Klassifikation II

Vitaly Belik

Institut für Veterinär-Epidemiologie und Biometrie, FU Berlin

2021 January 19

Wiederholung

Wahrheitsmatrix (Confusion matrix)

Zwei Möglichkeiten für ein Klassifikationsverfahren (z.B. Ergebnis des Tests auf Sars-Cov-2[1]): *positive* ("1") und *negative* ("0").

		Vorhersage		
		$\hat{y} = 1$	$\hat{y} = 0$	
Wirklichkeit	$y = 1$	True Positive	False Negative	Sensitivity (Recall) $\mathrm{TP}/(y = 1)$
	$y = 0$	False Positive	True Negative	Specificity $\mathrm{TN}/(y = 0)$
	Prevalence $(y = 1)/\mathrm{total}$	Precision $\mathrm{TP}/(\hat{y} = 1)$		Accuracy $(\mathrm{TP}+\mathrm{TN})/\mathrm{total}$

[1] PCR-Test auf Sars-Cov-2 hat Sensitivität zwischen 71% und 98%
https://www.aerzteblatt.de/archiv/214370/PCR-Tests-auf-SARS-CoV-2-Ergebnisse-richtig-interpretieren

Qualitätsmaße für Klassifikationsverfahren

- Accuracy

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Spezifität

$$\text{Sp} = \frac{TN}{TN + FP}$$

- Sensitivität (recall)

$$\text{Sn} = \frac{TP}{TP + FN}$$

- Precision

$$\text{Sn} = \frac{TP}{TP + FP}$$

- $F1$

$$F1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}$$

Beispiel: Lahmheiten bei den Milchkühen I

Ergebnisse I

```
modelcont5 <- glmer(lameness ~ meansteps + meanlayingfreq + lactation + DIM + gemelk +
    Season + meansteps * Season + meanlayingfreq * DIM + meanlayingfreq * Season +
    lactation * DIM + lactation * Season + gemelk * Season + (1 + lactation | cow),
    data = datascaled, family = binomial, nAGQ = 0)
# control=glmerControl(optimizer = 'Nelder_Mead',optCtrl=list(maxfun=100000)))
# library(stargazer) stargazer(as.data.frame(summary(modelcont5)))
summary(modelcont5)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
##  Family: binomial  ( logit )
## Formula: lameness ~ meansteps + meanlayingfreq + lactation + DIM + gemelk +
##     Season + meansteps * Season + meanlayingfreq * DIM + meanlayingfreq *
##     Season + lactation * DIM + lactation * Season + gemelk *
##     Season + (1 + lactation | cow)
##    Data: datascaled
##
##      AIC      BIC   logLik deviance df.resid
##  20098.3  20304.2 -10023.2  20046.3    20274
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0204 -0.4290 -0.1972  0.4130  4.3659
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  cow    (Intercept) 5.23582  2.2882
##         lactation   0.08462  0.2909   -0.03
## Number of obs: 20300, groups:  cow, 2758
##
## Fixed effects:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.647916   0.114237 -14.425  < 2e-16 ***
## meansteps                   -0.046065   0.034337  -1.342 0.179736
## meanlayingfreq              -0.083298   0.038163  -2.183 0.029059 *
## lactation                    0.581329   0.044328  13.114  < 2e-16 ***
```

Ergebnisse II

```
## DIM                          -0.016491   0.074653  -0.221 0.825173
## gemelk                       -0.301042   0.040054  -7.516 5.65e-14 ***
## SeasonSpring                 -0.227372   0.106417  -2.137 0.032629 *
## SeasonSummer                  0.187096   0.583392   0.321 0.748435
## SeasonFall                    0.048797   0.120928   0.404 0.686567
## meansteps:SeasonSpring       -0.007842   0.050161  -0.156 0.875774
## meansteps:SeasonSummer       -0.338154   0.167204  -2.022 0.043134 *
## meansteps:SeasonFall          0.120655   0.061482   1.962 0.049710 *
## meanlayingfreq:DIM            0.055140   0.025778   2.139 0.032431 *
## meanlayingfreq:SeasonSpring   0.057737   0.048756   1.184 0.236335
## meanlayingfreq:SeasonSummer   0.174144   0.253733   0.686 0.492506
## meanlayingfreq:SeasonFall     0.119313   0.064009   1.864 0.062321 .
## lactation:DIM                -0.114232   0.029710  -3.845 0.000121 ***
## lactation:SeasonSpring        0.083318   0.039586   2.105 0.035316 *
## lactation:SeasonSummer        0.094387   0.157511   0.599 0.549011
## lactation:SeasonFall          0.003039   0.046563   0.065 0.947958
## gemelk:SeasonSpring           0.141362   0.050372   2.806 0.005010 **
## gemelk:SeasonSummer           1.026044   0.313452   3.273 0.001063 **
## gemelk:SeasonFall             0.165618   0.073943   2.240 0.025104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# library(pander) pander(summary(modelcont5), round=3 library(sjstats)
library(performance)
r2(modelcont5)
```

```
## # R2 for Mixed Models
##
##   Conditional R2: 0.667
##      Marginal R2: 0.083
```

