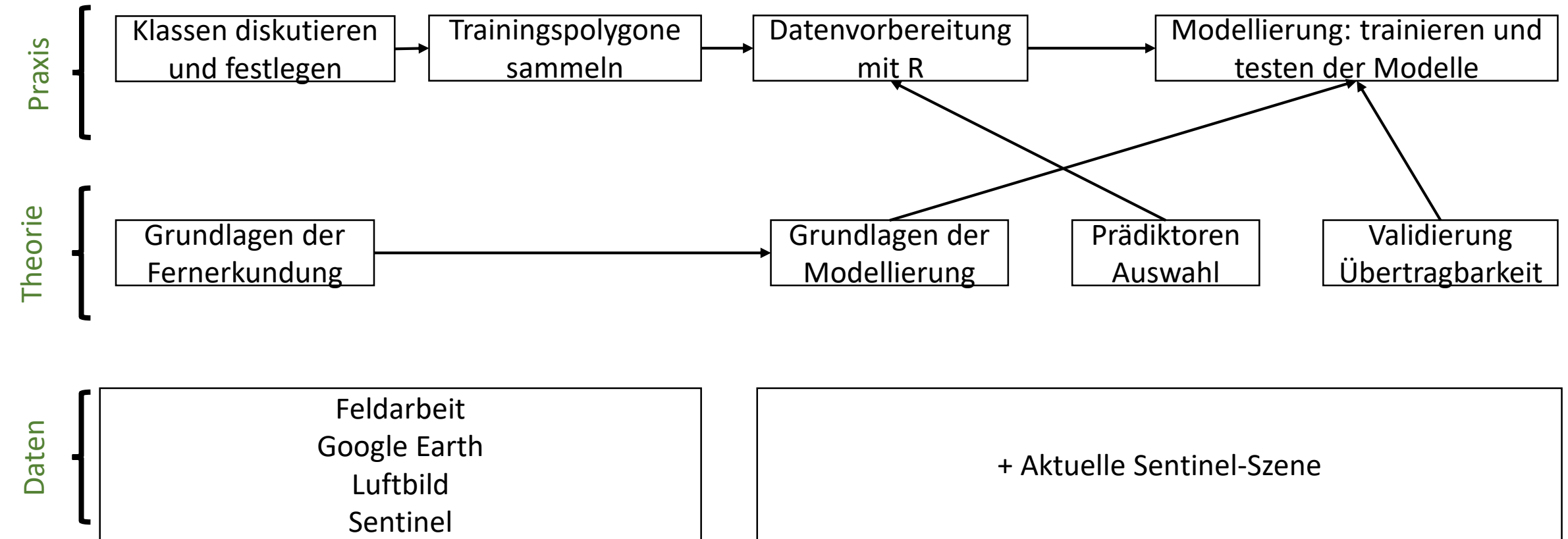


# Validierung...endlich

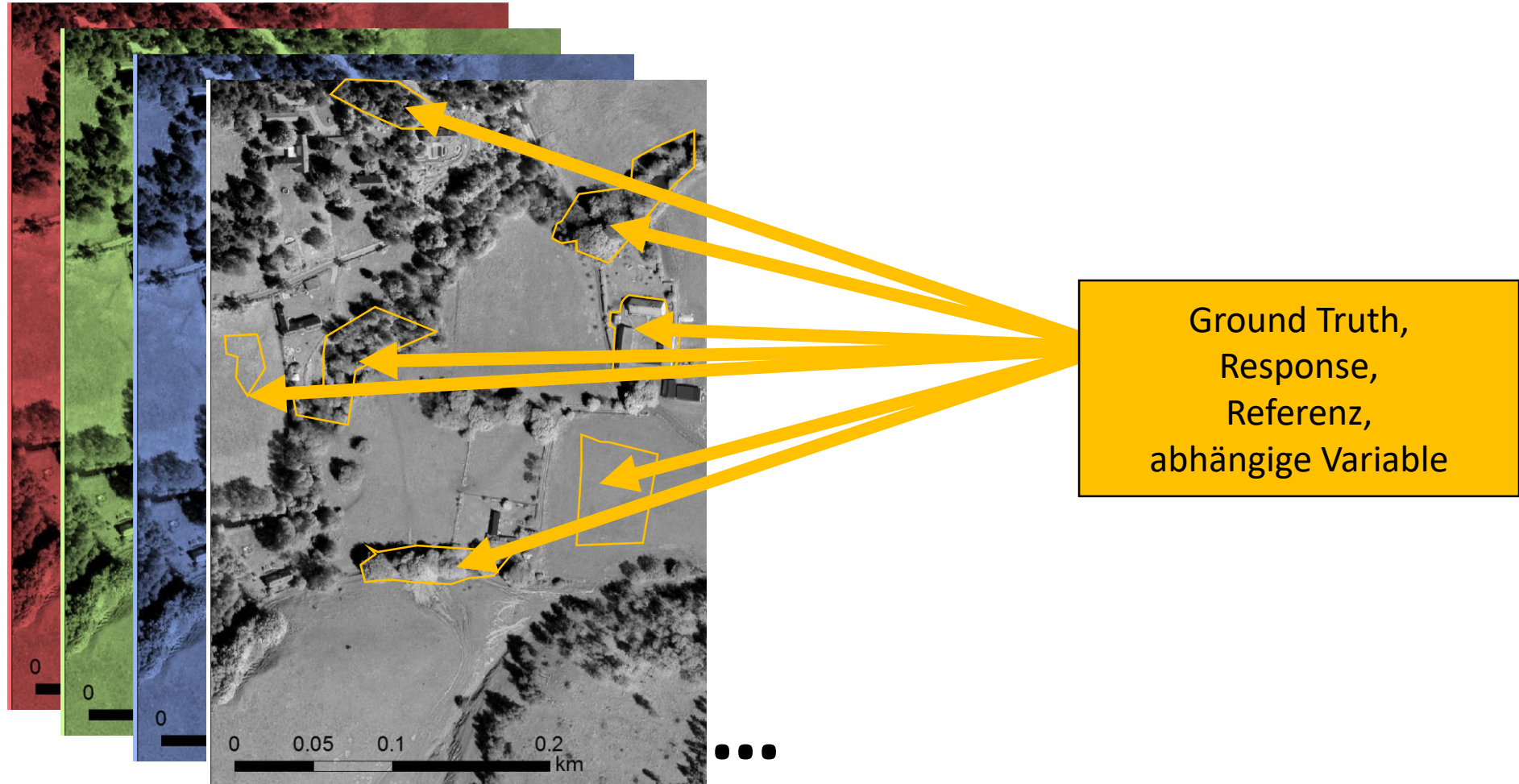
26.5.2019

Marvin Ludwig, Alice Ziegler, Hanna Meyer

# Workflow

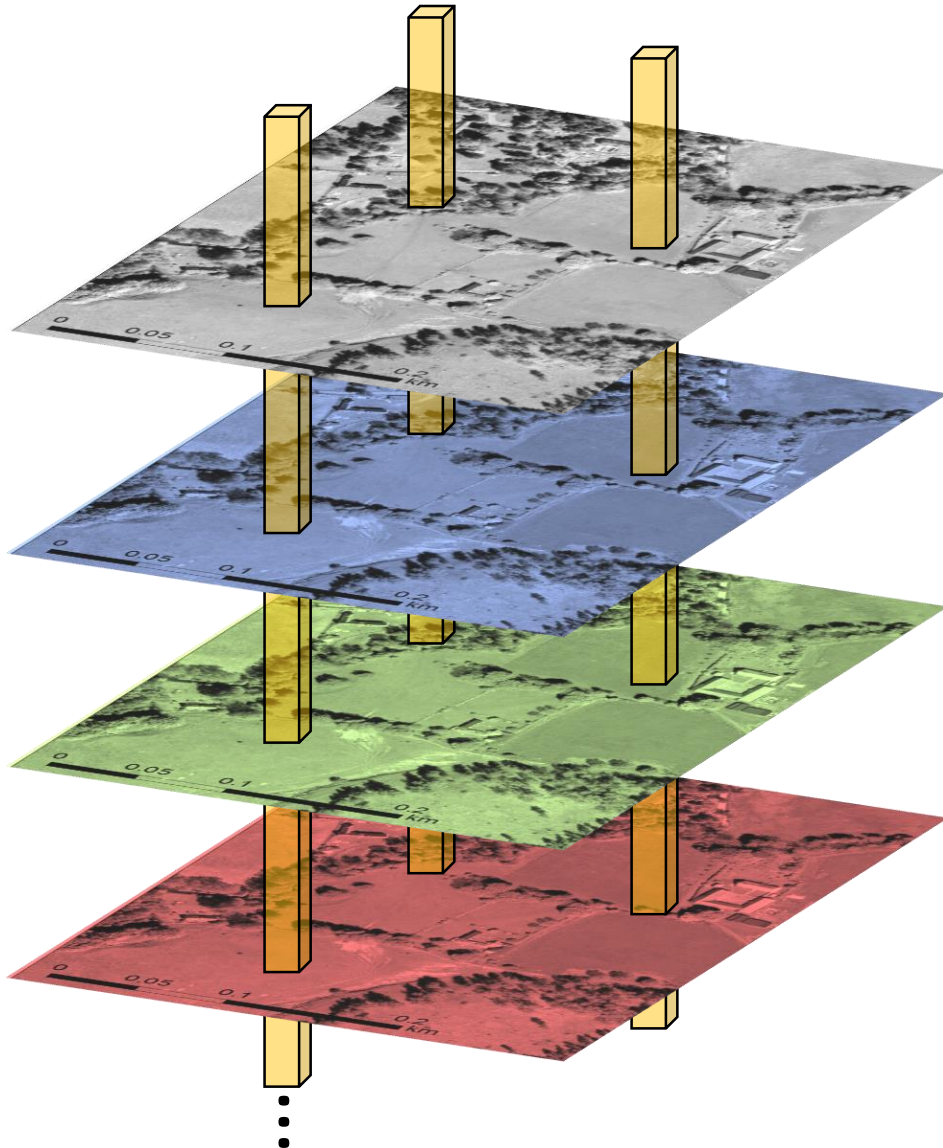


# Modellierung



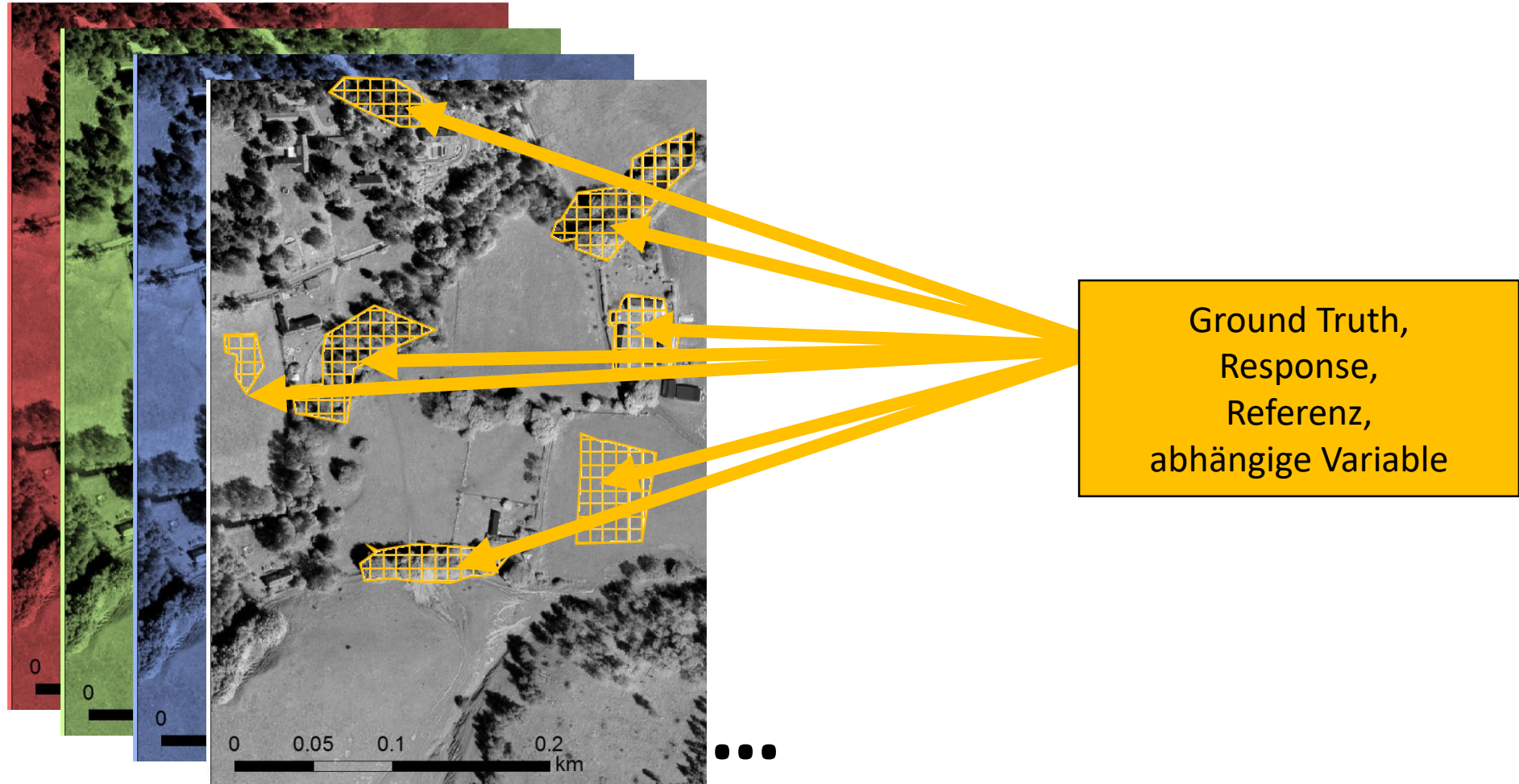
Quelle: Eigene Darstellung  
Bildaten: SPOT 5, Aufnahmedatum 2010 (ca)  
Eigene Abbildung (2012)

# Modellierung



Pix ID	R	G	B	NIR	Klasse
1	393	680	792	3550	Nadelwald
2	2167	1586	1476	2853	brach
3	859	940	1006	2434	Gebäude
4	705	1133	1012	4910	bewachsen
5	947	1024	1138	2253	Gebäude
...					
105	402	668	800	3467	?
106	698	1142	1003	5012	?
107	2203	1601	1386	2976	?
108	879	962	956	2312	?
...					

