

Landnutzungsklassifikation - Modellierung

Marvin Ludwig, Alice Ziegler

05/06/2019

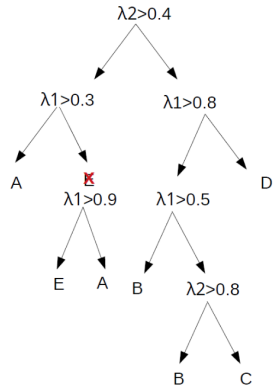
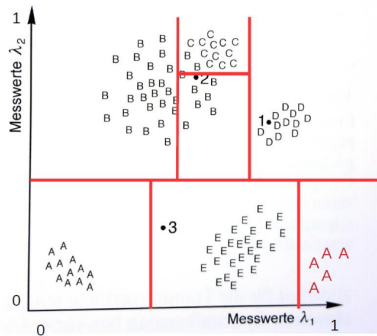
Wiederholung

Was machen wir eigentlich?

- ▶ Einteilen eines Datensatzes in Gruppen
 - ▶ Überwacht
 - ▶ Gruppen: Landnutzungsklassen (response)
 - ▶ Datensatz: Spektrale Information (predictor)
- ▶ Klassifikationsbaum arbeitet mit Grenzen
 - ▶ Wie errechnen sich diese Grenzen?

Wiederholung

Klassifikationen



Erstelle einen Klassifikationsbaum

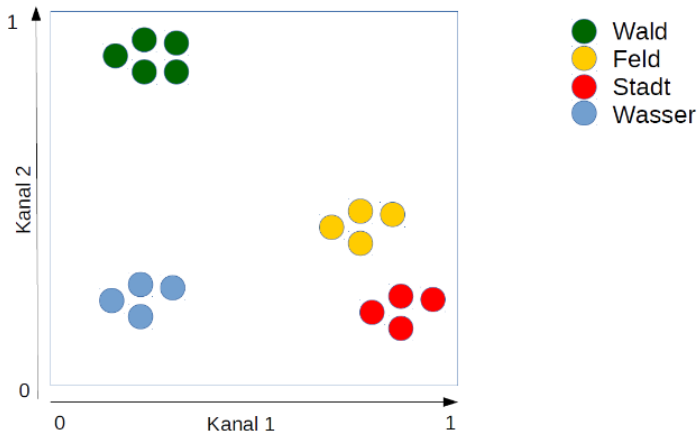


Figure 1: Uebung Klassifikationsbaum

Beispiel mit dem Iris Datensatz

```
data(iris)
iris <- iris[,c(3:5)]
head(iris)
```

##	Petal.Length	Petal.Width	Species
## 1	1.4	0.2	setosa
## 2	1.4	0.2	setosa
## 3	1.3	0.2	setosa
## 4	1.5	0.2	setosa
## 5	1.4	0.2	setosa
## 6	1.7	0.4	setosa

- ▶ Predictoren: Petal.Length, Petal.Width
- ▶ Response: Species

Beispiel mit dem Iris Datensatz

```
# install.packages("tree")
```

```
library(tree)
```

```
kb <- tree(Species ~ .,  
           data = iris,  
           split = "deviance")  
summary(kb)
```

```
##
```

```
## Classification tree:
```

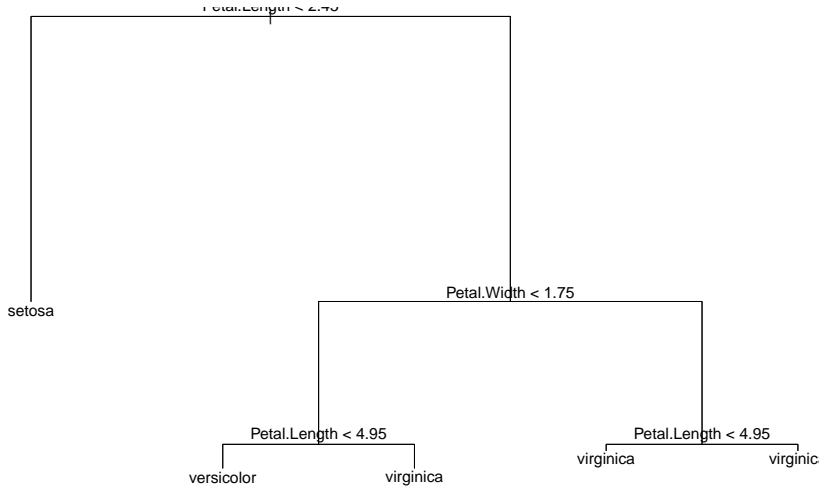
```
## tree(formula = Species ~ ., data = iris, split = "deviance")
```

```
## Number of terminal nodes: 5
```

