# Arbeitsblatt 5

## Aufgabe 1: Bestimmung des Geschlechtes von Krabben mit einem Klassifikationsbaum

Wir betrachten den `crabs` Datensatz, der im Paket `MASS` enthalten ist. Er enthält morphologische Informationen zu Krabben. Bei dieser Spezies von Krabben (blaue oder orange Leptograpsus variegatus) ist das Geschlecht nicht leicht ersichtlich. Hier wollen wir ein Klassifikationsmodell entwickeln, mit dem das Geschlecht einer Krabbe mithilfe verschiedener Variablen, die den Körperbau der Krabbe beschreiben, bestimmt werden kann.

a) Machen Sie sich mit dem Datensatz `crabs` vertraut (`library(MASS)` und `?crabs`). Erstellen Sie einen Klassifikationsbaum, der das Geschlecht der Krabbe anhand der vermessenen Features vorhersagt (`r.tree <- train(sex ~., data=crabs, method='rpart', tuneGrid=data.frame(cp=0.01))`). Das finale Modell ist dabei im Element `r.tree$finalModel` gespeichert.

   Stellen Sie den Klassifikationsbaum grafisch dar. Wenden Sie dazu die `plot` Funktion auf das finale Modell an (`plot(r.tree$finalModel, margin=0.2, uniform=TRUE)`). Hilfe zu dieser Plot-Funktion erhalten Sie unter (`?plot.rpart`). Um die Knoten zu beschriften müssen Sie die Abbildung noch mit Text ergänzen (`text(r.tree$finalModel, use.n=TRUE)`). Durch das Argument `use.n=TRUE` wird die zusätzlich die Verteilung der Klassen in den Endknoten abgebildet.

   Optional: Laden Sie dann die Pakete `rattle` und plotten Sie den Entscheidungsbaum mit `fancyRpartPlot(r.tree$finalModel)`. Haben beide Darstellungen den gleichen Informationsgehalt?

b) Benutzen Sie die Funktion `predict` um mit dem Baummodell das Geschlecht der Krabben vorherzusagen, welche Sie auch zum Trainieren des Baums benutzt haben (`predict(r.tree, data =crabs)`). Evaluieren Sie die "in-sample" Performance der Geschlechter-Klassifikation durch die Angabe und Interpretation der Konfusionsmatrix.

c) Im Klassifikationsbaum wurden als Knoten nur die Variablen RW und CL verwendet (z.B. direkt in `r.tree$finalModel` zu sehen). Erstellen Sie ein Streudiagramm mit diesen beiden Variablen und färben Sie die Punkte gemäss dem Geschlecht der Krabben ein. Zeichnen Sie die erste Auftrennung im Klassifikationsbaum als Linie im Streudiagramm (`abline(v=...)`). Welcher Nachteil von Klassifikationsbäumen wird hier ersichtlich?

d) Führen Sie mit den numerischen Variablen des Crabs-Datensatzes eine PCA durch. Würden Sie die Daten standardisieren oder nicht? Visualisieren Sie die Position der Datenpunkte im Streudiagramm der ersten beiden Hauptkomponenten. Färben Sie die Punkte gemäss dem Geschlecht der Krabben ein.

e) Fügen Sie die Hauptkomponenten als weitere Variablen an den Datensatz Crabs an (`cbind(crabs,pca$x)`). Trainieren Sie mit dem erweiterten Datensatz wieder einen Klassifikationsbaum und stellen Sie ihn dar. Welche Variablen wurden als Knoten verwendet? Stellen Sie die Variablen für die ersten beiden Knoten in einem Streudiagramm dar und zeichnen Sie die ersten Auftrennungen ein. Quantifizieren Sie die Performance auf dem Trainingsdatensatz durch eine Konfusionsmatrix. Hat sich die Performance durch das Hinzufügen der Hauptkomponenten als Variablen geändert?

## Aufgabe 2: Churn bei einem Telefonanbieter (Klassifikationsbäume und Random Forest)

In dieser Aufgabe arbeiten wir nochmals mit den Churn-Daten aus der Aufgabe 2 Arbeitsblatt 4.

a) Laden Sie die Daten. Verwenden Sie die ersten 80% der Daten zum Trainieren und die zweiten 20% zum Testen. Wie gross ist die Performance eines Klassifikationsbaums auf den Trainings- und Testdaten? Vergleichen Sie die Ergebnisse mit dem k = 1 NN Klassifizierer. Die Fehlerrate eines k = 1 NN Klassifizierers war 0 auf den Trainingdaten und etwa 20% auf den Testdaten. Im Unterschied zum nächste Nachbarn-Klassifizier müssen Sie hier die Faktorvariablen nicht ausschliessen.

b) Wiederholen Sie a) nun mit einem Random Forest und vergleichen Sie die Ergebnisse. Verwenden Sie dazu die Methode `rf` in der Funktion `train`.

c) Vergleichen Sie nun den out-of-bag error des Trainingsets mit dem Fehler auf dem Testset (`rf_fit$finalModel`).

d) Welche Variablen sind für die Vorhersagen wichtig. Visualisieren Sie die "Wichtigkeit" der Variablen mit der Funktion `varImp()`.

e) Können Sie Ihre Vorhersage mit einem stratifizierten Sampling noch verbessern?

## Aufgabe 3: Random Forest für Klassifikation mit 9 Klassen

Im Jahr 2015 hat die Otto Group einen Data Science Wettbewerb auf Kaggle gestartet. Ziel des Kaggle-Wettbewerbs mit dem Namen "Otto Group Product Classification Challenge" war es, einen Algorithmus zu entwickeln, der Produkte weltweit anhand bestimmter Merkmale automatisch und zuverlässig Produktkategorien zuordnet.

"*Um Analysen zum weltweiten Verkauf einzelner Artikel vorzunehmen, müssen die Produkte einheitlichen Kategorien zugeordnet werden. Eine manuelle Zuordnung aller Produkte ist aber nicht nur zeitaufwendig, sondern auch sehr fehleranfällig*", erläutert Josef Feigl, Data Scientist bei der Otto Group. Für den Wettbewerb wurden mehr als 200'000 Produkte mit unterschiedlichen Merkmalen zusammengestellt. Im Datensatz `otto.csv` finden Sie einen Ausschnitt aus den Daten.

a) Laden Sie das CSV `otto.rdata` und verschaffen Sie sich einen Überblick insbesondere über die Zielvariable `target` an. Mit welcher Häufigkeit treten die einzelnen Klassen auf?

b) Bei der Aufteilung der Daten in Trainings- und Testdaten wollen wir, dass sich die Verteilung der Zielvariable in beiden Datensätzen widerspiegelt. Die Funktion `createDataPartition` sorgt automatisch für ein balanciertes Sampling.

c) Trainieren Sie einen Random Forest mit der Zielvariable `target` und allen Features. Je nach Anzahl Bäumen und Tuning Grid kann die Simulation etwas länger dauern. Wie gut performt die Methode auf einem Testdatensatz. Schauen Sie sich an, wie gut die einzelnen Klassen klassifiziert werden. Was fällt Ihnen dabei auf?

d) Können Sie durch Stratifizierung die Klassifikation der selteneren vertretenen Klassen verbessern?