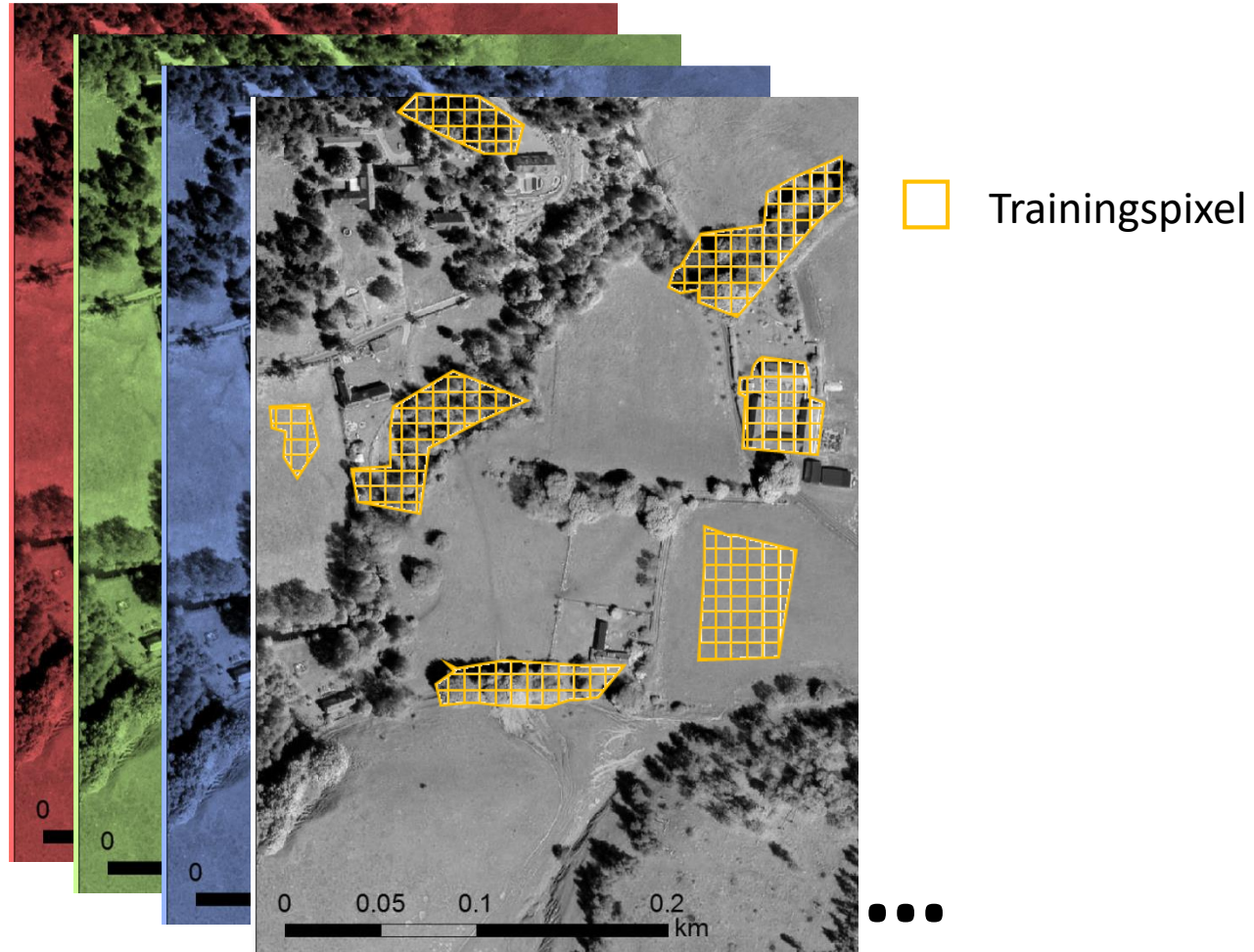
# Modellierung



Quelle: Eigene Darstellung  
Bilddaten: SPOT 5, Aufnahmedatum 2010 (ca)  
Eigene Abbildung (2012)



# Validierung



Quelle: Eigene Darstellung  
Bilddaten: SPOT 5, Aufnahmedatum 2010 (ca)  
Eigene Abbildung (2012)

# Validierung - Accuracy

		Referenz Daten							
		Wasser	Gebäude	anderes	brach	bewachsen	Laubwald	Nadelwald	Summe
Klassifikations Daten	Wasser	168	0	0	0	0	0	0	168
	Gebäude	0	1054	0	0	0	0	0	1054
	anderes	0	0	840	0	0	0	0	840
	brach	0	0	0	5099	0	0	0	5099
	bewachsen	0	0	0	0	6221	0	0	6221
	Laubwald	0	0	0	0	0	10470	0	10470
	Nadelwald	0	0	0	0	0	0	2191	2191
	Summe	168	1054	840	5099	6221	10470	2191	26043

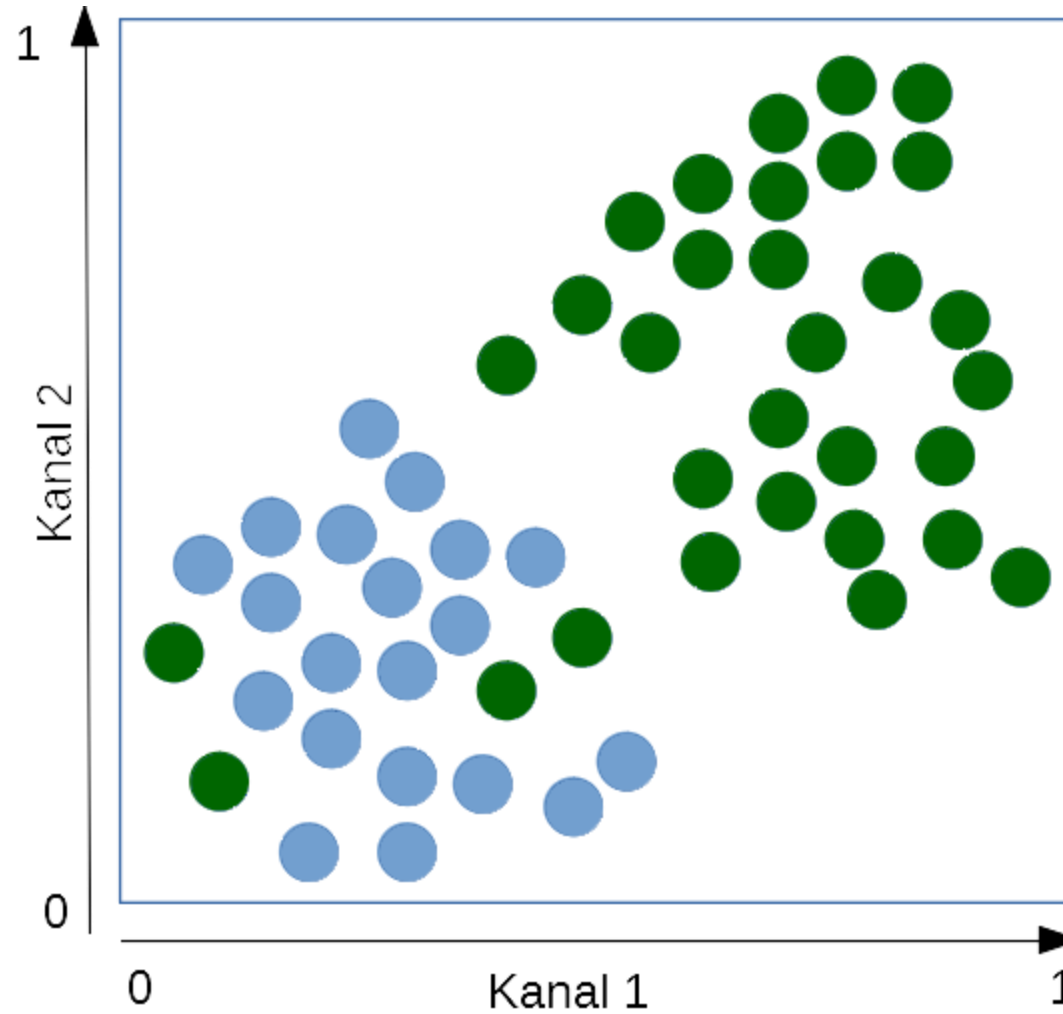
**Perfektes Modell??**

# Trainingsfehler vs. Testfehler

- Trainingsfehler gibt Hinweis wie gut das Modell die Trainingsdaten reproduzieren kann.
- Flexible Algorithmen (z.B. Random Forest) können Trainingsdaten I.d.R sehr gut fitten
- Keine Aussage, wie gut das Modell neue unbekannte Daten klassifizieren kann
- Deshalb: Validierung IMMER mit unabhängigen Daten (Daten, die nicht in das Modelltraining eingegangen sind)!

