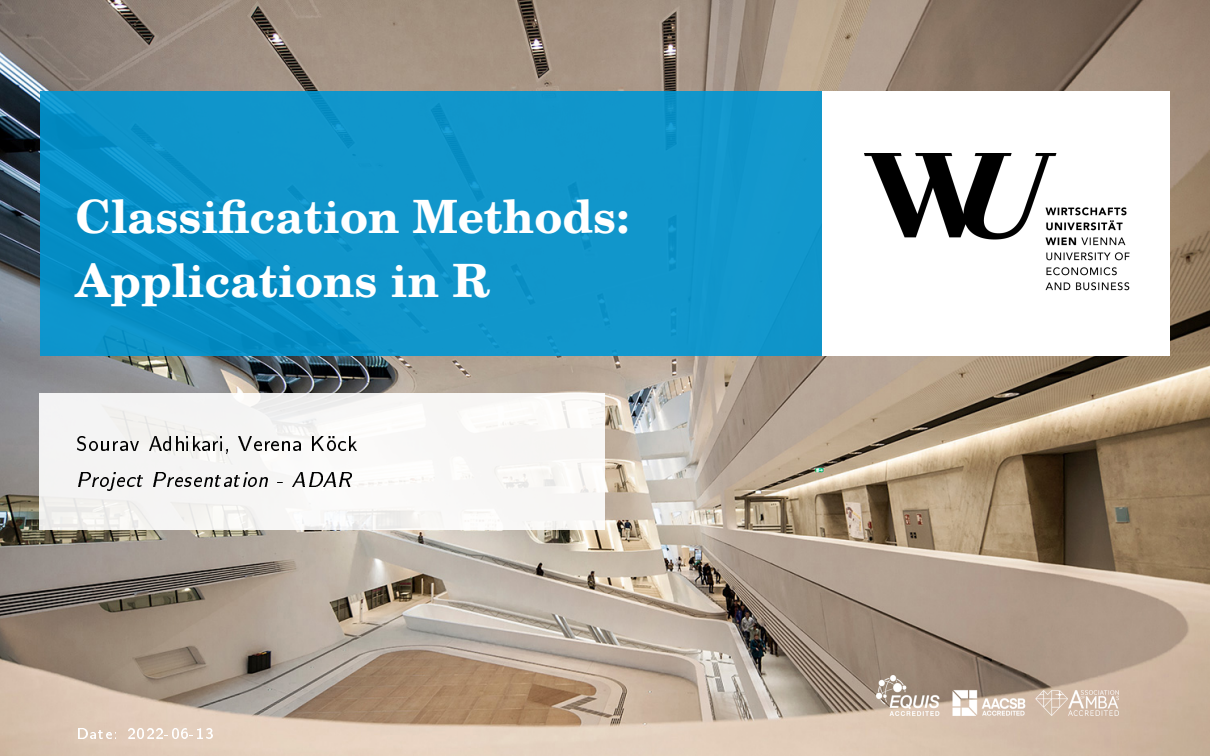
Classification Methods: Applications in R

Sourav Adhikari, Verena Köck

Project Presentation - ADAR



WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Date: 2022-06-13

(Statistical) Classification: What is it?

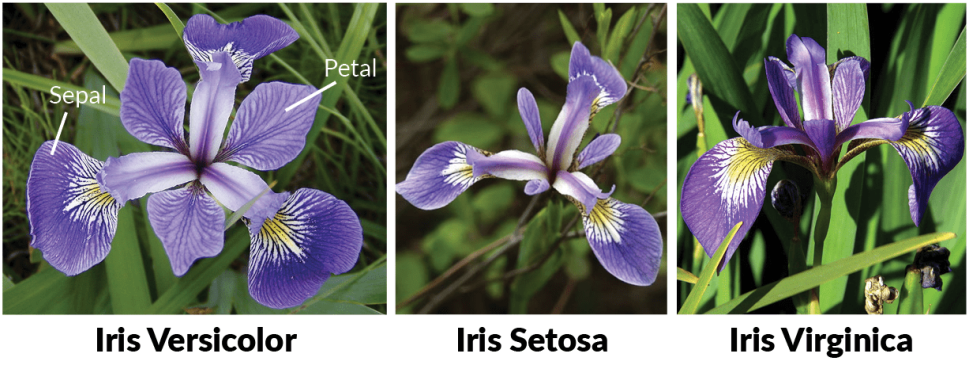
- The problem of identifying which of a set of categories an observation belongs to.
  - E.g. assigning an incoming email to "spam" or "inbox" mailbox.
- Classification can be thought of as two separate problems:
  - binary classification
  - multiclass classification.

