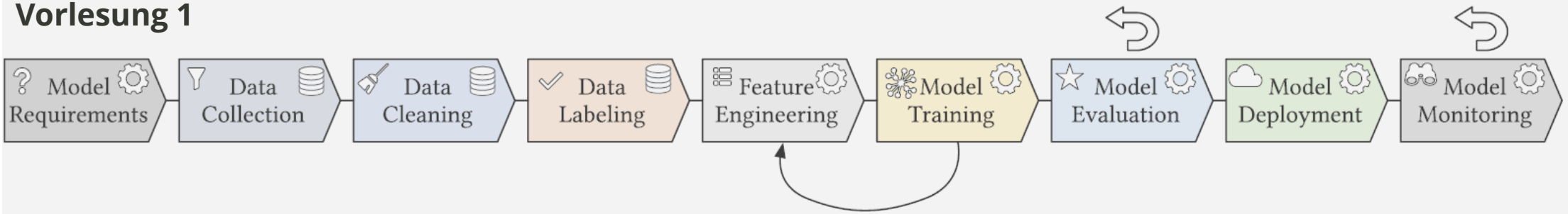
Dr. rer. nat. Valentin Khaydarov
Professur für Prozessleittechnik & Arbeitsgruppe Systemverfahrenstechnik

Klassifikation

Vorlesung 4, Lehrveranstaltung Experimentelle Prozessanalyse

Einordnung der Vorlesung

Vorlesung 1



Vorlesung 2

Vorlesung 3 – Regr.

Vorlesung 4 – Class.

Vorlesung 5 – Clust.

Vorlesung 6 - Zeitreihenanalyse

Vorlesung 7 – Neuronale Netze

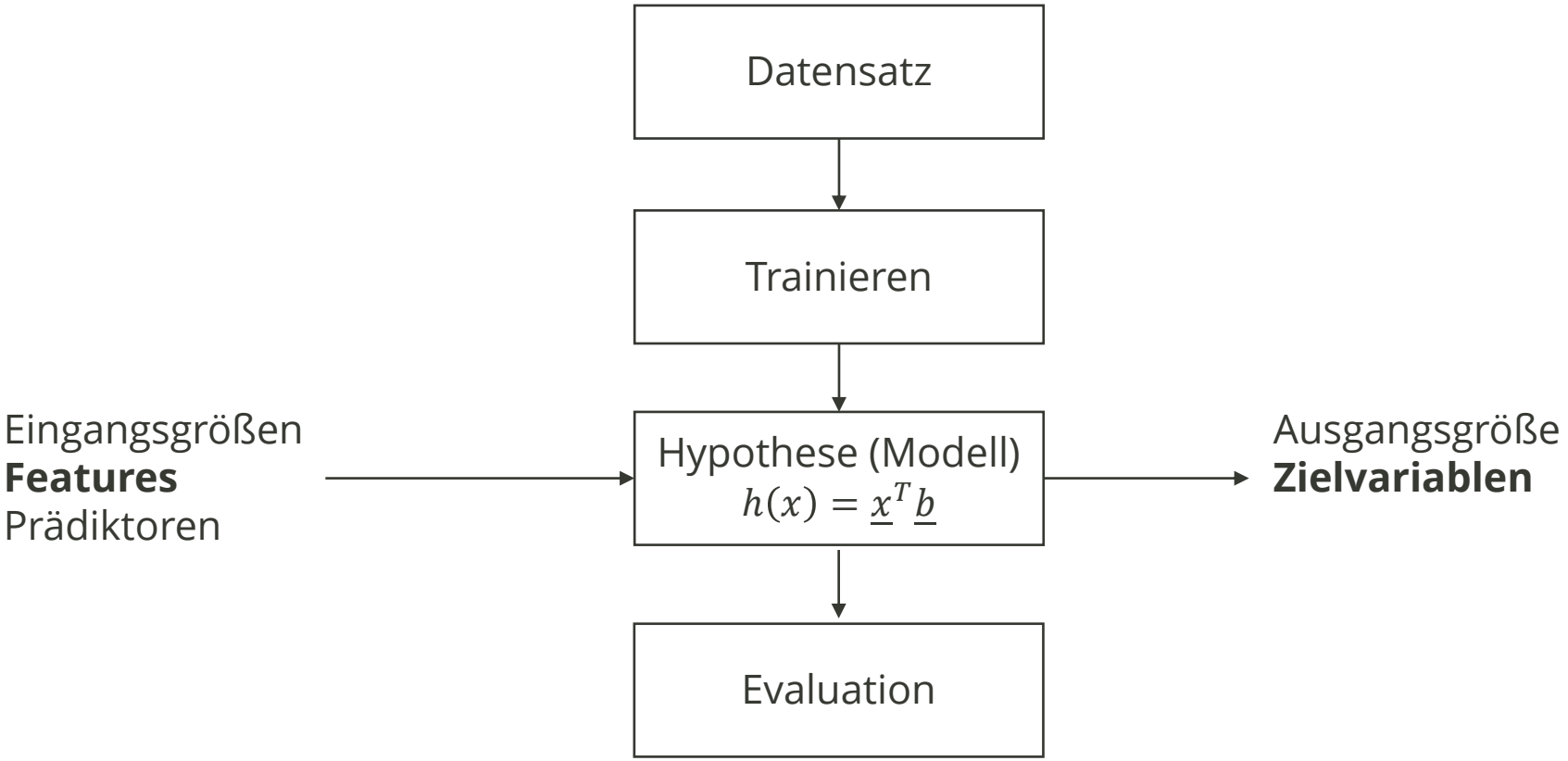
S. Amershi *et al.*, "Software Engineering for Machine Learning: A Case Study," in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, May 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042.

Agenda

- Wiederholung
- Logistische Regression
- Performance Metriken
- Klassifikatoren
  - Support Vector Machine
  - Decision Trees und Random Forests
- Multiclass, Multilabel und Multioutput
- Zusammenfassung und Ausblick

Wiederholung der letzten Vorlesung

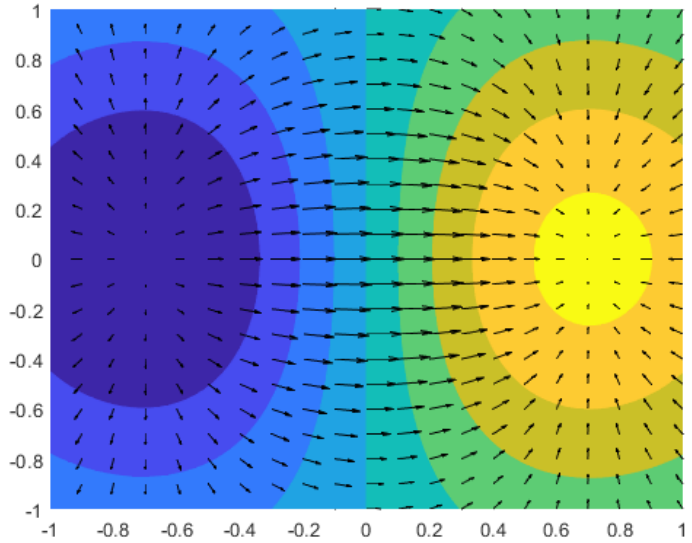
Regressionsanalyse



Numerische Optimierungsverfahren

Der Gradient einer Funktion $f(x)$:

$$grad\ f(x) = \nabla f(x) = \frac{df}{dx} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \dots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$



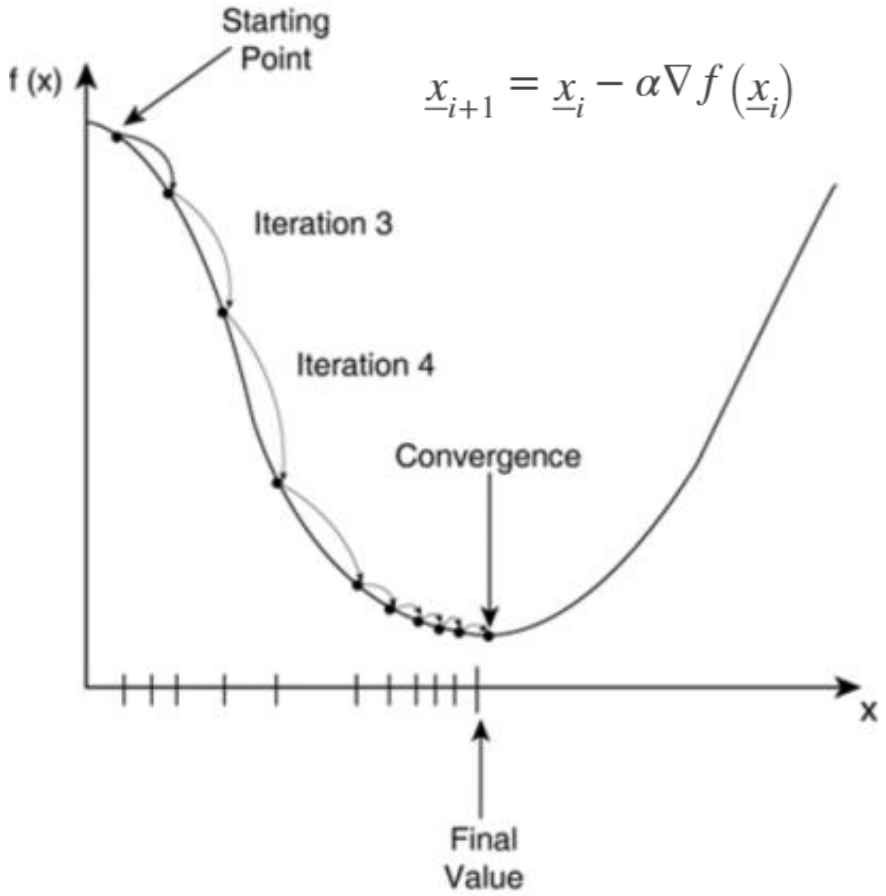
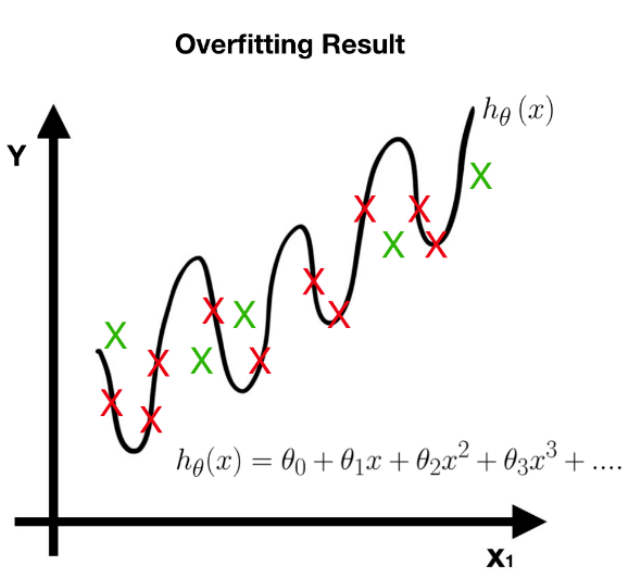
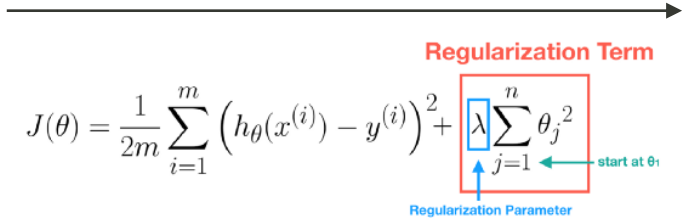
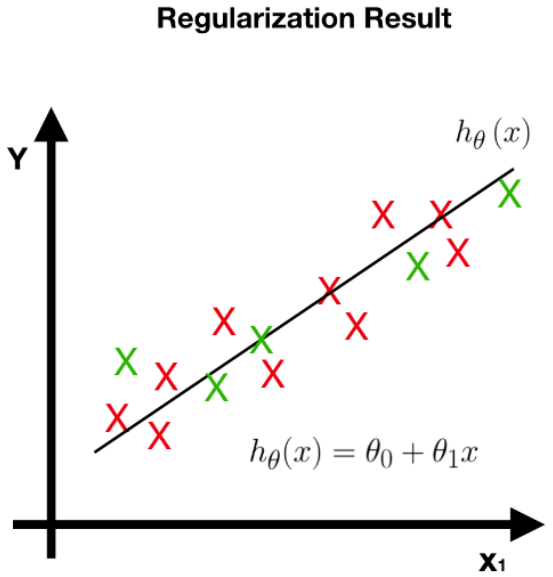