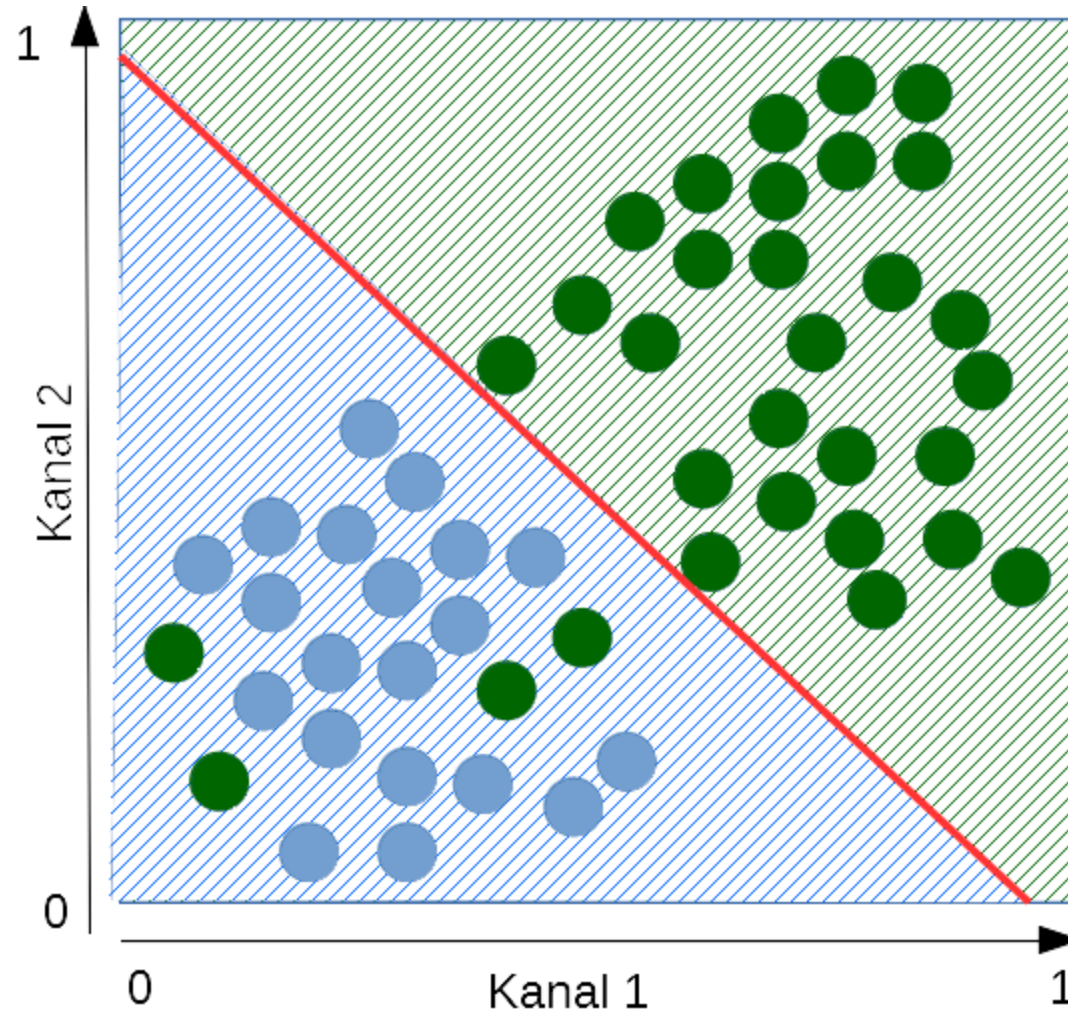


# Trainingsfehler vs. Testfehler



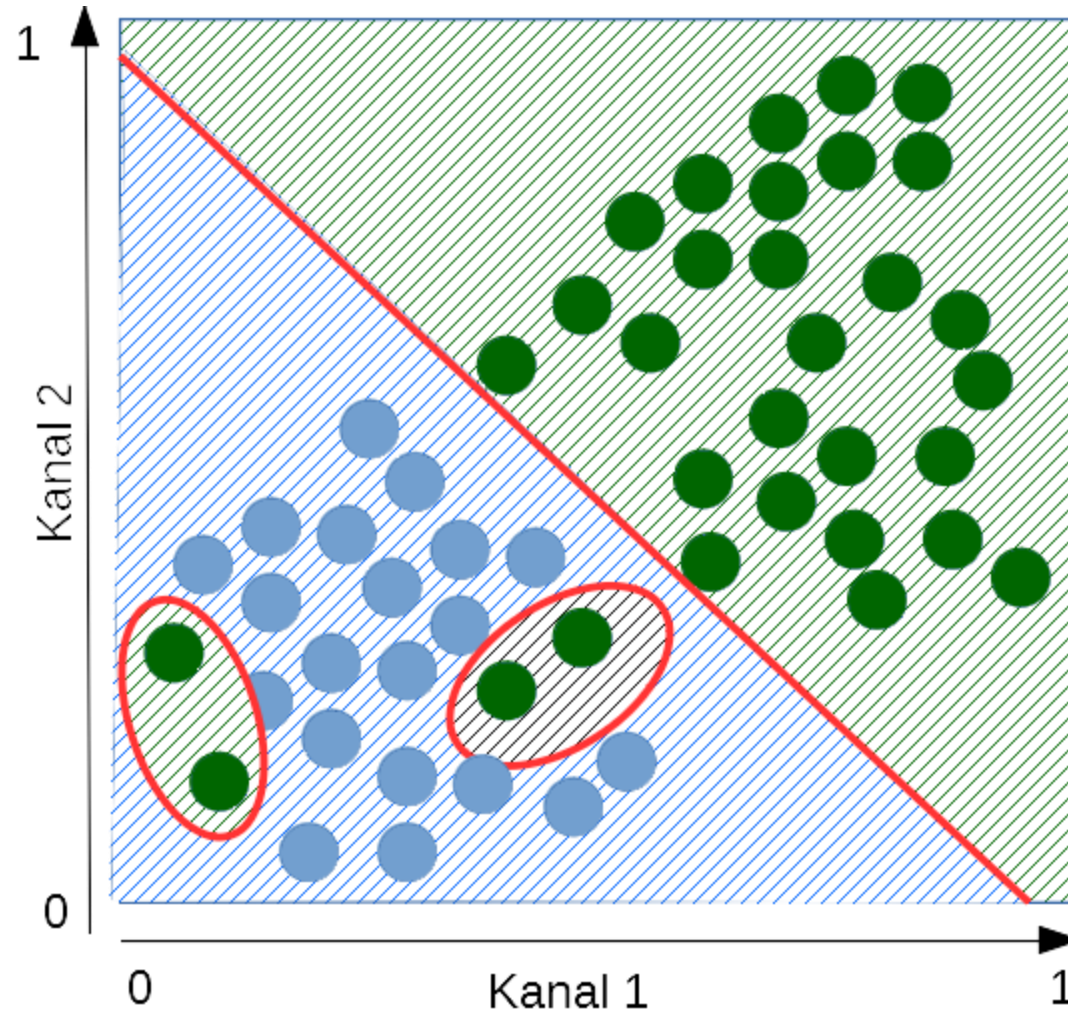
Beispiel für ein Klassifikationsproblem

# Trainingsfehler vs. Testfehler



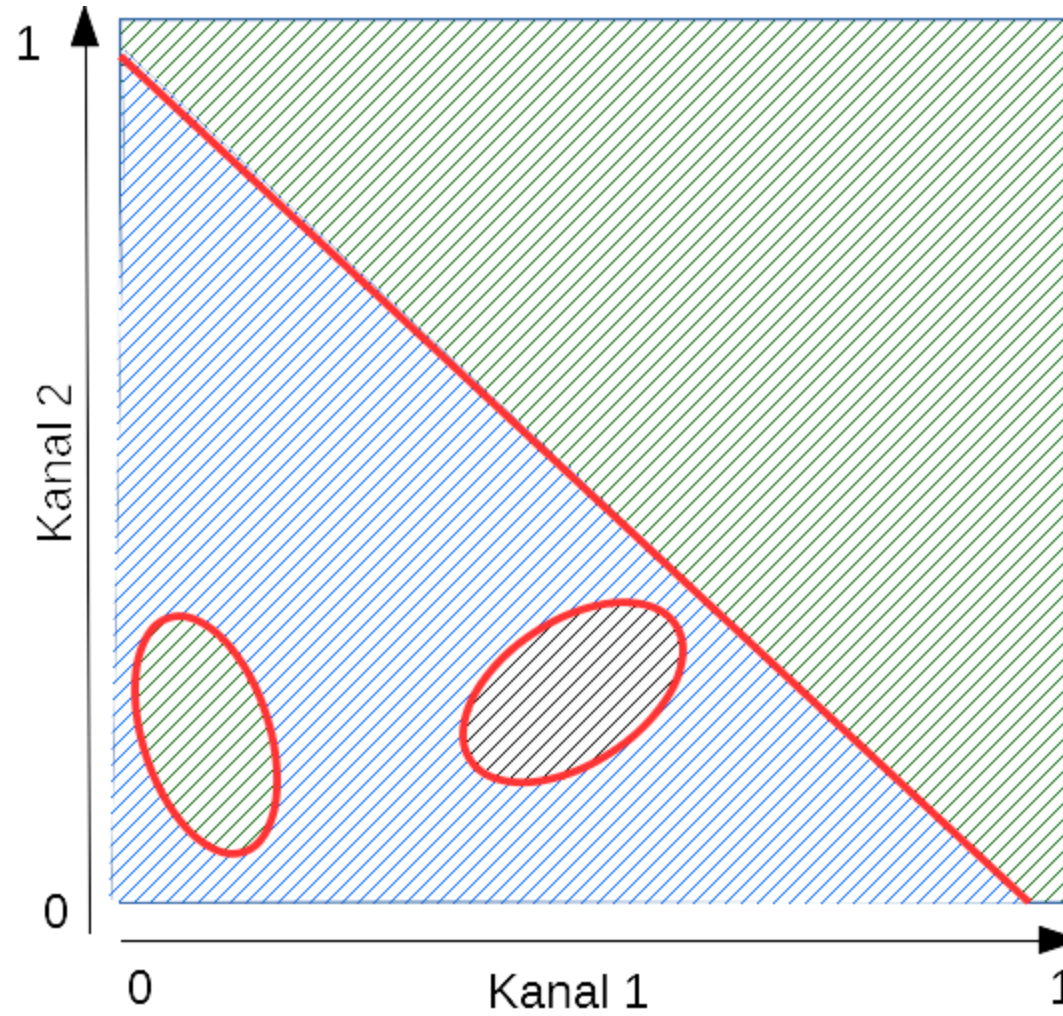
Modell #1: Einfaches Modell

# Trainingsfehler vs. Testfehler



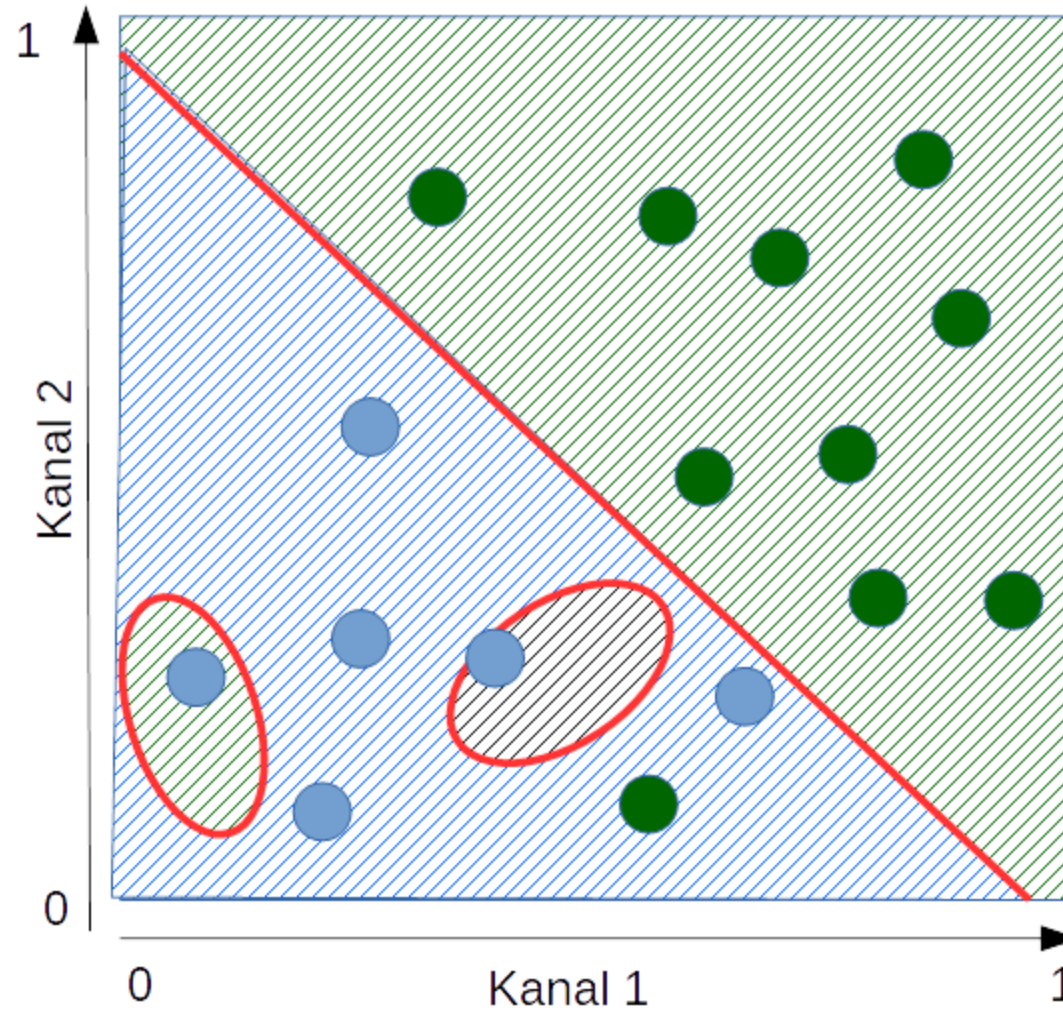
Modell #2: Komplexes Modell

# Trainingsfehler vs. Testfehler



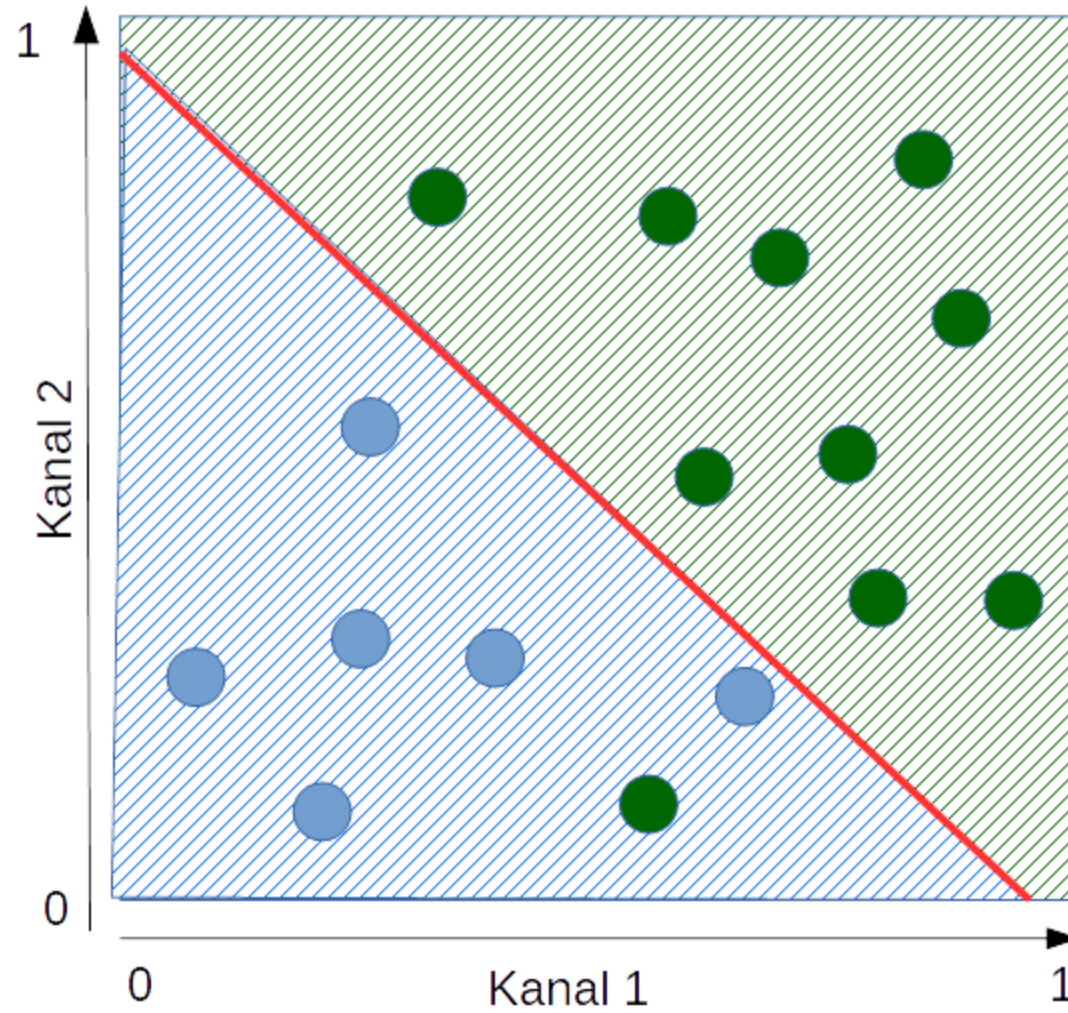
Modell #2: Komplexes Modell

# Trainingsfehler vs. Testfehler



Externe Validierung Modell #2

# Trainingsfehler vs. Testfehler



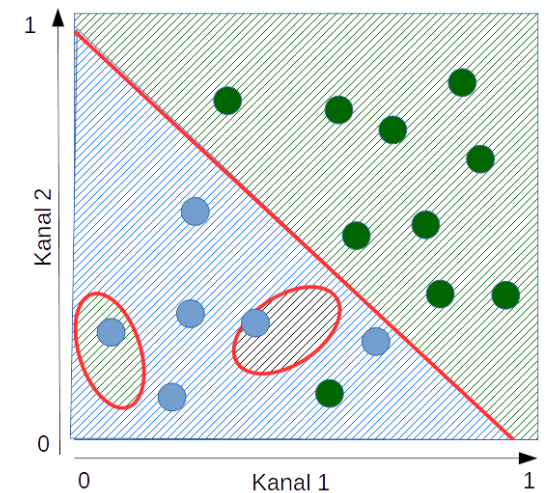
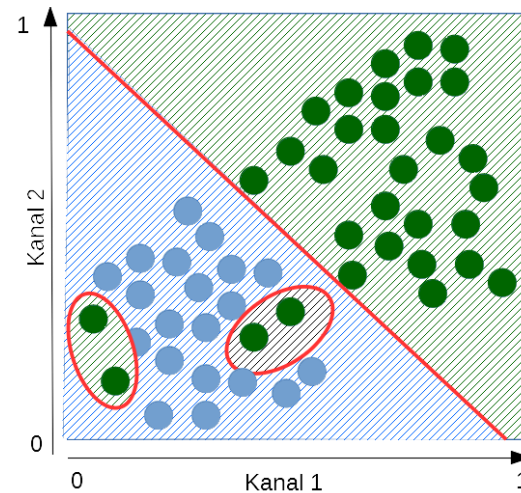
