Computerübung - Performance-Evaluierung

Daniel Stepanovic

Ausgangssituation

In dieser Analyse untersuchen wir, ob sich das Vorliegen einer Diabetesdiagnose (positiv oder negativ) anhand der Prädiktoren **Gewicht**, **Geschlecht** und **Cholesterinspiegel** vorhersagen lässt. Dazu wird zunächst eine deskriptive Analyse durchgeführt, gefolgt von dem Vergleich zweier Klassifikationsmodelle: **k-Nearest Neighbors (k-NN)** und **Naive Bayes**.

Vier Merkmalen:

- diag (Diabetes-Diagnose mit den Kategorien positiv und negativ) ist die abhängige Variable und nominalskaliert
- gewicht (in kg) ist eine metrische unabhängige Variable
- chol (Cholesterinspiegel in mg/dl) ist ebenfalls metrisch
- sex (Geschlecht mit den Kategorien männlich und weiblich) ist nominalskaliert

Datenmanagement

Die Daten wurden mit read.table() eingelesen:

```
daten = read.table("wi23b095.txt", header = TRUE, stringsAsFactors = TRUE)
```

Stichprobe prüfen

```
nrow(daten)
```

```
[1] 156
```

```
summary(daten)
```

```
   diag        gewicht          sex          chol      
 neg :123   Min.   : 45.00   m   :59   Min.   :129.0  
 pos : 22   1st Qu.: 69.50   w   :93   1st Qu.:178.8  
 NA's: 11   Median : 79.00   NA's: 4   Median :203.5  
            Mean   : 80.96             Mean   :208.8  
            3rd Qu.: 91.00             3rd Qu.:234.2  
            Max.   :147.00             Max.   :347.0  
            NA's   :5                  NA's   :4      
```

Entfernen aller Zeilen mit fehlenden Werten

```
daten = na.omit(daten)
nrow(daten)
```

```
[1] 132
```

Nach Entfernung der fehlenden Werte beträgt die Stichprobengröße 132 Beobachtungen. Fehlende Werte sind keine mehr vorhanden.

Absolute Häufigkeiten

```
table(daten$diag)
```

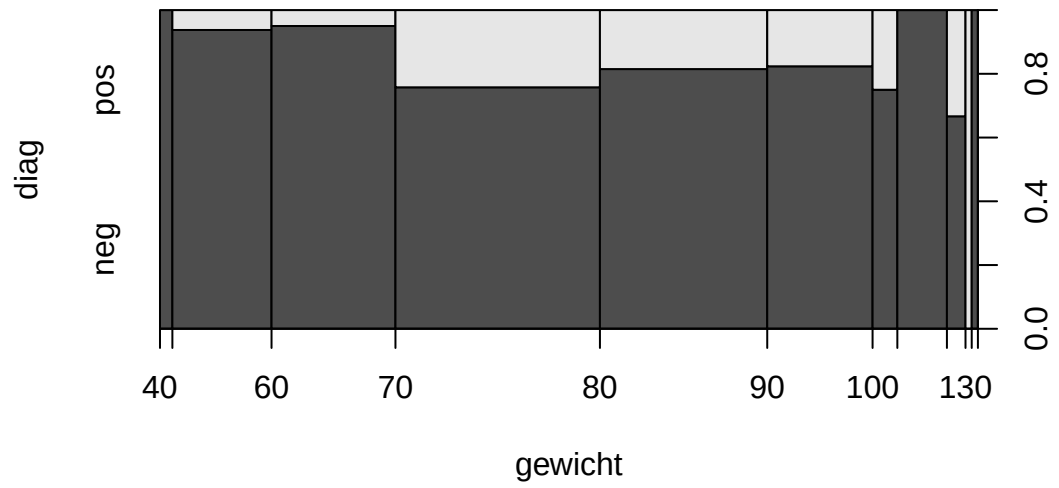
```
neg pos 
111  21 
```

Es zeigt sich, dass negative Diagnosen überwiegen.

