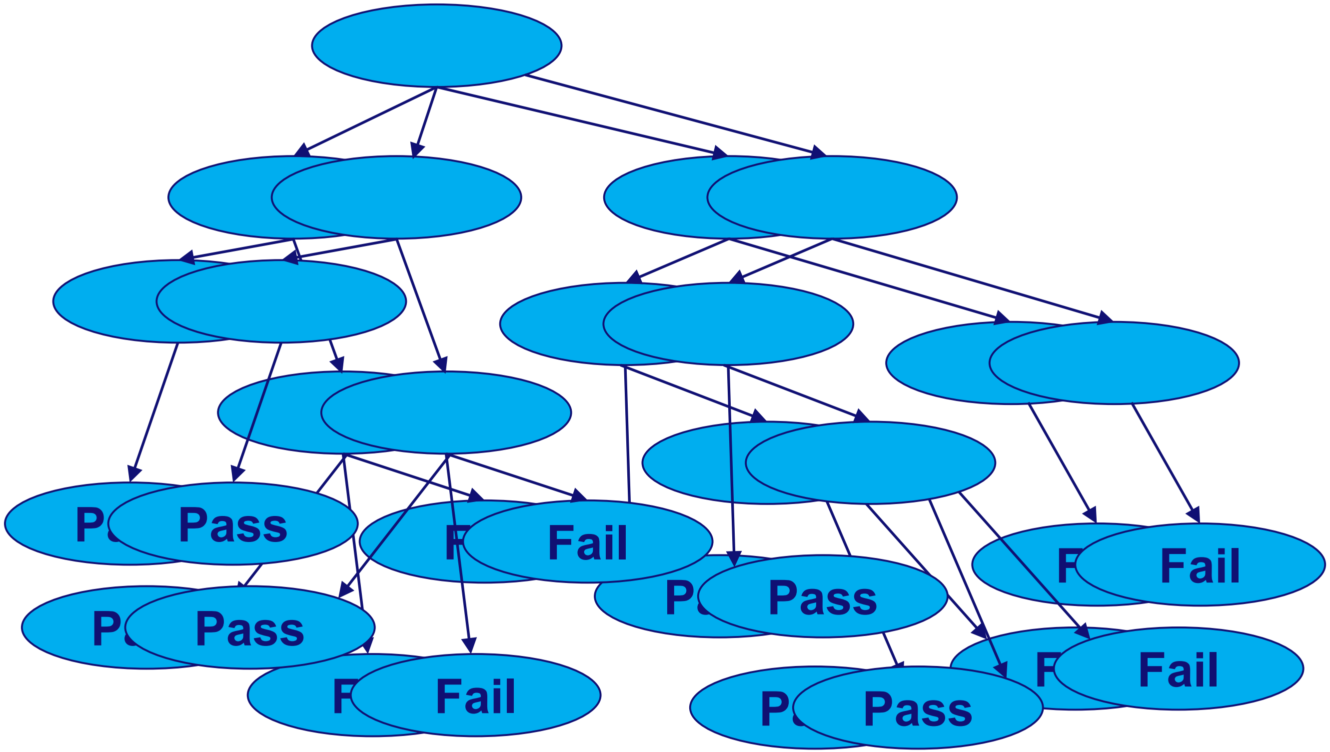


Diagram by Sachin Joglekar (Google).

Decision trees

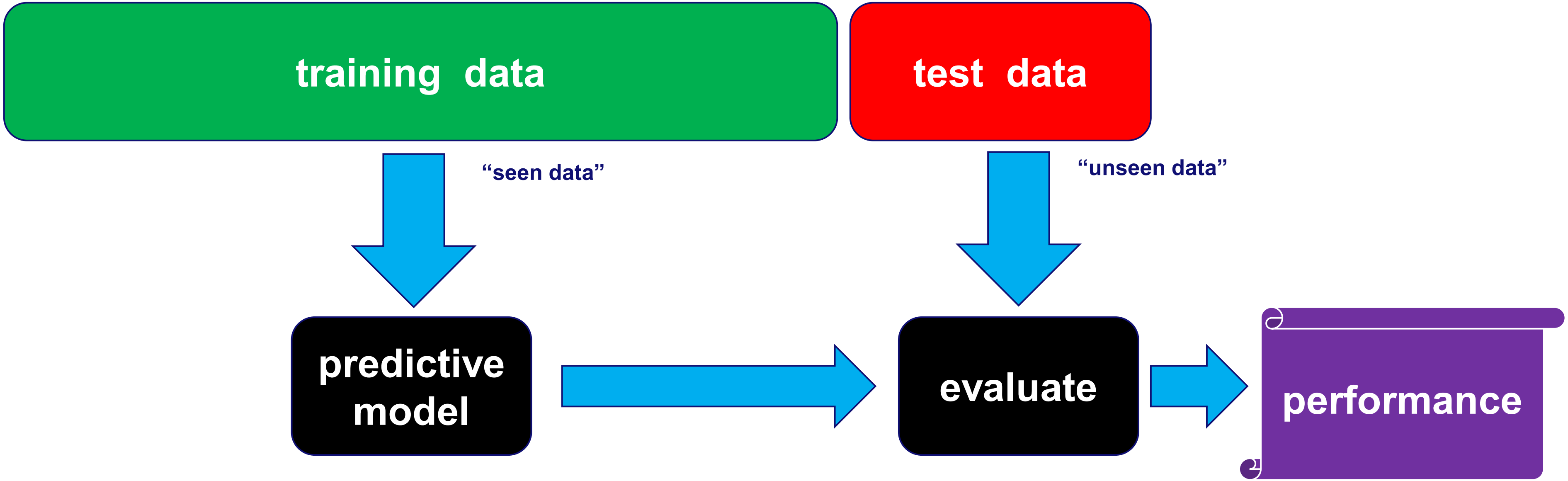
Overfitting



Underfitting

Chair of Process and Data Science

Hold-out test set



Sometimes the training data are split into a "true" training set and a validation set for parameter selection, hyperparameter tuning, stopping criteria, etc.

Example test results binary classification

spam = positive, ham = negative

ID	Target	Pred.	Outcome	ID	Target	Pred.	Outcome
1	spam	ham	FN	11	ham	ham	TN
2	spam	ham	FN	12	spam	ham	FN
3	ham	ham	TN	13	ham	ham	TN
4	spam	spam	TP	14	ham	ham	TN
5	ham	ham	TN	15	ham	ham	TN
6	spam	spam	TP	16	ham	ham	TN
7	ham	ham	TN	17	ham	spam	FP
8	spam	spam	TP	18	spam	spam	TP
9	spam	spam	TP	19	ham	ham	TN
10	spam	spam	TP	20	ham	spam	FP

		Prediction	
		positive	negative
Target	positive	*TP*	*FN*
	negative	*FP*	*TN*

Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Chair of Process and Data Science

Example test results binary classification

spam = positive, ham = negative

ID	Target	Pred.	Outcome
1	spam	ham	FN
2	spam	ham	FN
3	ham	ham	TN
4	spam	spam	TP
5	ham	ham	TN
6	spam	spam	TP
7	ham	ham	TN
8	spam	spam	TP
9	spam	spam	TP
10	spam	spam	TP

ID	Target	Pred.	Outcome
11	ham	ham	TN
12	spam	ham	FN
13	ham	ham	TN
14	ham	ham	TN
15	ham	ham	TN
16	ham	ham	TN
17	ham	spam	FP
18	spam	spam	TP
19	ham	ham	TN
20	ham	spam	FP

		Prediction	
		positive	negative
Target	positive	*TP*	*FN*
	negative	*FP*	*TN*

Confusion matrix:

		Prediction	
		'spam'	*'ham'*
Target	*'spam'*	6	3
	'ham'	2	9

Chair of Process and Data Science

Example test results binary classification

spam = positive, ham = negative

ID	Target	Pred.	Outcome
1	spam	ham	FN
2	spam	ham	FN
3	ham	ham	TN
4	spam	spam	TP
5	ham	ham	TN
6	spam	spam	TP
7	ham	ham	TN
8	spam	spam	TP
9	spam	spam	TP
10	spam	spam	TP

ID	Target	Pred.	Outcome
11	ham	ham	TN
12	spam	ham	FN
13	ham	ham	TN
14	ham	ham	TN
15	ham	ham	TN
16	ham	ham	TN
17	ham	spam	FP
18	spam	spam	TP
19	ham	ham	TN
20	ham	spam	FP

Confusion matrix:

		Prediction 'spam'	Prediction 'ham'
Target	'spam'	6	3
	'ham'	2	9

$$\text{misclassification accuracy} = \frac{(FP + FN)}{(TP + TN + FP + FN)}$$

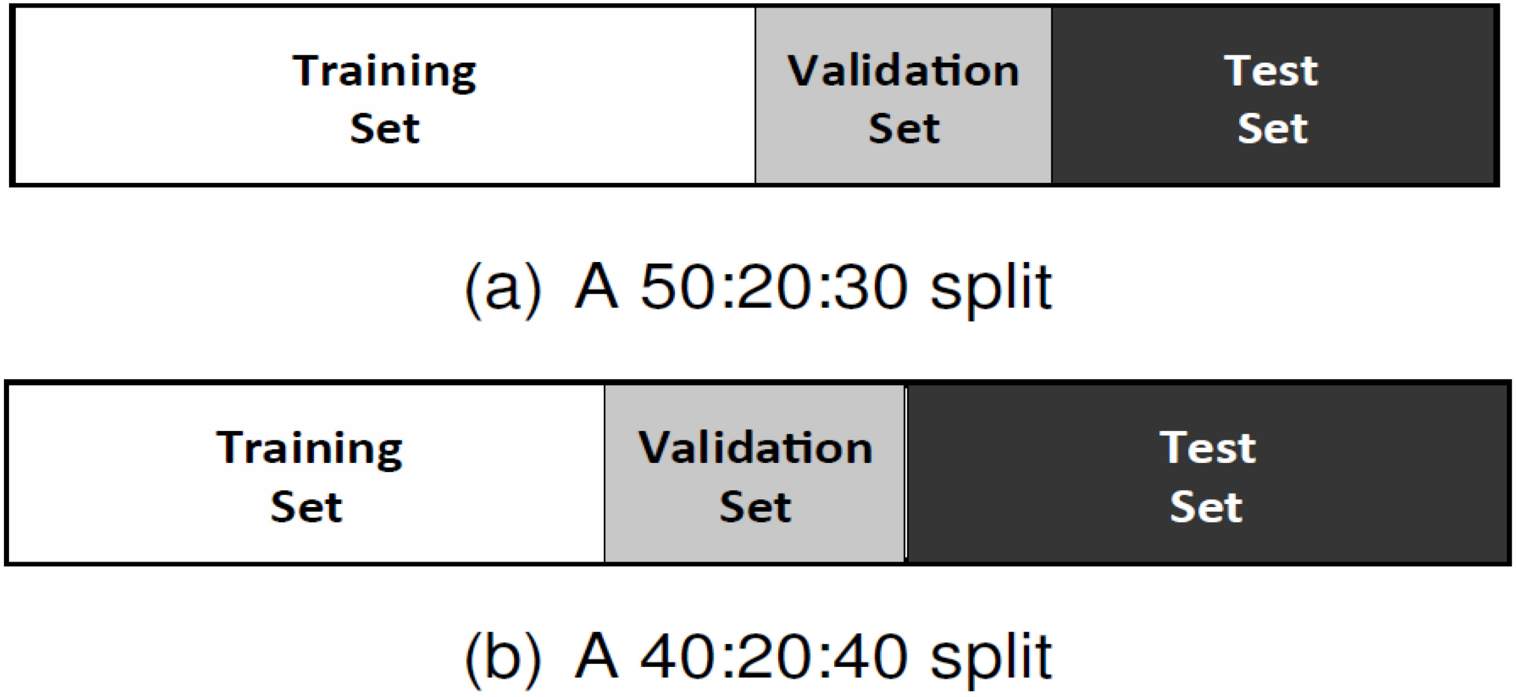
$$= \frac{(2+3)}{(6+9+2+3)} = 0.25$$

Chair of Process and Data Science

Validation set

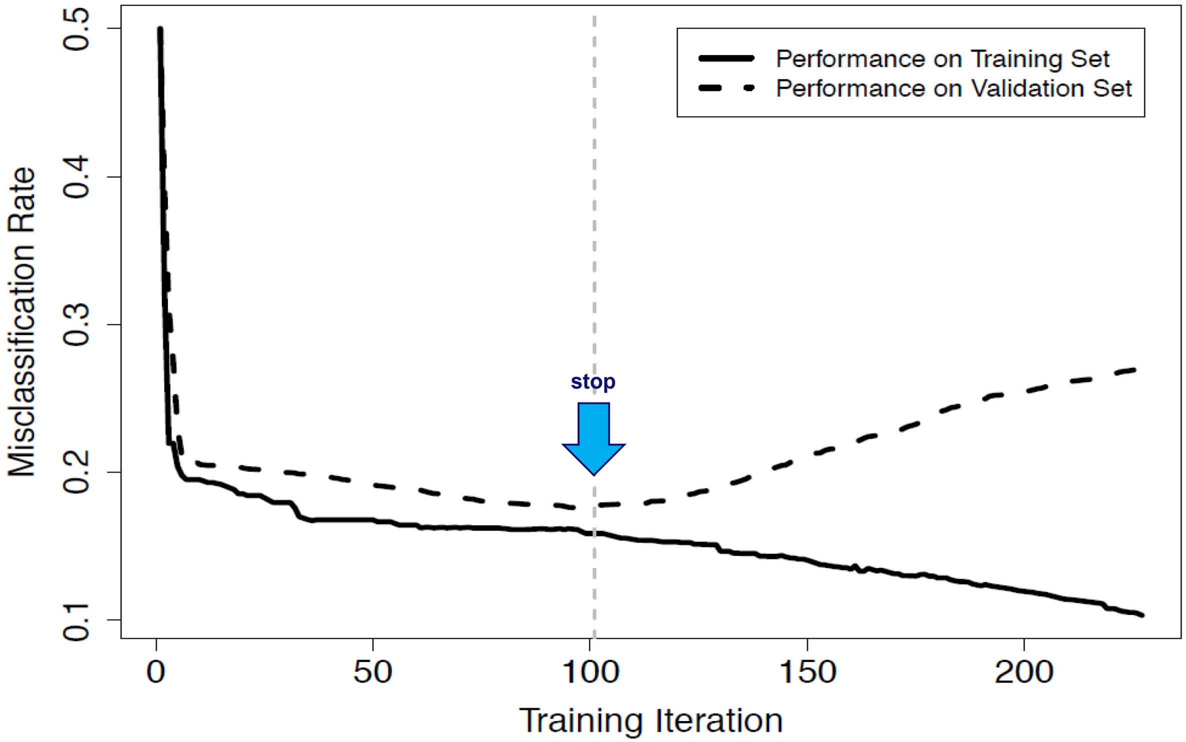
(not considered in remainder)

Validation set is used to "pre-evaluate" trained model (for parameter optimization or stopping criterion)



PADS Chair of Process and Data Science

Validation set is used to “pre-evaluate” trained model (for parameter optimization or stopping criterion)