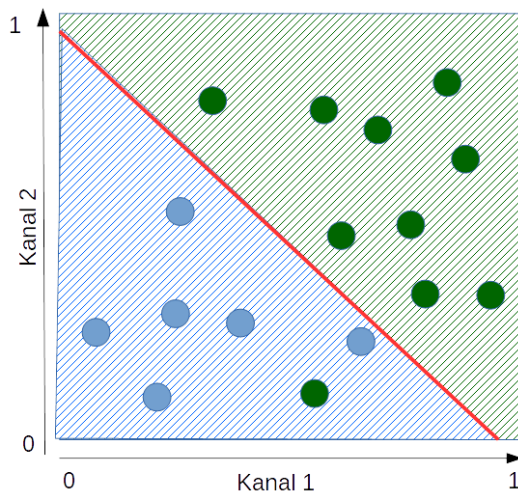
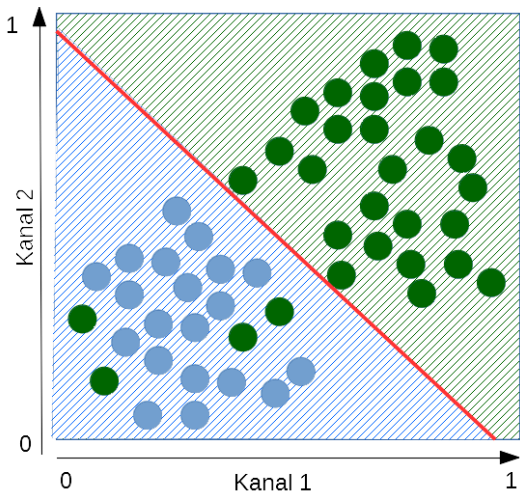
Externe Validierung Modell #1




# Trainingsfehler vs. Testfehler

## Alle noch dabei?

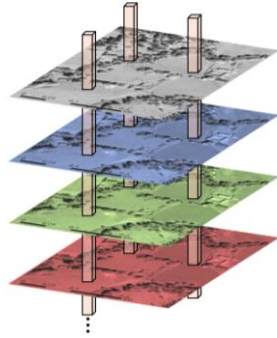
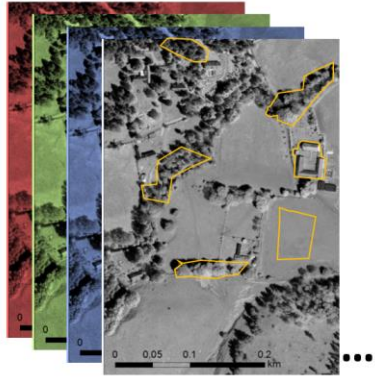
- Was ist der Unterschied zwischen Test- und Trainingsfehler?
- Welche Frage kann der Testfehler beantworten?
- Welche Frage kann der Trainingsfehler beantworten?
- Was ist Overfitting?
- Warum sind komplexere Modelle nicht immer besser?
- Warum sind generelle Modelle nicht immer besser?
- Wie finden wir heraus, welches Modell besser ist?



# Trainingsfehler vs. Testfehler

- Trainingsfehler ist nicht ausreichend um Modellperformanz zu beschreiben
- Deshalb: Validierung IMMER mit unabhängigen Daten (Daten, die nicht in das Modelltraining eingegangen sind)!
- Mehrere Möglichkeiten:
  - Neue Referenzdaten erstellen
  -  ▪ Trainingsdaten vor Modelltraining in Training/Test teilen
  - Kreuzvalidierung (wiederholtes Teilen in Training/Test)

# Modellierung



Pix ID	R	G	B	NIR	Klasse
1	393	680	792	3550	Nadelwald
2	2167	1586	1476	2853	brach
3	859	940	1006	2434	Gebäude
4	705	1133	1012	4910	bewachsen
5	947	1024	1138	2253	Gebäude
...					
105	402	668	800	3467	?
106	698	1142	1003	5012	?
107	2203	1601	1386	2976	?
108	879	962	956	2312	?
...					