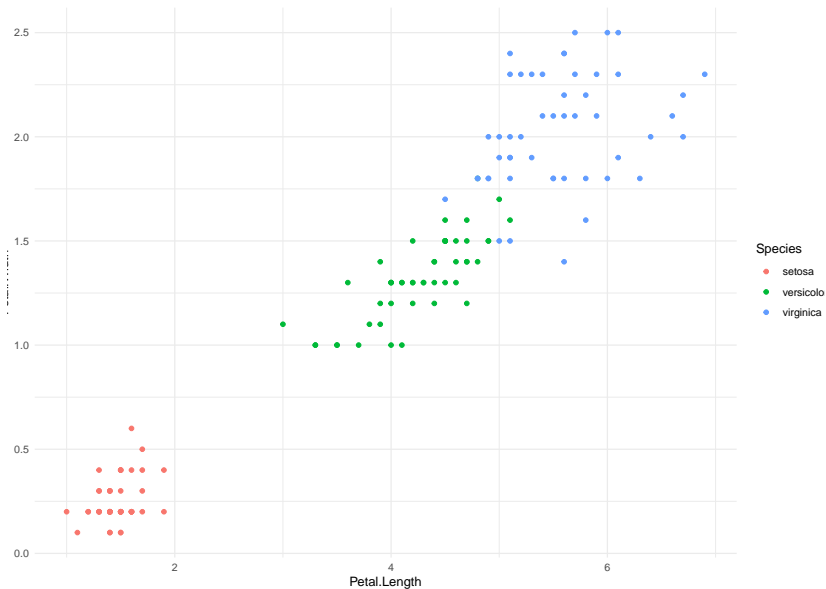
```
## Residual mean deviance: 0.157 = 22.77 / 145
```

```
## Misclassification error rate: 0.02667 = 4 / 150
```

Beispiel mit dem Iris Datensatz



Beispiel mit dem Iris Datensatz



Wie werden die Grenzen berechnet?

- ▶ Maschinelles Lernen
 - ▶ Lernt “selbstständig” Zusammenhänge in den Daten
 - ▶ Benutzt dafür Fehlermaße um optimale Modelle zu finden
 - ▶ Dafür wird der Datensatz aufgeteilt um zu testen, wie gut die Einteilung der Gruppen funktioniert

Nochmal mit dem Ganzen Iris Datensatz

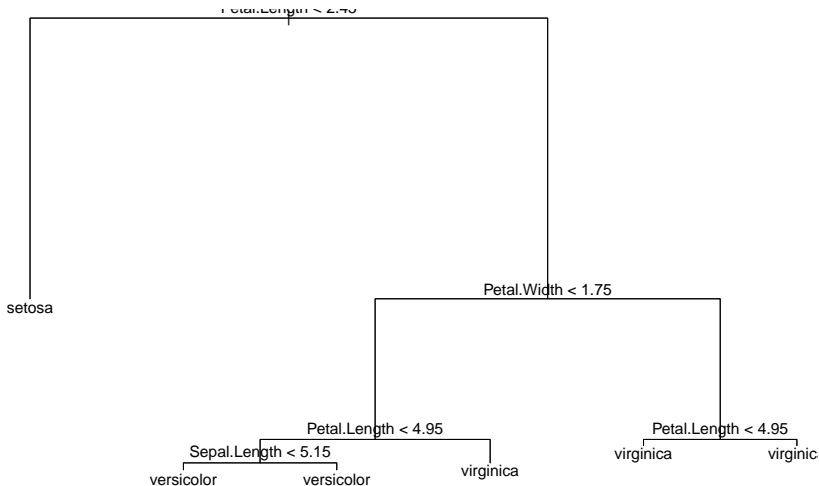
```
data(iris)
head(iris)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa

```
kb <- tree(Species ~ .,
           data = iris,
           split = "deviance")
```

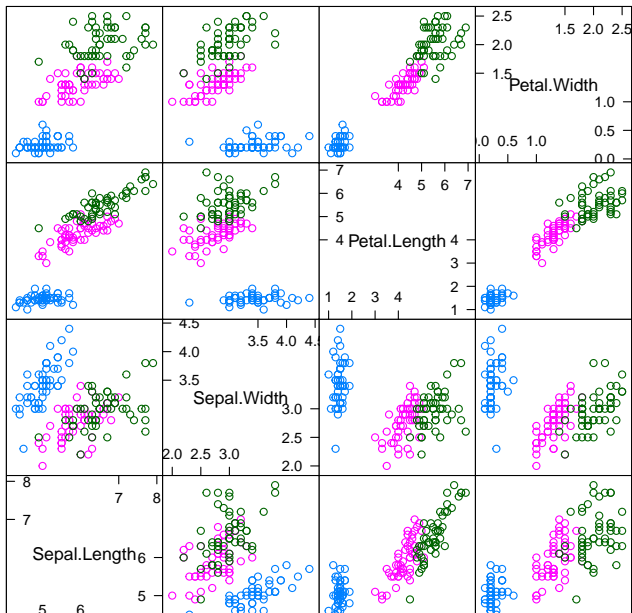
Nochmal mit dem ganzen Iris Datensatz