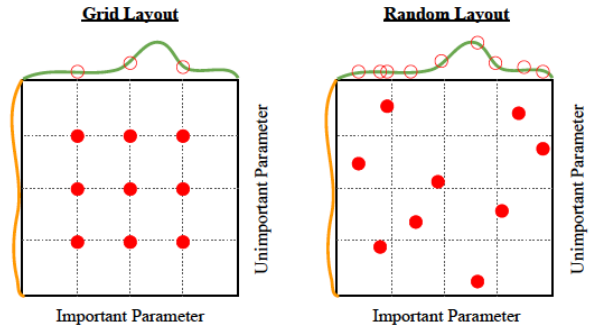
Bild: https://zkmablog.com/2017/02/11/was-ist-das-gradientenabstiegsverfahren/

Regularisierung und HPO





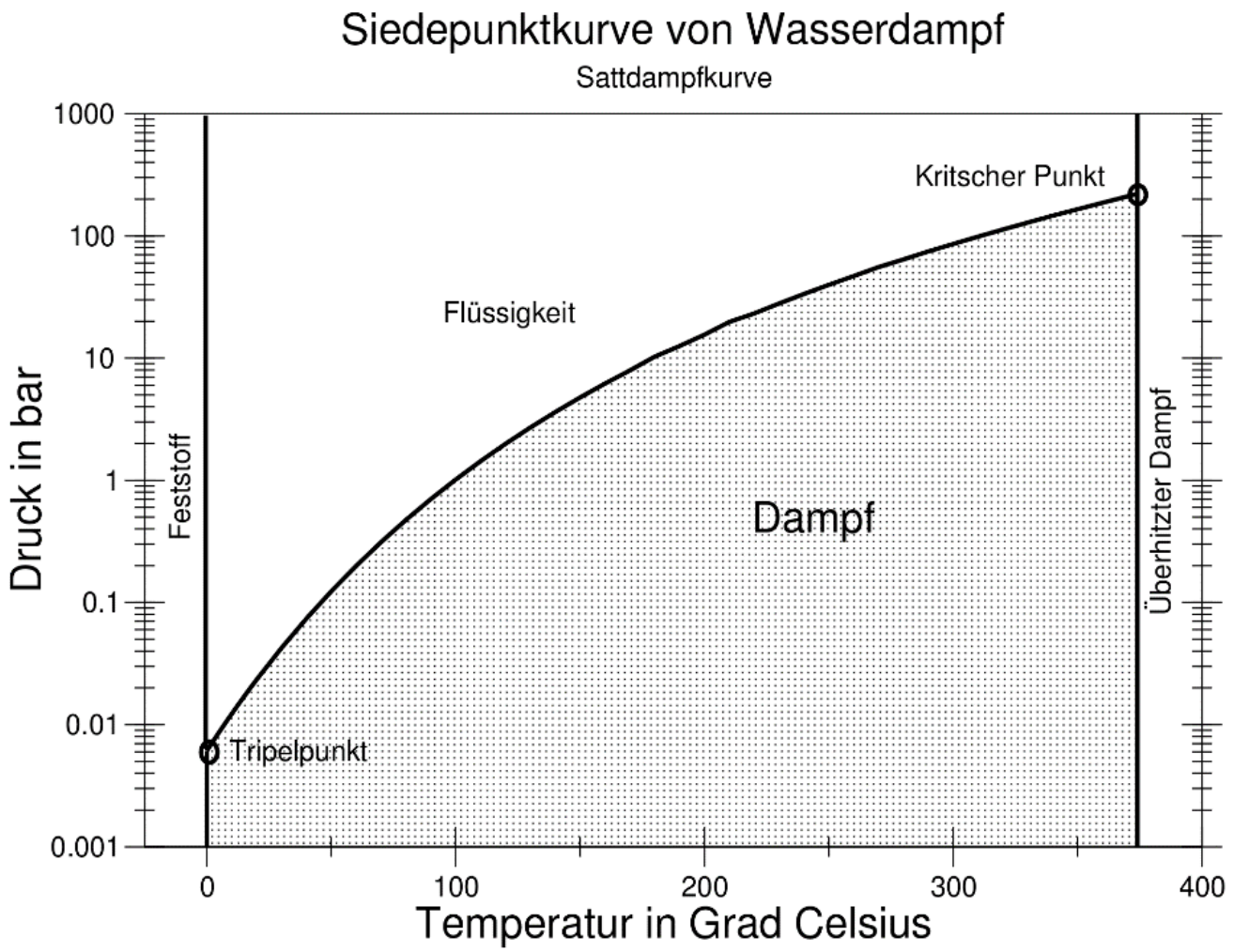




Bilder: https://medium.com/@qempsil0914/courseras-machine-learning-notes-week3-overfitting-and-regularization-partii-3e3f3f36a287

Logistische Regression

Beispiel: p-T Diagramm für Wasser



https://de.wikipedia.org/wiki/Datei:Dampfdruckkurve.svg

Binäre Klassifikation

Vorgegeben:

- n Beobachtungen mit m Eingangsgrößen $x_1, \ldots, x_m$ und einer Ausgangsgröße y
  - Eingangsgrößen können ebenfalls nominal oder ordinal sein
- eine Hypothese $y = f(b, x)$

Ziel: Ermittlung eines funktionalen Zusammenhang zwischen m Eingangsgrößen und einer ausgewählten Zielgröße y, wo y steht für eine Klasse

Beschränkung: **Zielvariable darf nur zwei Werte annehmen, entweder 0 oder 1**

Sigmoid-Funktion - Intuition

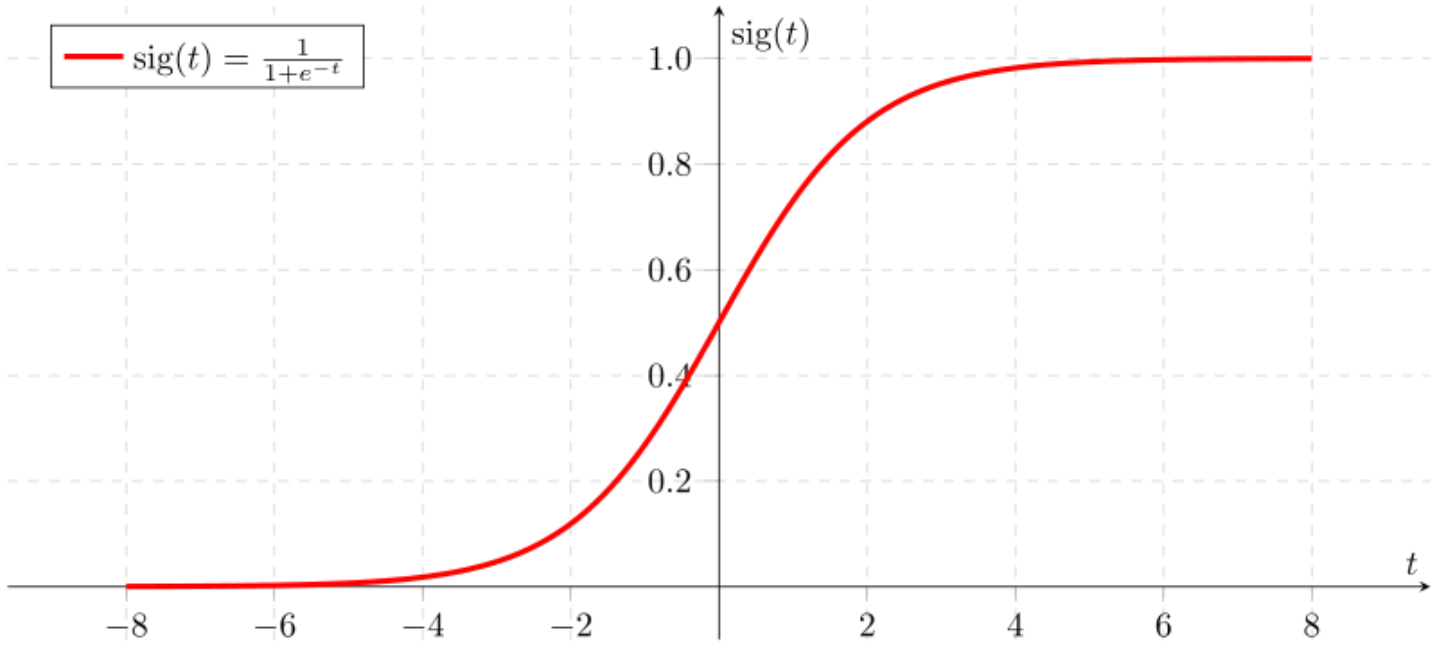
Problem beim Einsatz des Multiregressionsanalyse: Zielvariable darf Werte $(-\infty, +\infty)$ annehmen.

Ergebnis der Multiregressionsanalyse als **Wahrscheinlichkeit der Klassenzugehörigkeit**

→ **eine weitere Transformation erforderlich:** $(-\infty, +\infty) \rightarrow [0, 1]$

Sigmoid-Funktion

Logistische Funktion/Sigmoid



Fall 1 (Klasse 0):

$t \in (-\infty, 0)$, dann $\mathrm{sig}(t) < 0{,}5$

Fall 2 (Klasse 1):

$t \in (0, +\infty)$, dann $\mathrm{sig}(t) > 0{,}5$

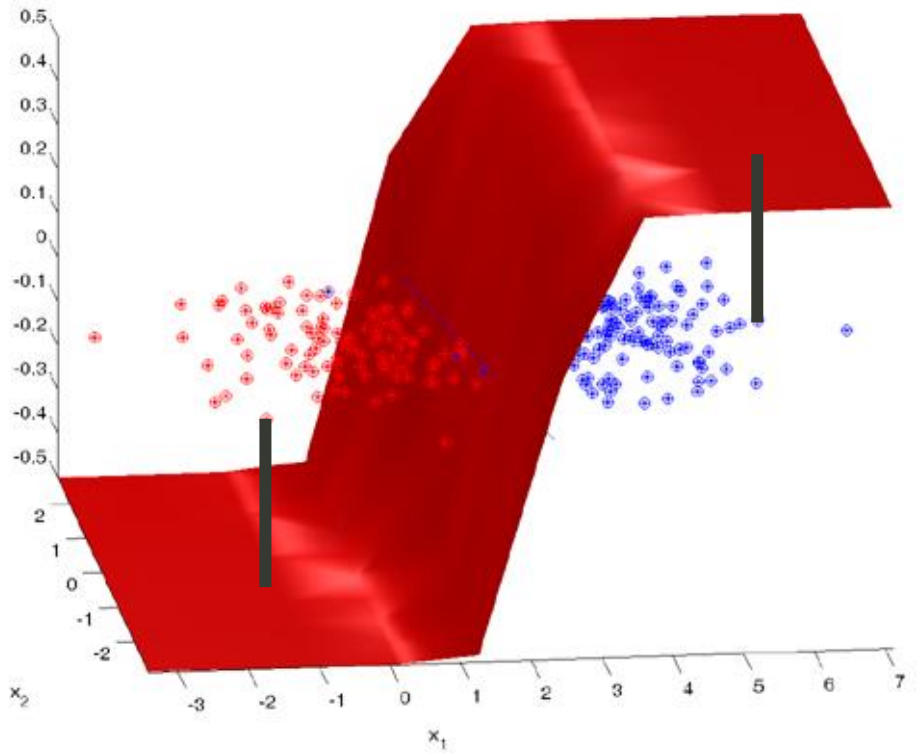
Fall 3 (gleichwertig Klasse 0 und 1):