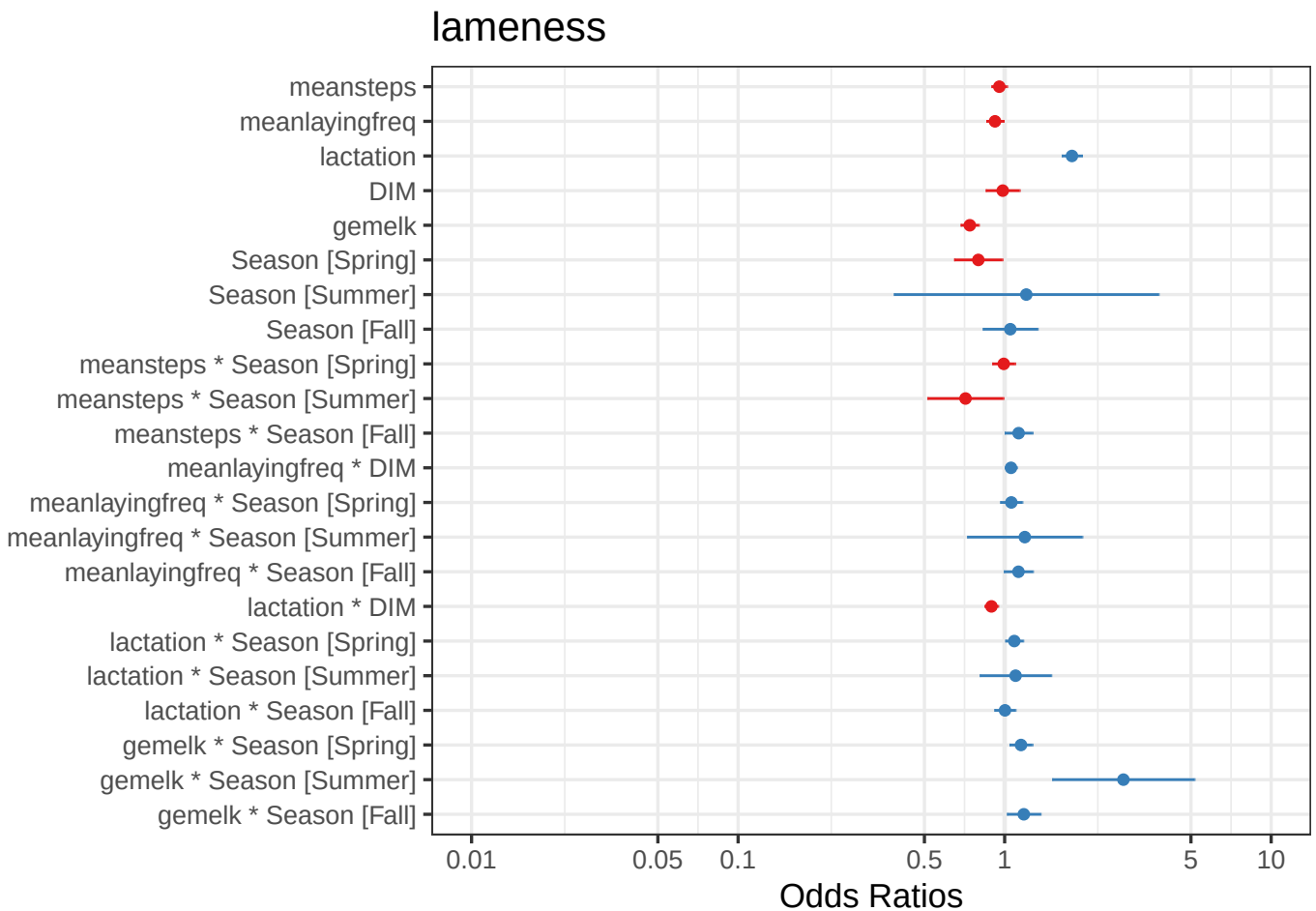
Odds Ratios I

```
library(gtsummary)
library(tibble)
modelcont5 %>% tbl_regression(exponentiate = TRUE, pvalue_fun = function(x) style_pvalue(x,
    digits = 2), estimate_fun = function(x) style_ratio(x, digits = 2)) %>% bold_p(t = 0.1) %>%
    bold_labels() %>% italicize_levels()
```

