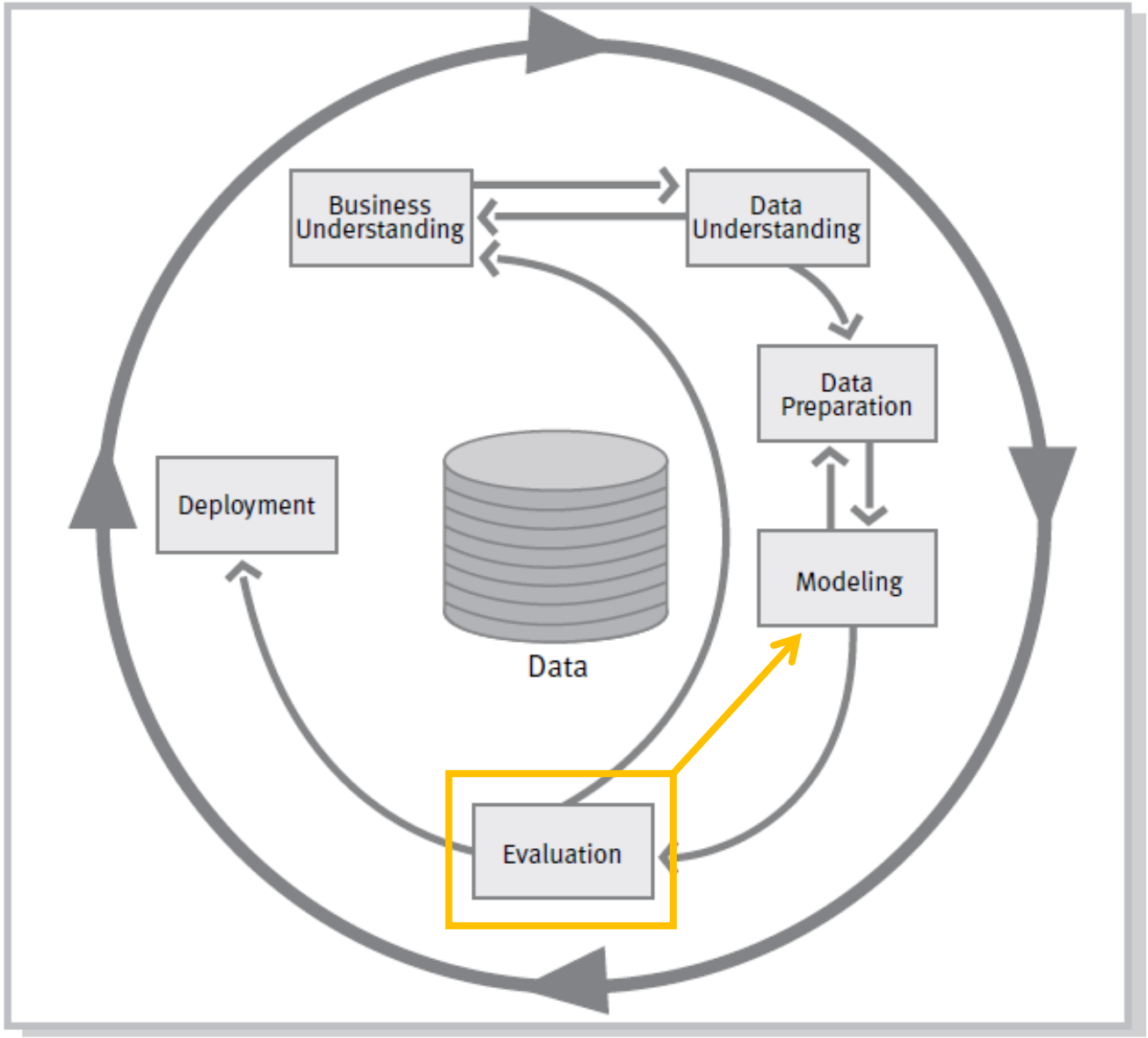
Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Maschinelles Lernen – Typische Probleme (1)