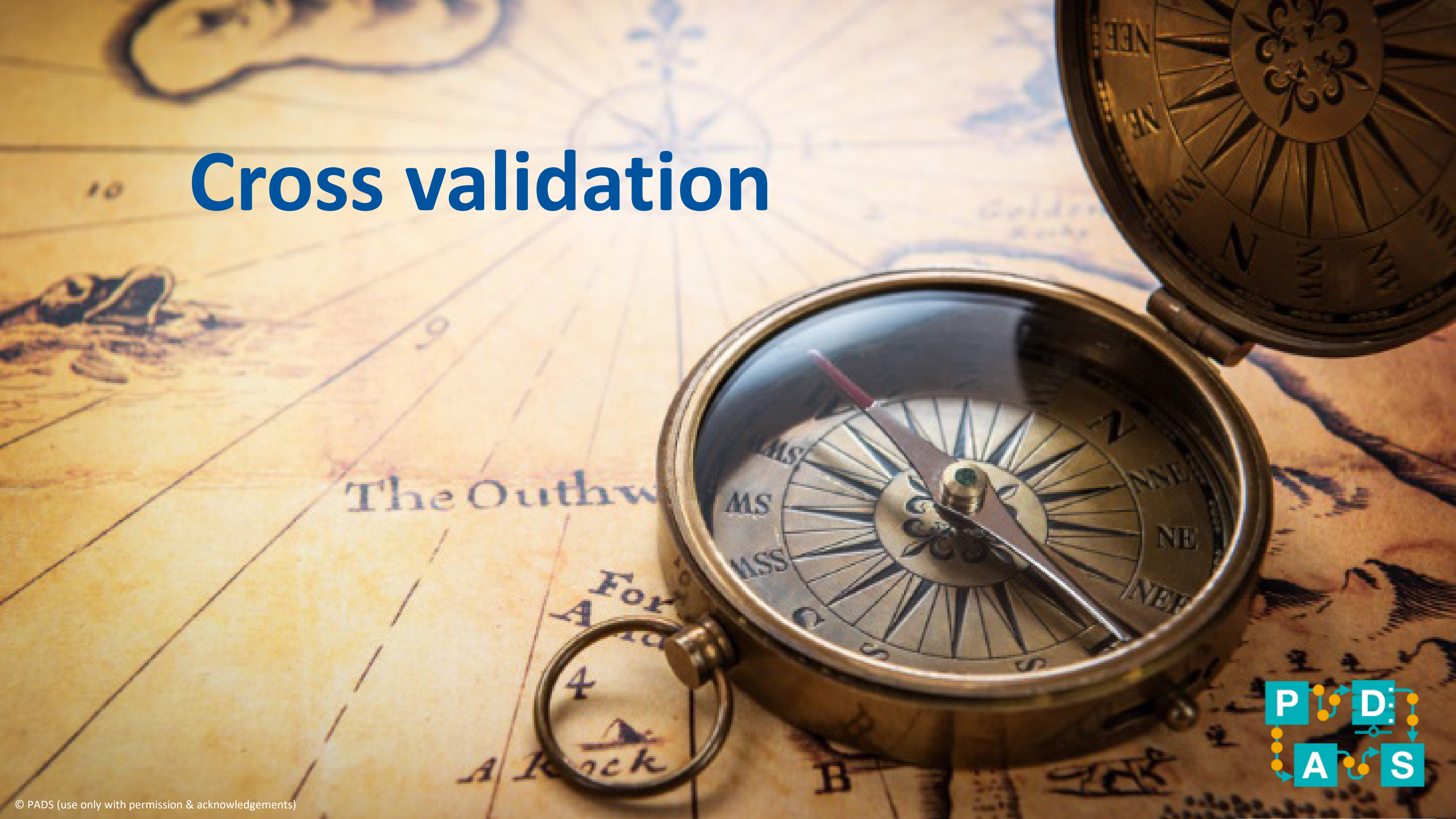


Validation set is used to avoid overfitting the test set.

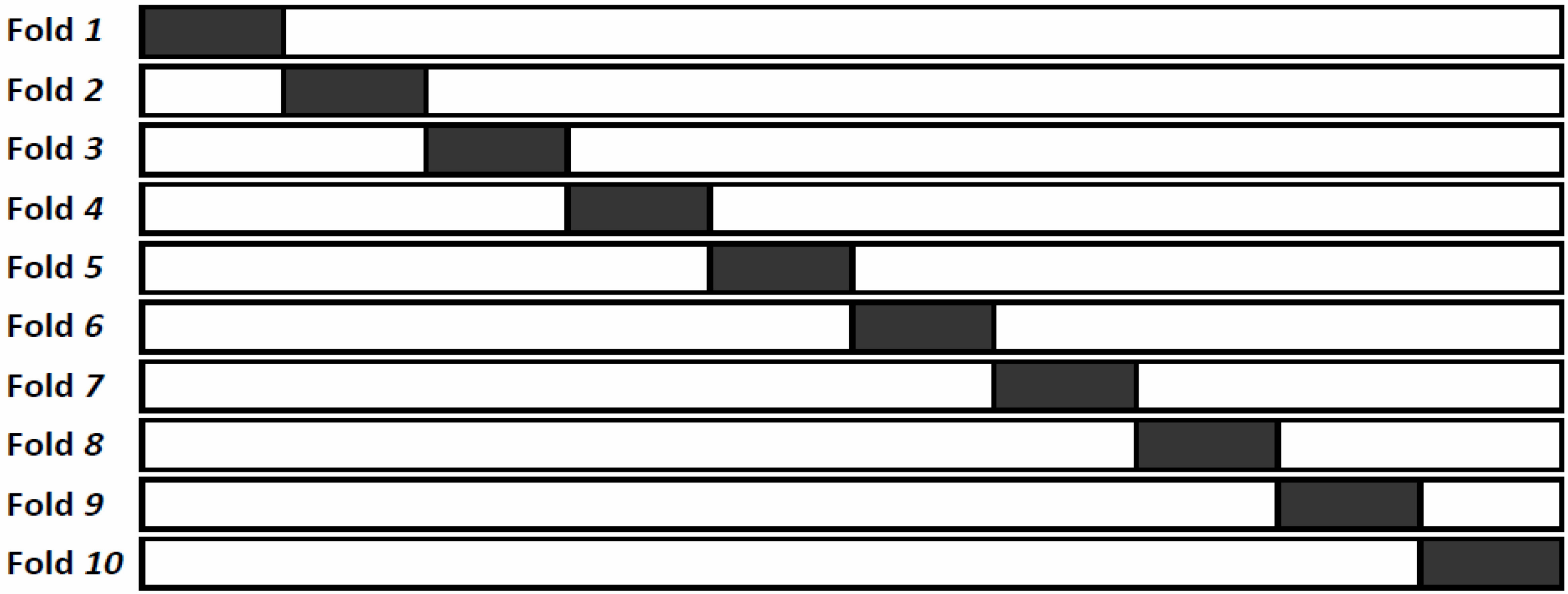
In the remainder we will abstract from this and just consider training and test data.

Chair of Process and Data Science

Cross validation



k-fold cross validation



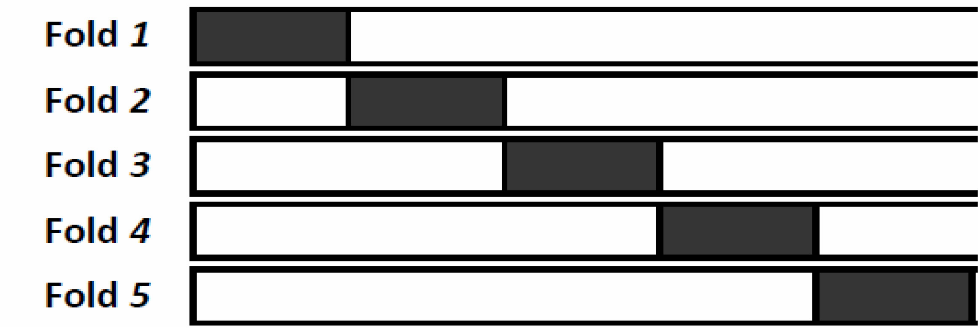
Typical value for k = 10

= test = training

Chair of Process and Data Science

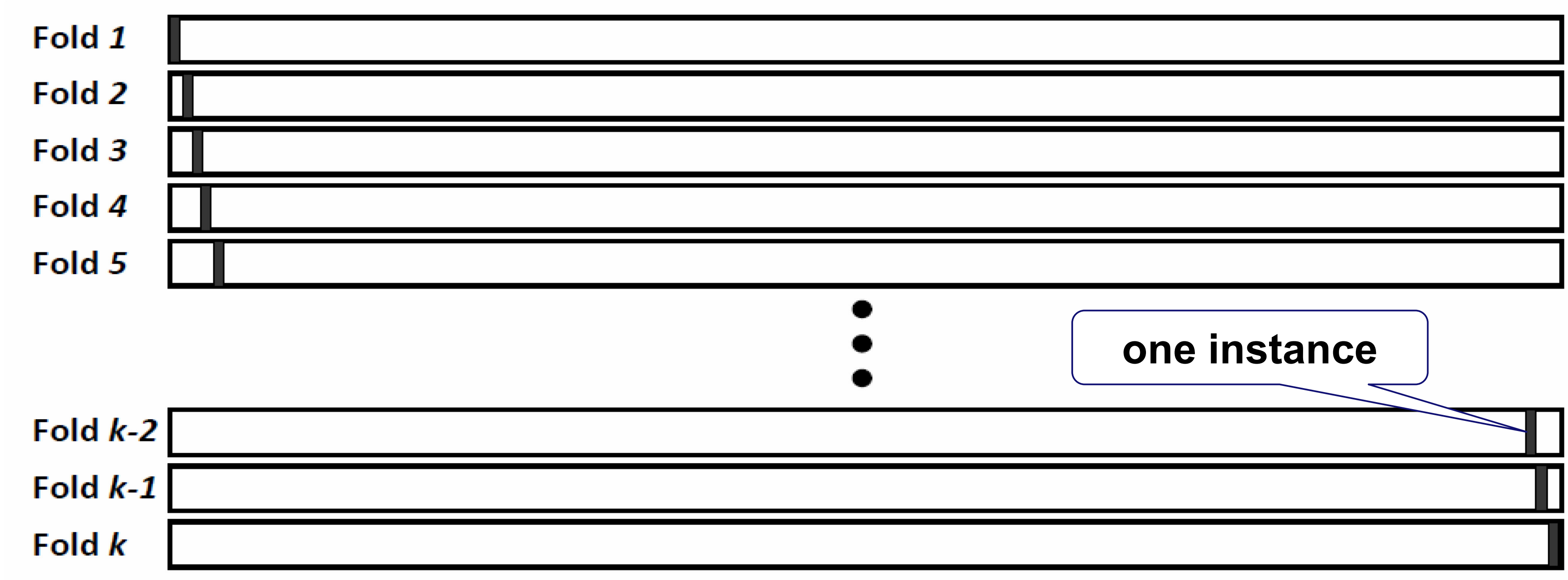
Example 5-fold cross validation

Fold	Confusion Matrix: Target	Prediction 'lateral'	Prediction 'frontal'	Class Accuracy
1	'lateral'	43	9	81%
	'frontal'	10	38	
2	'lateral'	46	9	88%
	'frontal'	3	42	
3	'lateral'	51	10	82%
	'frontal'	8	31	
4	'lateral'	51	8	85%
	'frontal'	7	34	
5	'lateral'	46	9	84%
	'frontal'	7	38	
Overall	'lateral'	237	45	84%
	'frontal'	35	183	



Chair of Process and Data Science

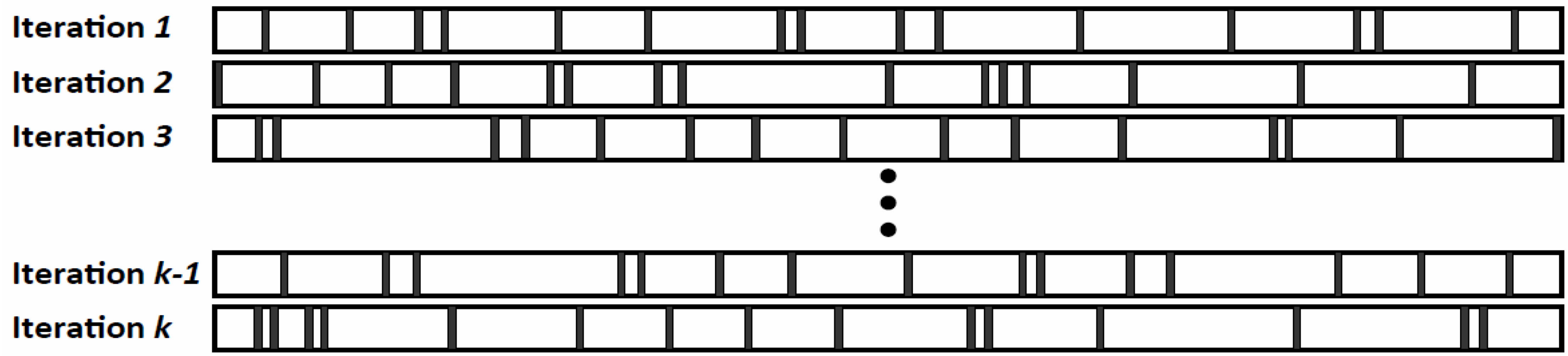
Leave-one-out cross validation



Also known as jackknifing: K-fold cross validation with test data being single instances. Used when less data is available.

Chair of Process and Data Science

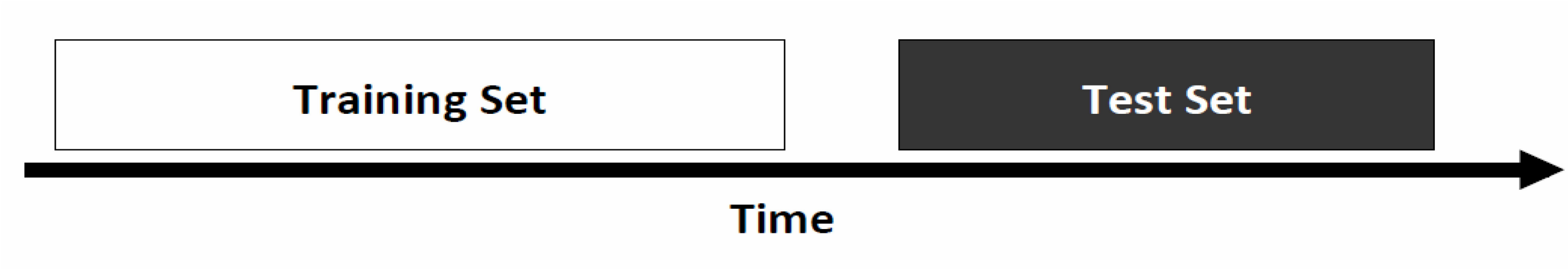
Bootstrapping



Repeatedly (*k* times), *m* randomly selected instances are selected as test set. Typically, used for smaller data sets, but k is typically much higher than for k-fold checking.

