Hochschule Bochum
Bochum University
of Applied Sciences

Maschinelles Lernen
Prof. Dr. Jörg Frochte
Ana Martínez
Campus Velbert/Heiligenhaus des FB E

Praktikum 1: CART, Random Forest und Bayes für die Klassifikation dreier Weinsorten

Bei diesem Praktikum, **CART, Random Forest und Bayes für die Klassifikation dreier Weinsorten** wird der Datensatz **Wine recognition data** aus dem Machine Learning Repository benutzt. Unter diesem Link finden Sie weitere Informationen zur Quelle, sowie den Originaldatensatz:

`https://archive.ics.uci.edu/ml/datasets/wine`.

Dieser Datensatz enthält die Ergebnisse chemischer Analysen verschiedener Weinsorten (von drei unterschiedlichen Kulturvarianten) aus der gleichen Region in Italien. Die Analyse besteht aus 13 Merkmalen, die in 178 protokollierten Datensätzen gefunden worden sind.

Nr.	Merkmal	Wertebereich
0	Weinsorte	$\{1, 2, 3\}$
1	Alcohol	[11, 15]
2	Malic acid	[0, 6]
3	Ash	[1, 4]
4	Alcalinity of ash	[10, 30]
5	Magnesium	[70, 162]
6	Total phenols	[0, 4]
7	Flavanoids	[0, 6]
8	Nonflavanoid phenols	[0, 1]
9	Proanthocyanins	[0, 4]
10	Color intensity	[1, 13]
11	Hue	[0, 2]
12	OD280/OD315 of diluted wines	[1, 4]
13	Proline	[278, 1680]

Die möglichen Zielwerte der Klassifikation werden mit den Zahlen 1, 2 und 3 kodiert.

Ziel: Unser Ziel ist es nun, zu lernen, zu welcher dieser Sorten ein bestimmter Wein gehört.

Nun stellt sich die Frage nach dem Trainings- und Testset. Den Datensatz haben wir für Sie vorbereitet und in zwei Dateien gesplittet. Sie bekommen in der Datei **Trainingsset.csv** 128 Daten, um den CART Algorithmus zu trainieren und in der Datei **Testset.csv** die 50 Restdaten, um die Genauigkeit Ihres Algorithmus zu testen, sowie den kompletten Datensatz **AllData.csv**. Wichtig ist, dass Sie mit den gegebenen Trainings- und Test-csv-Dateien arbeiten.

Aufgabenstellung:

1. Nutzen Sie für eine Klassifikation den CART-Algorithmus, um eine entsprechende Vorhersage des Modells zu bekommen. Nennen Sie diese Datei ***Aufgabe1.py***

   - Stellen Sie zunächst die Parameter: *xDecimals=5* und *threshold=0.1* ein. Optimieren Sie mittels des Parameters *minLeafNodeSize*.

   - Lassen Sie sich die Anzahl der Fehler Ihrer Konfiguration in der *Ipython console* anzeigen.
     Als Beispiel: *print('Fehler %e' % Fehler)*

   - Plotten Sie vier Scatterplotts aus dem ganzen Datensatz **AllData.csv** (wie im Beispiel auf Seite 72 aus dem Buch *Maschinelles Lernen* von Prof. Frochte) mit den Merkmalen:

     – Non flavanoid phenols / Proline (Subplot1)

     – Flavanoids / Color Intensity (Subplot2)

     – Alcohol / Flavanoids (Subplot3)

     – Alcohol / Color Intensity (Subplot4)

     und entscheiden Sie, welches dieser Merkmale eine bessere Eingruppierung ergibt.

   - Erzeugen Sie eine PDF-Datei mit den 4 Plots, in der Sie aus Ihrer Sicht bewerten, welche die besten zwei Merkmale für die Eingruppierung sind. Notieren Sie zusätzlich, welchen Wert Sie dem Parameter *minLeafNodeSize* übergeben haben.

2. CART mit 2 Merkmalen: Benennen Sie Ihre Datei ***Aufgabe2.py***

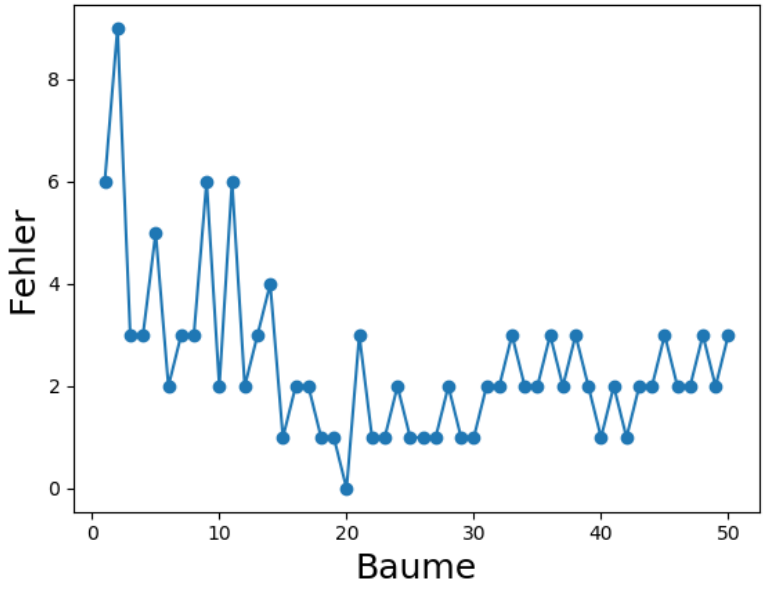
   - Nehmen Sie die aus Ihrer Sicht zwei besten Merkmale aus der letzten Aufgabe und trainieren Sie wieder den CART-Algorithmus mit den Parametern: *xDecimals=5* und *threshold=0.1*. Vergleichen Sie nun den Fehler mit dem aus Aufgabe 1.

   - Plotten Sie die drei Gebiete der Klassifikation *pcolormesh* wie auf Seite 76 aus dem Buch *Maschinelles Lernen* von Prof. Frochte angegeben:
     XX, YY = np.mgrid[XTrain.min():XTrain.max():0.005, XTrain.min():XTrain.max():0.005]
     X = np.array([XX.ravel(), YY.ravel()]).T
     Z = ...

Z = ...
ax.pcolormesh(...)

- Plotten Sie über den *pcolormesh* einen *Scatter Plot* der Testmenge dieser Aufgabe und überprüfen Sie, ob die Anzahl der Fehler, die Ihnen Ihr Algorithmus geliefert hat mit dem Plot passt.
- Kopieren Sie alle Plots dieser Aufgabe in die PDF-Datei unter Aufgabe 2 und kommentieren Sie die Ergebnisse im Vergleich zu Aufgabe 1.

3. Trainieren Sie nun ein Random Forest von 1 bis 50 Bäumen. Verwenden Sie für diese Aufgabe alle Merkmale des Dataset analog zu Aufgabe 1. Nennen Sie diese Datei ***Aufgabe3.py***. Zeigen Sie Ihre Ergebnisse mit Hilfe eines Plots, wobei der Fehler als Funktion der Anzahl der Bäume dargestellt wird. Als Beispiel, siehe nachfolgenden Plot:



- Kopieren Sie Ihren Plot in die PDF-Datei unter Aufgabe 3.