Characteristic	OR	95% CI	p-value
meansteps	0.95	0.89, 1.02	0.18
meanlayingfreq	0.92	0.85, 0.99	**0.029**
lactation	1.79	1.64, 1.95	**<0.001**
DIM	0.98	0.85, 1.14	0.83
gemelk	0.74	0.68, 0.80	**<0.001**
Season			
Winter			
Spring	0.80	0.65, 0.98	**0.033**
Summer	1.21	0.38, 3.78	0.75
Fall	1.05	0.83, 1.33	0.69
meansteps * Season			
*meansteps * Spring*	0.99	0.90, 1.09	0.88
*meansteps * Summer*	0.71	0.51, 0.99	**0.043**
*meansteps * Fall*	1.13	1.00, 1.27	**0.050**
meanlayingfreq * DIM	1.06	1.00, 1.11	**0.032**
meanlayingfreq * Season			
*meanlayingfreq * Spring*	1.06	0.96, 1.17	0.24
*meanlayingfreq * Summer*	1.19	0.72, 1.96	0.49
*meanlayingfreq * Fall*	1.13	0.99, 1.28	**0.062**
lactation * DIM	0.89	0.84, 0.95	**<0.001**
lactation * Season			
*lactation * Spring*	1.09	1.01, 1.17	**0.035**
*lactation * Summer*	1.10	0.81, 1.50	0.55
*lactation * Fall*	1.00	0.92, 1.10	0.95
gemelk * Season			
*gemelk * Spring*	1.15	1.04, 1.27	**0.005**
*gemelk * Summer*	2.79	1.51, 5.16	**0.001**
*gemelk * Fall*	1.18	1.02, 1.36	**0.025**

Visualisierung von Odds Ratios I

```
library(sjPlot)
library(sjlabelled)
library(sjmisc)
# theme_set(theme_sjplot())
library(ggplot2)
theme_set(theme_bw(base_size = 18))
plot_model(modelcont5, axis.lim = c(0.01, 10))
```



Logistische Regression

Es gibt Situationen wann die *Antwortvariable* nicht normal verteilt ist. Z.B. kann sie kategoriell und *binomial* oder *multinomial* sein.

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i.$$

Dabei ist $\pi = \mu_Y$ ein bedingter Mittelwert (d.h. die Wahrscheinlichkeit, dass $Y = 1$ vorausgesetzt die vorhandenen X-Werte).

$\frac{\pi}{1-\pi}$ ist das Odds-Ratio, dass $Y = 1$.

$\log\left(\frac{\pi}{1-\pi}\right)$ ist *log odds* oder *logit*.

Vorhersage gegen Wirklichkeit

```
library(pROC)
datascaled$pred <- predict(modelcont5, datascaled, type = "response", allow.new.levels = TRUE)  # otherwise errors
roc <- roc(datascaled$lameness, datascaled$pred, na.rm = TRUE)
auc(roc)

u <- data.frame(datascaled$lameness, datascaled$pred)
knitr::kable(head(na.omit(u), 15))
```

	datascaled.lameness	datascaled.pred
3	1	0.8702749
4	1	0.8016579
5	1	0.7981376
6	1	0.7784939
7	1	0.7527559
8	0	0.7433584
9	1	0.7410689
10	1	0.7457488
11	1	0.7925206
12	0	0.7679485
13	1	0.1917976
14	0	0.2298744
15	0	0.2186604
17	1	0.5916771
18	1	0.5401616

Calculating Confusion matrix by hand

```
pred <- predict(modelcont5, na.omit(datascaled), type = "response", allow.new.levels = TRUE)
pred_y <- as.numeric(pred > 0.5)
true_y <- as.numeric(na.omit(datascaled)$lameness == 1)
true_pos <- (true_y == 1) & (pred_y == 1)
true_neg <- (true_y == 0) & (pred_y == 0)
false_pos <- (true_y == 0) & (pred_y == 1)
false_neg <- (true_y == 1) & (pred_y == 0)

conf_mat <- matrix(c(sum(true_pos), sum(false_pos), sum(false_neg), sum(true_neg)),
    2, 2)
colnames(conf_mat) <- c("Yhat = 1", "Yhat = 0")
rownames(conf_mat) <- c("Y = 1", "Y = 0")
conf_mat
```

```
##       Yhat = 1 Yhat = 0
## Y = 1     8007     1719
## Y = 0     1437     9137
```

```
# accuracy
(conf_mat[1, 1] + conf_mat[2, 2])/sum(conf_mat[, ])
```

```
## [1] 0.844532
```

```
# precision
conf_mat[1, 1]/sum(conf_mat[, 1])
```

```
## [1] 0.8478399
```