- **Examples** for classification methods are:
  - Naive Bayes
  - k-Nearest Neighbors
  - Neural Networks
  - Others: Decision Trees, Random Forest, Logistic Regression, SVM, etc.

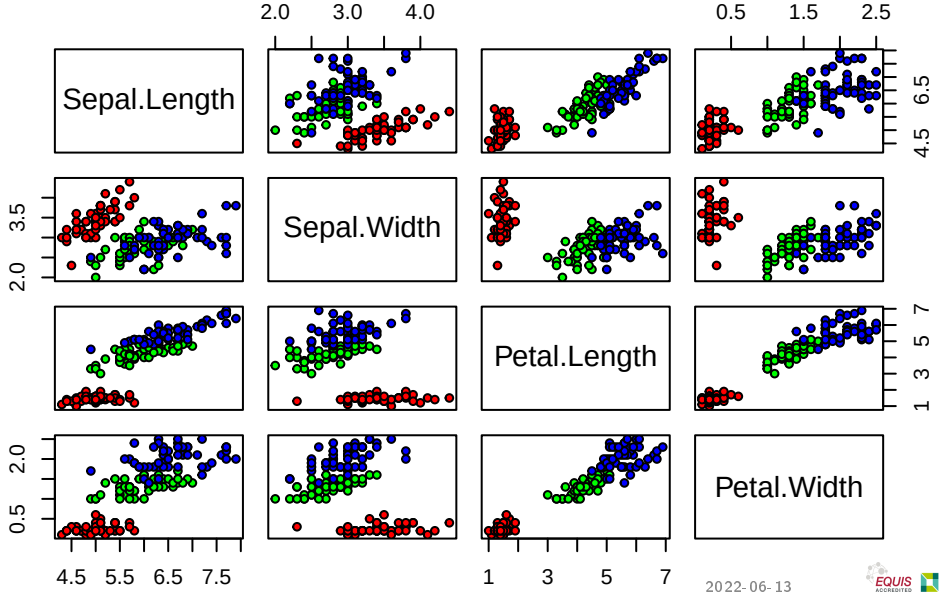
- **This project:** We explain and present results from first three methods: Naive Bayes, k- Nearest Neighbors and Neural Networks.

The IRIS dataset I

- The data contains 4 measurements for 150 flowers from each of three species of *iris*:
  - `Sepal.Length`, `Sepal.Width`, `Petal.Length` and `Petal.Width` in cm
  - `Species`: setosa, virginica and versicolor

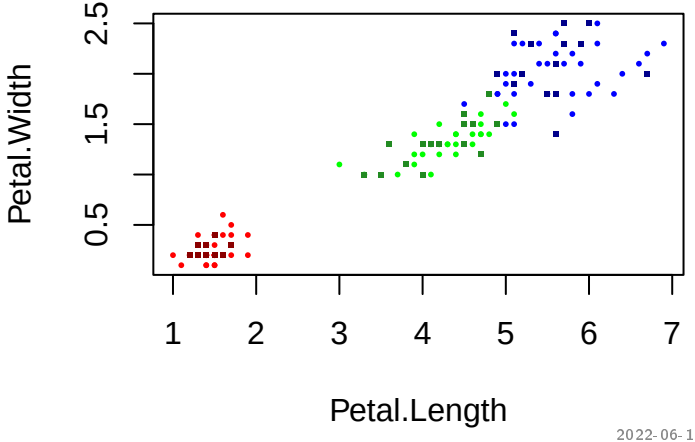


IRIS Data: setosa(red), versicolor(gr.), virginica(bl.)