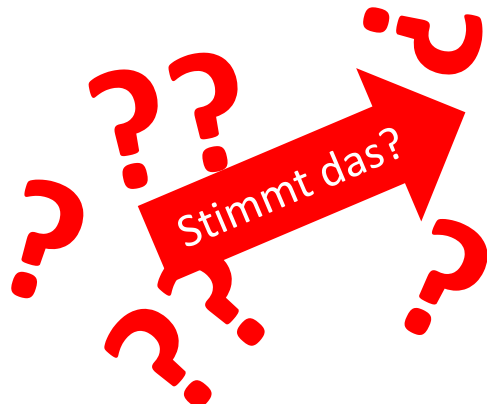
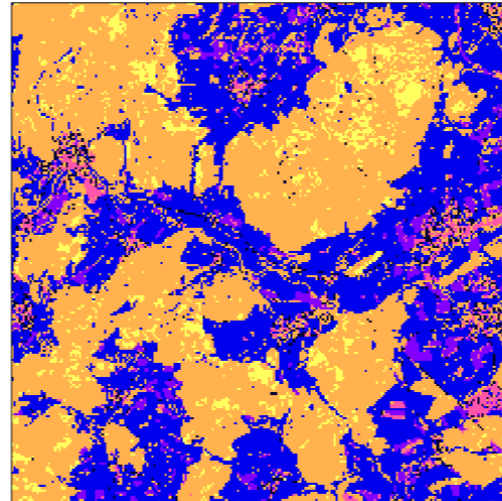
Algorithmus lernt  
Zusammenhänge



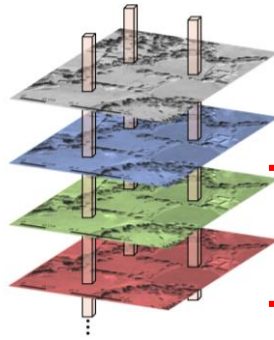
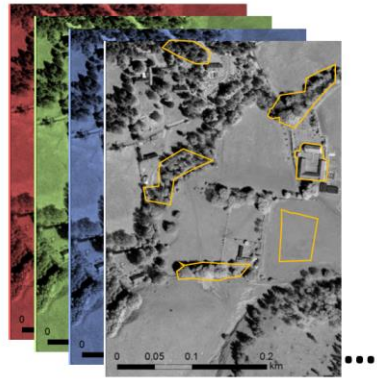
räumliche  
Vorhersage



Modellierung



# Modellierung



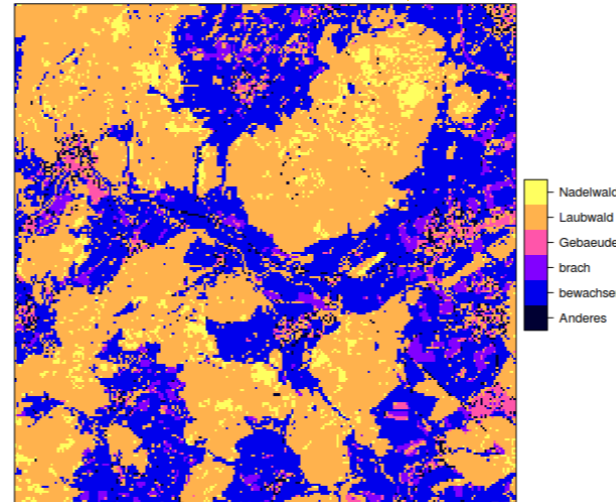
Pix ID	R	G	B	NIR	Klasse
1	393	680	792	3550	Nadelwald
2	2167	1586	1476	2853	brach
3	859	940	1006	2434	Gebäude
...					
4	705	1133	1012	4910	bewachsen
5	947	1024	1138	2253	Gebäude
...					
107	2203	1601	1386	2976	?
108	879	962	956	2312	?
...					

Training  
Test  
restliche Pixel

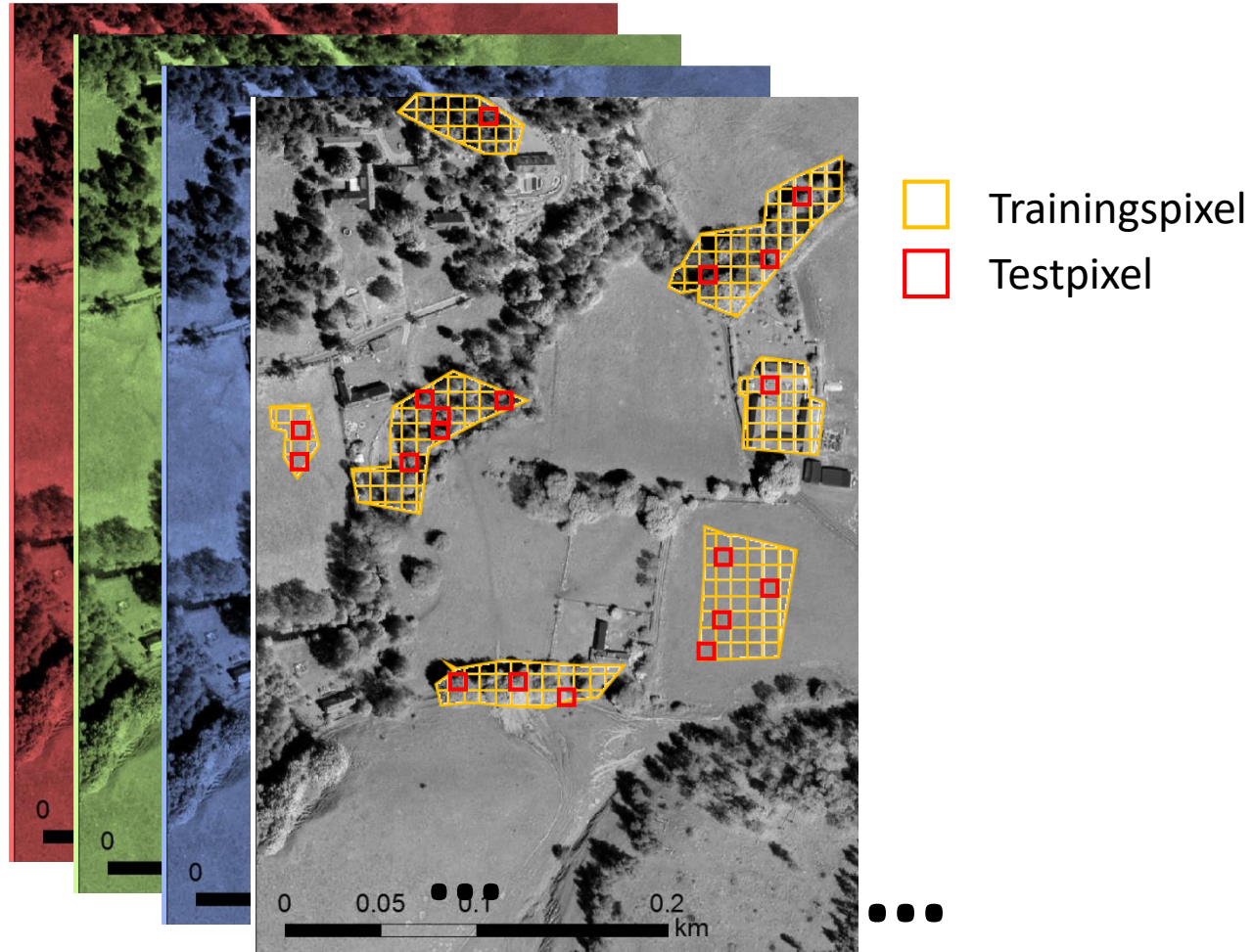
Algorithmus lernt Zusammenhänge

räumliche Vorhersage

Modellierung



# Validierung



Quelle: Eigene Darstellung  
Bildaten: SPOT 5, Aufnahmedatum 2010 (ca)  
Eigene Abbildung (2012)