Chair of Process and Data Science

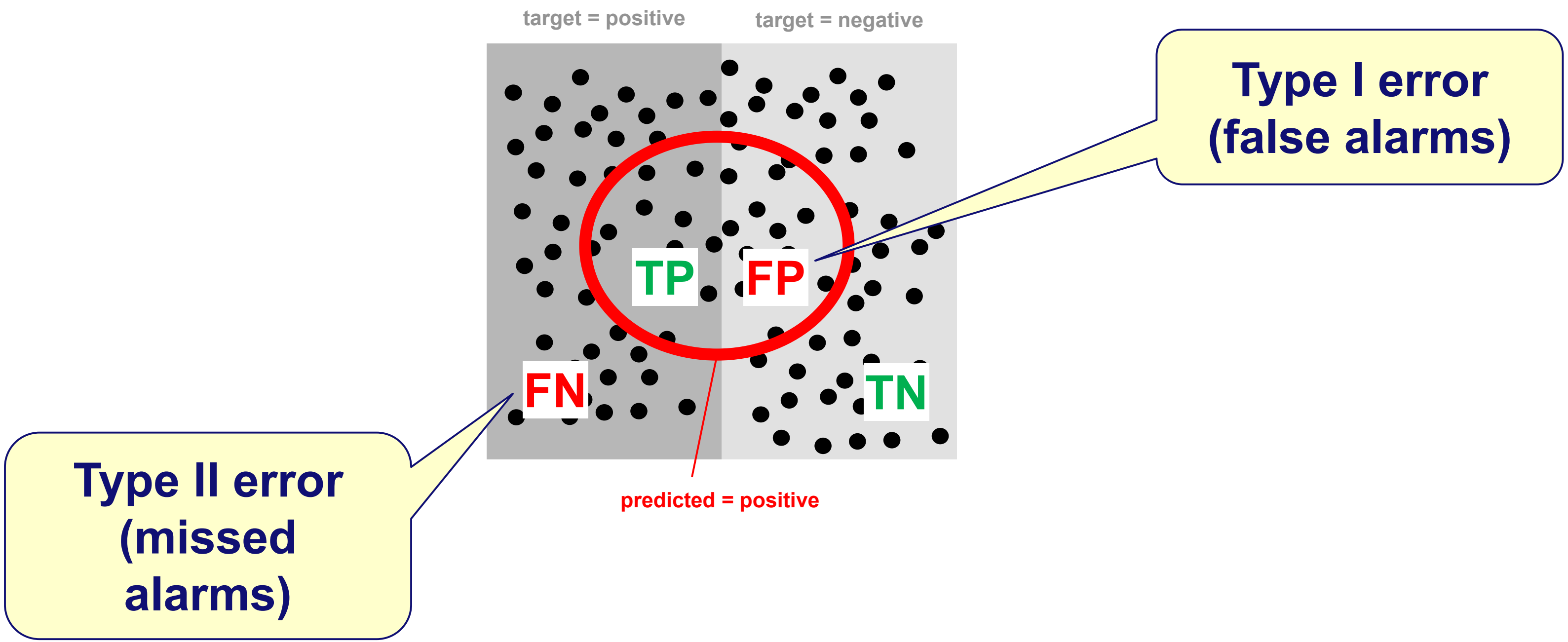
Out-of-time sampling



- **Random sampling works best if the instances are independent. However, in case of concept drift the time dimension plays a key role.**
- **Imagine seasonal effects, backlog, etc.**

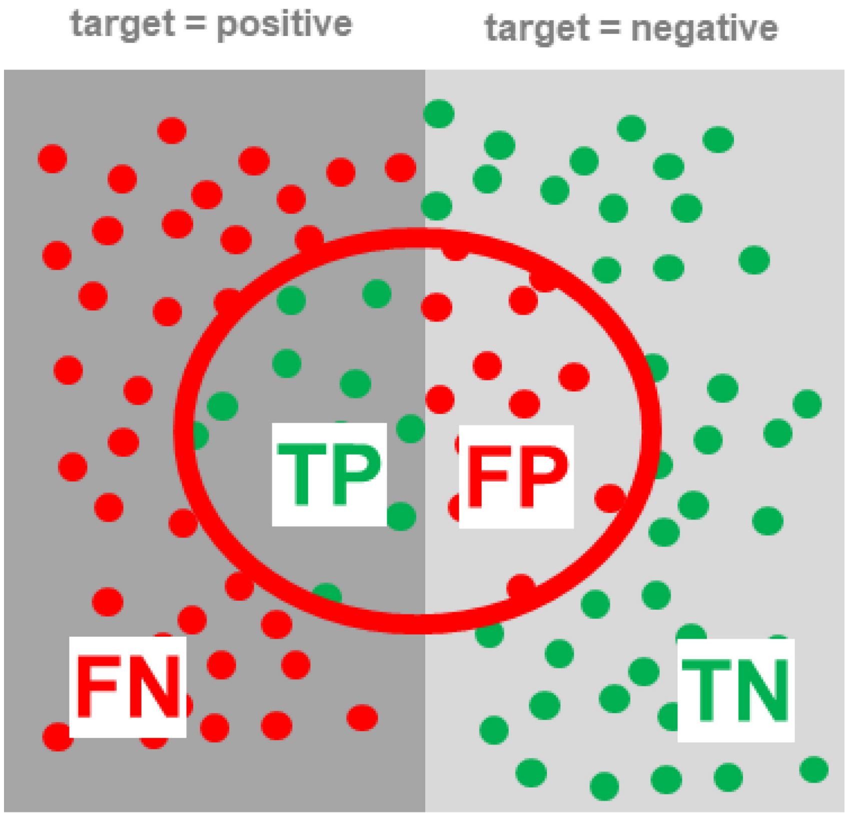
Categorical target features

Confusion matrix



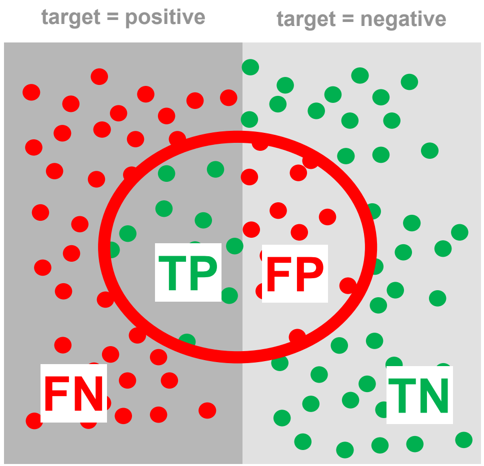
Chair of Process and Data Science

Confusion matrix



Confusion matrix-based performance measures

- True Positive Rate: $TPR = \frac{TP}{TP+FN}$
- True Negative Rate: $TNR = \frac{TN}{TN+FP}$
- False Negative Rate: $FNR = \frac{FN}{FN+TP}$
- False Positive Rate: $FPR = \frac{FP}{FP+TN}$
- Accuracy: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$
- Misclassification: $1 - ACC$



$TPR = 1 - FNR$

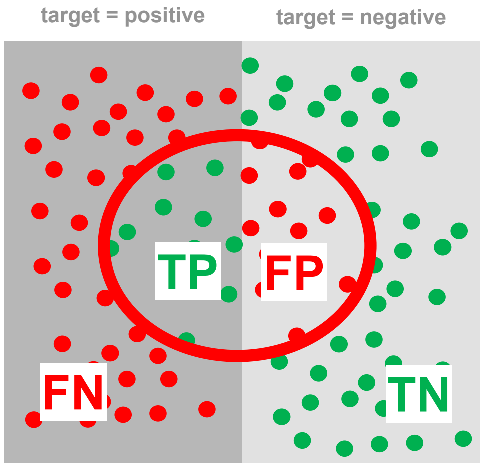
$TNR = 1 - FPR$

Chair of Process and Data Science

Precision, Recall and F_1-measure

- **Precision:** $precision = \frac{TP}{TP+FP}$

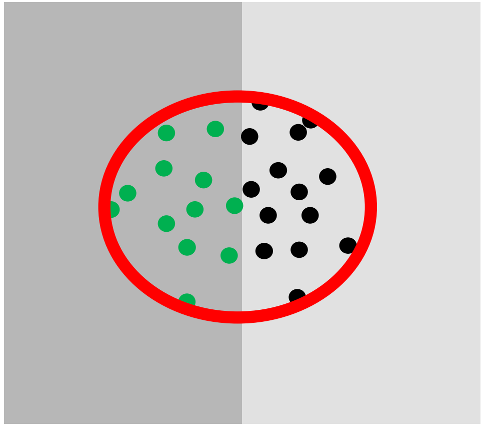
- **Recall:** $recall = \frac{TP}{TP+FN}$



Precision, Recall and F_1-measure

- **Precision:** $precision = \frac{TP}{TP+FP}$

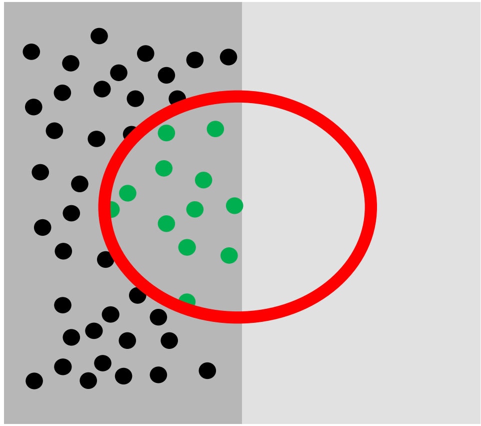
- **Recall:** $recall = \frac{TP}{TP+FN}$



Precision, Recall and F_1-measure

- **Precision:** $precision = \frac{TP}{TP+FP}$

- **Recall:** $recall = \frac{TP}{TP+FN}$



Note: True Positive Rate $TPR = \frac{TP}{TP+FN} = recall$

Example test results binary classification

spam = positive, ham = negative

ID	Target	Pred.	Outcome
1	spam	ham	FN
2	spam	ham	FN
3	ham	ham	TN
4	spam	spam	TP
5	ham	ham	TN
6	spam	spam	TP
7	ham	ham	TN
8	spam	spam	TP
9	spam	spam	TP
10	spam	spam	TP

ID	Target	Pred.	Outcome
11	ham	ham	TN
12	spam	ham	FN
13	ham	ham	TN
14	ham	ham	TN
15	ham	ham	TN
16	ham	ham	TN
17	ham	spam	FP
18	spam	spam	TP
19	ham	ham	TN
20	ham	spam	FP

Confusion matrix:

		Prediction 'spam'	Prediction 'ham'
Target	'spam'	6	3
	'ham'	2	9

$$\text{precision} = \frac{6}{(6+2)} = 0.75$$

$$\text{recall} = \frac{6}{(6+3)} = 0.667$$

$$F_1\text{-measure} = 2 \times \frac{\left(\frac{6}{(6+2)} \times \frac{6}{(6+3)}\right)}{\left(\frac{6}{(6+2)} + \frac{6}{(6+3)}\right)} = 0.706$$

Chair of Process and Data Science

Precision, Recall and F_1-measure

- $$F_1 measure = 2 \times \frac{precision \times recall}{precision + recall}$$

precision	recall	mean	F_1-measure
1	0	0.5	0
0	1	0.5	0
0.5	0.5	0.5	0.5
0.8	0.2	0.5	0.32
0.1	0.9	0.5	0.18

Imbalanced data

- Suppose 99% is positive in a data set with 100 instances.
- Always predicting positive leads to the following

- Precision: $\boldsymbol{precision = \frac{99}{99+1} = 0.99}$
- Recall: $\boldsymbol{recall = \frac{99}{99+0} = 1}$

PADS Chair of Process and Data Science

Average class accuracy

$$\text{average class accuracy} = \frac{1}{|levels(t)|} \sum_{l \in levels(t)} \text{recall}_l$$

- **Each class has the same weight independent of size**

Chair of Process and Data Science

Example

- Suppose 99% is positive in a data set with 100 instances.
- Always predicting positive leads to the following

$$\text{average class accuracy} = \frac{1}{|levels(t)|} \sum_{l \in levels(t)} \text{recall}_l$$

$$= \frac{1}{2}\left(\frac{99}{99} + \frac{0}{1}\right) = 0.5$$

Chair of Process and Data Science

Average class accuracy (harmonic mean)

$$\text{average class accuracy}_{\text{HM}} = \frac{1}{\frac{1}{|levels(t)|} \sum_{l \in levels(t)} \frac{1}{\text{recall}_l}}$$

- **Each class has the same weight independent of size**

Chair of Process and Data Science

Example

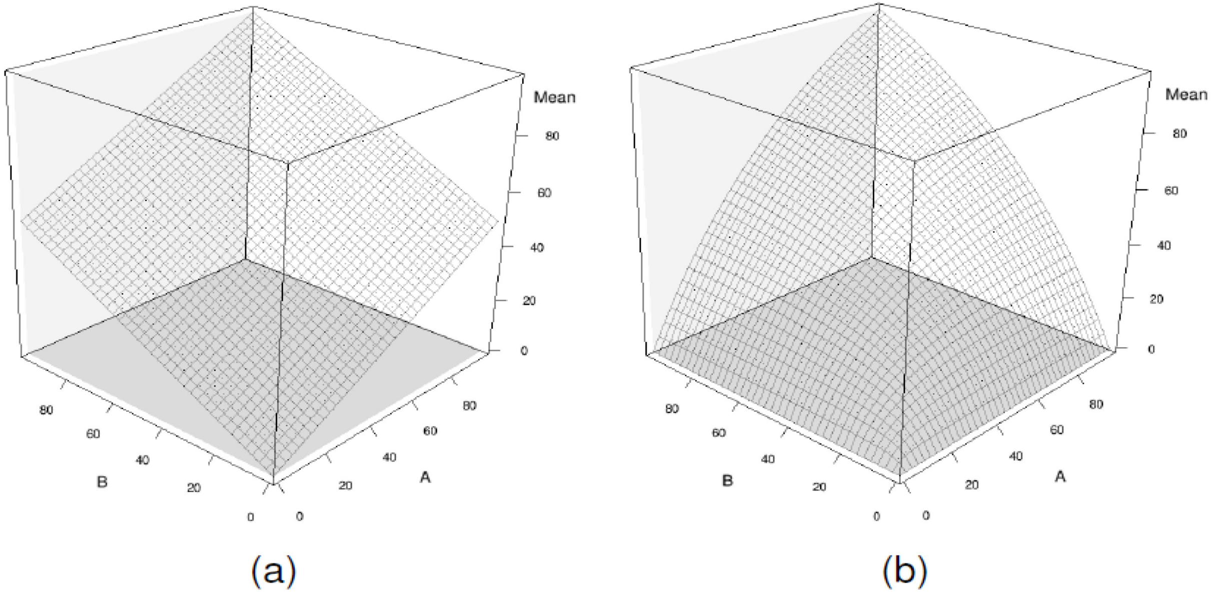
- **Suppose 99% is positive in a data set with 100 instances.**
- **Always predicting positive leads to the following**

$$\text{average class accuracy}_{HM} = \frac{1}{\frac{1}{|levels(t)|} \sum_{l \in levels(t)} \frac{1}{\text{recall}_l}}$$

$$= \frac{\mathbf{1}}{\frac{\mathbf{1}}{\mathbf{2}}\left(\frac{\mathbf{99}}{\mathbf{99}} + \frac{\mathbf{1}}{\mathbf{0}}\right)} = \mathbf{0.0}$$

$\frac{\mathbf{1}}{\mathbf{0}} = \infty$ **in the limit**

Effect of using the harmonic mean



Both need to be good for the overall HM measure to be good.