– Hans Friedrich Witschel, Andreas Martin

Bias und Variance bzw. Over- vs. Underfitting

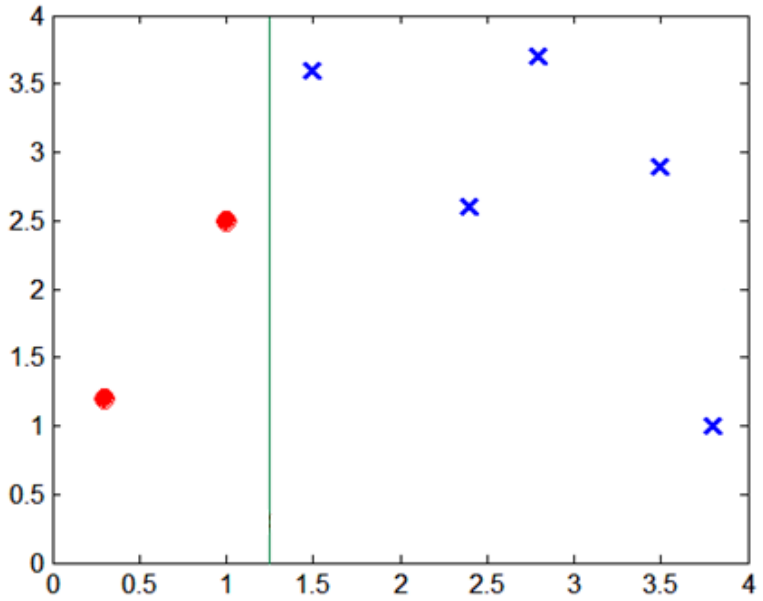
- Grundprinzip:

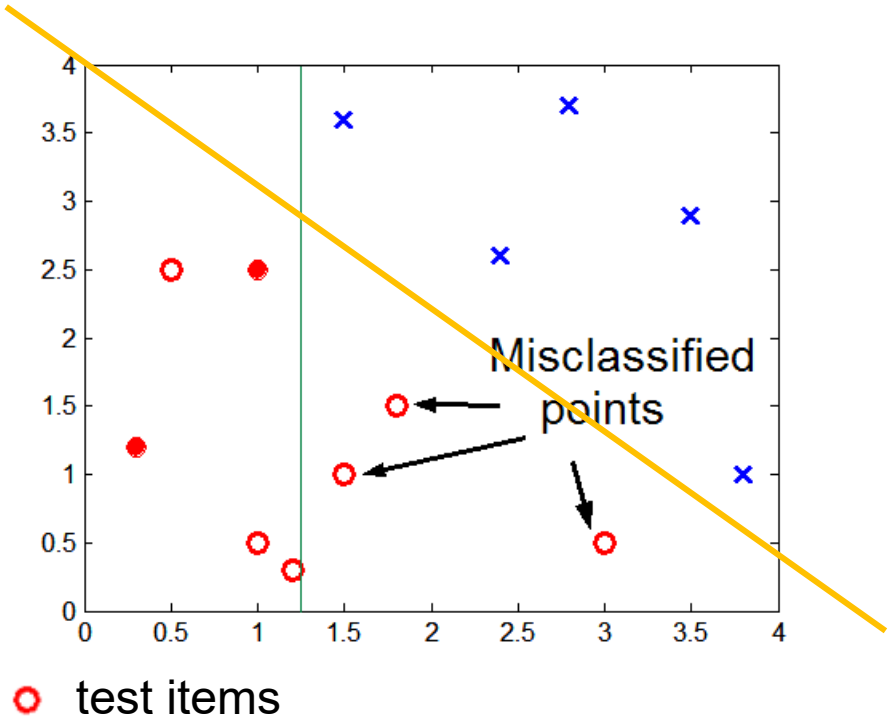
a. Modelle mit niedriger Komplexität sind robuster, aber manchmal zu einfach
   → wir sprechen von **Underfitting** bzw. grossem **Bias**

b. Modelle mit hoher Komplexität sind genauer, generalisieren aber manchmal nicht
   → wir sprechen von **Overfitting** bzw. grosser **Variance**

Underfitting / grosser Bias

- Underfitting: Modell ist zu einfach, z.B. aufgrund zu kleiner Trainingsmenge





- **Diagnose**: schlechte Vorhersagen auf Testmenge, z.B.: ein Modell mit höherer Komplexität oder ein Mensch machen wesentlich genauere Vorhersagen

