8 Prädiktive Modellierung

8.1 Mathematische Grundlagen

Gegeben sei ein *binärer Klassifikationsdatensatz* mit einer Menge von *Trainingsdatenpunkten*

$$\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\} \ , \tag{1}$$

wobei $x_i \in \mathbb{R}^m$, $i = 1, ..., n$ ein m-dimensionaler *Featurevektor* sei und $y_i \in \{0, 1\}$, $i = 1, ..., n$ den Wert einer skalaren *Labelvariable* darstelle, wobei der Wert 1 für "positive Fälle" und der Wert 0 für "negative Fälle" stehe.

Die Bezeichnungen "positiv" und "negativ" sind aber lediglich formal zu verstehen. Beispiele für vorherzusagende Labelvariablen (1/0) etwa wären "erkrankt/gesund", "erzielt Therapieerfolg/bleibt erkrankt" oder "ältere/jüngere Person".

Ziel der *prädiktiven Modellierung* ist die Kalibrierung einer Funktion $f(x) : \mathbb{R}^m \to \{0, 1\}$, die prädizierte Labels $f(x_i) = \hat{y}_i$ ergibt, wenn man Featurevektoren einsetzt. Mit der Methode der *leave-one-out cross-validation* (LOO-CV) wird ein prädiktives Modell wiederholt an einem Datensatz trainiert und getestet, wobei in jeder Wiederholung der Trainingsdatensatz aus allen experimentellen Einheiten bis auf eine besteht und die ausgelassene experimentelle Einheit den Testdatensatz bildet.

Liegen für einen Trainingsdatensatz wahre Labels y_i und prädizierte Labels $\hat{y}_i = f(x_i)$ vor, kann die Klassifikationsperformanz des prädiktiven Modells evaluiert werden. Dazu werden die Einträge der binären Konfusionsmatrix bestimmt

$$\begin{aligned} \text{TN} &= |\{(x_i, y_i) \mid y_i = 0 \wedge f(x_i) = 0\}| \\ \text{FP} &= |\{(x_i, y_i) \mid y_i = 0 \wedge f(x_i) = 1\}| \\ \text{FN} &= |\{(x_i, y_i) \mid y_i = 1 \wedge f(x_i) = 0\}| \\ \text{TP} &= |\{(x_i, y_i) \mid y_i = 1 \wedge f(x_i) = 1\}| \ , \end{aligned} \tag{2}$$

wobei $|\{\cdot\}|$ die Anzahl der Elemente der entsprechenden Menge bezeichne (z.B. der Menge der tatsächlich negativen Fälle, $y_i = 0$, die als positiv klassifiziert wurden, $f(x_i) = 1$; false positives, FP).

Folgende Klassifikationsmetriken, die in der Vorlesung detailliert besprochen wurden, können aus den Einträgen der Konfusionsmatrix berechnet werden:

- true positive rate (TPR), false positive rate (FPR), true negative rate (TNR) und false negative rate (FNR)
- positive predictive value (PPV), false discovery rate (FDR), negative predictive value (NPV) und false omission rate (FOR)
- accuracy (ACC), balanced accuracy (BAC), Precision, Recall, F1-Score (F1)

8.2 Analyse in R

Die Datei `FADE_SAME.csv` enthält den in der ersten Seminarsitzung vorgestellten Datensatz. Erklären Sie die Funktion des folgenden R-Codes:

```
# Daten einlesen
fname = 'FADE_SAME.csv'                          # Dateiname
D     = read.csv(fname)                          # Dataframe

# Datenmatrix extrahieren
rows  = startsWith(D$subject, 'subA')            # Personen aus Studie A
cols  = c('novelty.SAME', 'memory.SAME')         # Definition Variablen
X     = t(as.matrix(D[rows,cols]))               # Datenmatrix
y     = t(as.matrix(D$age[rows]))                # gruppendefinierende Variable
y     = 0*(y <= 35) + 1*(y > 35)                 # Labelvariable
print(dim(X))                                    # Überprüfung Datenmatrix
print(dim(y))                                    # Überprüfung Labelvariable
```

8.3 Erste Programmieraufgabe

Visualisieren Sie die in der Datenmatrix enthaltenen Variablen getrennt nach, d.h. unter Berücksichtigung der Werte der Labelvariable. Gehen Sie dazu wie folgt vor:

- Orientieren Sie sich für diese Aufgabe am Abschnitt 1.4 des Arbeitsblatts (1) *Multiple Regression*, weichen Sie jedoch wie folgt vom dortigen Vorgehen ab:
- Die Variable auf der x-Achse soll die erste Zeile von X, die Variable auf der y-Achse soll die zweite Zeile von X sein.
- Die Variablen sollen getrennt für junge Probanden (indiziert durch `y==0`) und ältere Probanden (indiziert durch `y==1`) dargestellt werden.
- Es sollen keine Regressionsgeraden dargestellt werden.
- Beide Achsen sollen jeweils von -3 bis $+3$ gehen.