Validation of the training procedure

- We split the data into training (points) and testing set (squares) in the ratio 67:33.
- We perform k-fold cross validation.



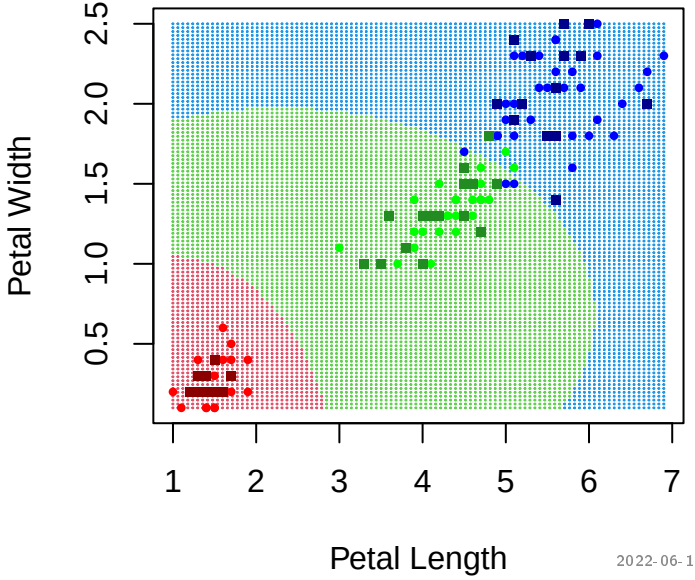
Naive Bayes

- Naive Bayes classifiers are simple "probabilistic classifiers" based on Bayes' theorem.
- *Disadvantage*: (**Strong**) assumption, that the features are independent (i.e presence of one particular feature does not affect the other). Hence the adjective **naive**.
- *Advantage*: Requires only a small number of training data to estimate the parameters.
- Let y be the category variable, and X the features, then **Bayes theorem** is:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)},$$

- **Steps:**
  1. Estimate prior probability $P(X)$: Compute the relative frequency of each class/species.
  2. Assume normal distribution for each class (species). Estimate μ and σ^2 for each class.
  3. For a new observation, apply Bayes theorem (and normalize) to get a vector of probabilities, e.g. $(\mathbf{0.5}, \mathbf{0.25}, \mathbf{0.25})$!

