Allgemeines

Die vorgesehene Bearbeitungszeit für das Übungsblatt endet am Freitag, 18.12.2020, um 12:00 Uhr. Kurze Zeit später werden die Lösung und das nächste Übungsblatt hochgeladen.

Falls Sie anonymes Peer-Feedback geben und erhalten wollen, laden Sie Ihre Lösung zu den Programmieraufgaben innerhalb der Frist in das ILIAS-Übungsobjekt "Abgaben" hoch. Feedback kann erst nach Ablauf der Abgabefrist gegeben werden. Geben Sie das Feedback bis zum Mittwoch, 23.12.2020, 23:55 Uhr. Erst nach Ablauf dieser Feedbackfrist wird dann das erhaltene Feedback sichtbar. Weitere Details zum Peer-Feedback finden Sie in der Datei "Feedback_geben.pdf" im ILIAS.

1 Online-Test 3

Der dritte Online-Test steht ab dem 07.12.2020, 00:00 Uhr im ILIAS bereit. Er beschäftigt sich mit den Vorlesungskapiteln 4 (Klassifikation mit Entscheidungsbäumen) und 5 (Evaluation von Datenanalyseverfahren).

2 Entscheidungsbäume

Ziel dieser Aufgabe ist es, einen Entscheidungsbaum zur Klassifikation einzusetzen und die Ergebnisse auszuwerten. Hierzu verwenden wir den bereits bekannten `iris`-Datensatz. Die Art der Blume, gespeichert in der Spalte `Species`, ist unsere Zielvariable.

a) **[Training-Test-Partitionierung]** Teilen Sie die Datenobjekte in 70%-Trainingsdaten und 30% Testdaten auf. Hierbei könnte die Funktion `sample()` helfen. Warum machen wir überhaupt so eine Aufteilung? *(Basis)*

b) **[Training]** Nutzen Sie die Funktion `rpart()` aus dem Paket `rpart`, um einen Entscheidungsbaum zu trainieren. Diese Funktion hat einige Parameter, die den Trainingsprozess steuern. Für den Anfang sollten aber die voreingestellten Werte ausreichen. Sie müssen also nur die Trainingsdaten und eine Formel, die den Zusammenhang zwischen Zielvariable und Features beschreibt, übergeben. *(Basis)*

c) **[Modell]** Geben Sie den Entscheidungsbaum auf die Konsole aus. Erstellen Sie einen Plot, z.B. mit dem Paket `rpart.plot`. Wie hängt der Entscheidungsbaum mit den Plots des ersten Übungsblatts und den Ergebnissen der Entropie-Aufgabe des zweiten Übungsblatts zusammen? *(Basis)*

d) **[Evaluation]** Nutzen Sie die Funktion `predict()`, um Vorhersagen auf Trainings- und Testdaten zu machen. Berechnen Sie die Erfolgsquote (Accuracy) der Vorhersagen. Wie bewerten Sie die Ergebnisse? *(Basis)*

e) **[Alternative Pakete]** Verwenden Sie ein anderes Paket, welches Entscheidungsbäume enthält. Es eignen sich beispielsweise `party` (Funktion `ctree()`) und `C50`. Wie unterscheidet sich der notwendige Code für Training, Vorhersagen und Plots? Unterscheidet sich der gelernte Entscheidungsbaum, und wenn ja, warum? *(Vertiefung)*

3 Evaluation von Klassifikationsmodellen

Ziel dieser Aufgabe ist es, die Vorhersagequalität eines Entscheidungsbaums auf verschiedene Arten auszuwerten. Wir verwenden dazu weiterhin `rpart`-Entscheidungsbäume. Außerdem nutzen wir eine Abwandlung des `iris`-Datensatzes mit binärer Zielvariable. Führen Sie dazu folgenden Code aus:

```
dataset <- iris[, colnames(iris) != "Species"]
dataset$virginica <- factor(c("no", "yes")[(iris$Species=="virginica") + 1])
```

a) **[Baselines (1)]** Bestimmen Sie die Erfolgsquote, wenn einfach konstant die häufigere Klasse vorhergesagt wird. Welche Aussage ermöglicht uns dies über den Einsatz weiterer Vorhersagemodelle? *(Basis)*

b) **[Baselines (2)]** Setzen Sie als nächstes ein One-Rules-Modell mittels des Paketes `OneR` ein und bestimmen Sie dessen Erfolgsquote. Wie geht dieses Modell mit numerischen Daten um? Inwieweit ist das Modell für den gegebenen Datensatz geeignet? *(Basis)*

c) **[Holdout-Validierung]** Werten sie die Erfolgsquote für verschiedene stratifizierte Training-Test-Partitionierungen aus, z.B. von 90%:10% bis 10%:90%. Wie erklären Sie sich die Ergebnisse? *(Basis)*

d) **[Kreuzvalidierung]** Implementieren Sie eine k-fache Kreuzvalidierung und betrachten Sie die Ergebnisse für verschiedene Werte von k. Insbesondere ist interessant, wie sich der Mittelwert und die Standardabweichung der Erfolgsquote über die Partitionen (Folds) verhalten, wenn Sie k variieren. Wie erklären Sie sich die Ergebnisse? Welchen Wert von k würden Sie empfehlen? *(Vertiefung)*

e) **[Regularisierung]** Versuchen Sie, Entscheidungsbäume verschiedener Komplexität zu lernen, indem Sie beim Aufruf von `rpart()` beispielsweise die Mindestzahl an Datenobjekten pro Blattknoten kontrollieren. Wie wirkt sich dies auf die Erfolgsquote aus? *(Vertiefung)*

f) **[ROC-Kurve]** Sagen Sie Wahrscheinlichkeiten anstelle von Klassenlabels voraus und plotten Sie die ROC-Kurve, beispielsweise mit dem Paket `ROSE` oder `ROCR`. Welche Erkenntnisse können Sie aus dem Plot ziehen? Berechnen Sie wahlweise die Punkte, an denen die ROC-Kurve einen Knick hat, und heben Sie diese auf dem Plot hervor. Woraus ergeben sich diese Punkte? Berechnen Sie abschließend die Fläche unter der Kurve (=AUC). *(Vertiefung)*