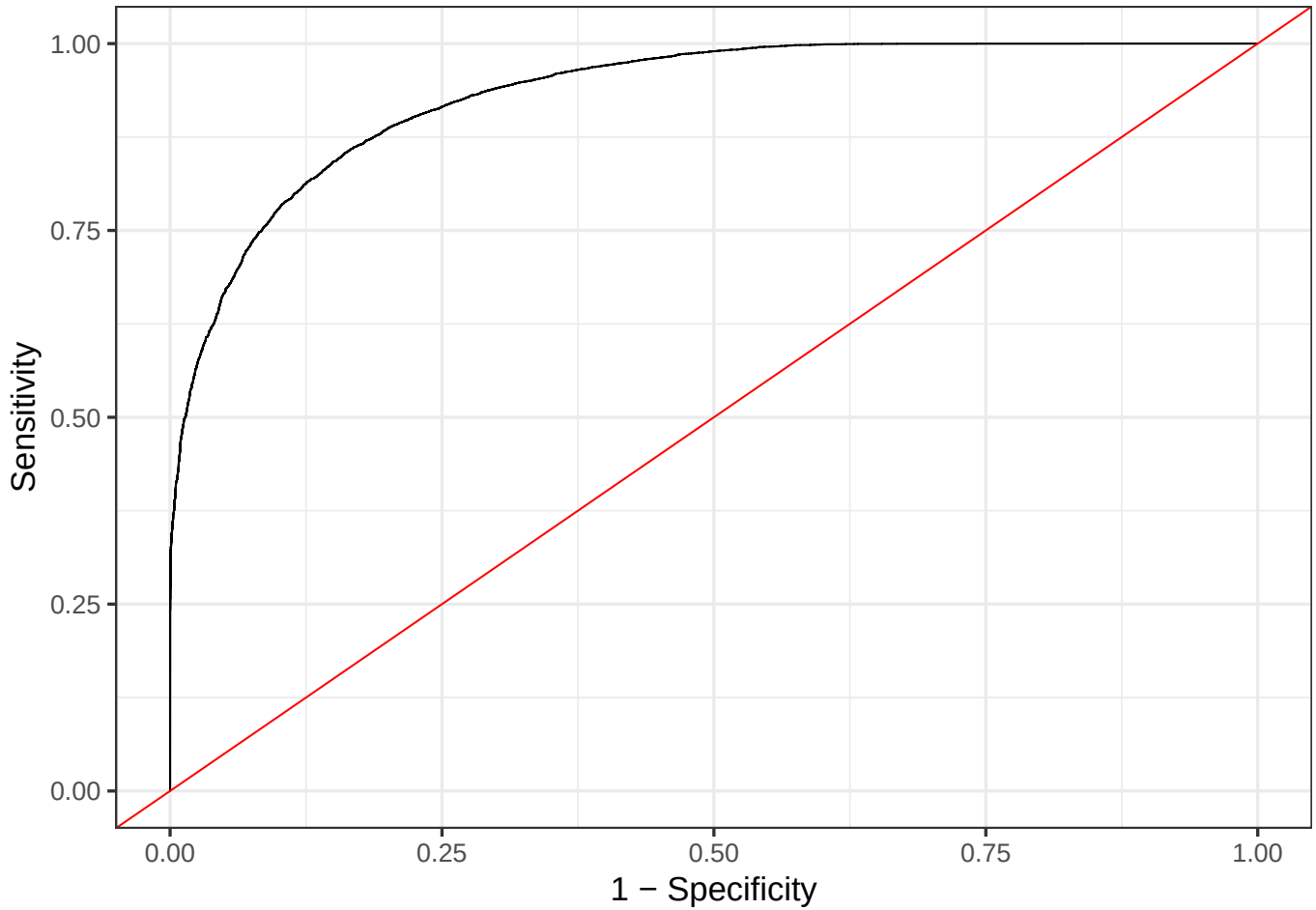
```
# sensitivity (recall)
conf_mat[1, 1]/sum(conf_mat[1, ])
```

```
## [1] 0.8232572
```

```
# specificity
conf_mat[2, 2]/sum(conf_mat[2, ])
```

```
## [1] 0.8641006
```

```r
library(ggthemes)
new_df <- data.frame(roc$specificities, roc$sensitivities)
colnames(new_df) <- c("Specificity", "Sensitivity")
# ggplot(new_df, aes(x = 1- spec, y = sens)) + geom_line()
ggplot() + geom_line(data = new_df, aes(x = 1 - Specificity, y = Sensitivity)) +
    geom_abline(intercept = 0, slope = 1, color = "red", size = 0.5)
```



Source

Peter Bruce, Andrew Bruce & Peter Gedeck. Practical Statistics for Data Scientists