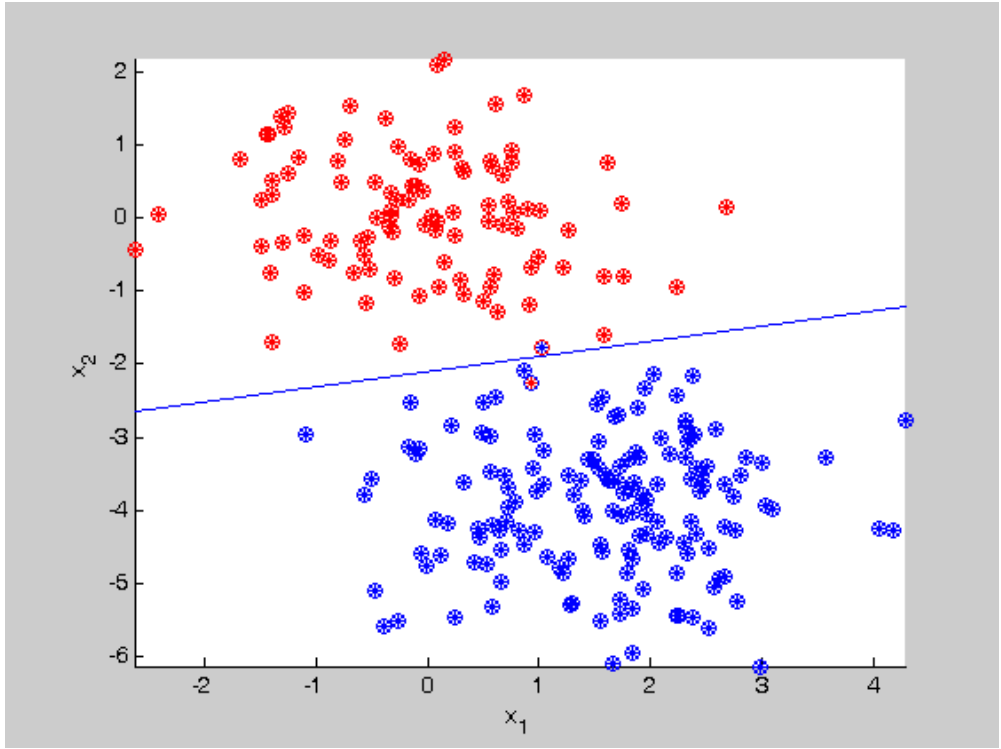
$t = 0$, dann $\mathrm{sig}(t) = 0{,}5$ – **in der Praxis mit dem Fall 2 kombiniert**

https://de.wikipedia.org/wiki/Datei:Sigmoid-function-2.svg

Parameterschätzung - Intuition

Ziel: **eine optimale Trennungshyperebene zu finden**





http://strijov.com/sources/demoDataGen.php

Parameterschätzung

$$\hat{p} = h(x) = \sigma\left(\underline{x}^T \underline{b}\right)$$

Das Modell ist durch $\underline{b}$ parametrisiert.

Was ist ein guter Vektor $\underline{b}$?

Wie lässt sich ein guter Klassifikator beurteilen?

Welche Methoden gibt es um einen guten/optimalen Vektor $\underline{b}$ zu finden?

Optimierungskriterium

Ziel des Modell-Training:

- höhe Wahrscheinlichkeiten für positive Beobachtungen $y = 1$ **und**
- niedrige Wahrscheinlichkeiten für negative Beobachtungen $y = 0$

Cost-Funktion: $c(\underline{b}) = \begin{cases} -\log(\hat{p}), wenn\ y = 1 \\ -log(1 - \hat{p}), wenn\ y = 0 \end{cases}$

https://de.wikipedia.org/wiki/Datei:Sigmoid-function-2.svg

Optimierungskriterium

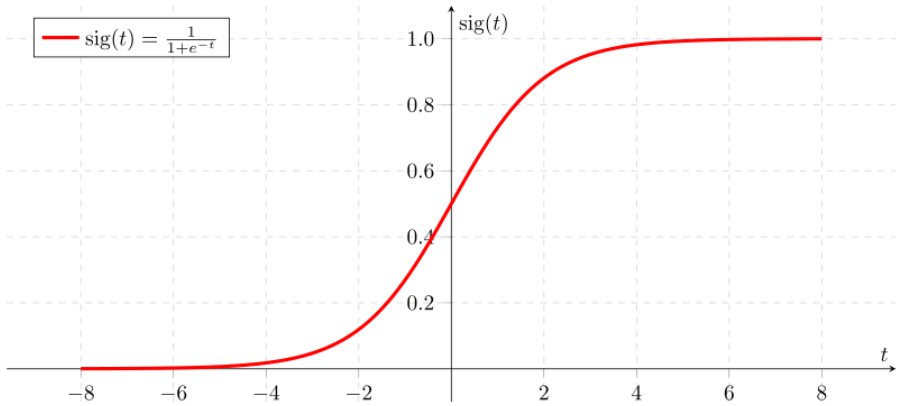
Fall 1 (**TP**, $y = 1,\ \hat{p} = 0{,}7$):

$c = -\log(0.7) = 0{,}3567 \quad c = -\log(1) = 0$

Cost-Funktion: $c(\underline{b}) = \begin{cases} -\log(\hat{p}), wenn\ y = 1 \\ -\log(1 - \hat{p}), wenn\ y = 0 \end{cases}$

Fall 2 (**TN**, $y = 0,\ \hat{p} = 0{,}1$):

$c(\underline{b}) = -\log(1 - 0.1) = 0{,}1054 \quad c = -\log(1 - 0) = 0$



Fall 3 (**FP**, $y = 0,\ \hat{p} = 0{,}6$):
$c(\underline{b}) = -\log(1 - 0{,}6) = 0{,}9163$

Fall 4 (**FN**, $y = 1,\ \hat{p} = 0{,}6$):
$c(\underline{b}) = -\log(0{,}6) = 0{,}5108$

https://de.wikipedia.org/wiki/Datei:Sigmoid-function-2.svg

Optimierungskriterium

Zielfunktion, nach dem Summieren für m Beobachtungen:

$$J(\underline{b}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{p}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-\left(\hat{p}^{(i)}\right)\right)\right]$$

Keine analytische Lösung des Minimierungsproblems, aber Funktion ist konvex

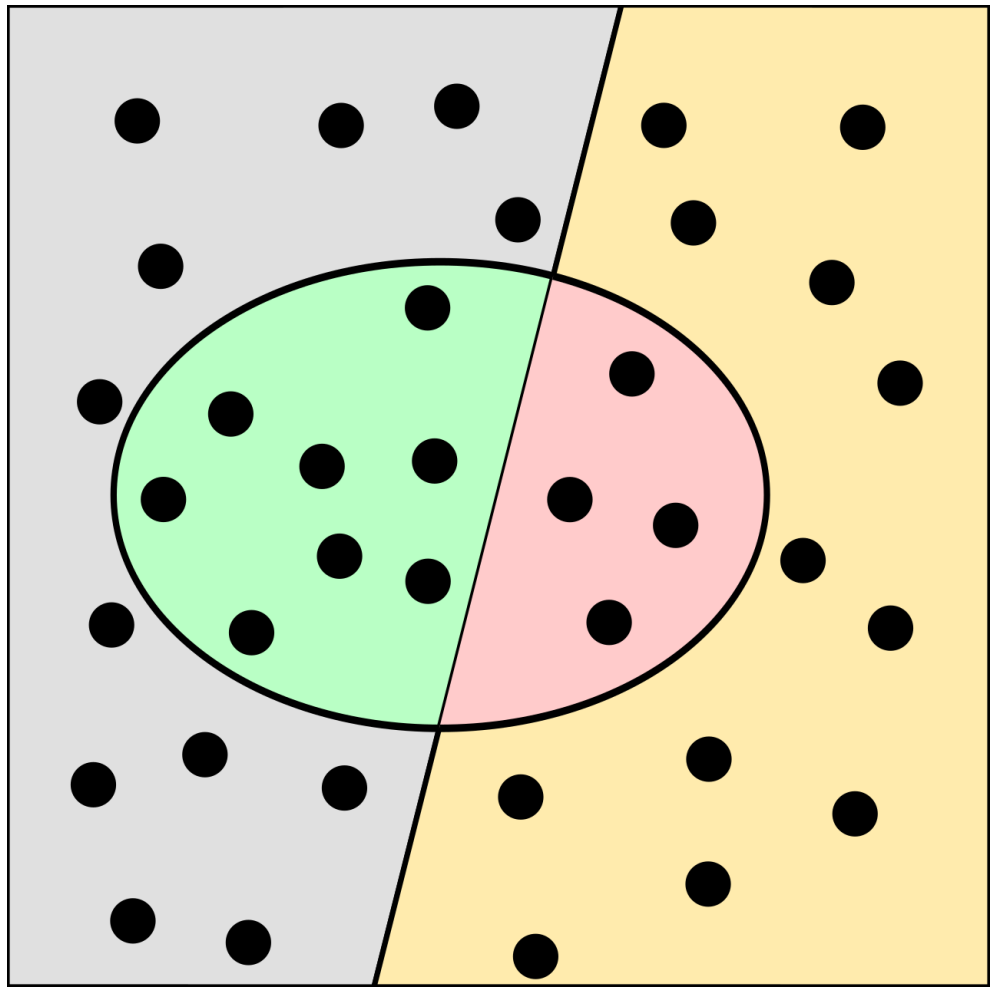
→Optimierungsverfahren findet das **globale Minimum**

Partielle Ableitungen erster Ordnung:

$$\frac{\partial}{\partial b_j}J(\underline{b}) = \frac{1}{m}\sum_{i=1}^{m}\left[\left(\sigma\left(\underline{b}^T\underline{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\right]$$

Performance Metriken

Konfusionsmatrix



		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive	False Positive
	Negative	False Negative	True Negative

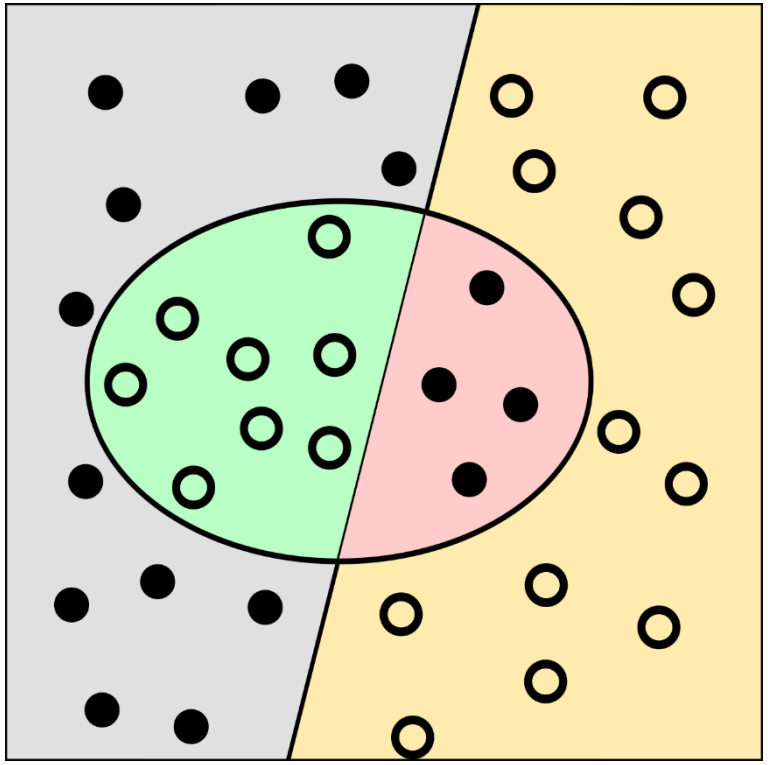
https://de.wikipedia.org/wiki/Datei:Binary-classification-file.svg

Accuracy

Treffergenauigkeit gibt den Anteil aller Objekte an, die korrekt klassifiziert werden:

$$Accuracy = \frac{richtig\ klassifiziert}{alle\ Fälle} = \frac{TP + TN}{TP + TN + FP + FN}$$

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive	False Positive
	Negative	False Negative	True Negative



