Voraussetzungen

- Python IDE eingerichtet
- Jupyter Notebook mit Python-kernel ist lauffähig
- Ihnen steht der bereinigte Umfrage-Datensatz zur Verfügung: `Umfrage_EML_2024_cleaned.csv`

Material

Machen Sie sich vertraut mit den folgenden Python-Klassen:

- from sklearn.preprocessing import StandardScaler
- from sklearn.preprocessing import MinMaxScaler
- from sklearn.preprocessing import PowerTransformer
- from sklearn.preprocessing import LabelEncoder

Übung -- 1. Teil

Projekt einrichten

- **Optional**:
  - Erzeugen Sie ein neues github-Repo, in dem Sie arbeiten
- Erzeugen Sie folgende Ordnerstruktur lokal:
  - `data/`
  - `notebook/`
- Speichern Sie die Umfragedaten unter `data/` ab
- Legen Sie auch das Aufgabenblatt (PDF) Dokument unter `./` ab

Daten einlesen

- Lesen mit der Python-Bibliothek "pandas" die Umfragedaten in einen DataFrame ein
- Lassen Sie sich die ersten Zeilen des DataFrames anzeigen
- Lesen Sie die Anzahl der Spalten und Zeilen des DataFrames aus und speichern Sie diese in Variablen:
  - Nutzen Sie dazu die `shape`-Methode des DataFrames.
  - Wenn der Datensatz 18 Zeilen und 16 Spalten hat, *printen* Sie das alles OK ist, andernfalls geben Sie eine Fehlermeldung aus.

Fehlwerte identifizieren und bereinigen

- Lassen Sie sich die Anzahl der Fehlwerte pro Spalte anzeigen.
- Führen Sie das Kommando `df.dropna()` aus. Was passiert?
- Führen Sie das Kommando `df.dropna(axis=1)` aus. Was passiert?
- Führen Sie das Kommando `df.dropna(how='all')` aus. Was passiert?

- Führen Sie das Kommando `df.dropna(subset=['semester'])` aus. Was passiert?
- Welche Vor- und Nachteile kann das Entfernen von Zeilen haben?

Fehlwerte imputieren

- Machen Sie sich mit der **SimpleImputer**-Klasse vertraut: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html
- Welche Strategien gibt es, um fehlende Werte zu imputieren?
- Ersetzen Sie den Fehlwert für `koepergroesse` durch den Durchschnittswert der Spalte:

```
from sklearn.impute import SimpleImputer
mean_imputer = SimpleImputer(strategy='mean')

VARIABLE_TO_IMPUTE = "koerpergroesse"  # Wähle eine Variable aus

result_mean_imputer = mean_imputer.fit_transform(
    df[VARIABLE_TO_IMPUTE].values.reshape(-1, 1))

# Erstelle ein DataFrame mit den imputierten Werten
df_imputed = pd.DataFrame(result_mean_imputer, columns=[
                          VARIABLE_TO_IMPUTE], index=df.index)

# Vergleich der beiden Spalten
pd.concat([df[VARIABLE_TO_IMPUTE], df_imputed], axis=1).head()
```

Versuchen Sie den Code nachzuvollziehen. Was passiert in den einzelnen Schritten?

- Der Fehlwert für die Körpergröße fehlt für eine Frau. Da die Körpergröße durchaus vom Geschlecht abhängt, wäre es besser, den Fehlwert durch die mittlere weibliche Körpergröße zu ersetzen. Versuchen Sie dies zu umsezusetzen.

Übung -- 2. Teil

Kategorische Variablen enkodieren

- Machen Sie sich mit der Pandas-Funktion `get_dummies()` vertraut: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html
- Erzeugen Sie Dummy-Variablen für die Spalte `geschlecht`.
- Für welche Variablen bietet es sich noch an, Dummy-Variablen zu erzeugen?
- Machen Sie sich mit dem Label-Encoder des sklearn-Moduls vertraut: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html
- Erzeugen Sie eine neue Spalte `abschluss_encoded` mit Hilfe des LabelEncoders.