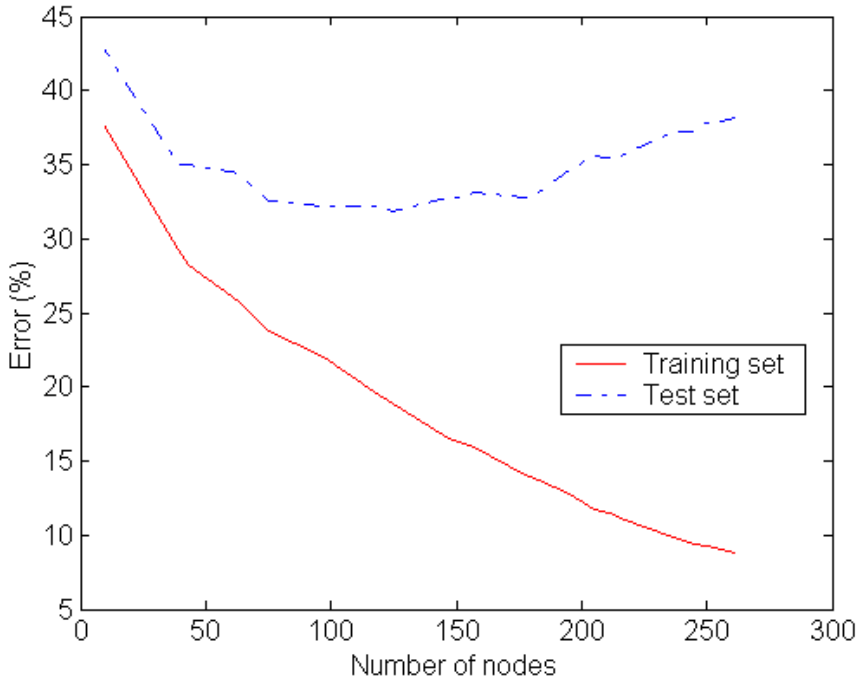
Underfitting – was hilft?

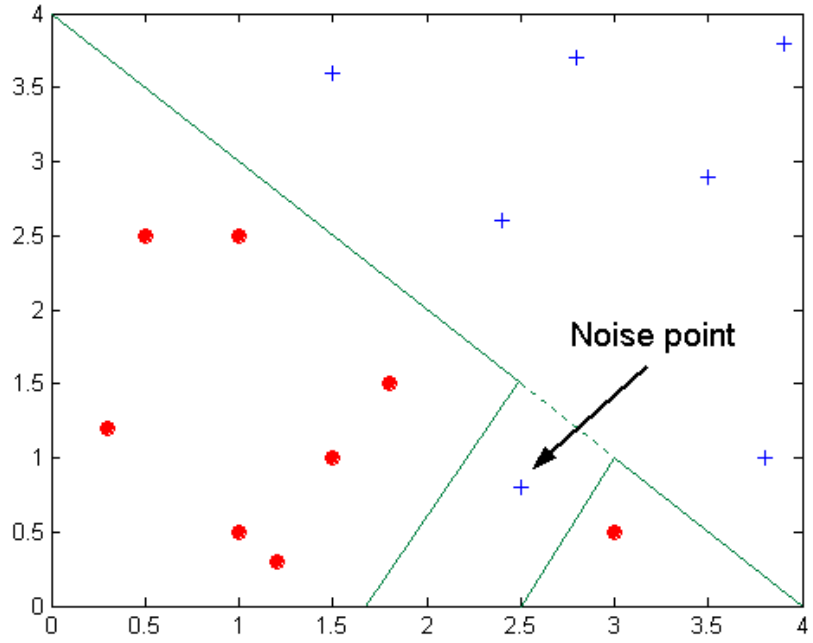
- Mehr Trainingsdaten
- Komplexeres Modell wählen!

Algorithmus	K	R	Komplexität	Optionen zur Komplexitätssteuerung
Entscheidungsbaum	X	X	Mittel	Pruning, Begrenzung der Tiefe
Logistische Regression	X		Gering	Regularisierung (Ridge oder Lasso)
Knn	X	X	Mittel	
Gradient Boosting	X	X	Hoch	Anzahl Bäume, Tiefe der Bäume
Lineare Regression		X	Gering	Regularisierung (Ridge oder Lasso)

Overfitting / grosse Variance

- Modell zu komplex:





- Erklärung: Modell passt sich z.B. an «Noise Points» an
- Diagnose: Wert der gewählten Metrik (z.B. AUC) auf Trainingsmenge mit Wert auf Testmenge vergleichen!