- Achsenbeschriftungen und Legende sollen den dargestellten Variablen angepasst werden.
- Erzeugen und speichern Sie die Abbildung. Mutmaßen Sie auf Grundlage des dargestellten Trainingsdatensatzes, inwieweit die Variable "Altersgruppe" sich aus den Variablen "Novelty-SAME" und "Memory-SAME" klassifizieren lässt. Sie sollten in etwa folgende Abbildung erhalten:

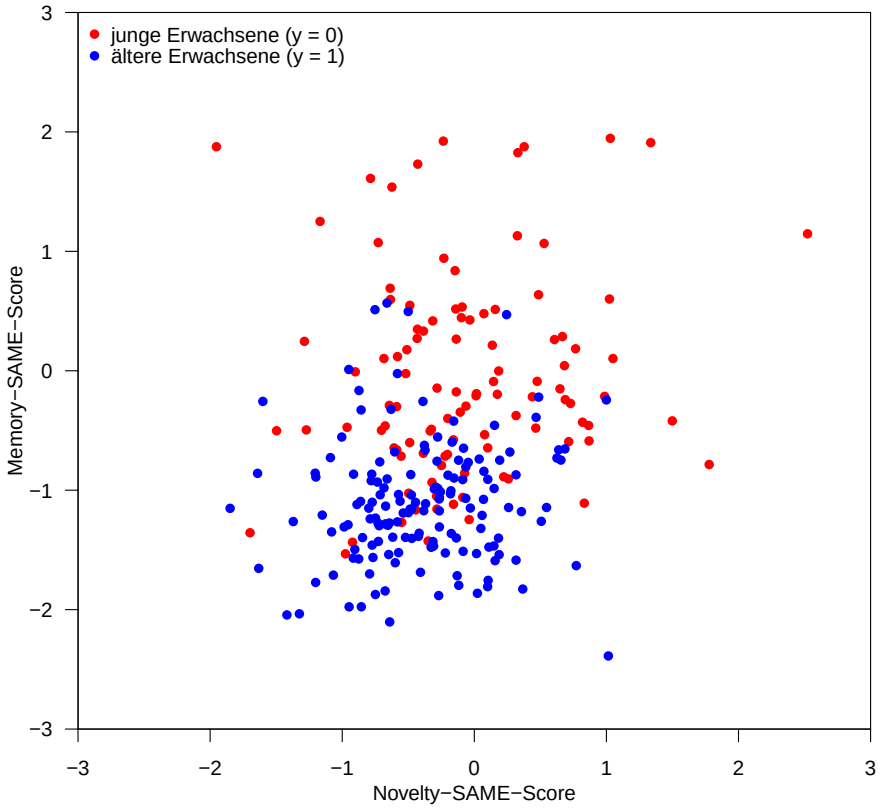

Abbildung 1. Novelty-SAME-Score und Memory-SAME-Score, getrennt nach Altersgruppe.

8.4 Analyse in R

Im zweiten Teil wollen wir nun eine prädiktives Modell auf Grundlage der Datenmatrix X und der Labelvariable y schätzen. Erklären Sie dazu zunächst den folgenden R-Code und die Analyse, die er durchführt:

```
# Vorbereitung Datenanalyse
m       = nrow(X)                                    # Anzahl Features
n       = length(y)                                  # Anzahl Datenpunkte
L       = c(0,1)                                     # Klassenlabels
y_pred  = matrix(rep(NaN,n*2), nrow = n)             # wahre und prädizierte Label

# leave-one-out cross-validation
for (i in 1:n) {                                     # Iteration über Datenpunkte

    # Datensatz partitionieren
    x_train     = as.matrix(X[,-i])                  # i-ter Featurevektor nicht im Trainingsdatendatz
    y_train     = as.matrix(y[,-i])                  # i-tes Label nicht im Trainingsdatendatz
    x_test      = as.matrix(X[, i])                  # i-ter Featurevektor als Testdatenpunkt
    y_test      = as.matrix(y[, i])                  # i-ites Label als Testdatenpunkt
    y_pred[i,1] = y_test                             # wahres Label des i-ten Datenpunkts

    # "Training": klassenspezfische Mahalanobisdistanz berechnen
    D           = matrix(rep(NaN,2), ncol = 2)       # Vektor klassenspezifischer Mahalanobisdistanzen
    for (l in L) {                                   # Iteration über Klassen
        Xl      = x_train[,y_train == l]             # x^{(i)} für Label l
        nl      = ncol(Xl)                           # Anzahl Realisierungen
        I_nl    = diag(nl)                           # Einheitsmatrix I_n
        J_nl    = matrix(rep(1,nl^2), nrow = nl)     # 1_{nn}
        x_bar   = (1/nl)* Xl %*% J_nl[,1]            # Stichprobenmittel
        C       = (1/(nl-1)) *                       # Stichprobenkovarianzmatrix
                  (Xl %*% (I_nl-(1/nl)*J_nl) %*% t(Xl))
        D[l+1]  = t(x_test - x_bar) %*% solve(C) %*%# Mahalanobisdistanz
                    (x_test - x_bar)
    }

    # "Test": Label des Testdatenpunkts prädizieren
    if (D[1] <= D[2]) {y_pred[i,2] = 0}              # prädiziertes Label
    else              {y_pred[i,2] = 1}
}
```