# Validierung - Accuracy

Klassifikations Daten	Referenz Daten							
	Wasser	Gebäude	anderes	brach	bewachsen	Laubwald	Nadelwald	Summe
	69	0	1	0	0	1	0	71
	0	372	33	38	8	0	0	451
	0	65	203	30	62	0	0	360
	0	9	1	2148	26	0	0	2184
	0	17	6	20	2605	18	0	2666
	0	0	0	0	32	4444	10	4486
	0	0	0	0	0	19	920	939
Summe	69	463	244	2236	2733	4482	930	11157

$$Accuracy = \frac{\sum_{i=1}^k x_{ii}}{N} = \frac{\text{Summe aller richtig klassifizierten Objekte}}{\text{Summe aller Objekte}}$$

$$Accuracy = \frac{69 + 372 + 203 + 2148 + 2605 + 4444 + 920}{11157} = 0.96450659$$

$$Producer's Accuracy = \frac{\text{Summe aller richtig klassifizierten Objekte einer Klasse}}{\text{Anzahl Referenzobjekte dieser Klasse}}$$

$$User's Accuracy = \frac{\text{Summe aller richtig klassifizierten Objekte einer Klasse}}{\text{Anzahl klassifizierte Objekte dieser Klasse}}$$



# Validierung - Kappa

Klassifikations Daten	Referenz Daten								Summe	User's
	Wasser	Gebäude	anderes	brach	bewachsen	Laubwald	Nadelwald			
	Wasser	69	0	1	0	0	1	0	71	0.97
	Gebäude	0	372	33	38	8	0	0	451	0.82
	anderes	0	65	203	30	62	0	0	360	0.56
	brach	0	9	1	2148	26	0	0	2184	0.98
	bewachsen	0	17	6	20	2605	18	0	2666	0.98
	Laubwald	0	0	0	0	32	4444	10	4486	0.99
	Nadelwald	0	0	0	0	0	19	920	939	0.98
Summe	69	463	244	2236	2733	4482	930	11157		
Producer's	1.00	0.80	0.83	0.96	0.95	0.99	0.99			

Accuracy = 0.96  
Kappa = 0.95

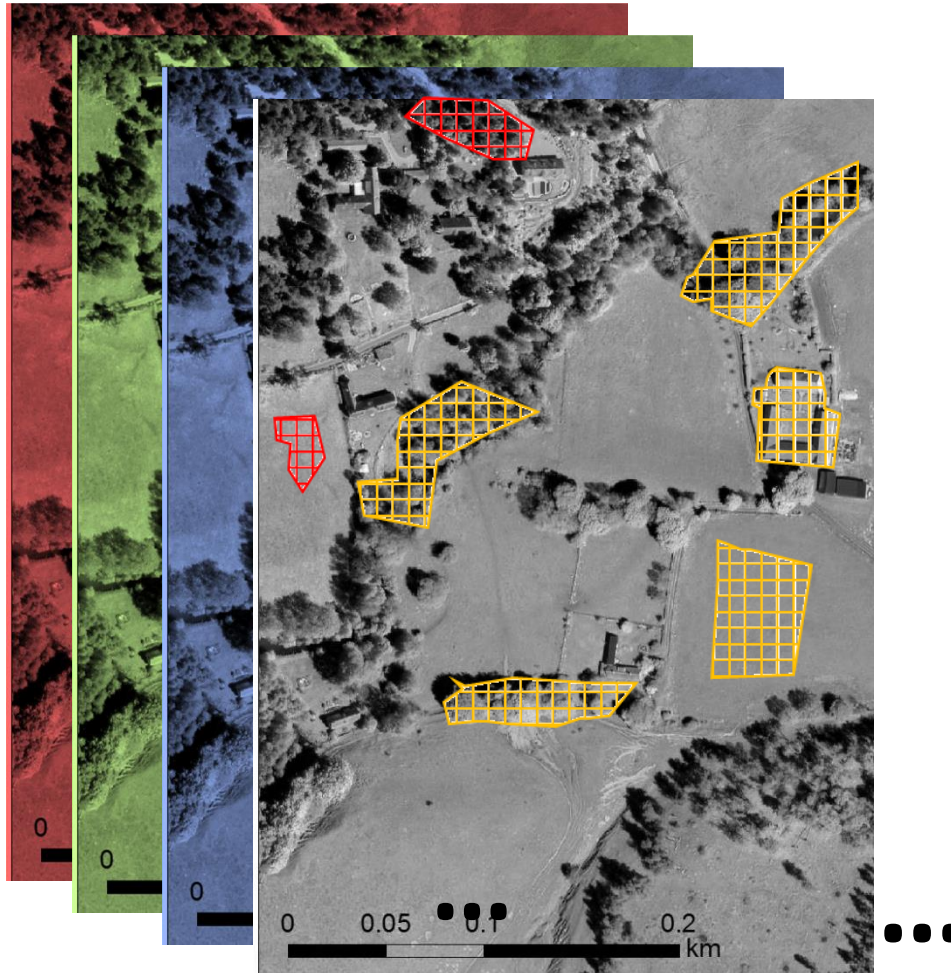
$$Kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

$p_o$  = observed agreement = Accuracy

$$p_e = \text{expected agreement} = \frac{1}{N^2} \sum_{x=1}^k x_{i1} * x_{i2}$$

$$Kappa = 1 - \frac{1 - 0.96450659}{1 - \frac{1}{11157^2} * ((69 * 71) + (463 * 451) + (244 * 360) + (2236 * 2184) + (2733 * 2666) + (4482 * 4486) + (930 * 939))} = 0.951463579$$

# Validierung



Quelle: Eigene Darstellung  
Bilddaten: SPOT 5, Aufnahmedatum 2010 (ca)  
Eigene Abbildung (2012)

# Räumliche Validierung – mit Polygonen

		Referenz Daten							Klassifikations Daten
		Wasser	Gebäude	anderes	brach	bewachsen	Laubwald	Nadelwald	
Wasser	Summe	150	10	6	0	0	3	0	
Gebäude	Summe	0	464	302	64	12	0	0	
anderes	Summe	0	47	148	35	42	2	0	
brach	Summe	0	39	55	2764	238	0	0	
bewachsen	Summe	0	9	143	190	2148	420	0	
Laubwald	Summe	0	0	0	0	28	9370	13	
Nadelwald	Summe	0	0	3	0	0	69	2131	
Summe	Summe	150	569	657	3053	2468	9864	2144	

User's
0.89
0.55
0.54
0.89
0.74
1.00
0.97

Accuracy = 0.91  
Kappa = 0.87

Producer's	1.00	0.82	0.23	0.91	0.87	0.95	0.99
------------	------	------	------	------	------	------	------

Kappa	Interpretation
<0.0	Keine Übereinstimmung
0 - 0.2	Kaum Übereinstimmung
>0.2 - 0.4	Schwache Übereinstimmung
>0.4 – 0.6	Mäßige Übereinstimmung
>0.6 – 0.8	Gute Übereinstimmung
>0.8 - 1	Fast perfekte Übereinstimmung

**ABER:**  
Keine allgemeingültige Interpretation möglich!  
Abhängig von Fragestellung und erwarteter Übereinstimmung.

# Validierung

## Vorhersage Testdaten

- Sagen Sie Ihren Testdatensatz vorher (?predict)
- Erstellen Sie aus den tatsächlichen Klassen und den vorhergesagten Klassen Ihres Testdatensatzes eine Kreuztabelle (?table)
- Berechnen Sie per Hand/R/Excel die Accuracy Ihres Datensatzes (keine Funktionen)

# Validierung

## Vorhersage Testdaten

- Überprüfen Sie Ihr Ergebnis und berechnen Sie Kappa (?confusionMatrix)

# Validierung

## Räumliche Validierung

- Teilen Sie Ihren Datensatz wieder im Verhältnis 3:7 in Test- und Trainingsdatensatz auf. Beachten Sie diesmal, dass Sie die Polygone als ganze Objekte auf die beiden Datensätze aufteilen. (?unique, ?sample)
- Gehen Sie analog zu der pixelbasierten Vorgehensweise vor und trainieren Sie ein Modell (?train), erstellen Sie eine Vorhersage (?predict) und berechnen Sie Accuracy und Kappa (?confusionMatrix).
- Vergleichen Sie die verschiedenen Fehlermaße der verschiedenen Vorhersagen.