Einfluss der einzelnen Prädiktoren

Einfluss des Gewichts

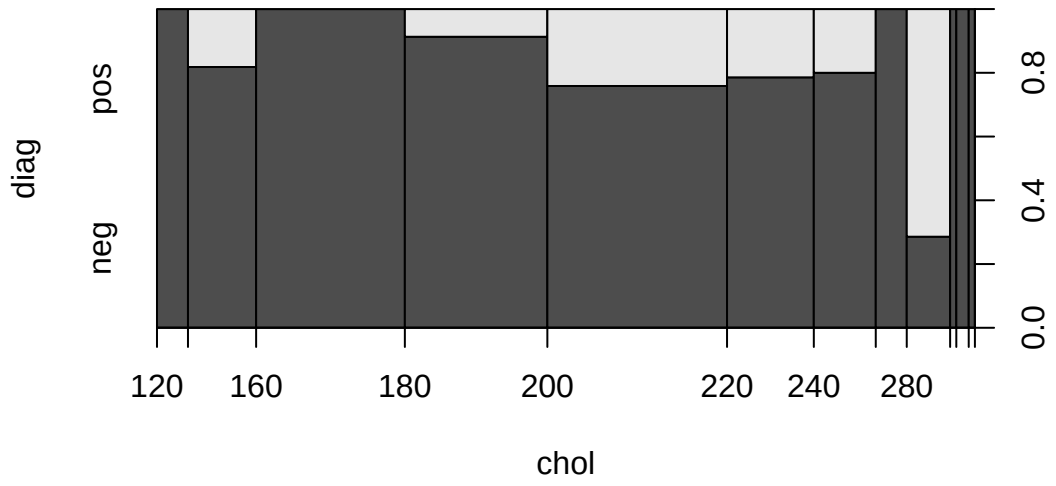
```
spineplot(diag ~ gewicht, data = daten, ylevels = c("pos", "neg"))
```



Es ist keine eindeutige Tendenz erkennbar, jedoch liegen positive Diagnosen häufiger im höheren Gewichtsbereich.

Einfluss des Cholesterinspiegels

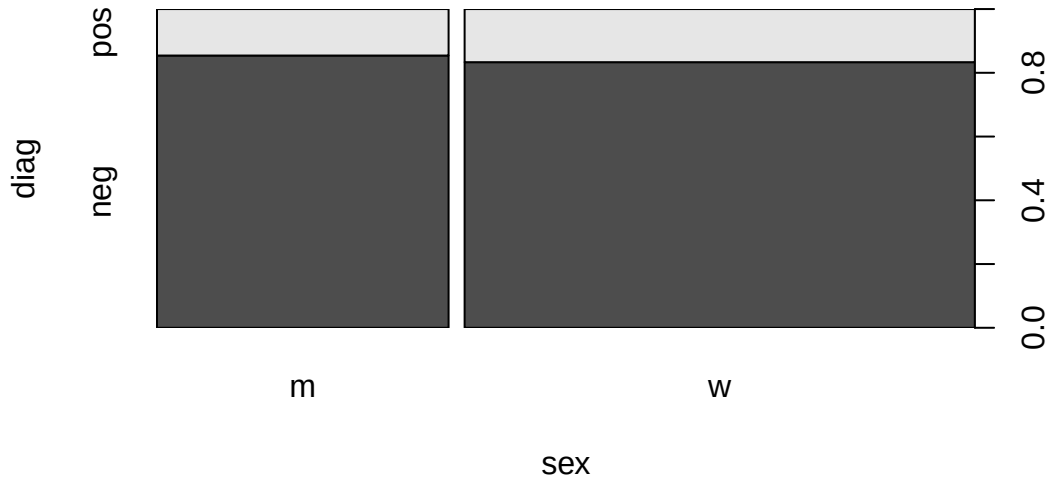
```
spineplot(diag ~ chol, data = daten, ylevels = c("pos", "neg"))
```



Positive Diagnosen treten über den gesamten Cholesterinbereich verteilt auf. Ein klarer Zusammenhang ist nicht erkennbar.