8.5 Zweite Programmieraufgabe

Evaluieren Sie die Klassifikationsperformanz, indem Sie die sich aus den wahren und prädizierten Labels ergebenden Klassifikationsmetriken berechnen. Gehen Sie dazu wie folgt vor:

- Berechnen Sie die Einträge der 2×2 Konfusionsmatrix, indem Sie diejenigen Fälle summieren, in denen die prädizierten Labels in der zweiten Spalte von `y_pred` den Wert 0 oder 1 haben, unter der Bedingung, dass die wahren Labels in der ersten Spalte von `y_pred` den Wert 0 oder 1 haben. Die Anzahl der true negatives (TN) beispielsweise ergibt sich als `sum(y_pred[y_pred[,1] == 0, 2] == 0)`. Berechnen Sie auf dieselbe Art und Weise die Anzahl der false positives (FP), false negatives (FN) und true positives (TP).
- Berechnen Sie mithilfe der in der Vorlesung gegebenen Formeln die Klassifikationsmetriken true positive rate (TPR), false positive rate (FPR), true negative rate (TNR) und false negative rate (FNR).
- Berechnen Sie mithilfe der in der Vorlesung gegebenen Formeln die Klassifikationsmetriken positive predictive value (PPV), false discovery rate (FDR), negative predictive value (NPV) und false omission rate (FOR).
- Berechnen Sie mithilfe der in der Vorlesung gegebenen Formeln die Klassifikationsmetriken accuracy (ACC), balanced accuracy (BAC) und F1-Score (F1).
- Geben Sie die Resultate ihrer Analyse aus. Sie sollten folgende Ergebnisse erhalten:

```
true positive rate (sensitivity) :  0.7451
false positive rate              :  0.1792
true negative rate (specificity) :  0.8208
false negative rate              :  0.2549
positive predictive value        :  0.8571
false discovery rate             :  0.1429
negative predictive value        :  0.6905
false omission rate              :  0.3095
Accuracy                         :  0.7761
Balanced Accuracy                :  0.7829
F1 Score                         :  0.7972
```

8.6 Lückentext

Füllen Sie mit den in der Übung gewonnenen Erkenntnissen den folgenden Lückentext aus und präsentieren Sie die Ergebnisse im Seminar:

Lückentext: Ein binärer Klassifikationsdatensatz besteht aus einer Menge von Trainingsdatenpunkten, wobei jeder Trainingsdatenpunkt aus einem _____ x_i und einem _____ y_i besteht. Eine beliebte Methode zur Evaluation eines prädiktiven Modells am Trainingsdatensatz ist _____. Vorhergesagte Werte von y_i bezeichnet man als _____. Der vorliegende Trainingsdatensatz besteht aus _____ Datenpunkten, wobei jeder Featurevektor _____ Einträge hat. Nach *Nearest-Neighbor-Klassifikation* der Variable _____ aus den Variablen _____ und _____ mittels *leave-one-out cross-validation* ergibt sich eine true positive rate (TPR, "Sensitivität") von _____, eine true negative rate (TNR, "Spezifizität") von _____ und eine accuracy (ACC, "Genauigkeit") von _____.

8.7 Mögliche Klausurfrage

Präsentieren Sie im Seminar folgende Klausurfrage und erklären Sie die richtige Antwort:

Frage: Aus einer binären Klassifikation ergeben sich TN = 57, FP = 25, FN = 43, TP = 75 (TN = true negative, FP = false positive, FN = false negative, TP = true positive). Wie hoch ist der positive predictive value (PPV)?

a) 0.57

b) 0.25

c) 0.43

d) 0.75

8.8 Kinderwitz

Was mögen Autos am liebsten?

Antwort: Parkplätzchen!