Overfitting am Beispiel einfache Regression

Leider ist es aber meist so: wenn die Variance sinkt, steigt der Bias – und umgekehrt…!

Overfitting – was hilft?

- Komplexität reduzieren…
  a. Einfachere Modellart (siehe Tabelle.)
  b. Gleiches Modell aber vereinfacht (regularisieren, prunen, … → siehe auch Tabelle)
    – Verwendung von Validierungsmenge («Dev set»), um die richtigen Werte der entsprechenden Parameter zu finden

Algorithmus	**K**	**R**	**Komplexität**	**Optionen zur Komplexitätssteuerung**
Entscheidungsbaum	X	X	Mittel	Pruning, Begrenzung der Tiefe
Logistische Regression	X		Gering	Regularisierung (Ridge oder Lasso)
Knn	X	X	Mittel	
Gradient Boosting	X	X	Hoch	Anzahl Bäume, Tiefe der Bäume
Lineare Regression		X	Gering	Regularisierung (Ridge oder Lasso)

Class Imbalance

- Problem: «Nadeln im Heuhaufen finden», interessante Klasse hat nur wenig Beispiele
- Beispiele: Marketing, Fraud Detection, Vorhersage von: Spitaleinweisung, Churn, …
- Wichtig: Korrekt evaluieren

a. Keine Accuracy / Fehlerrate! → warum nochmal?

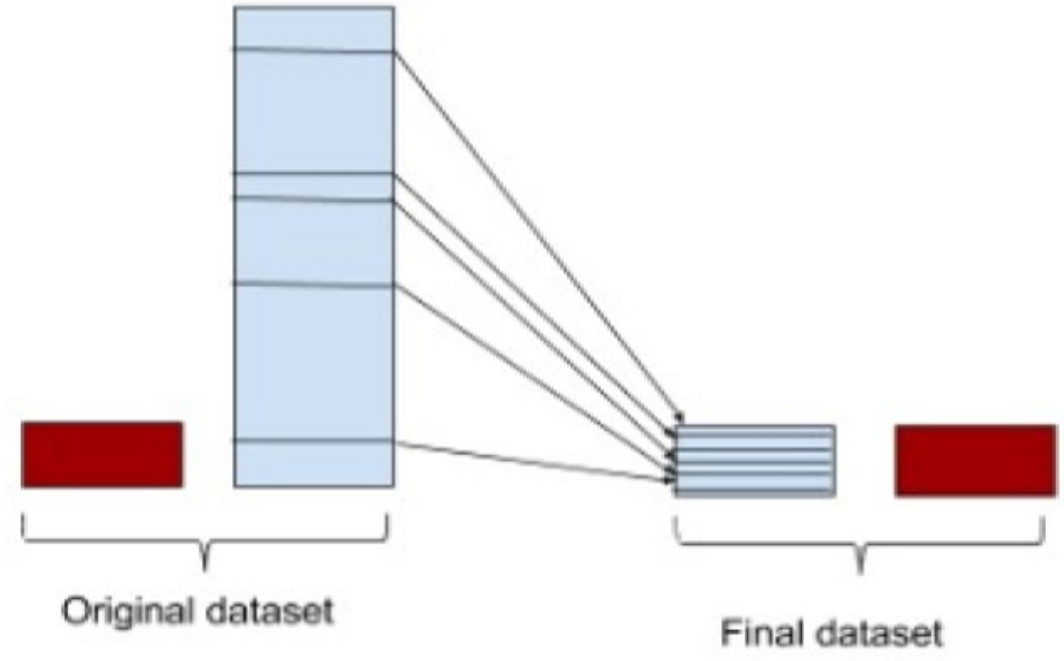
b. Wenn Kosten quantifiziert werden können: kostenbasiert

