Projektarbeit: Titel

Kurs: Maschinelles Lernen (MaLe-AD23-HS24)
Autor/in: Maxine Muster
Datum: TT.MM.JJJJ

Einführung

Anleitung: *Hier folgt eine kurze Zusammenfassung des Projektinhalts in wenigen Sätzen.*

- *Welches Problem wird gelöst (ev. mit ein bisschen Kontext)?*
- *Welche Daten werden verwendet?*
- *Welches ist die Zielvariable?*
- *Welche Features werden verwendet?*
- *...andere nützliche Informationen*

Die Zielgruppe eures Projekts sind andere Studierende.
Verwendet bitte eine knappe aber präzise Sprache.

Markdown: *Folgende Tutorials helfen vielleicht weiter falls ihr euch noch nicht so gut auskennt mit Markdown:*

- *Kurze Überisicht*
- *Markdown à la GitHub*
- *Detailierteres Tutorial*

Übrigens: Man kann in Markdown Zellen einfaches HTML verwenden (erkennbar an den <key>...</key> Blöcken) um etwas mehr Kontrolle über die Darstellung zu haben. Ich verwende solche. HTML-Elemente um die Anleitung visuell hervorzuheben. Für euch ist das aber keine Pflicht!

Setup

Anleitung: *In diesem Abschnitt geht es darum, das Jupyter Notebook zu konfigurieren. Ihr braucht hier nicht viel zu machen.*

In [14]:
```
# Basic imports
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris

# Enable vectorized graphics
%config InlineBackend.figure_formats = ["svg"]

# Setup plotting
PALETTE = [ (0.341, 0.648, 0.962, 1.0),
            (0.990, 0.476, 0.494, 1.0),
            (0.281, 0.749, 0.463, 1.0),
            (0.629, 0.802, 0.978, 1.0),
            (0.994, 0.705, 0.715, 1.0),
            (0.595, 0.858, 0.698, 1.0),
            (0.876, 0.934, 0.992, 1.0),
            (0.998, 0.901, 0.905, 1.0),
            (0.865, 0.952, 0.899, 1.0) ]

# For more color palettes, see here:
# https://seaborn.pydata.org/tutorial/color_palettes.html
# https://matplotlib.org/stable/users/explain/colors/colormaps.html
#PALETTE = sns.color_palette("husl", 8)
#PALETTE = sns.color_palette("viridis", 10)

print("Our color palette:")
sns.palplot(PALETTE, size=0.5)

sns.set_style("whitegrid")
plt.rcParams["axes.prop_cycle"] = plt.cycler(color=PALETTE)
plt.rcParams["figure.dpi"] = 300    # High-res figures (DPI)
plt.rcParams["pdf.fonttype"] = 42   # Editable text in PDF
```

```
Our color palette:
```

Präprozessierung

Anleitung: *In diesem Abschnitt werden die Daten geladen und aufbereitet.*

In [15]:
```
# Code to load and preprocess the data
...
```

Explorative Datenanalyse

Anleitung: *Hier untersucht ihr eure Daten ein erstes Mal, visualisiert sie und versucht, Muster zu erkennen. Zeigt hier, dass ihr neugierig seid und euch mit den Daten auseinandersetzt.*

In [ ]:
```
# Code to perform the exploratory data analysis
...
```

Feature Engineering und Dimensionalitätsreduktion

Anleitung: *Beim Feature Engineering versucht ihr aus den verfügbaren Daten nützliche neue Features zu generieren. Falls ihr bereits viele Prädiktoren habt, versucht ihr die Dimensionalität des Problems mittels eines geeigneten Verfahrens zu reduzieren. Je nach Datensatz und Problem fällt dieser Abschnitt länger oder kürzer aus.*

In [ ]:
```
# Code to perform the feature engineering
...
```

Modellieren, Trainieren und Validieren

Anleitung: *Es folgt der für diese Projektarbeit zentrale Abschnitt. Hier trainiert Datenmodelle, und validiert diese. Folgende Punkte sind zu beachten:*

- *Es sollen mindestens zwei verschiedene Datenmodelle trainiert und verglichen werden.*
- *Die Hyperparameter sollen mittels Kreuzvalidierung ermittelt werden.*
- *Die Vorhersagegenauigkeit der Modelle müssen mit einem separaten Testdatensatz abgeschätzt werden.*
- *Die Resultate der Trainings- und Testphasen sollen visualisiert werden (z.B. mittels ROC-Kurve oder Residuen-Plots).*
- *Bonuspunkte gibt es, falls*
  - *der Generalisierungsfehler des Modells robust ermittelt wird (durch wiederholtes Validieren, so dass die Kennzahlen für Vorhersagegenauigkeit als "$\mu \pm \sigma$" angeben werden kann).*
  - *die Vorhersagefehler der Modelle auf Muster untersucht werden.*
  - *falls eine Standardmodellierung basierend auf Beobachtungen verfeinert und weiterentwickelt wird.*

Wir empfehlen, die verschiedenen Modelle in scikit-learn als Pipelines aufzubauen.

In [ ]:
```
# Code to perform the model training and validation
...
```

Diskussion und Fazit

Anleitung: *Diskutiert hier kurz eure Erkenntnisse aus eurer Projektarbeit. Folgende Punkte müsst ihr addressieren:*

- *Wie interpretiert eure Ergebnisse:*
  - *Welches Datenmodell funktioniert am besten?*
  - *Wie gut löst es das formulierte Problem?*
  - *Entsprechen die Ergebnisse euren Erwartungen?*
- *Habt ihr Verbesserungsvorschläge für eure Datenmodelle?*
- *Beschreibt eure Lernerlebnisse. Was waren eure wichtigsten Erkenntnisse im Verlauf dieses Projekts?*