Figure: Surfaces generated by calculating (a) the **arithmetic mean** and (b) the **harmonic mean** of all combinations of features A and B that range from 0 to 100.

Chair of Process and Data Science

Profit matrix: Quantifying good and bad

		Prediction positive	Prediction negative
Target	positive	TP_{Profit}	FN_{Profit}
	negative	FP_{Profit}	TN_{Profit}

FPs and FNs do not always have the same costs!

Example from book

(a) *k*-NN model

		Prediction 'good'	Prediction 'bad'
Target	'good'	57	3
	'bad'	10	30

(b) decision tree

		Prediction 'good'	Prediction 'bad'
Target	'good'	43	17
	'bad'	3	37

two confusion matrices

		Prediction 'good'	Prediction 'bad'
Target	'good'	140	−140
	'bad'	−700	0

profit matrix

profits

(a) *k*-NN model

		Prediction 'good'	Prediction 'bad'
Target	'good'	7 980	−420
	'bad'	−7 000	0
	Profit		**560**

(b) decision tree

		Prediction 'good'	Prediction 'bad'
Target	'good'	6 020	−2 380
	'bad'	−2 100	0
	Profit		**1 540**

Receiver Operating Characteristic (ROC) Curves

Prediction score

- What if the outcome is not binary, but a value between 0 and 1.
- 0 = “pretty sure” the predicted class is negative
- 1 = “pretty sure” the predicted class is positive
- 0.5 = ??

Where to put the threshold?

Threshold = 0.5

ID	Target	Pred-iction	Score	Out-come
7	ham	ham	0.001	TN
11	ham	ham	0.003	TN
15	ham	ham	0.059	TN
13	ham	ham	0.064	TN
19	ham	ham	0.094	TN
12	spam	ham	0.160	FN
2	spam	ham	0.184	FN
3	ham	ham	0.226	TN
16	ham	ham	0.246	TN
1	spam	ham	0.293	FN

More TP (and FP)
Less FN (and TN)

ID	Targe	[illegible]	e	Out-come
5	ham	[illegible]	2	TN
14	ham	ham	0.348	TN
17	ham	spam	0.657	FP
8	spar	[illegible]	[illegible]	TP
6	spar	[illegible]	[illegible]	TP
10	[illegible]	[illegible]	[illegible]	TP
18	spam	[illegible]	.833	TP
20	ham	spam	0.877	FP
9	spam	spam	0.960	TP
4	spam	spam	0.963	TP

More TN (and FN)
Less FP (and TP)

Chair of Process and Data Science

Playing with the threshold

(a) Threshold: 0.75

		Prediction 'spam'	Prediction 'ham'
Target	'spam'	4	4
	'ham'	2	10

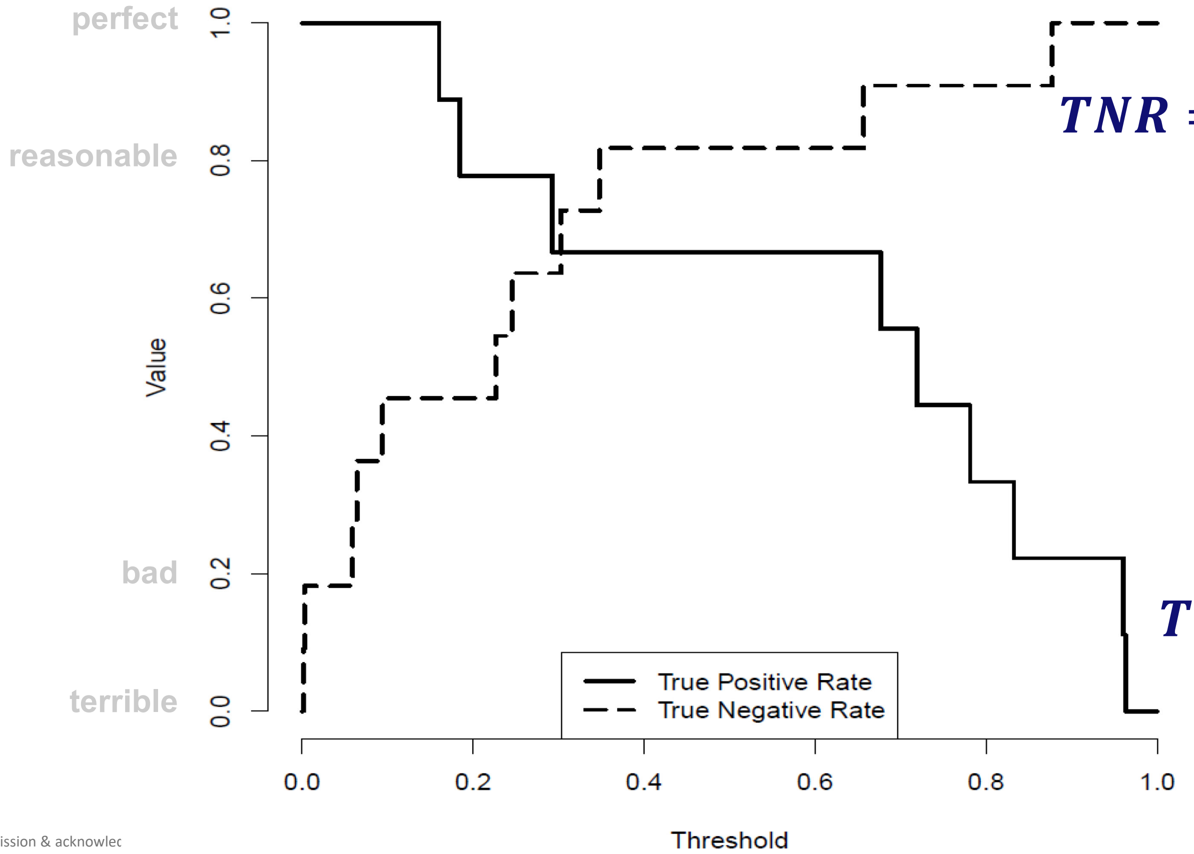
(b) Threshold: 0.25

		Prediction 'spam'	Prediction 'ham'
Target	'spam'	7	2
	'ham'	4	7

ID	Target	Score	Pred. (0.10)	Pred. (0.25)	Pred. (0.50)	Pred. (0.75)	Pred. (0.90)
7	ham	0.001	ham	ham	ham	ham	ham
11	ham	0.003	ham	ham	ham	ham	ham
15	ham	0.059	ham	ham	ham	ham	ham
13	ham	0.064	ham	ham	ham	ham	ham
19	ham	0.094	ham	ham	ham	ham	ham
12	spam	0.160	spam	ham	ham	ham	ham
2	spam	0.184	spam	ham	ham	ham	ham
3	ham	0.226	spam	ham	ham	ham	ham
16	ham	0.246	spam	ham	ham	ham	ham
1	spam	0.293	spam	spam	ham	ham	ham
5	ham	0.302	spam	spam	ham	ham	ham
14	ham	0.348	spam	spam	ham	ham	ham
17	ham	0.657	spam	spam	spam	ham	ham
8	spam	0.676	spam	spam	spam	ham	ham
6	spam	0.719	spam	spam	spam	ham	ham
10	spam	0.781	spam	spam	spam	spam	ham
18	spam	0.833	spam	spam	spam	spam	ham
20	ham	0.877	spam	spam	spam	spam	ham
9	spam	0.960	spam	spam	spam	spam	spam
4	spam	0.963	spam	spam	spam	spam	spam
Misclassification Rate			0.300	0.300	0.250	0.300	0.350
True Positive Rate (TPR)			1.000	0.778	0.667	0.444	0.222
True Negative rate (TNR)			0.455	0.636	0.818	0.909	1.000
False Positive Rate (FPR)			0.545	0.364	0.182	0.091	0.000
False Negative Rate (FNR)			0.000	0.222	0.333	0.556	0.778

ID	Target	Score	Pred. (0.10)	Pred. (0.25)	Pred. (0.50)	Pred. (0.75)	Pred. (0.90)
7	ham	0.001	ham	ham	ham	ham	ham
11	ham	0.003	ham	ham	ham	ham	ham
15	ham	0.059	ham	ham	ham	ham	ham
13	ham	0.064	ham	ham	ham	ham	ham
19	ham	0.094	ham	ham	ham	ham	ham
12	spam	0.160	spam	ham	ham	ham	ham
2	spam	0.184	spam	ham	ham	ham	ham
3	ham	0.226	spam	ham	ham	ham	ham
16	ham	0.246	spam	ham	ham	ham	ham
1	spam	0.293	spam	spam	ham	ham	ham
5	ham	0.302	spam	spam	ham	ham	ham
14	ham	0.348	spam	spam	ham	ham	ham
17	ham	0.657	spam	spam	spam	ham	ham
8	spam	0.676	spam	spam	spam	ham	ham
6	spam	0.719	spam	spam	spam	ham	ham
10	spam	0.781	spam	spam	spam	spam	ham