c. Sonst: AUC, Precision, Recall, F-Measure

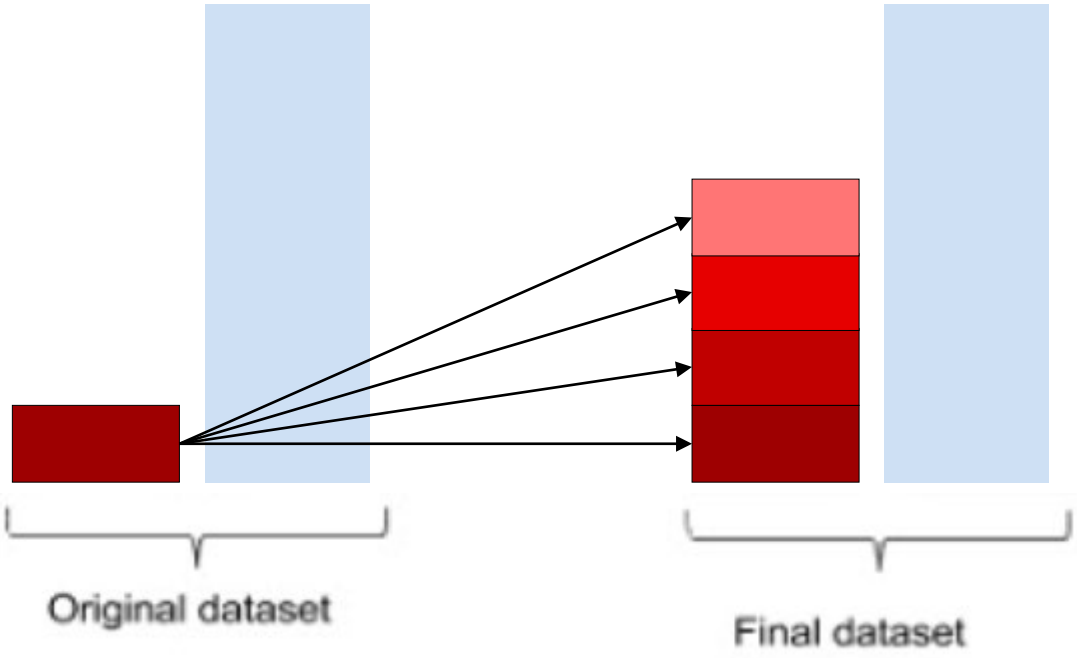
Für die Minderheitsklasse
(die «Nadeln»)!

Over- bzw. Under-Sampling

Undersampling majority class



Oversampling minority class
(manchmal: Erzeugung synthetischer Beispiele)



- Undersampling: nur gut, wenn insgesamt viele Daten vorliegen!

Over-/Undersampling – Konsequenzen

- WICHTIG: die Testdaten MÜSSEN die originale Verteilung aufweisen!

- Zu erwartende Effekte: das Modell hat mehr «Mut» die Minderheitsklasse vorherzusagen. Effekte für die Metriken:

Genauigkeit (Accuracy)	↓
AUC	↑
F1	↑
Kosten	↓
Confusion Matrix	Mehr TP / FP

Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Modelle tunen – ein paar Tipps

Evaluierungsstrategie wählen

- Metrik(en) wählen

  a. Den Business-Stakeholdern gut zuhören: was ist wichtig? Kann man Kosten quantifizieren?

  b. Optimieren vs. Erfüllen
  - Versucht nur eine Metrik zu **optimieren**!
  - Falls **Bedingungen** an andere Metriken gestellt werden (z.B. Zeit für Training oder Vorhersagen): versucht sie zu erfüllen, nicht zu optimieren!

- Beispiel: «möglichst genau vorhersagen, muss aber schnell genug sein!»

  → rausfinden, was «schnell genug» heisst; dann unter allen «genug schnellen» Modellen das genaueste finden

Fehleranalysen

- Tuning-Prozess:
  a. Bias + Variance analysieren
  b. Unbedingt auch: Confusion Matrix anschauen!!!
  c. Einzelne Fehler anschauen
     - ... eine kleine, zufällige Stichprobe von falsch klassifizierten Datenobjekten nehmen
     - ... analysieren, warum sie falsch klassifiziert wurden, Gründe sammeln (**beachtet**: man könnte sogar ein interpretierbares Modell lernen, das Fehler vorhersagt…!)
     - ... die Anzahl der Vorkommen jedes Grunds (= Fehlerkategorie) zählen
     - ... zuerst die häufigsten Fehlerkategorien angehen

Trainings-, Validierungs- und Testmengen – einige Regeln

- Trainingsmenge: alles ist erlaubt, was die Performance steigert
  - Re-Sampling (s.o.)
  - Hinzufügen ähnlicher Daten aus externen Quellen

- Validierungs- und Testmenge: sollten die «wahren» Charakteristika des Anwendungsfalls widerspiegeln
  - z.B. die Verteilung des Klassenattributs
  - z.B. die Verteilung anderer Attribute