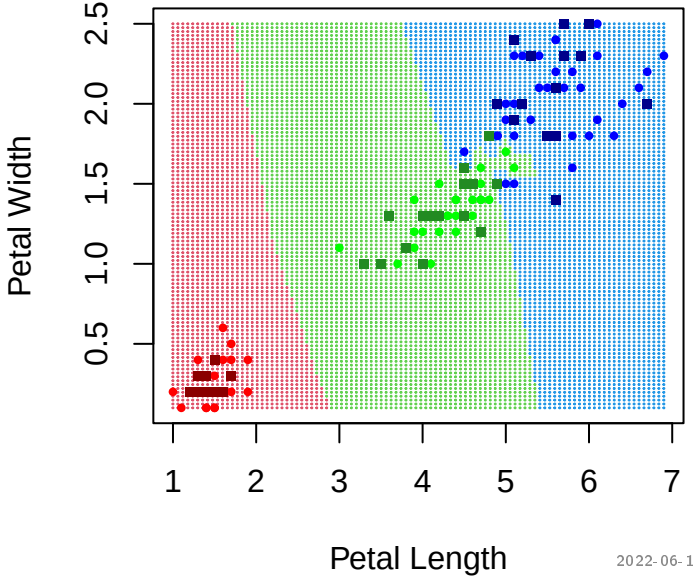
Naive Bayes in R



K-nearest neighbors

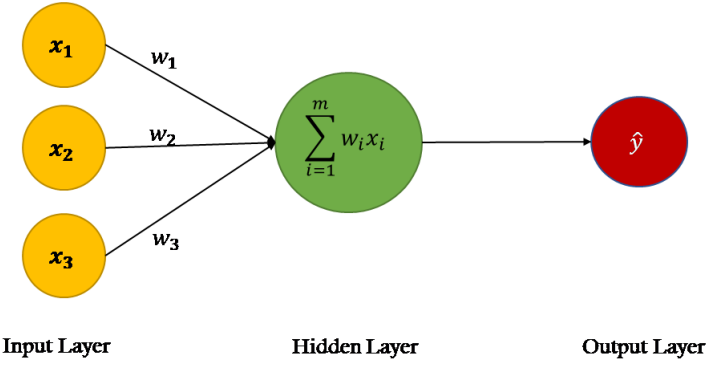
- A non-parametric supervised learning method
- Uses a distance metric to make classifications or predictions about the grouping of an individual data point.
- Object is assigned to the class it is most common with among its k nearest neighbors.
- *Advantages*: Easy to understand and implement, no assumptions required
- *Disadvantages*: Curse of Dimensionality

1-nearest neighbour



Neural Networks

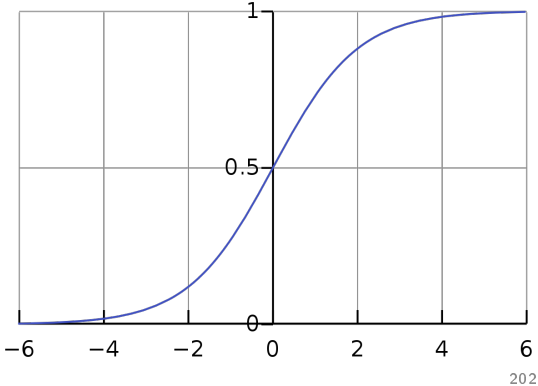
- Neural networks (NNs) are computing systems inspired by the biological neural networks that constitute animal brain.
- They learn forming probability-weighted associations between “input" and “result".



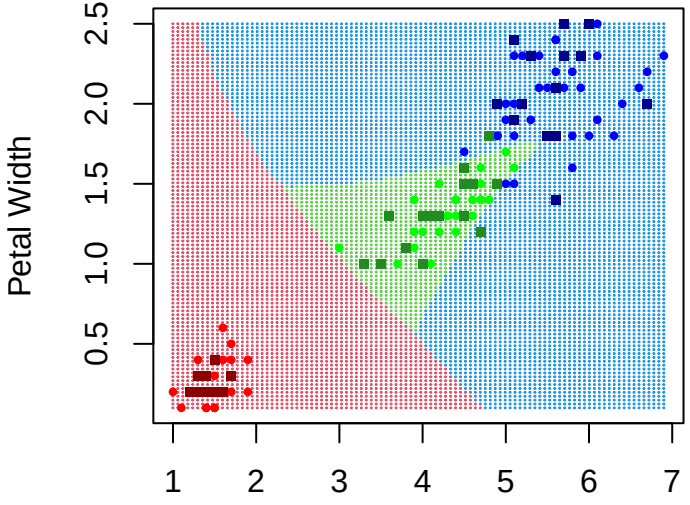
Neural Networks

- For classification tasks, NNs utilize an activation function, for example a logistic function:

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}}$$



Neural network



Cross Validation

- A resampling procedure for model validation.
- Assessing how the results of a statistical analysis will generalize to an independent data set.
- Estimates how accurately a predictive model will perform in practice.
- For the Iris dataset, we specifically perform the k-fold cross validation.

k-fold Cross Validation

- Shuffle the dataset randomly.
- Split the dataset into k groups.
- For each unique group:
  - Take the group as a hold out or test data set.
  - Take the remaining groups as a training data set.
  - Fit a model on the training set and evaluate it on the test set.
  - Retain the evaluation score and discard the model.
- Summarize the skill of the model using the sample of model evaluation scores.
- Note: The k value must be chosen carefully for the data sample.