Playing with the threshold

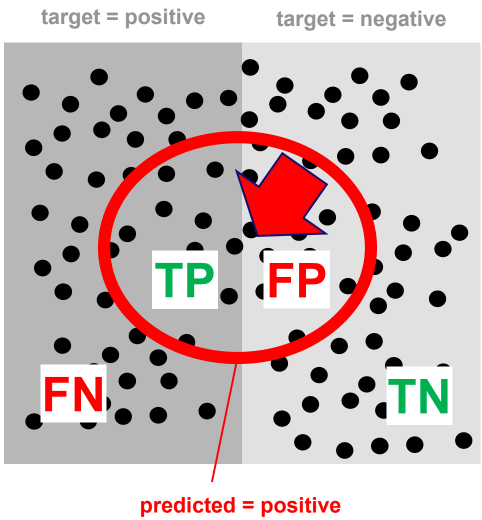


$$TNR = \frac{TN}{TN + FP}$$

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

Chair of Process and Data Science

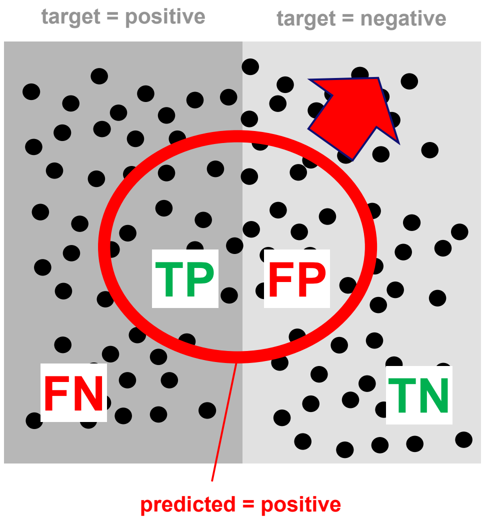
Playing with the threshold



$TNR = \frac{TN}{TN+FP}$ goes up

$FPR = \frac{FP}{TN+FP}$ goes down

$TPR = \frac{TP}{TP+FN}$ goes down



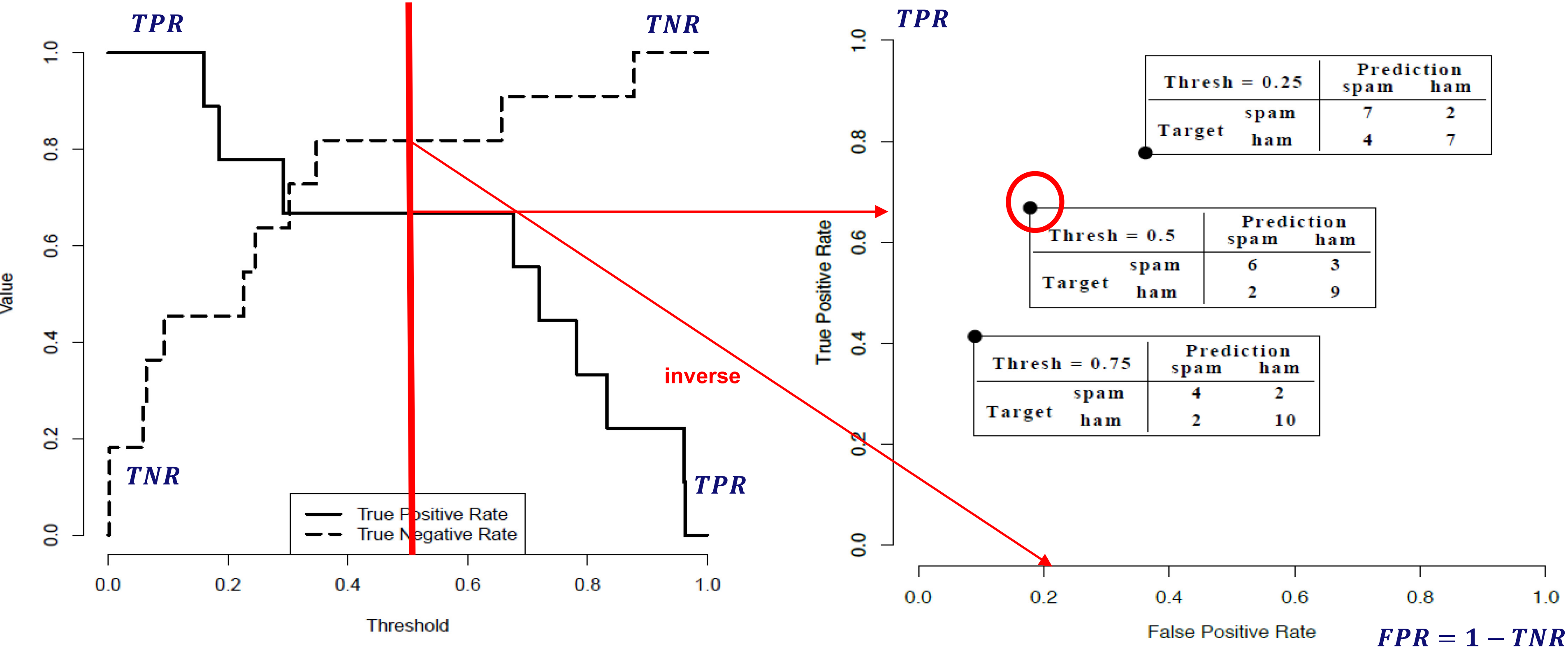
$TNR = \frac{TN}{TN+FP}$ goes down

$FPR = \frac{FP}{TN+FP}$ goes up

$TPR = \frac{TP}{TP+FN}$ goes up

Chair of Process and Data Science

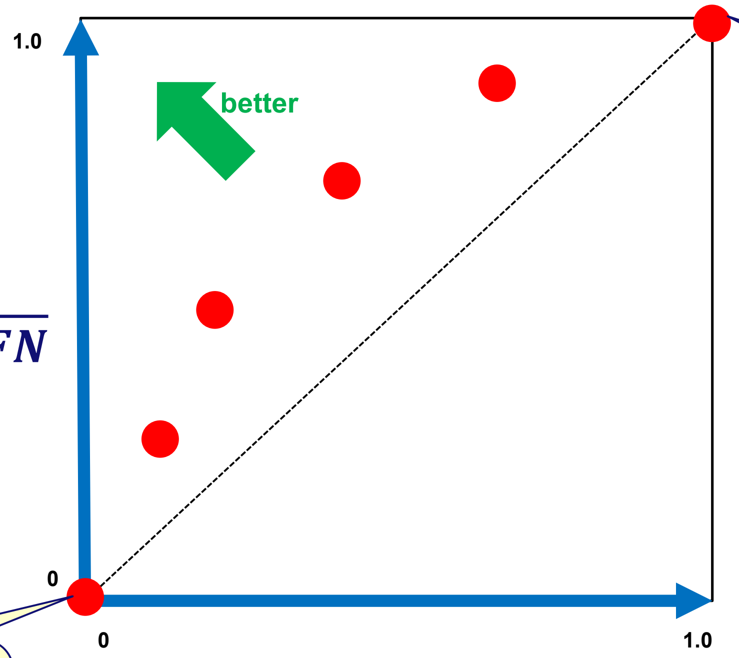
Trade-off



$FPR = 1 - TNR$

Receiver Operating Characteristic (ROC) Curve

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



predict all to be true

$TN = 0 \Rightarrow FPR = 1$

$FN = 0 \Rightarrow TPR = 1$

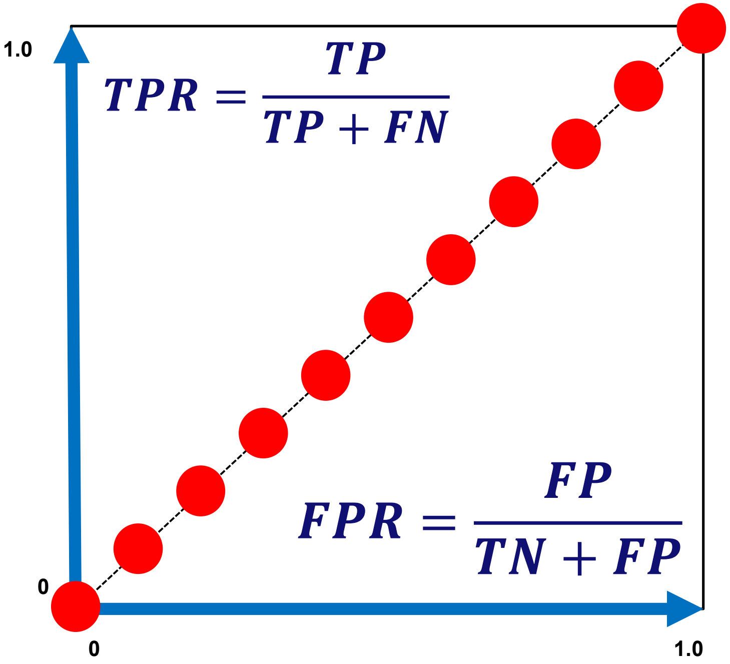
predict all to be false

$FP = 0 \Rightarrow FPR = 0$

$TP = 0 \Rightarrow TPR = 0$

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

Receiver Operating Characteristic (ROC) Curve: Beating random guessing

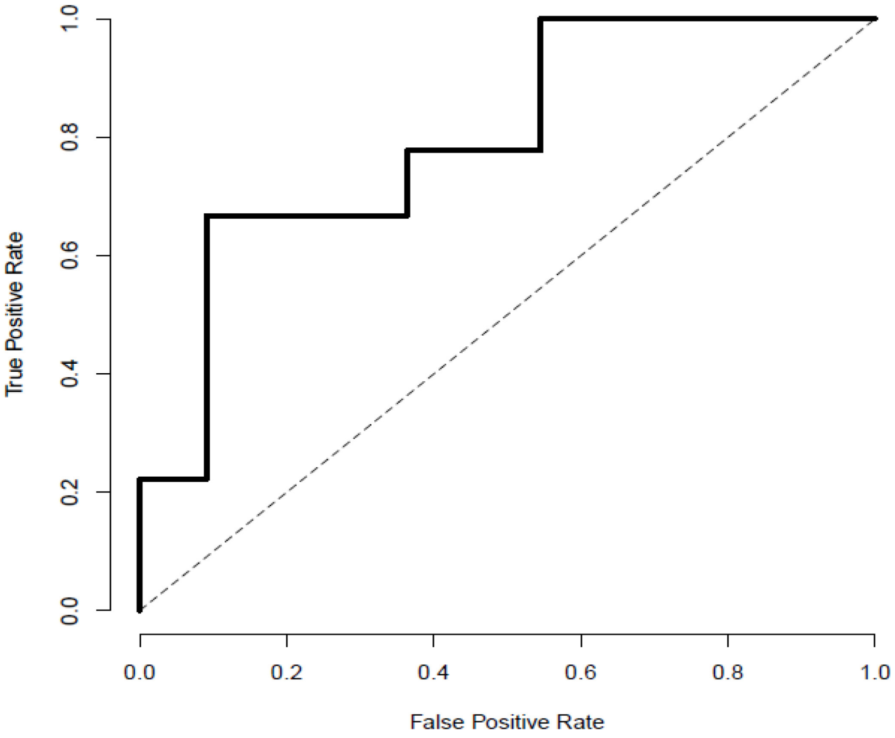


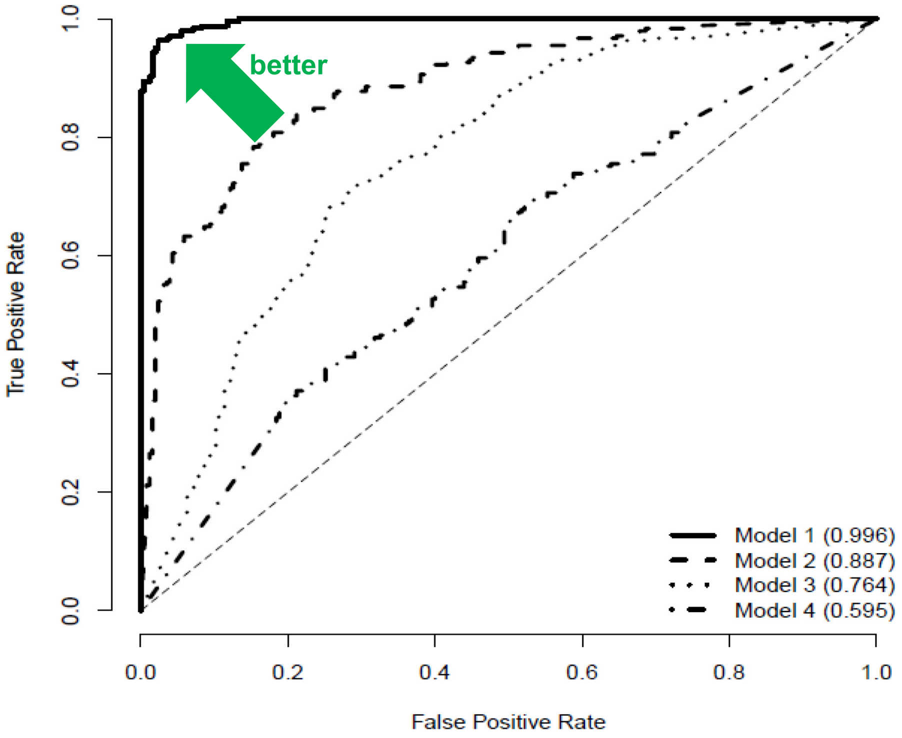
- Assume fraction q is actually positive and fraction $1-q$ is negative.
- Guess with probability p that an instance is positive and with probability $1-p$ negative.
- $TP = pq$, $TN = (1-p)(1-q)$, $FP = p(1-q)$, $FN = (1-p)q$ (just fractions; multiply with N)
- $TPR = \frac{TP}{TP+FN} = \frac{pq}{pq+(1-p)q} = p.$
- $FPR = \frac{FP}{TN+FP} = \frac{p(1-q)}{(1-p)(1-q)+p(1-q)} = p.$

Hence, $TPR = FPR = p$ for any p.

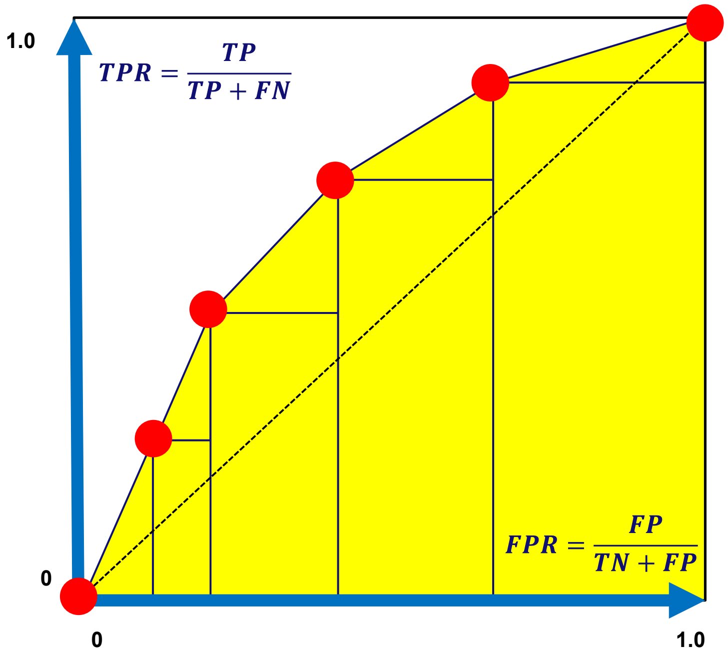
Chair of Process and Data Science

Example ROC curves





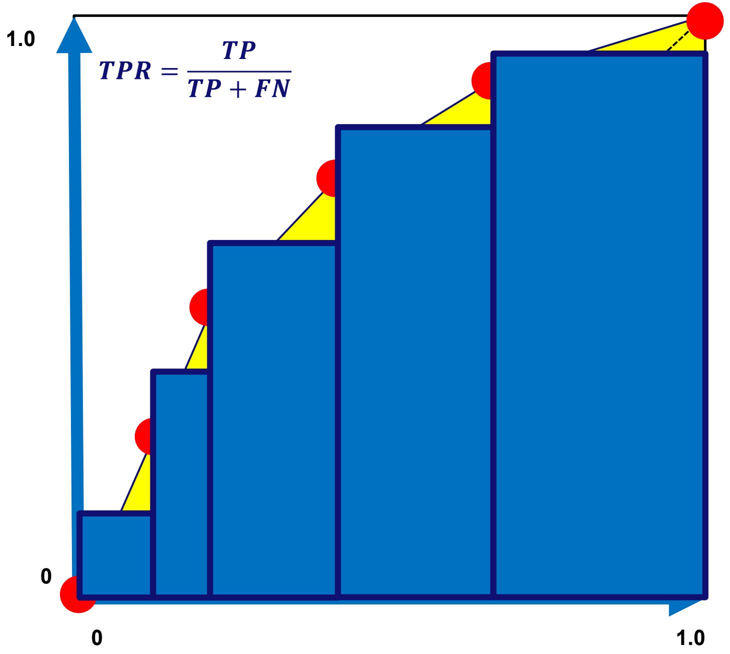
Area Under the Curve (AUC)



$$\text{ROC index} = \sum_{i=2}^{|\mathbf{T}|} \frac{(FPR(\mathbf{T}[i]) - FPR(\mathbf{T}[i-1])) \times (TPR(\mathbf{T}[i]) + TPR(\mathbf{T}[i-1]))}{2}$$

Chair of Process and Data Science

Area Under the Curve (AUC)



$$\text{ROC index} = \sum_{i=2}^{|\mathbf{T}|} \frac{(FPR(\mathbf{T}[i]) - FPR(\mathbf{T}[i-1])) \times (TPR(\mathbf{T}[i]) + TPR(\mathbf{T}[i-1]))}{2}$$

Multinomial targets

Confusion matrix

		Prediction				Recall
		'durionis'	'ficulneus'	'fructosus'	'pseudo.'	
Target	'durionis'	5	0	2	0	0.714
	'ficulneus'	0	6	1	0	0.857
	'fructosus'	0	1	10	0	0.909
	'pseudo.'	0	0	2	3	0.600
	Precision	1.000	0.857	0.667	1.000	

$$\text{precision(l)} = \frac{TP(l)}{TP(l) + FP(l)}$$

$$\text{recall(l)} = \frac{TP(l)}{TP(l) + FN(l)}$$

Confusion matrix

		Prediction				Recall
		'durionis'	'ficulneus'	'fructosus'	'pseudo.'	
Target	'durionis'	5	0	2	0	0.714
	'ficulneus'	0	6	1	0	0.857
	'fructosus'	0	1	10	0	0.909
	'pseudo.'	0	0	2	3	0.600
	Precision	1.000	0.857	0.667	1.000	

$$\text{precision}(l) = \frac{TP(l)}{TP(l) + FP(l)}$$

$$\text{recall}(l) = \frac{TP(l)}{TP(l) + FN(l)}$$

$$\boldsymbol{precision(fructosus) = \frac{10}{10 + (2 + 1 + 2)} = \frac{10}{15} = 0.667}$$

Chair of Process and Data Science

Confusion matrix

		Prediction				Recall
		'durionis'	*'ficulneus'*	*'fructosus'*	*'pseudo.'*	
Target	*'durionis'*	5	0	2	0	0.714
	'ficulneus'	0	6	1	0	0.857
	'fructosus'	0	1	10	0	0.909
	'pseudo.'	0	0	2	3	0.600
	Precision	1.000	0.857	0.667	1.000	

$$\text{precision(l)} = \frac{TP(l)}{TP(l) + FP(l)}$$

$$\text{recall(l)} = \frac{TP(l)}{TP(l) + FN(l)}$$

$$\boldsymbol{recall(fructosus) = \frac{10}{10 + (1)} = \frac{10}{11} = 0.909}$$

Chair of Process and Data Science

Continuous target features

Sum of squared errors (we have seen this before)

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

actual target value

predicted value

descriptive features

Mean squared error (we have seen this before)

actual target value

predicted value

descriptive features

$$\text{mean squared error} = \frac{\sum_{i=1}^{n} (t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}$$

Chair of Process and Data Science

Root Mean Squared Error (RMSE)

$$\text{root mean squared error} = \sqrt{\frac{\sum_{i=1}^{n} (t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}}$$

Chair of Process and Data Science

Mean Absolute Error (MAE)

$$\text{mean absolute error} = \frac{\sum_{i=1}^{n} abs(t_i - \mathbb{M}(\mathbf{d}_i))}{n}$$

Easy interpretation: average error

… but absolute error

R^2 coefficient (1= perfect and 0=terrible)

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

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

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

Compare performance with guessing the overall average.