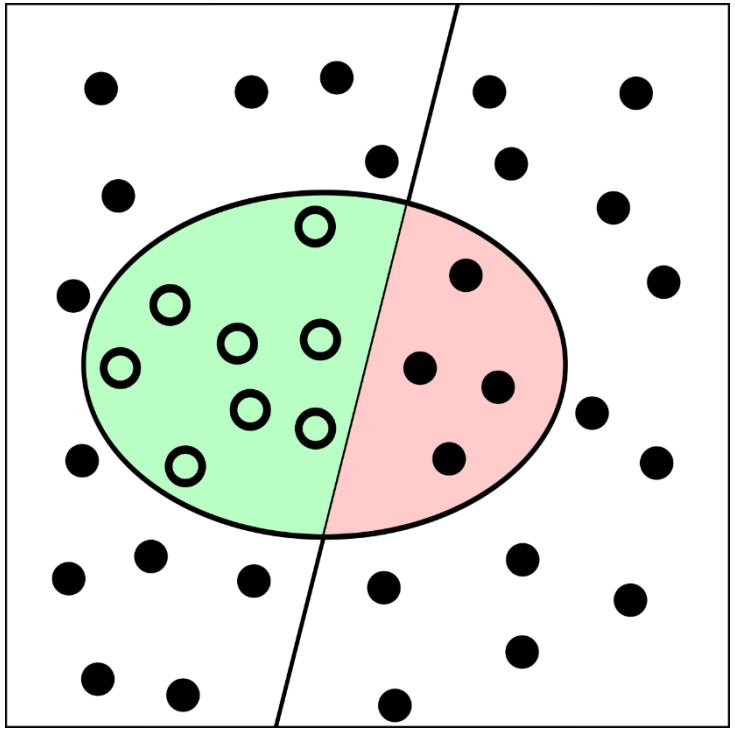
https://de.wikipedia.org/wiki/Datei:Binary-classification-file_ccr.svg

Precision

Genauigkeit gibt den Anteil der korrekt als positiv klassifizierten Beobachtungen an der Gesamtheit der als positiv klassifizierten Beobachtungen an:

$$P = \frac{richtig\ klassifizierte\ positive\ Fälle}{alle\ als\ positive\ klassifizierte\ Fälle} = \frac{TP}{TP + FP}$$

		Actual: Positive	Actual: Negative
Predicted	Positive	True Positive	False Positive
Predicted	Negative	False Negative	True Negative



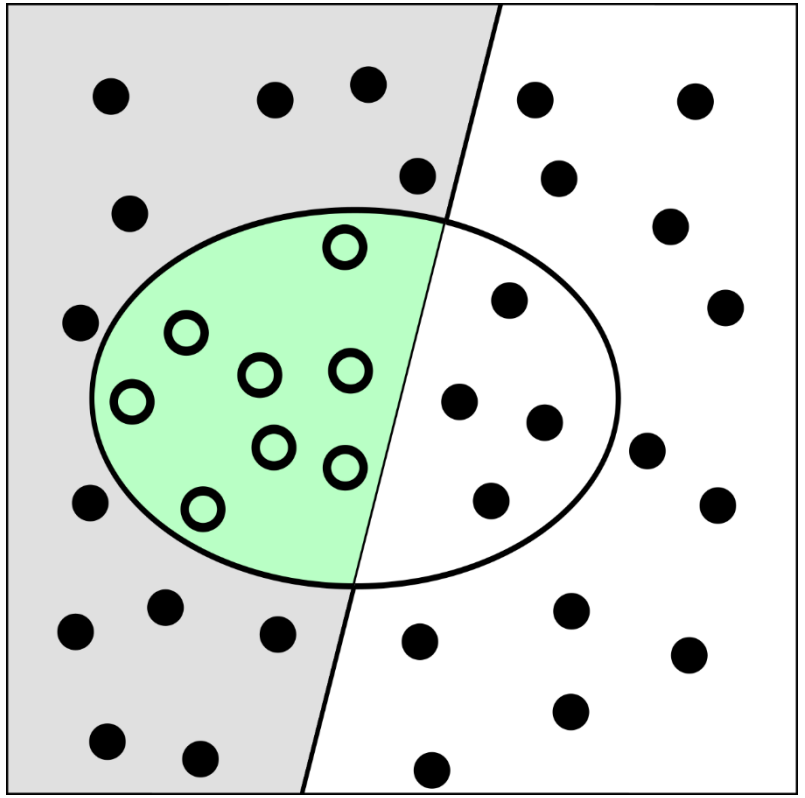
https://de.wikipedia.org/wiki/Datei:Binary-classification-file_ccr.svg

Recall

Trefferquote/Sensitivität gibt die Wahrscheinlichkeit an, mit der ein positives Objekt korrekt als positiv klassifiziert wird:

$$R = \frac{richtig\ klassifizierte\ positive\ Fälle}{alle\ positive\ Fälle} = \frac{TP}{TP + FN}$$

		Actual: Positive	Actual: Negative
Predicted	Positive	True Positive	False Positive
Predicted	Negative	False Negative	True Negative



https://upload.wikimedia.org/wikipedia/commons/8/87/Binary-classification-file_sensitivity.svg

Trennschärfe des Klassifikators

F-Score

Problem: mehre statistische Gütekriterien

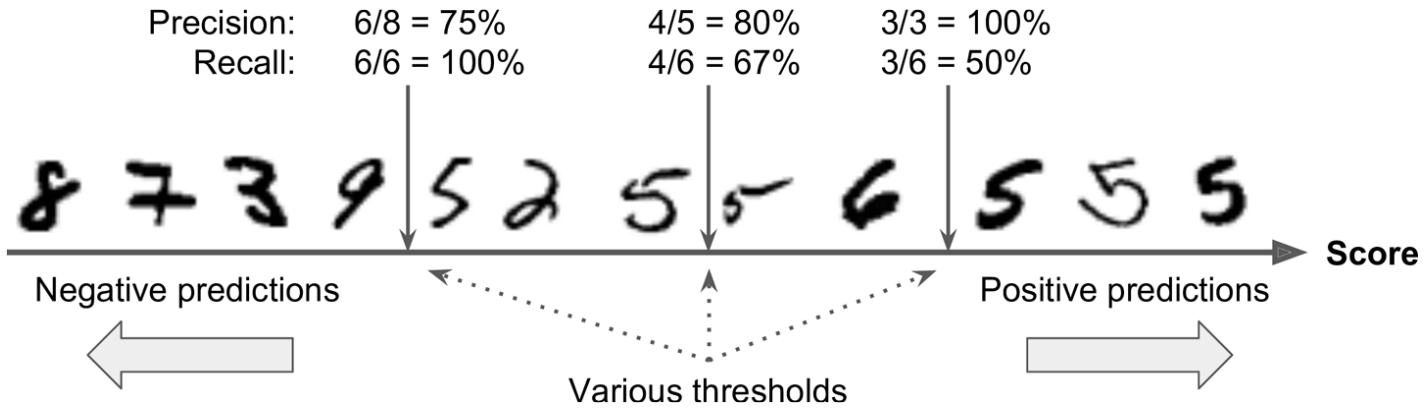
Harmonisches Mittel aus Precision und Recall, F1-Score:

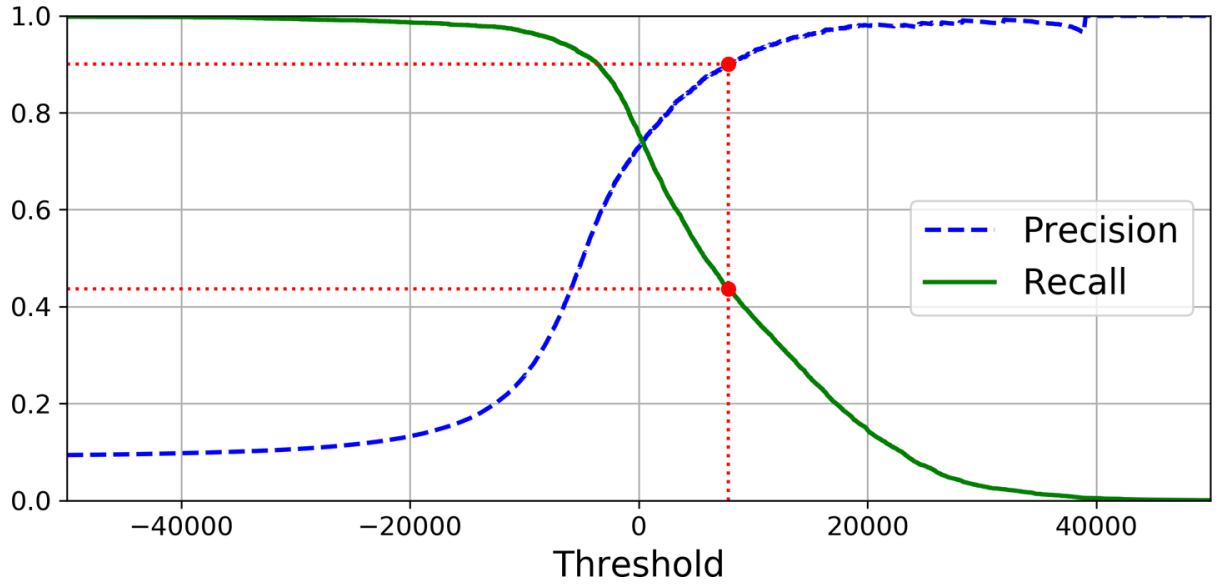
$$F_1 = 2\frac{P \cdot R}{P + R}$$

Hohe Werte, wenn beide Metriken, Precision und Recall, ähnliche Werte annehmen.