k-fold Cross Validation

- Performed using the function `train()` included in the R package `caret`.

```
library(caret)

## Lade nötiges Paket:  ggplot2
## Lade nötiges Paket:  lattice

tc <- trainControl(method = "cv", number = 10)
fit <- train(Species  .,data = data.frame(iris.training,"Species"= iris.trainingtarget), method
= "nb",trControl = tc, metric = "Accuracy")
```

- k=10 is used.
- Qualitative aspect is measured by Cohen's kappa score and the accuracy measure.
- Measures the agreement between two raters who each classify N items into C mutually exclusive categories:

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

k-fold Cross Validation: Naive Bayes

```
library(caret)
set.seed(1)
tc <- trainControl(method = "cv", number = 10)
fit <- train(Species ~.,data = data.frame(iris.training,"Species"=
  iris.trainingtarget),method = "nb",trControl = tc, metric = "Accuracy")
fit$results

##   usekernel fL adjust  Accuracy     Kappa AccuracySD    KappaSD
## 1     FALSE  0      1 0.9607071 0.9403631  0.0510286 0.07750075
## 2      TRUE  0      1 0.9507071 0.9252115  0.0522565 0.07933520

pred <- predict(fit, iris.test)
result<-confusionMatrix(iris.testtarget, pred)
sum(diag(result$table))/sum(result$table)

## [1] 0.9574468
```

k-fold Cross Validation: k-nearest Neighbours

```
set.seed(1)
knn_fit <- train(Species ~.,data = data.frame(iris.training,"Species"=
  iris.trainingtarget), method = "knn", trControl=tc, metric = "Accuracy")
knn_fit$results

##   k  Accuracy     Kappa AccuracySD    KappaSD
## 1 5 0.9609091 0.9403883 0.05053433 0.07705315
## 2 7 0.9609091 0.9403883 0.05053433 0.07705315
## 3 9 0.9800000 0.9696970 0.04216370 0.06388440

pred <- predict(knn_fit,iris.training)
result<-confusionMatrix(iris.trainingtarget, pred)
sum(diag(result$table))/sum(result$table)

## [1] 0.9805825
```

k-fold Cross Validation: Neural Network

```
set.seed(1)
nnet_fit <- train(Species ~.,data = data.frame(iris.training,"Species"=
  iris.trainingtarget), method = "nnet", trControl=tc, metric = "Accuracy",trace = FALSE)
nnet_fit$results

##   size decay  Accuracy     Kappa AccuracySD    KappaSD
## 1    1 0e+00 0.8212121 0.7208569 0.13431685 0.21313940
## 2    1 1e-04 0.7517172 0.5951669 0.26779376 0.44643182
## 3    1 1e-01 0.9497980 0.9223883 0.07103029 0.10906072
## 4    3 0e+00 0.9536364 0.9280673 0.08891242 0.13915422
## 5    3 1e-04 0.9618182 0.9419596 0.04938557 0.07510644
## 6    3 1e-01 0.9800000 0.9694639 0.04216370 0.06437816
## 7    5 0e+00 0.9309091 0.8930024 0.09459206 0.14926891
## 8    5 1e-04 0.9618182 0.9425365 0.04938557 0.07436975
## 9    5 1e-01 0.9800000 0.9694639 0.04216370 0.06437816

pred <- predict(nnet_fit,iris.training); result<-confusionMatrix(iris.trainingtarget, pred)
sum(diag(result$table))/sum(result$table)

## [1] 0.9902913
```

Summary

Method	Accuracy Score	Accuracy Test
Naive Bayes	0.9607	0.9574
9-NN	0.9800	0.9806
Neural Network	0.9800	0.9903

- Among the methods discussed, kNN with k = 9 offers the highest rate of accuracy.
- There is no “one method fits all" approach when undertaking classification tasks.

References

[1] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, "An Introduction to Statistical Learning : with Applications in R", Springer, 2013.

[2] McCulloch, Warren, Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics. 5 (4): 115–133, 1943.