Chair of Process and Data Science

R^2 coefficient: Scale [0,1]

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

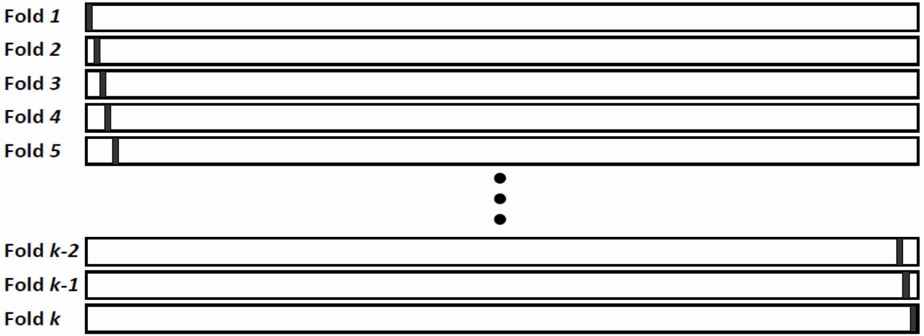
$R^2 = 0$ implies that one is not better than the baseline (guessing the average)

$R^2 = 1$ implies all predictions are perfect

Chair of Process and Data Science

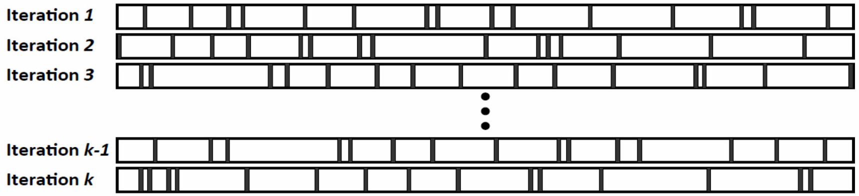
Cross Validation (as before)

k-fold cross validation



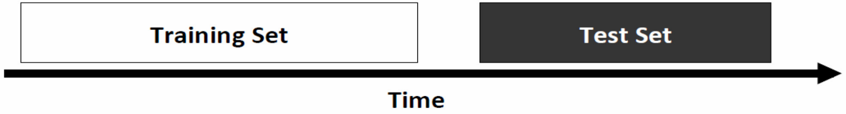
leave-one-out cross validation

(k times leave out 1)



bootstrapping

(k times leave out random m)



out-of-time sampling

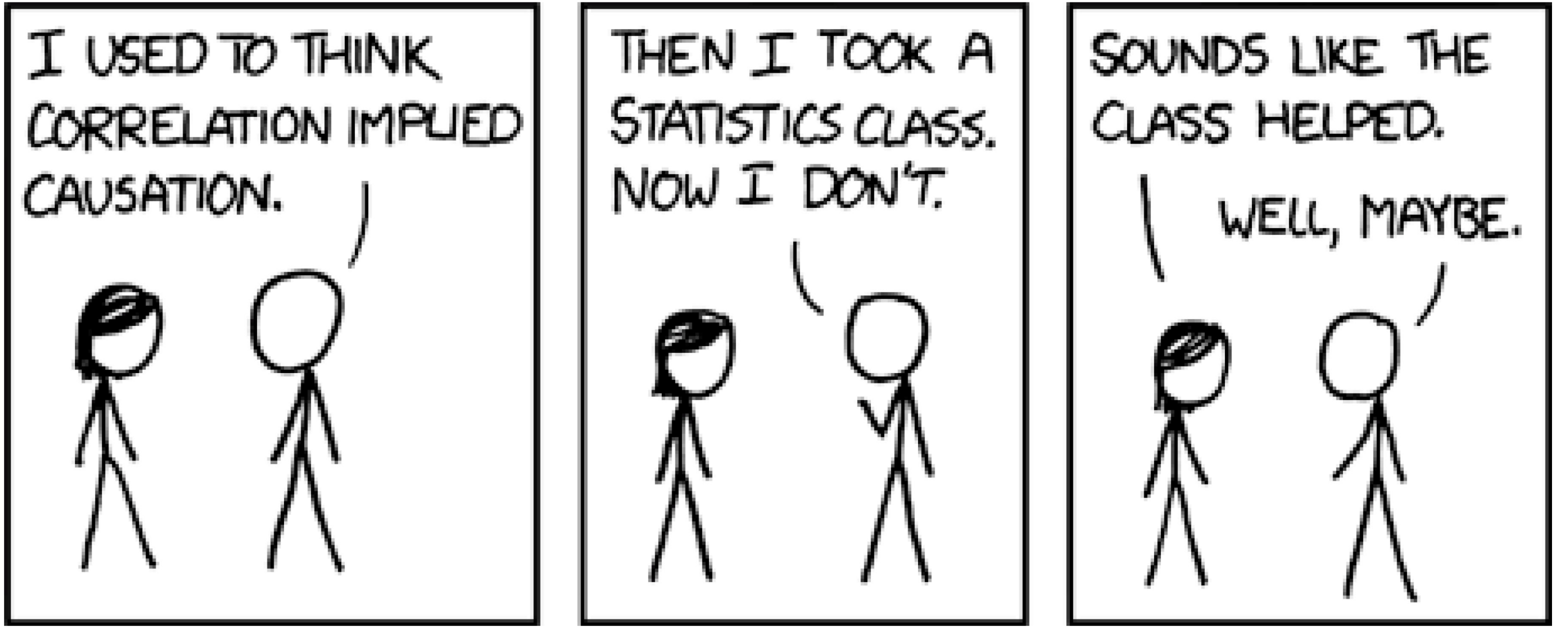
Chair of Process and Data Science

A/B Testing

Turning predictions into recommendations

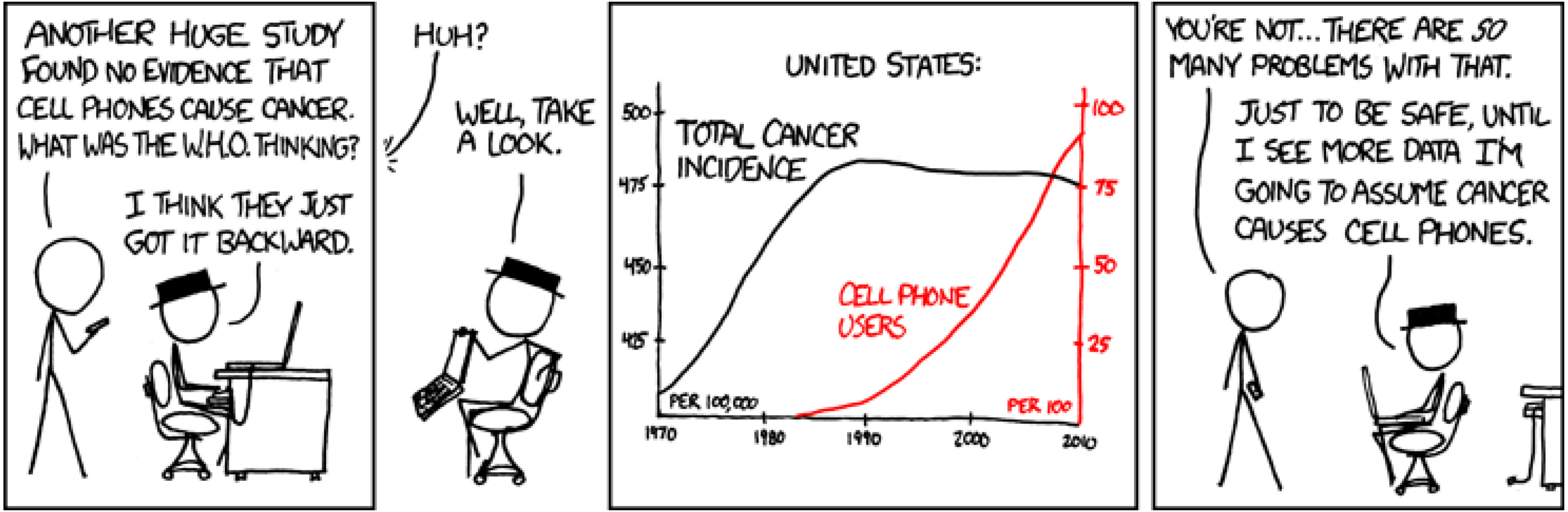
- Supervised learning helps to turn data into predictive models.
- Predictions can be turned into recommendations:
  - We predict that you will fail, so start working …
  - If you “change this descriptive feature”, we predict a more desirable outcome.
  - Customers like you ended up buying …
- Issues:
  - Cause and effect? What is controllable and what not?
    - Wealthy people drive Porsches, so let’s buy one.
    - If I wear sunglasses, it will not rain.
  - Recommendations change the reality they are based on.

Correlation does not imply causation



Randall Munroe, CC BY-NC 2.5, https://xkcd.com/552/

Correlation does not imply causation



Randall Munroe, CC BY-NC 2.5, https://xkcd.com/925/

Chair of Process and Data Science

A/B testing

- Offer randomly two variants: A and B.
- For example, intervene based on prediction or not.
- Conduct statistical hypothesis testing.

Chair of Process and Data Science

Concept drift

Concept drift: Process changes over time

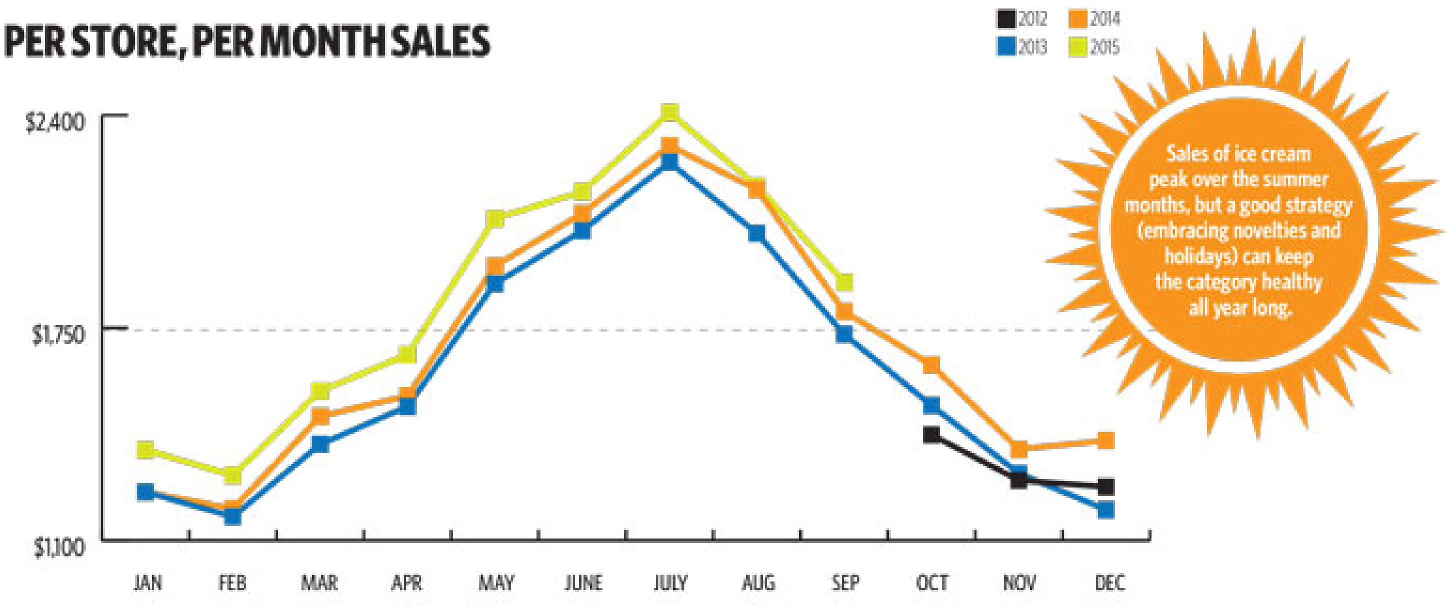
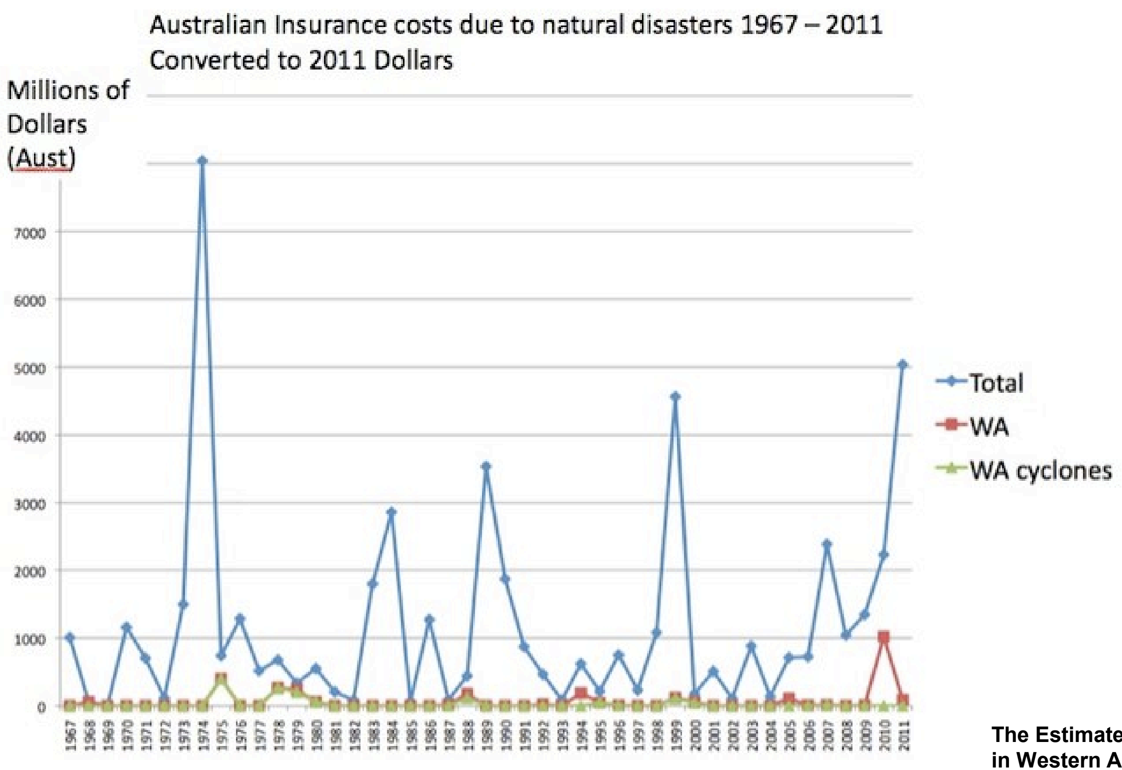