Für verschiedene Probleme lässt sich Precision oder Recall bevorzugen

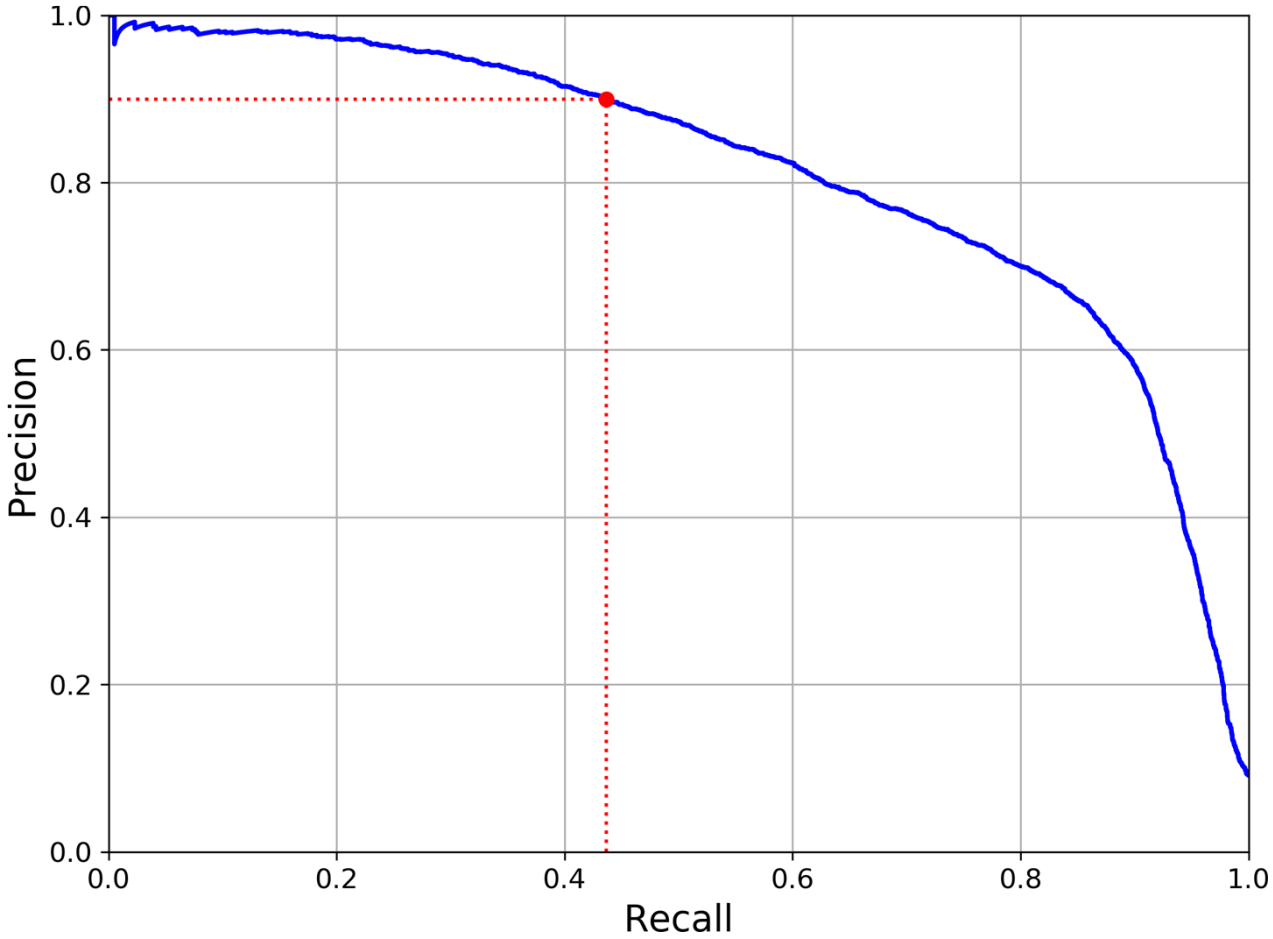
Precision/Recall Trade-off





A. Géron, 2019, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media

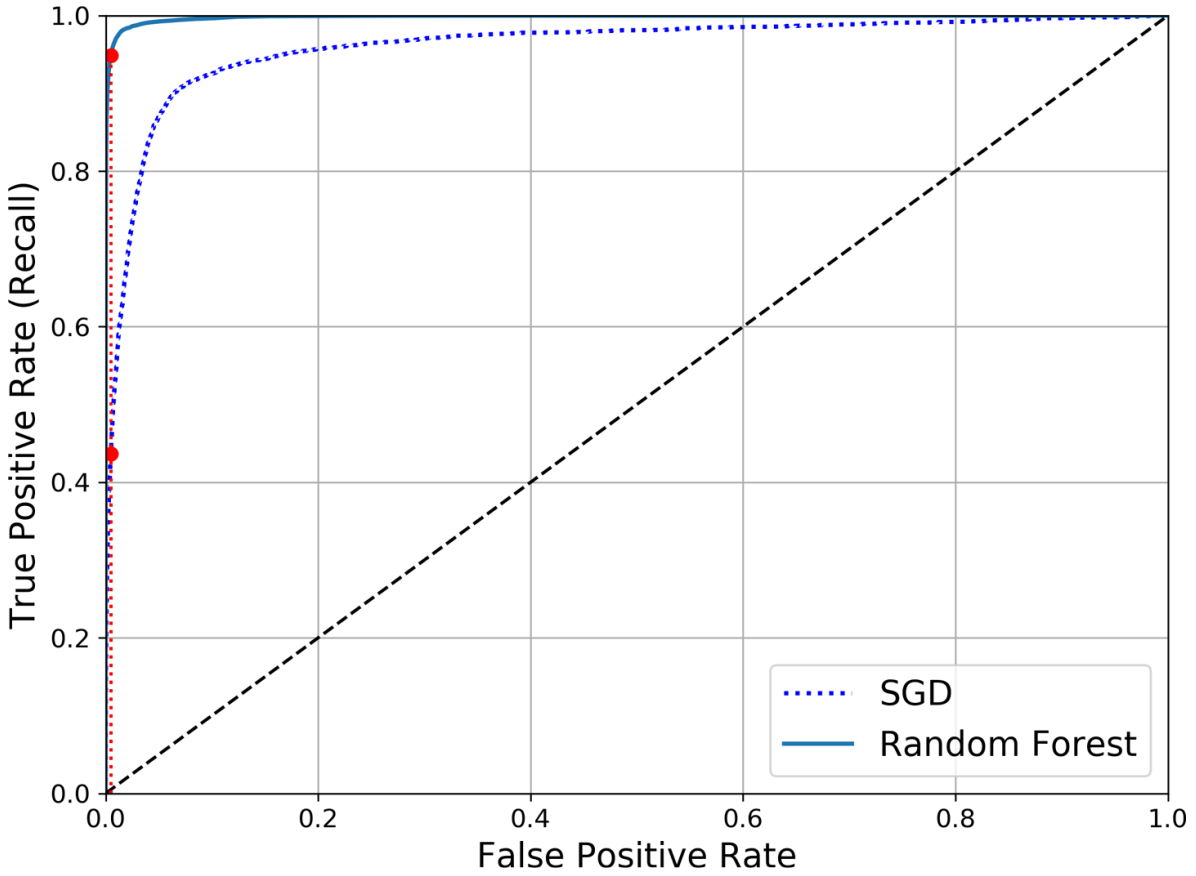
Precision/Recall Trade-off



A. Géron, 2019, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media

ROC Curve und AUC

Receiver Operating Characteristic (ROC), TP-Rate vs. FP-Rate

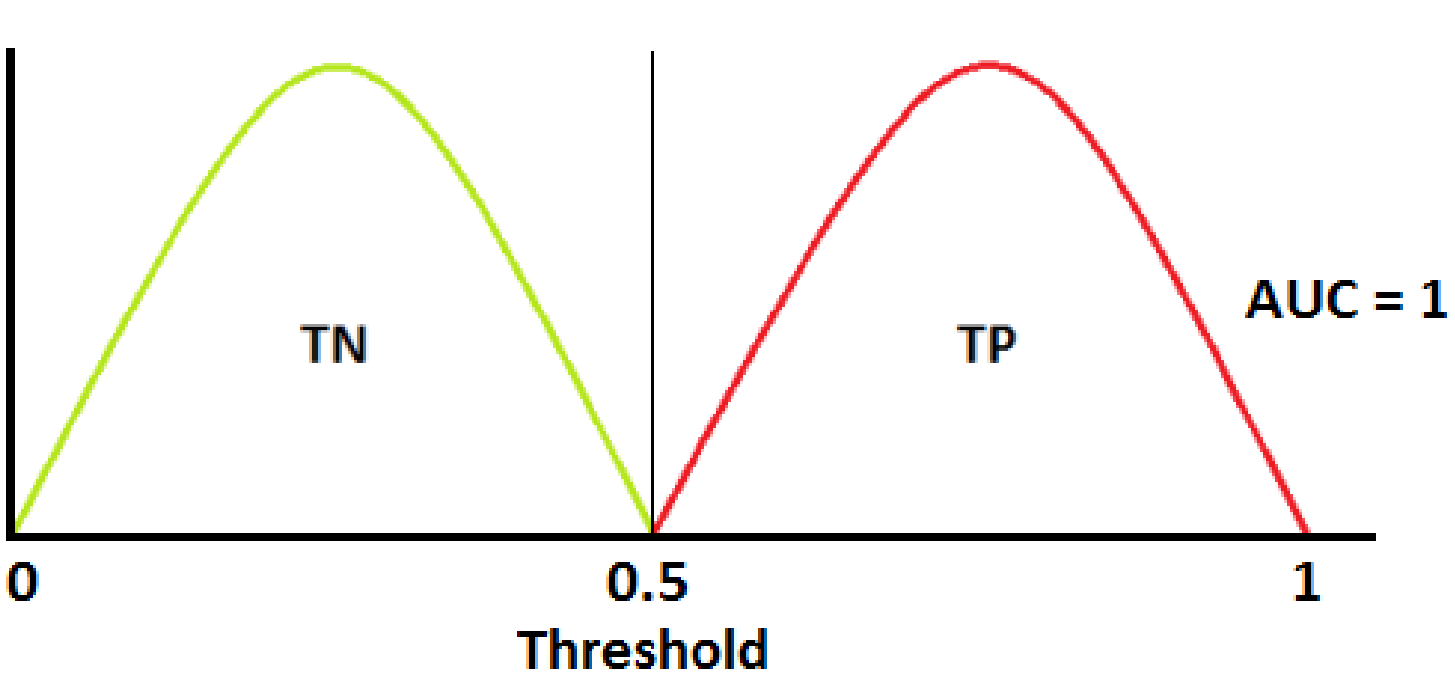


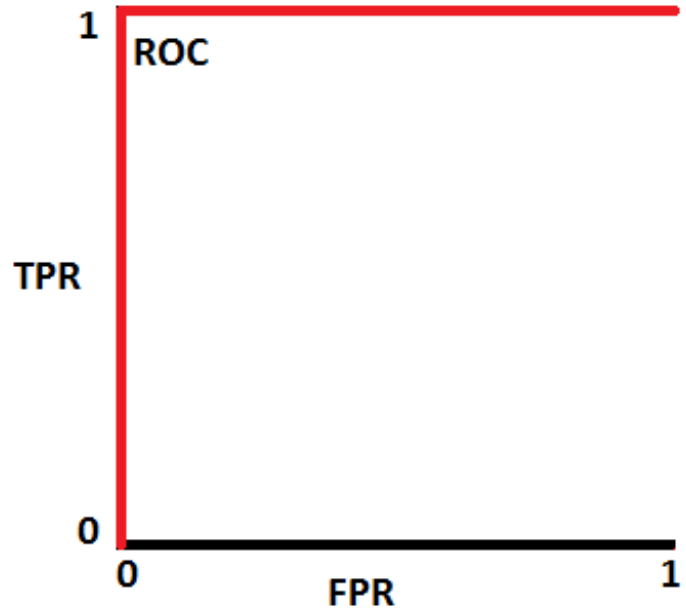
$$TPR = \frac{TP}{TP + FN}$$

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

ROC Curve und AUC

Idealer Klassifikator

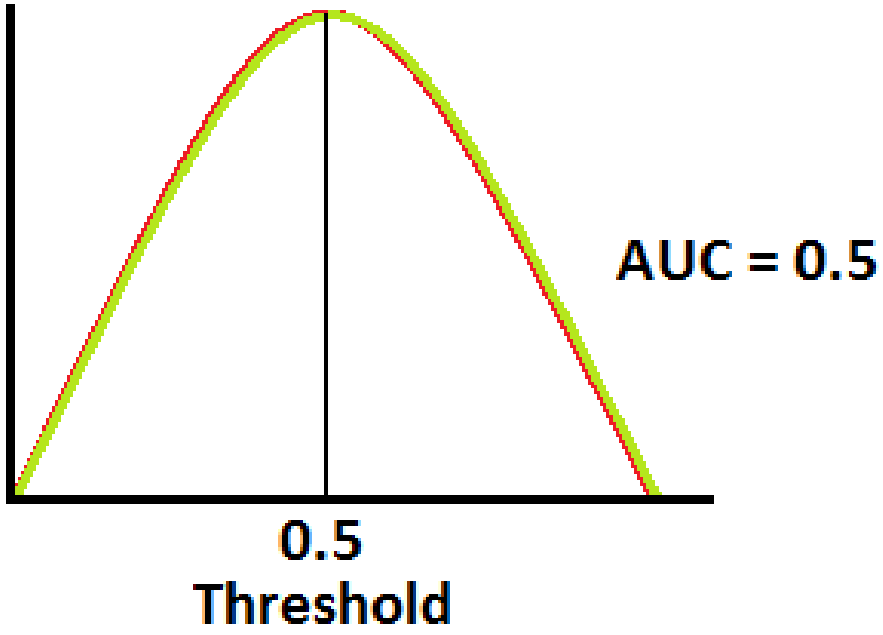


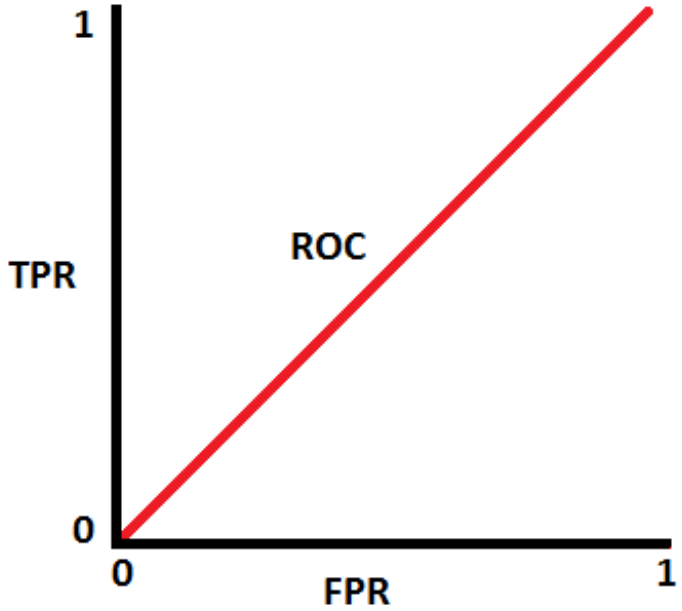


https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

ROC Curve und AUC

Zufälliger Klassifikator

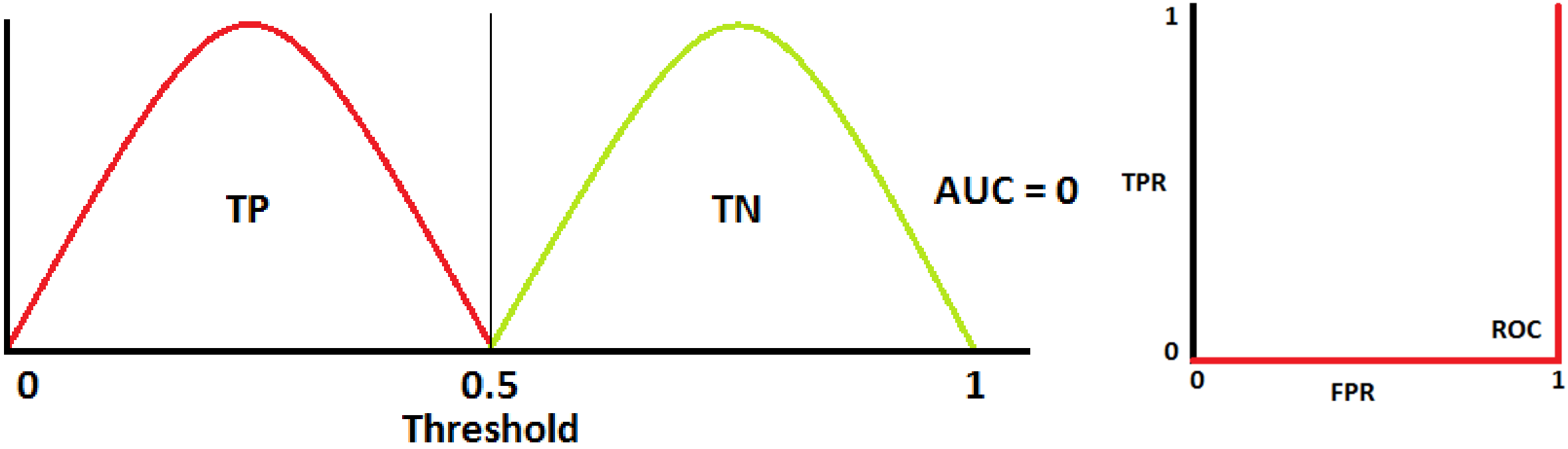




https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