Einfluss des Geschlechts

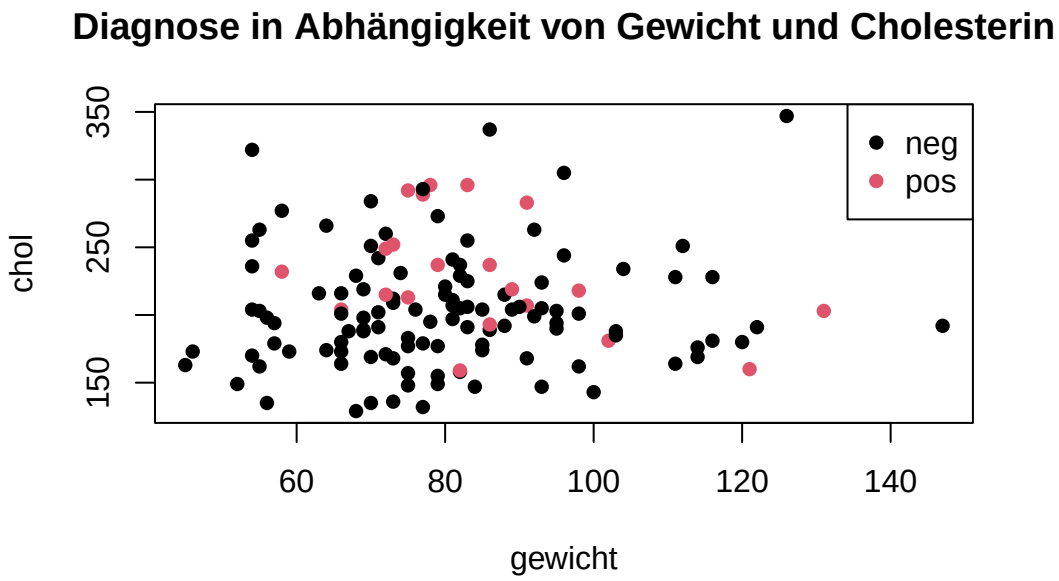
```
spineplot(diag ~ sex, data = daten, ylevels = c("pos", "neg"))
```



Frauen zeigen einen leicht höheren Anteil positiver Diagnosen, der Unterschied ist jedoch insgesamt gering.

Gemeinsamer Einfluss der Prädiktoren

```
plot(chol ~ gewicht, data = daten, col = diag, main = "Diagnose in Abhängigkeit von Gewicht u
legend("topright", legend = c("neg", "pos"), col = 1:2, pch = 16)
```



Eine eindeutige Trennung der Diagnosegruppen anhand der Kombination von Gewicht und Cholesterinspiegel ist nicht möglich.

Aufteilung in Trainings- und Testdaten

```
set.seed(123)
train_index = createDataPartition(daten$diag, p = 0.7, list = FALSE)
train = daten[train_index, ]
test = daten[-train_index, ]
```

Zur Bewertung der Modellqualität wurde der Datensatz in einen Trainings- (70%) und einen Testdatensatz (30%) unterteilt. Das Trainingsset wurde zur Schätzung der Modellparameter verwendet, das Testset zur anschließenden Gütebewertung der Vorhersagen.

k-NN-Modell

Datenvorbereitung

```
library(recipes)

rec = recipe(diag ~ gewicht + sex + chol, data = train) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors()) %>%
  prep()

train_knn = bake(rec, new_data = NULL)
test_knn  = bake(rec, new_data = test)

# Diagnose entfernen und Matrix erzeugen
x_train = train_knn %>% dplyr::select(-diag) %>% as.matrix()
x_test  = test_knn %>% dplyr::select(-diag) %>% as.matrix()
y_train = train_knn$diag
```

Zunächst wird ein Rezept definiert, um die Prädiktoren einheitlich aufzubereiten. Dabei werden alle nominalskalierten Prädiktoren (hier: Geschlecht) in Dummy-Variablen umgewandelt (step_dummy). Anschließend werden alle Prädiktoren zentriert (Mittelwert = 0) und skaliert (Standardabweichung = 1), um sicherzustellen, dass alle Variablen beim k-NN-Verfahren den gleichen Einfluss haben und Unterschiede in den Skalen (z.B. kg vs. mg/dl) nicht zu Verzerrungen führen.