Figure by Sarah Hamaker, NACS Magazine, 2014

Chair of Process and Data Science

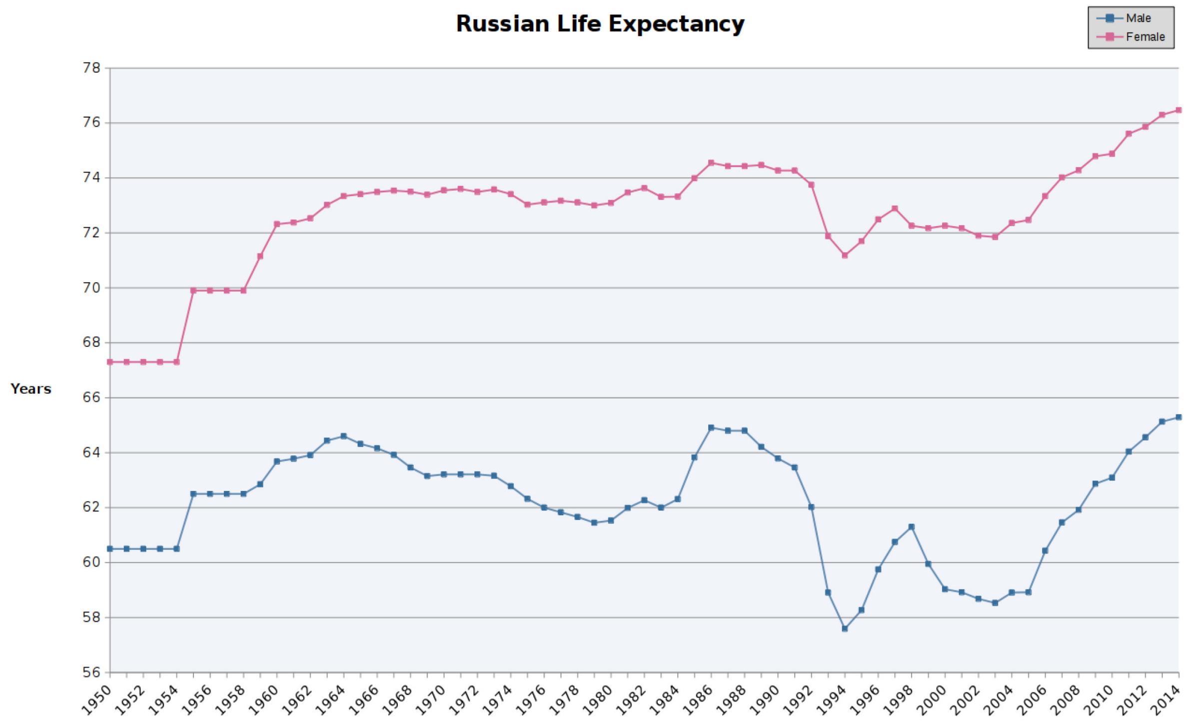
Concept drift: Process changes over time



The Estimated Cost of Tropical Cyclone Impacts in Western Australia, July 2012, Government of Western Australia, John Mcbride

Chair of Process and Data Science

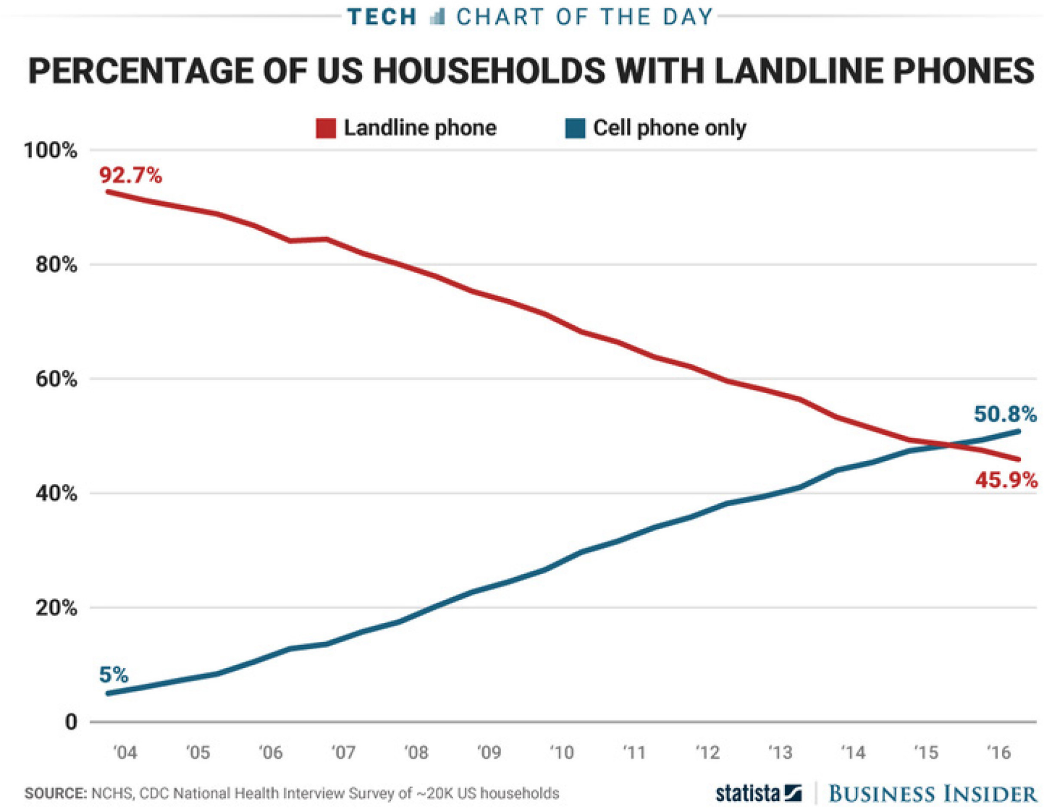
Concept drift: Process changes over time



Russian male and female life expectancy from 1950-2014. Rosstat.

Chair of Process and Data Science

Concept drift: Process changes over time



Chair of Process and Data Science

System Stability Index (SSI)

$$\text{stability index} = \sum_{l \in levels(t)} \left(\left(\frac{|\mathcal{A}_{t=l}|}{|\mathcal{A}|} - \frac{|\mathcal{B}_{t=l}|}{|\mathcal{B}|} \right) \times log_e \left(\frac{|\mathcal{A}_{t=l}|}{|\mathcal{A}|} / \frac{|\mathcal{B}_{t=l}|}{|\mathcal{B}|} \right) \right)$$

fraction of instances in the original test set classified as l

fraction of instances in the new data set classified as l

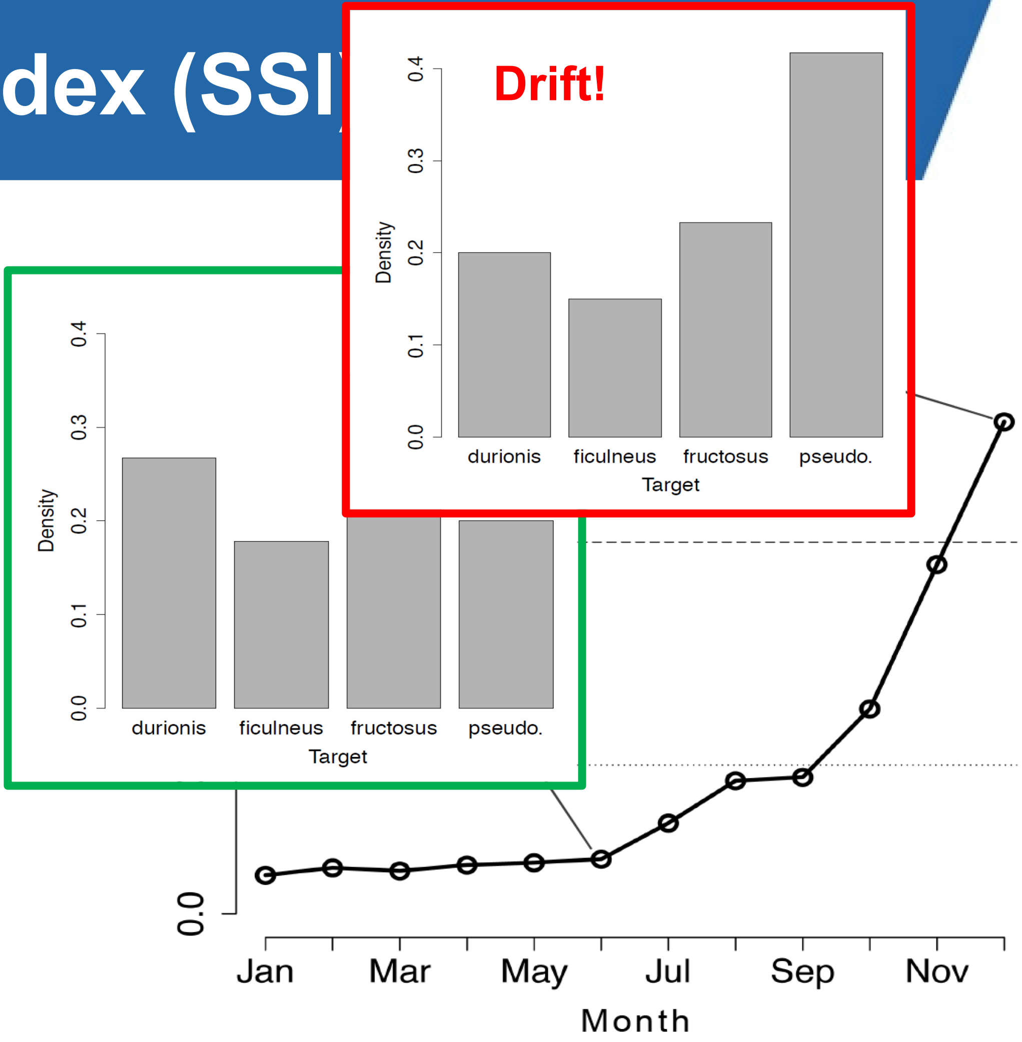
Informal interpretation:

- **SSI < 0.10: no significant drift**
- **0.10 ≤ SSI < 0.25: moderate drift**
- **SSI ≥ 0.25: significant drift**

Chair of Process and Data Science

System Stability Index (SSI)

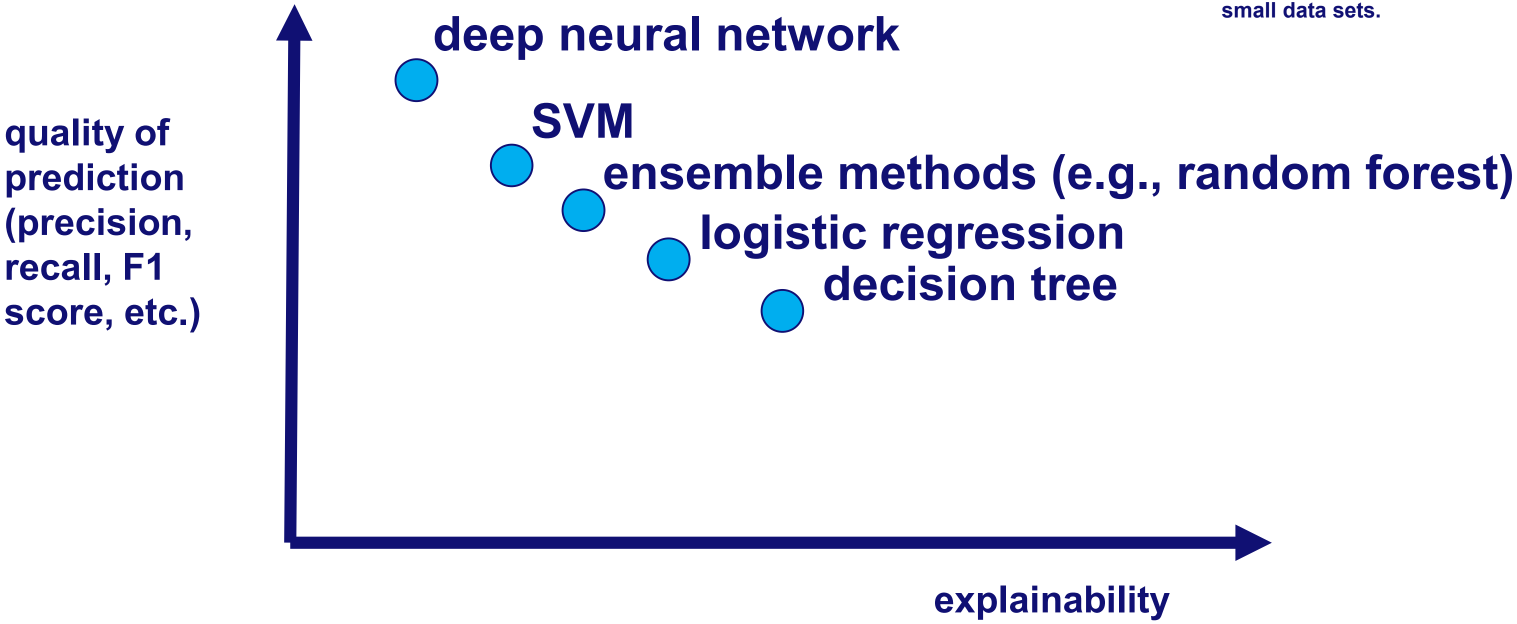
original distribution



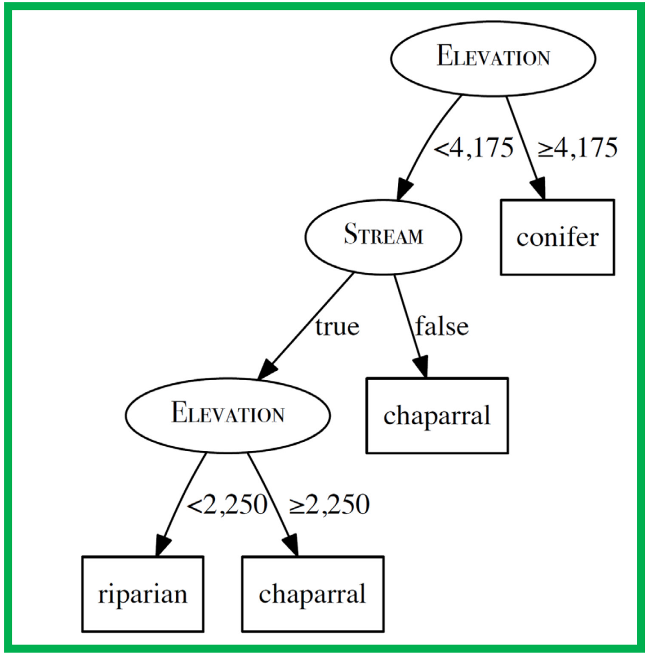
Explainability

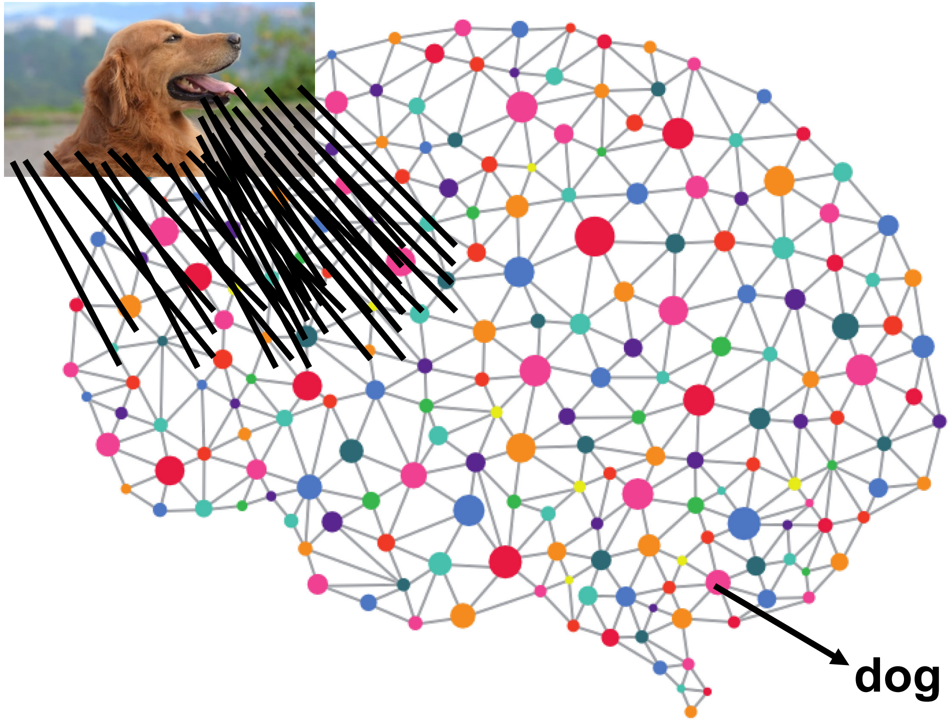
There may be a trade-off

Positions are just an indication and this depends on may factors (data set, parameters, etc.). Note that deep neural networks can perform poorly on small data sets.