ROC Curve und AUC

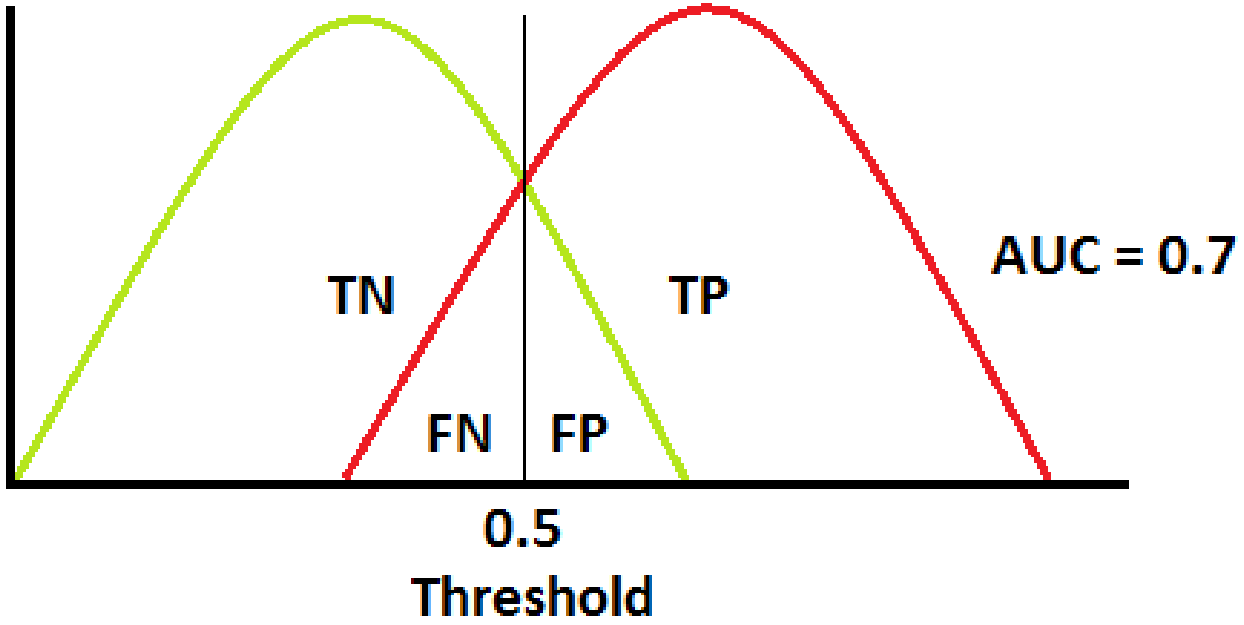
Der „schlechteste“ Klassifikator

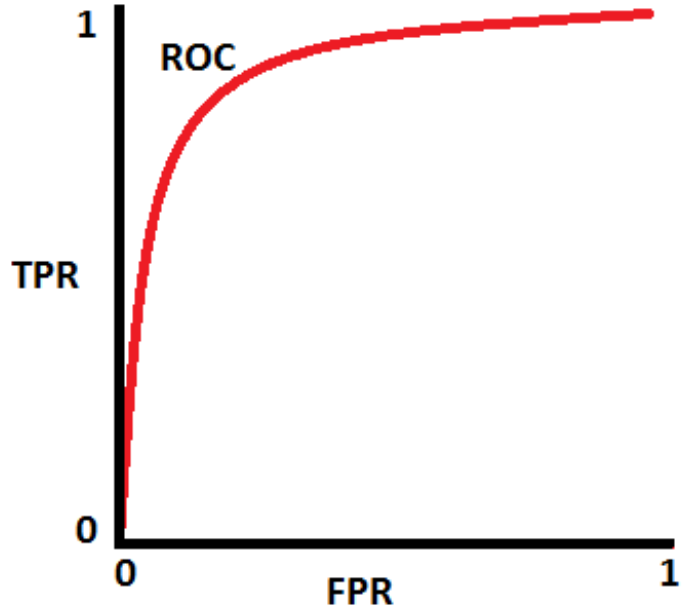


https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

ROC Curve und AUC

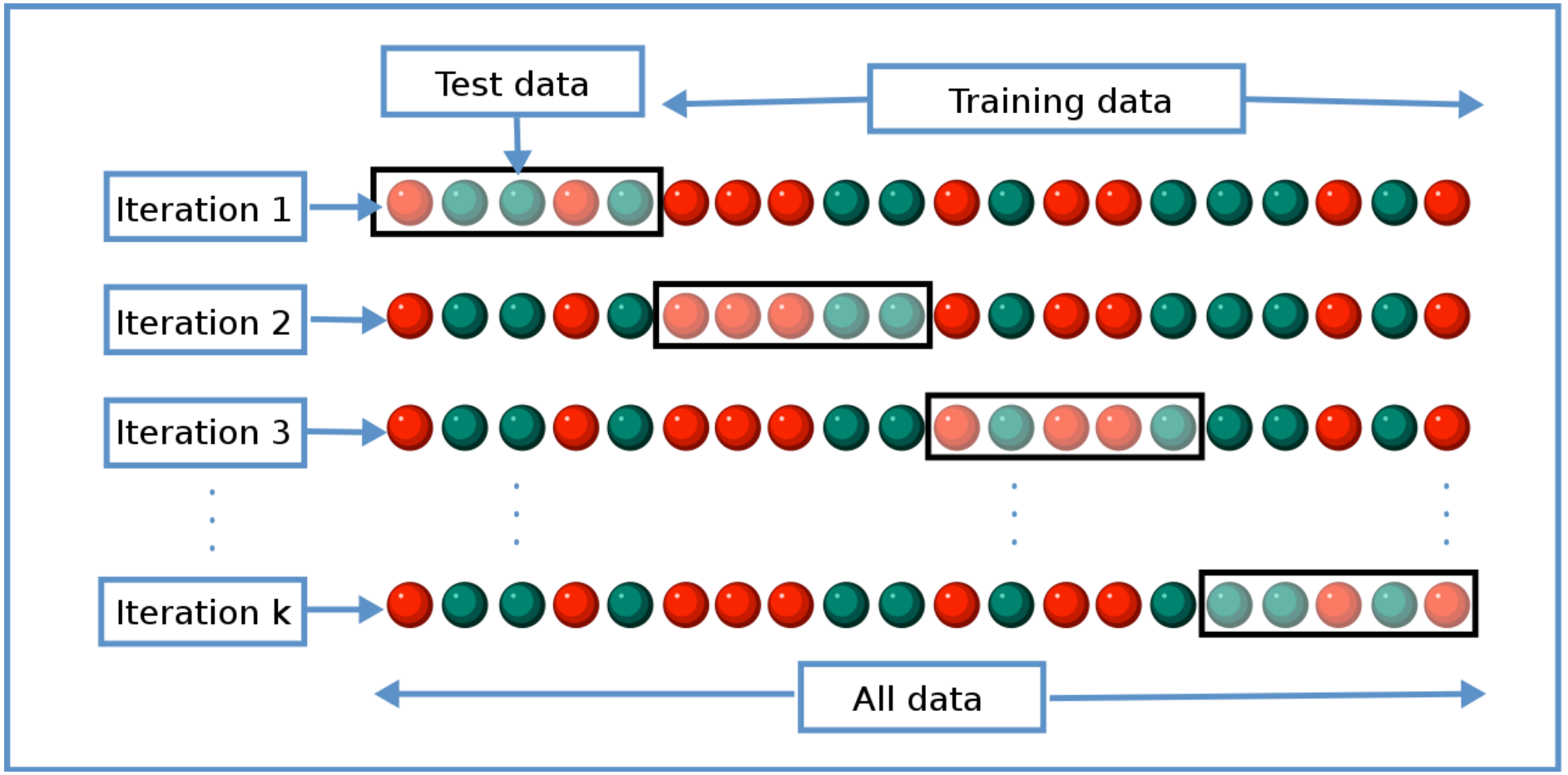
Realer Klassifikator





https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Cross-validation



https://en.wikipedia.org/wiki/Cross-validation_(statistics)#/media/File:K-fold_cross_validation_EN.svg

Klassifikatoren

SVM

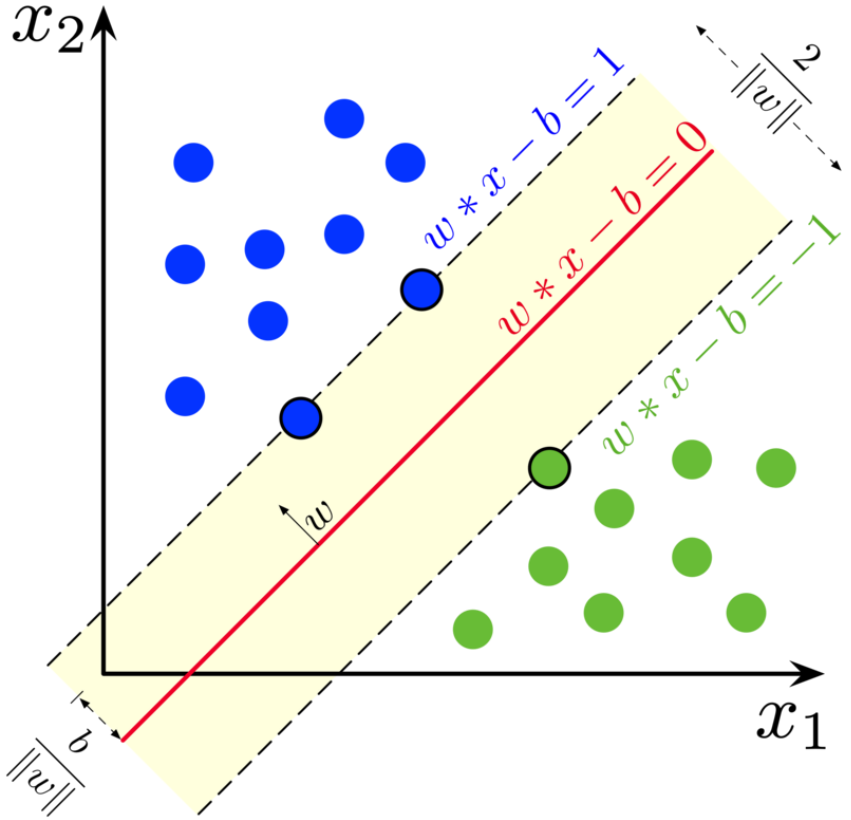
Arbeitsprinzip:

Finde eine **Hypereben** (Entscheidungsgrenze), die den **<u>Abstand zu Punkten möglichst groß hält</u>** und sich auf der Seite der Klasse befindet.

$$\min_{\underline{w},b} \frac{1}{2} \|\underline{w}\|_2^2$$

so dass die Nebenbedingung $y^{(i)}\left(\underline{w}^T \underline{x} - b\right) \geq 1$ für alle i gilt

Punkte, die am Rand der Streifen liegen heißen Support-Vektoren.