```
plot(kb); text(kb)
```



Nochmal mit dem ganzen Iris Datensatz

setosa ○
versicolor ○
virginica ○



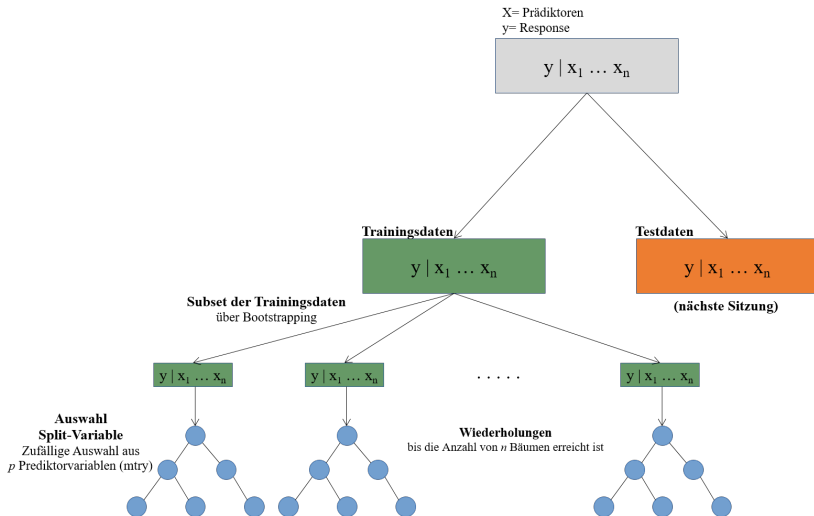
Random Forest

- ▶ 2001 von Leo Breiman
- ▶ Klassifikationsbaum ist die Grundlage

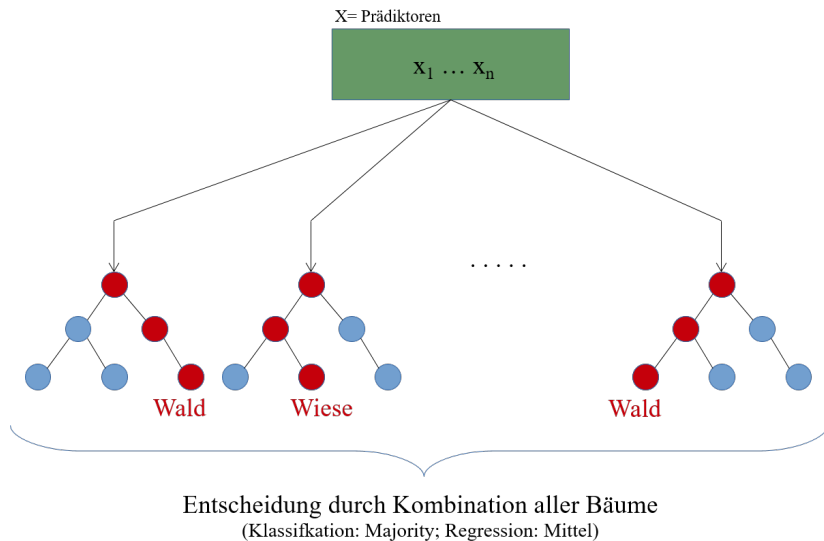
Was ist random an Random Forest?

- ▶ Bootstrap Sample fuer jeden Baum
 - ▶ “Ziehen mit zuruecklegen”
 - ▶ Für jeden Klassifikationsbaum wird also eine andere Datengrundlage verwendet
- ▶ Praediktorenauswahl in jedem Baum
 - ▶ Anzahl der betrachteten Praediktoren wird festgelegt (mtry)
 - ▶ mtry ist der sogenannte “tuning parameter” für Random Forest
 - ▶ Ein Ziel des maschinellen Lernens ist es diesen Parameter optimal zu bestimmen

Was ist random an Random Forest?



Was ist random an Random Forest?



Random Forest in R

- Installieren Sie die Pakete caret und ranger

Random Forest in R

- ▶ Laden Sie die Datentabelle (.RDS Datei) von letzter Woche ein
- ▶ Splitten Sie den Datensatz in einen Training- und Testdatensatz
 - ▶ Training soll zufällig 70% der Daten beinhalten
 - ▶ Test dementsprechend die übrigen 30%
 - ▶ `?set.seed()`
 - ▶ `?createDataPartition()`

Random Forest in R

- ▶ Trainieren sie ein Random Forest Model
 - ▶ Nutzen Sie dafür den Trainingsdatensatz
 - ▶ `caret::train()`
 - ▶ Speichern Sie das Modell als RDS-Datei ab

Random Forest in R

- ▶ Wenden Sie das Modell auf das gesamte Untersuchungsgebiet an
 - ▶ `?raster::predict()`
- ▶ Schauen Sie sich das Ergebnis an
 - ▶ `?spplot()`

Projektarbeit

Welche Spezialisierungen wären für einen Artikel denkbar?

- ▶ onlinequestions.org
- ▶ Event number: 14831