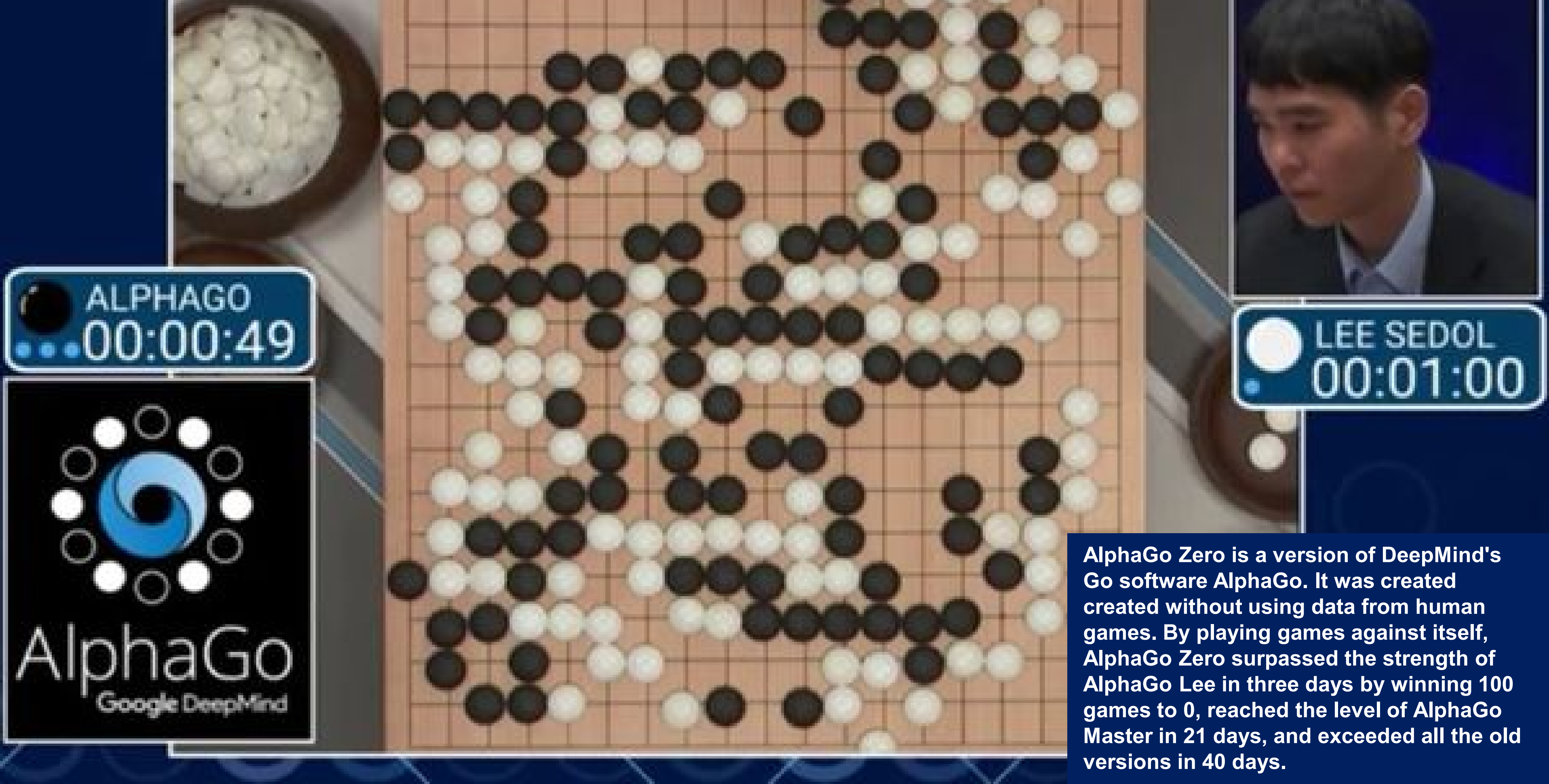
Chair of Process and Data Science





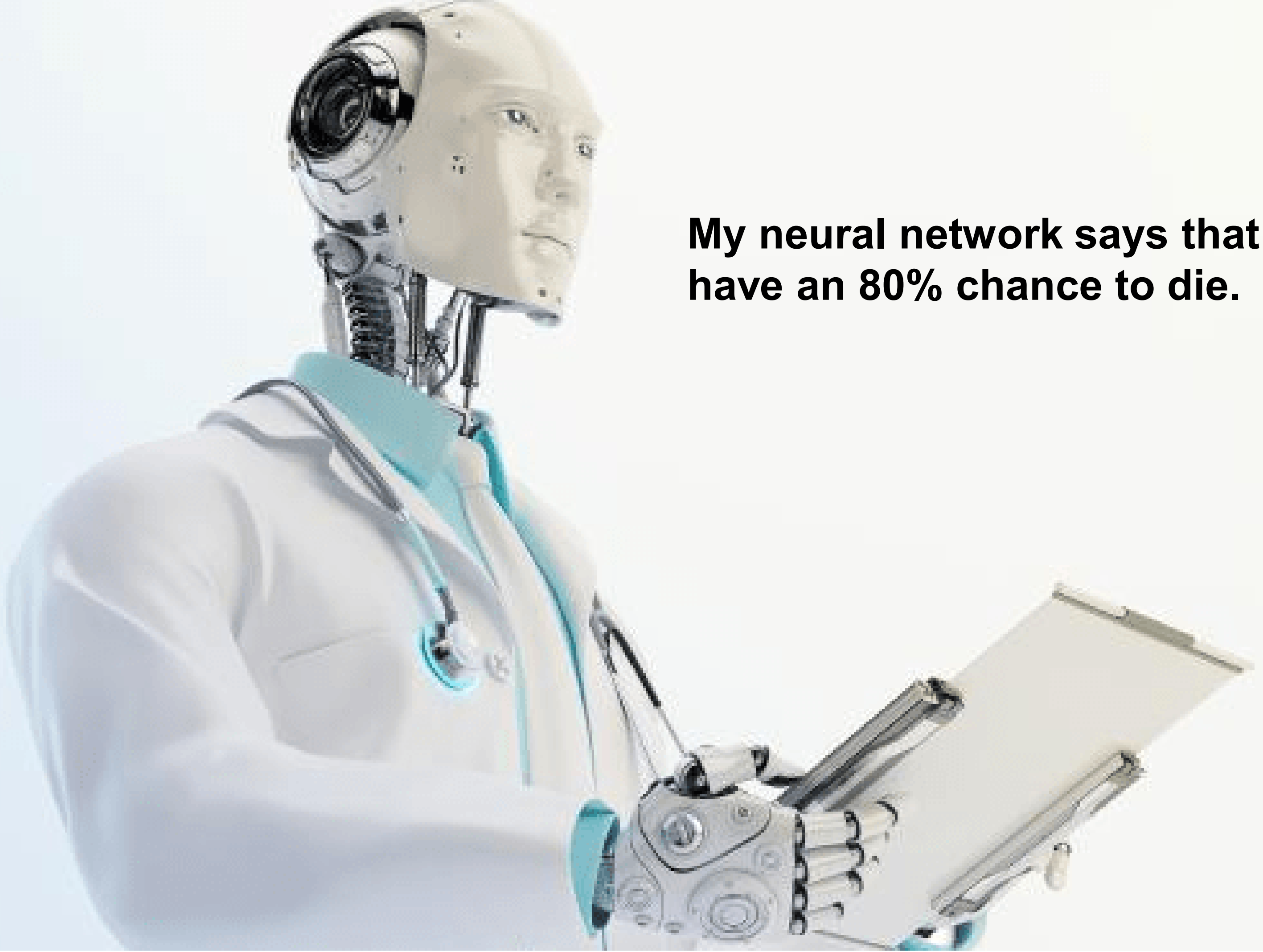
$$\textsc{Growth} = 0.3707 \times \phi_0(\textsc{Rain}) + 0.8475 \times \phi_1(\textsc{Rain}) + -1.717 \times \phi_2(\textsc{Rain})$$

AlphaGo vs Lee Sedol 4:1 (March 2016)



AlphaGo Zero is a version of DeepMind's Go software AlphaGo. It was created created without using data from human games. By playing games against itself, AlphaGo Zero surpassed the strength of AlphaGo Lee in three days by winning 100 games to 0, reached the level of AlphaGo Master in 21 days, and exceeded all the old versions in 40 days.


Guilty because of X & Y & Z
If X & Y & Z, then guilty.



My neural network says that you have an 80% chance to die.

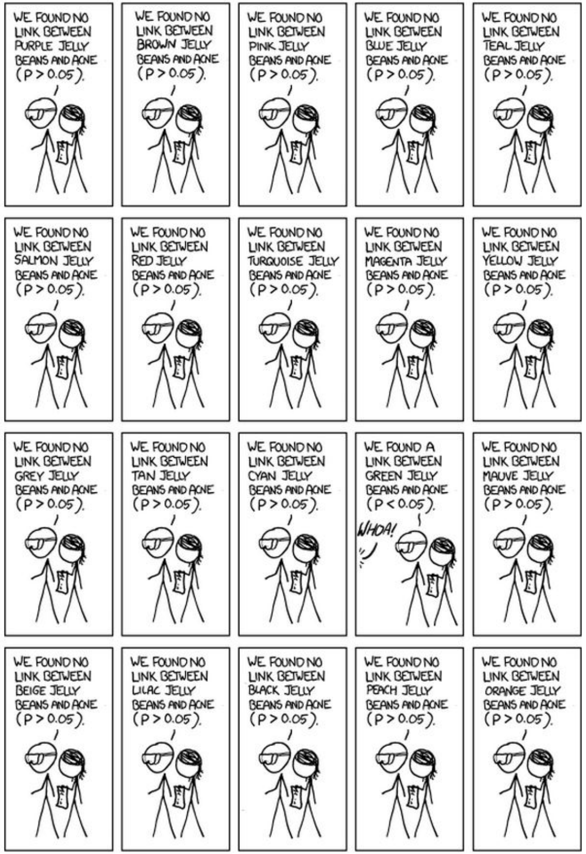
Conclusion

Short summary of lecture

Torture the data, and it will confess to anything!

Ronald Harry Coase (1910-2013)
British economist.

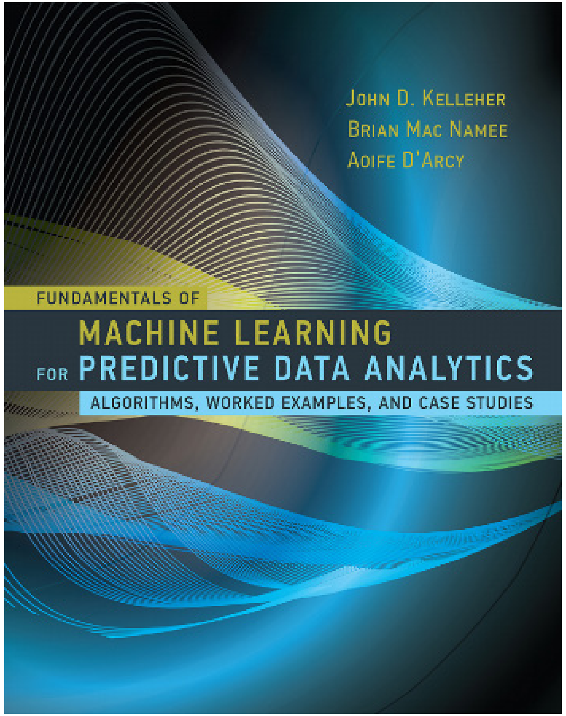
cross validation is essential !



Key concepts

- Different forms of CV: k-fold cross validation, leave-one-out cross validation, bootstrapping, out-of-time sampling.
- Confusion matrix
- TP, TN, FP, FN, accuracy, precision, recall, F_1 score, etc.
- ROC curve, AUC, etc.
- MAE, RMSE, R^2, etc.
- Drift detection, A/B testing, etc.

Relevant Literature



Chapter 8 of Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

First assignment handed out tomorrow

#	Lecture	date	day
Lecture 1	Introduction	10/10/2018	Wednesday
Lecture 2	Crash Course in Python	11/10/2018	Thursday
Instruction 1	*Python*	*12/10/2018*	*Friday*
Lecture 3	Basic data visualisation/exploration	17/10/2018	Wednesday
Lecture 4	Decision trees	18/10/2018	Thursday
Instruction 2	*Decision trees and data visualization/exploration*	*19/10/2018*	*Friday*
Lecture 5	Regression	24/10/2018	Wednesday
Lecture 6	Support vector machines	25/10/2018	Thursday
Instruction 3	*Regression and support vector machines*	*26/10/2018*	*Friday*
Lecture 7	Neural networks (1/2)	31/10/2018	Wednesday
Instruction 4	*Neural networks and supervised learning*	*02/11/2018*	*Friday*
Lecture 8	Neural networks (2/2)	07/11/2018	Wednesday
Lecture 9	Evaluation of supervised learning problems	08/11/2018	Thursday
Instruction 5	*Neural networks and supervised learning*	*09/11/2018*	*Friday*
Lecture 10	Clustering	14/11/2018	Wednesday
Lecture 11	Frequent items sets	15/11/2018	Thursday
Lecture 12	Association rules	21/11/2018	Wednesday
Lecture 13	Sequence mining	22/11/2018	Thursday
Instruction 6	*Clustering, frequent items sets, association rules*	*23/11/2018*	*Friday*
Lecture 14	Process mining (unsupervised)	28/11/2018	Wednesday
Lecture 15	Process mining (supervised)	29/11/2018	Thursday
Instruction 7			
Lecture 1			
Instruction 8			
Lecture 1			
Lecture 1			
Lecture 1			
backup			
Instruction 9			
Lecture 2			
Lecture 2			
Instruction 10			
Lecture 2			
Lecture 2			
Instruction 11			
Lecture 2			
backup			
Instruction 12	*Example exam questions*	*25/01/2018*	*Friday*
backup		30/01/2019	Wednesday
backup		31/01/2019	Thursday
extra	*Question hour*	*01/02/2019*	*Friday*

Lecture 8	Neural networks (2/2)	07/11/2018	Wednesday
Lecture 9	Evaluation of supervised learning problems	08/11/2018	Thursday
Instruction 5	*Neural networks and supervised learning*	*09/11/2018*	*Friday*
Lecture 10	Clustering	14/11/2018	Wednesday
Lecture 11	Frequent items sets	15/11/2018	Thursday
Lecture 12	Association rules	21/11/2018	Wednesday
Lecture 13	Sequence mining	22/11/2018	Thursday
Instruction 6	*Clustering, frequent items sets, association rules*	*23/11/2018*	*Friday*