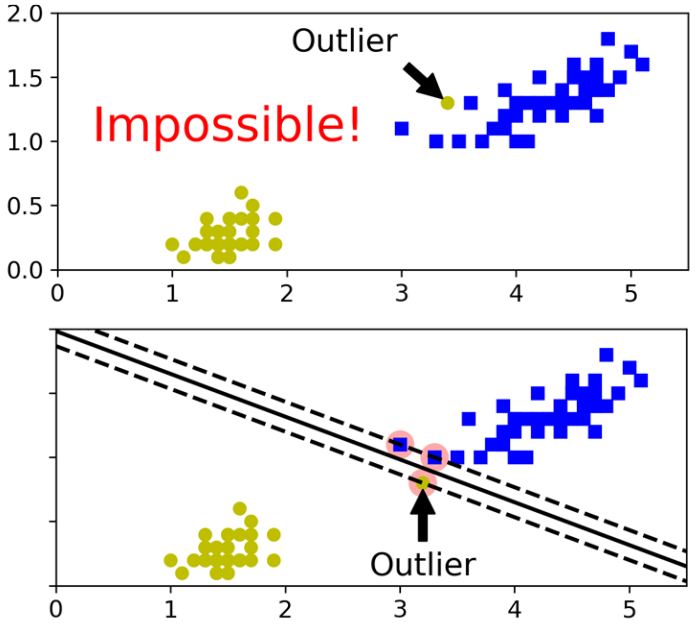
https://en.wikipedia.org/wiki/File:SVM_margin.png

SVM – Soft-Margin und Regularizierung

Probleme des **Hard-Margin**-Ansatzes:

- Der funktioniert nur dann, wenn Daten linear trennbar sind.
- Ausreißer haben einen zu großen Einfluss.

Soft-Margin-Ansatz führt einen C-Zusatzparameter zum Tolerieren von Ausreißer ein.



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

$$\min_{\underline{w},b} C\left\|\underline{w}\right\|_2^2 + \left[\frac{1}{n}\sum_{i=1}^{n} max\left(0, 1 - y^{(i)}\left(\underline{w}^T\underline{x} - b\right)\right)\right]$$

Hinge-Loss

SVM – Kernel Trick

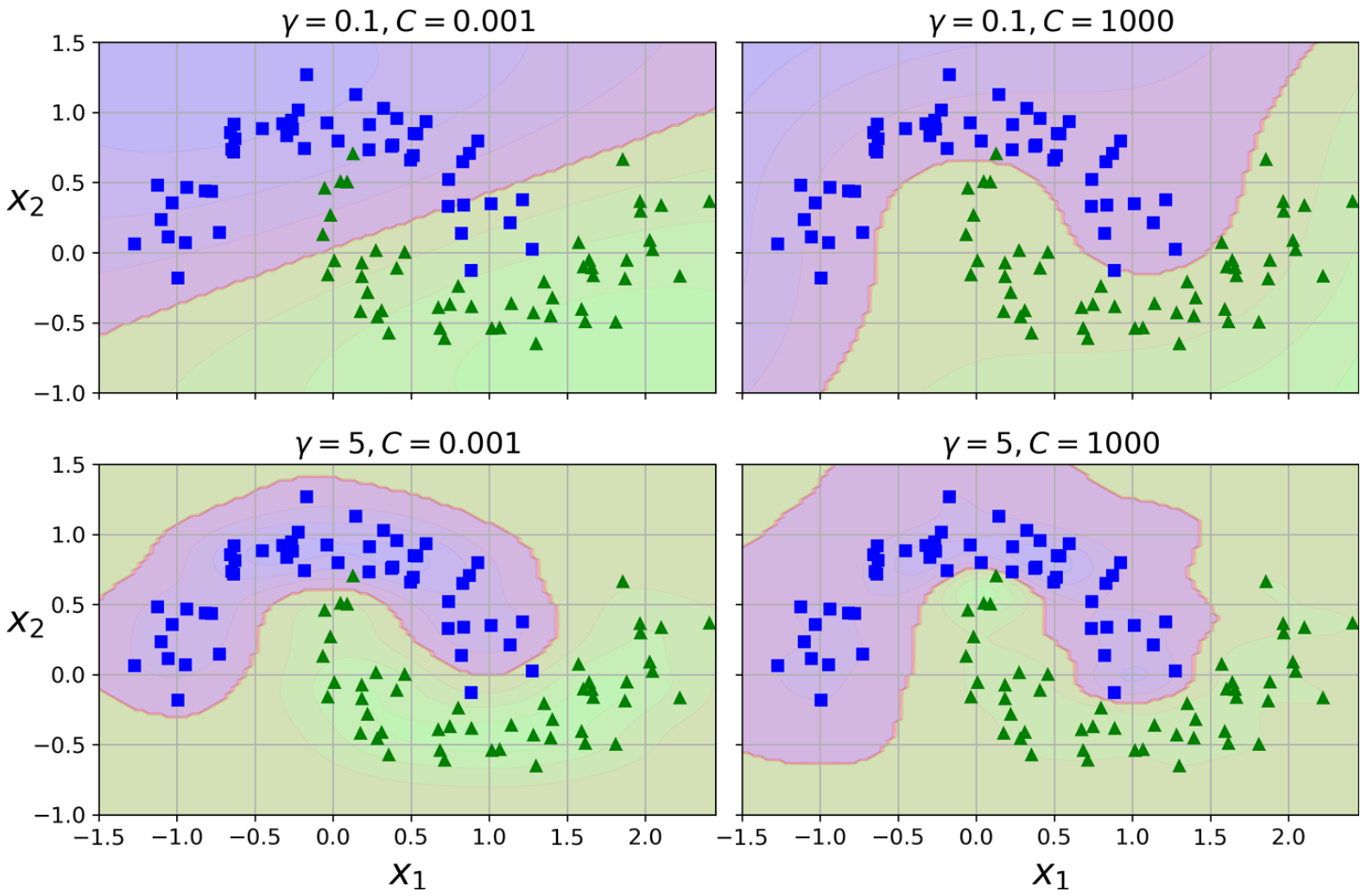
SVM wurde ursprünglich nur für linear trennbare Klassen entwickelt (Vapnik und Chervonenkis, 1963)

Anwendung des Kernel-Tricks ermöglicht Abbildung nicht-linearer Grenzen:

- Polynomial
- Radial-Basis-Funktion
- Tangens hyperbolicus
- usw.

Kernel-Trick entsprich das Hinzufügen zusätzlicher Features

SVM – Kernel Trick



A. Géron, 2019, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media

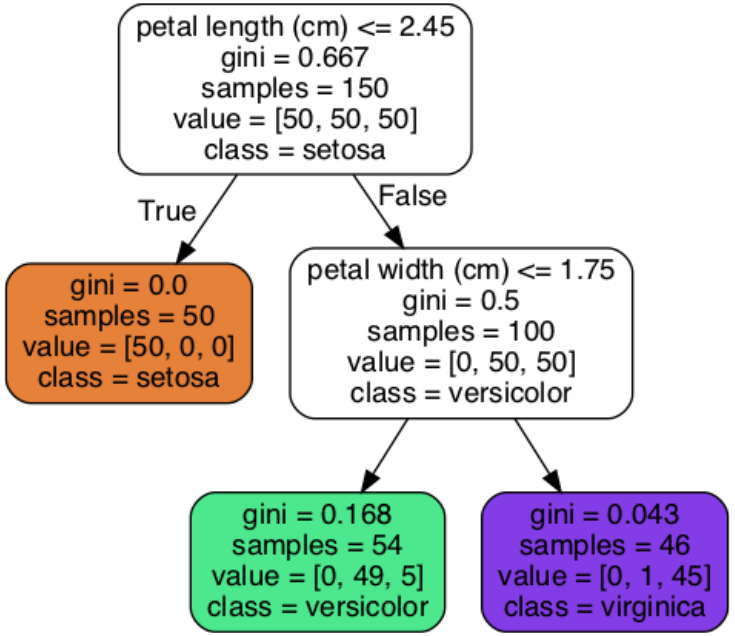
Entscheidungsbäume

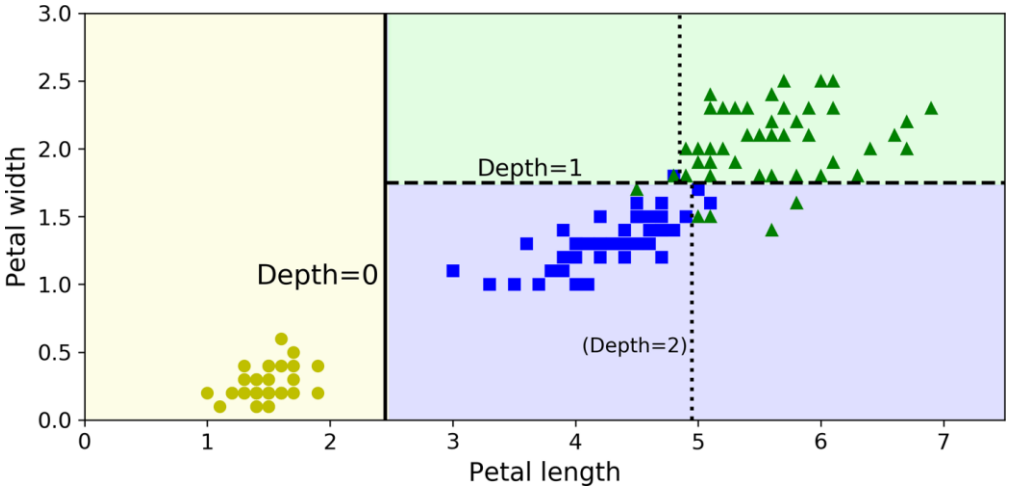
Arbeitsprinzip (für Induktion/Trainieren des Models):

- Top-Down
- Auswahl eines Features, das das beste Aufteilen ermöglicht
  - Maß ist sog. **Information-Gain** wie z.B. **Gini-Score** oder **Kreuzentropie**
- Das ausgewählte Feature wird zur Aufteilung genutzt

Rekursive Wiederholung der Auswahl und Aufteilung bis:

- keine Klassifikation mehr möglich oder
- die vorgegebene Tiefe erreicht ist



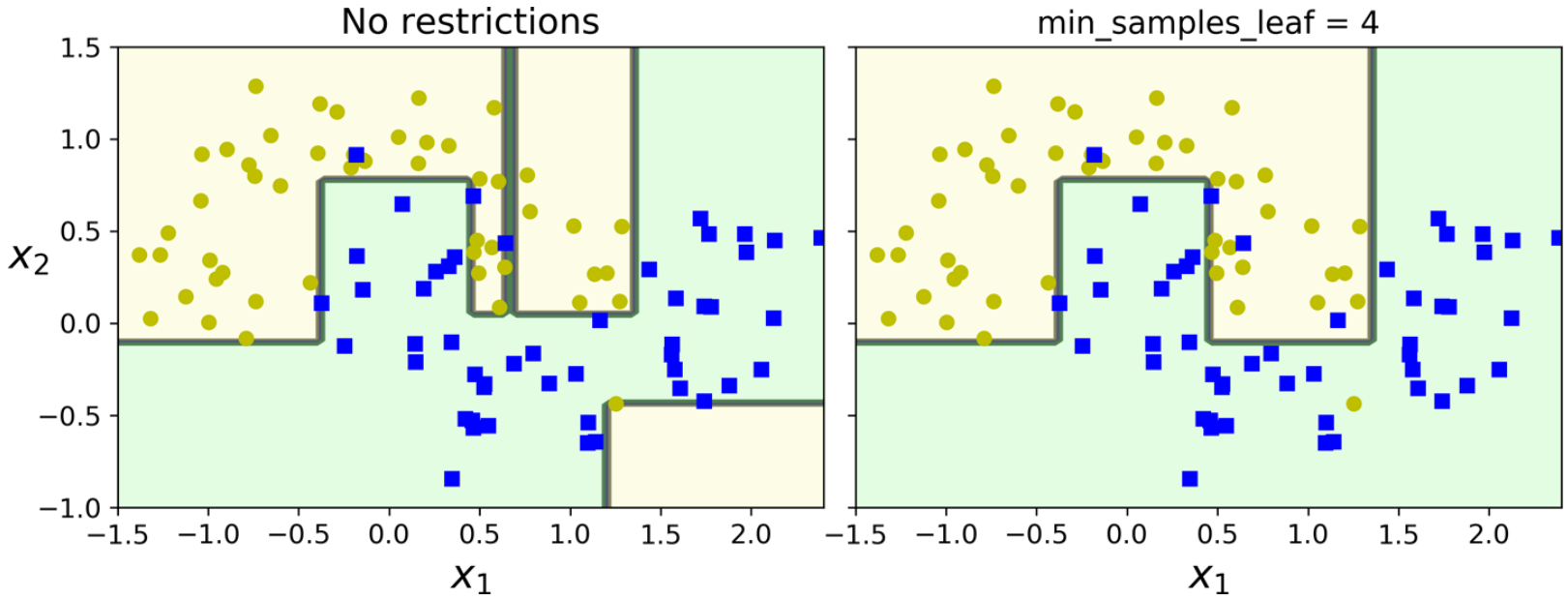


A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Entscheidungsbäume

Regularisierung:

Post-Pruning zum Entfernen von Blättern mit niedriger Relevanz (zur Bewältigung des Overfitting-Problems)

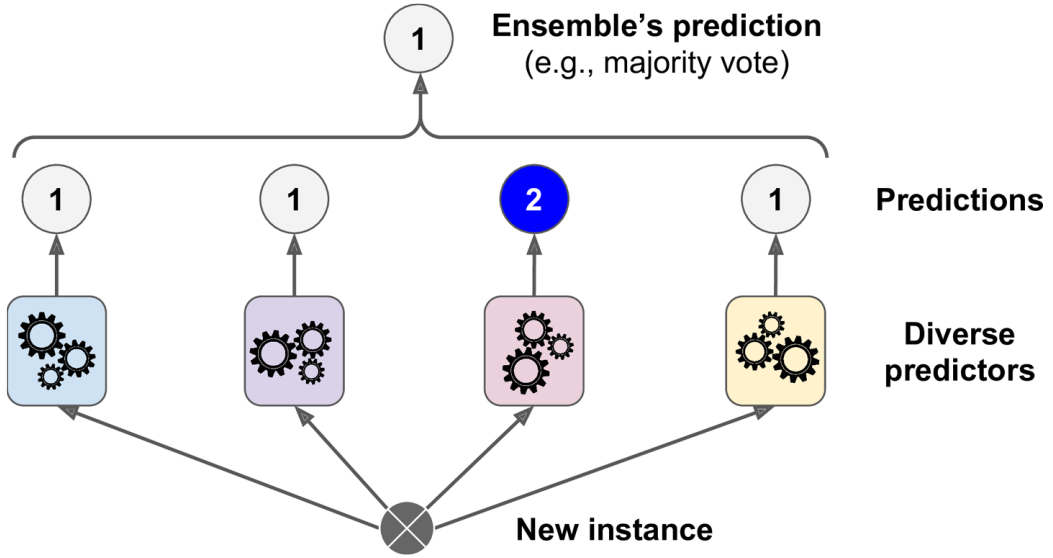


A. Géron, 2019, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media

Ensemble Learning

Kombinieren von schwachen (unabhängigen) Klassifikatoren, um bessere Ergebnisse zu erhalten

Bagging - Parallelschaltung

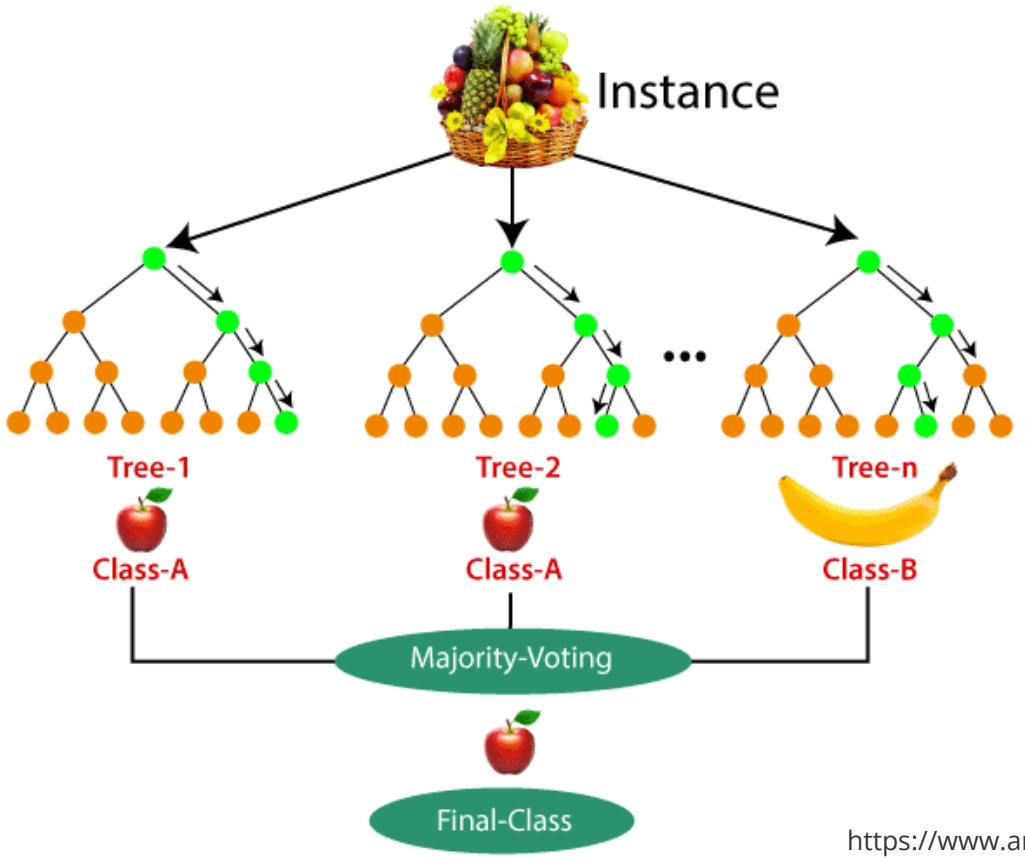


Boosting – sequenzielles Trainieren von Modellen: **Adaboost, Gradient Boosting**

A. Géron, 2019, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media

Random Forest

Random-Forest ist eine Methode, wo einzelne Entscheidungsbäume einen Ensemble-Klassifikator bilden.



https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Multiclass, Multilabel und Multioutput

Multiclass classification

Mehrklassen-Klassifikator für Klassifizierung in drei oder mehr Klassen

Nur bestimmte Algorithmen z.B. SGD, Random Forest unterstützen Mehrklassen-Klassifikation, andere erfordern künstliche Anpassungen, sog. Strategien:

- One-vs.-Rest
  - K Klassifikatoren werden als One-vs.-All trainiert
- One-vs.-One
  - K(K-1)/2 Klassifikatoren werden als One-vs.-One trainiert

Probleme: Skalierung von Konfidenzscore, nicht bilanzierte Klassen

- oder Erweiterung von Modellen, e.g. für SVM

One-hot encoding

One-Hot-Kodierung stellt **kategoriale** Variablen in ein Format, das deren einwandfreien Einsatz als Input- und Outputvariablen in ML Algorithmen erlaubt.

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

https://medium.com/@michaeldelsole/what-is-one-hot-encoding-and-how-to-do-it-f0ae272f1179

Softmax Regression

Normalisierung der Ausgabe eines Multiclass-Klassifikators

$$\hat{p}_k = \frac{exp(f_k)}{\sum_{j=1}^{K} exp(f_j)}$$

dabei gilt $\sum_{j=1}^{K} \hat{p}_j = 1$

In der Regel verfolgt von einem *arg max* Operator, der eine Klasse mit der größten Wahrscheinlichkeit ausgibt

Multilabel und Multioutput

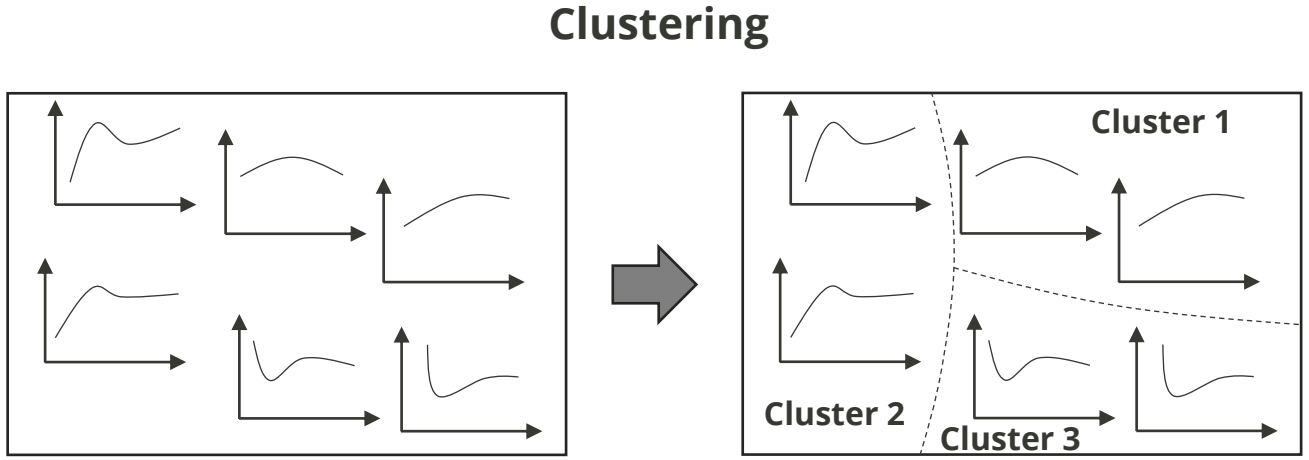
Multiclass – mehr als zwei Klassen für alle Beobachtungen, aber nur eine Klasse von einer Beobachtung möglich

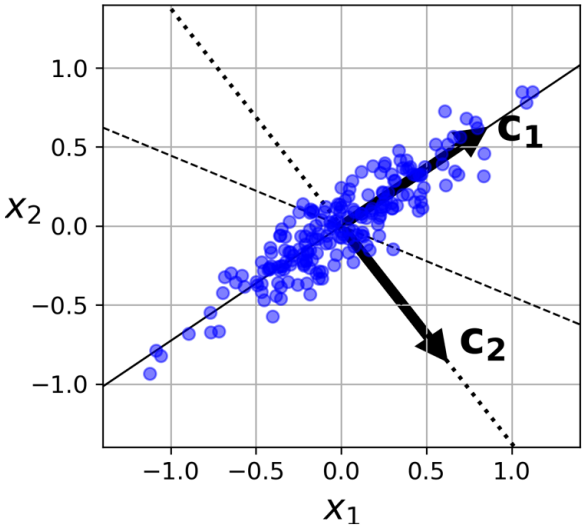
Multilabel – mehr als zwei Klassen pro Beobachtung

Multioutput – Multiclass + Multilabel

Zusammenfassung

Nächste Vorlesung





Adaptiert aus A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.
https://www.ml-science.com/k-means-clustering

Dr. rer. nat. Valentin Khaydarov
Email: valentin.khaydarov@tu-dresden.de
Telefon: 0351 463 33387

Vielen Dank für Ihre Aufmerksamkeit!