SEN P2: Klassifikation - Teil 1

In diesem Versuch entwickeln und testen Sie verschiedene Systeme zur Klassifikation von Daten in Python. Das Datenmaterial basiert auf zwei Merkmalen und ist kategorisiert in zwei Klassen. Die Daten stehen Ihnen im beiliegenden CSV-File `database.csv` zur Verfügung. Der Einfachheit halber werden in diesem Versuch die Klassen als „Klasse 0“ bzw. „Klasse 1“ und die Merkmale als „Merkmal A“ bzw. „Merkmal B“ bezeichnet.

- Lesen Sie die Daten aus dem CSV-File in Python ein. Dies lässt sich am einfachsten mittels der Bibliothek *Pandas* bewerkstelligen, wie im nachfolgenden Codeausschnitt dargestellt.[1]

```
import pandas as pd          # Importieren der Bibliothek
data = pd.read_csv("database.csv")  # Einlesen des Files
X = data.values[:,0:2]              # Merkmalswerte
y = data.values[:,2]                # Klassenwerte
```

- Visualisieren Sie den Datenbestand, bspw. als Scatterplot oder Scattermatrix.
- Unterteilen Sie den Datenbestand in Trainings- und Testdaten. Verwenden Sie die Trainingsdaten nur zum trainieren der Klassifikatoren und die Testdaten nur zum testen (siehe Anmerkung weiter unten).
- Erzeugen und trainieren Sie die nachfolgend aufgelisteten Klassifikatoren. Überlegen Sie sich, welcher dieser Klassifikationen für die vorliegende Aufgabe überhaupt Sinn macht. Vergleichen Sie die Klassifikationsergebnisse der unterschiedlichen Klassifikatoren. Visualisieren Sie die Entscheidungsbereiche der jeweiligen Klassifikatoren im Merkmalsraum (hierzu können Sie sich der Funktion `plot_2d_seperator.py` im beigefügten File `plot_2d_classifier_functions` bedienen) und interpretieren Sie die Diagramme. Überlegen Sie sich (sofern praktikabel) sinnvolle Parameterwerte zur Regularisierung der jeweiligen Klassifikatoren und spielen Sie mit den Parameterwerten.
  - Nearest Centroid Klassifikator
  - Naiver Bayes Klassifikator
  - Entscheidungsbaum
  - Random Forest
  - k-Nearest-Neighbors Klassifikator
  - Support Vector Machine mit Kernel

[1] Nach dem Einlesen des CSV-Files mittels `pd.read_csv()` liegt die Datenbank als sog. *Pandas DataFrame* (hier mit dem Namen `data`) vor. Die Daten selbst liegen in `data.values`. Die letzten beiden Zeilen des Codeausschnitts dienen der Umwandlung des DataFrames in Arrays (Merkmalswerte `X` und Klassen `y`).

Anmerkung: Sie können sich die Erfolgsquote des trainierten Klassifikators (also den Anteil der korrekt getroffenen Entscheidungen; sog. *Score* bzw. *Accuracy*) mit der Scikit-Learn Function `clf.score(X_test, y_test)` berechnen lassen, wobei `clf` die Instanz des Klassifikators darstellt (also z.B. `nc`, für den Fall, dass Sie den Klassifikator mittels `nc = NearestCentroid()` instanziiert haben). Die Variablen `X_test` und `y_test` stellen die Merkmalsmatrix bzw. den Ergebnisvektor des Testdatensatzes dar. Selbstverständlich kann durch ändern der Übergabeparameter auf `X_train` und `y_train` auch der mit den Trainingsdaten erzielte Score ermittelt werden.

- Erzeugen und trainieren Sie ggf. weitere Klassifikatorentypen und prüfen Sie, ob sich die Klassifikationsergebnisse weiter verbessern lassen.