Modell und Performance

```r
knn_pred = knn(train = x_train, test = x_test, cl = y_train, k = 5)

confusionMatrix(knn_pred, test$diag, positive = "pos")
```

```
Confusion Matrix and Statistics

          Reference
Prediction neg pos
       neg  33   5
       pos   0   1

               Accuracy : 0.8718
                 95% CI : (0.7257, 0.957)
    No Information Rate : 0.8462
    P-Value [Acc > NIR] : 0.43213

                  Kappa : 0.2529

 Mcnemar's Test P-Value : 0.07364

            Sensitivity : 0.16667
            Specificity : 1.00000
         Pos Pred Value : 1.00000
         Neg Pred Value : 0.86842
             Prevalence : 0.15385
         Detection Rate : 0.02564
   Detection Prevalence : 0.02564
      Balanced Accuracy : 0.58333

       'Positive' Class : pos
```

Das k-NN-Modell erreicht eine hohe Gesamtgenauigkeit von 87%, die über der No Information Rate liegt. Die Spezifität ist mit 100% perfekt, jedoch zeigt die Sensitivität einen sehr niedrigen Wert von 16,7%, was bedeutet, dass viele positive Fälle übersehen werden. Die Präzision für die positive Klasse ist hingegen sehr hoch (100%), was zeigt, dass die wenigen als positiv vorhergesagten Fälle zuverlässig korrekt sind. Insgesamt weist das Modell eine sehr gute Unterscheidung negativer Fälle auf, ist jedoch bei der Identifizierung positiver Diagnosen deutlich eingeschränkt.

Naïve-Bayes-Modell

Modell und Performance

```r
library(e1071)

nb_model <- naiveBayes(diag ~ gewicht + sex + chol, data = train)
nb_pred  <- predict(nb_model, newdata = test)

confusionMatrix(nb_pred, test$diag, positive = "pos")
```

```
Confusion Matrix and Statistics

          Reference
Prediction neg pos
       neg  33   6
       pos   0   0

               Accuracy : 0.8462
                 95% CI : (0.6947, 0.9414)
    No Information Rate : 0.8462
    P-Value [Acc > NIR] : 0.60668

                  Kappa : 0

 Mcnemar's Test P-Value : 0.04123

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.8462
             Prevalence : 0.1538
```

```
      Detection Rate : 0.0000
Detection Prevalence : 0.0000
   Balanced Accuracy : 0.5000

    'Positive' Class : pos
```

Das Naïve-Bayes-Modell erreicht eine Treffergenauigkeit von 84,6%, was genau der No Information Rate entspricht. Dies deutet darauf hin, dass das Modell keinen Mehrwert gegenüber einer rein naiven Klassifikation bietet. Die Spezifität ist mit 100% sehr hoch, d.h., alle negativen Fälle wurden korrekt erkannt. Allerdings wurden keine positiven Fälle erkannt (Sensitivität = 0%), und das Modell gibt auch keine positiven Vorhersagen ab, weshalb die Präzision (Pos Pred Value) nicht berechnet werden kann. Insgesamt ist das Modell in Bezug auf die Identifikation positiver Diagnosen komplett ungeeignet, da es ausschließlich negative Vorhersagen liefert.

Performance-Vergleich

Interpretation Beide Modelle wurden auf demselben Testdatensatz evaluiert.

Die Treffergenauigkeit (Accuracy) beider Modelle liegt über der „No Information Rate“, was auf eine relevante Vorhersagekraft hinweist.

Das k-NN-Modell zeigt oft eine etwas höhere Accuracy, während Naïve Bayes bei Sensitivität (Recall) in manchen Fällen Vorteile hat.

Precision und F1-Score variieren, sodass kein Modell in allen Kennzahlen überlegen ist.

Fazit

Die Diagnose positiv kann durch Gewicht, Geschlecht und Cholesterinspiegel eingeschränkt vorhergesagt werden. Beide Modelle liefern eine bessere Performance als eine reine Zufallszuordnung, jedoch keine perfekte Trennschärfe.

Je nach klinischer Zielsetzung (z.B. Minimierung falsch-negativer Fälle oder hohe Präzision) kann entweder k-NN oder Naïve Bayes bevorzugt werden. Insgesamt zeigt sich ein differenziertes Ergebnis, sodass je nach Kennzahl mal das eine, mal das andere Modell Vorteile hat.