Matthias Mruzek-Vering, M.Sc.

Hochschule Karlsruhe
University of
Applied Sciences

DSCB 230 - Aufgabenblatt 5

Die Aufgabenblätter werden in dieser Veranstaltung in Jupyter Notebooks veröffentlicht und bearbeitet. Diese finden Sie in der Github Organisation für Data Science 2 unter dem Repository *dscb230-tutorial* (https://github.com/hka-mmv/dscb230-tutorial). Die Musterlösung wird ebenfalls in Form eines Jupyter Notebook in Github hochgeladen.

Aufgabenteil 1: Data Cleaning + Matplotlib

Lesen Sie die Beispielhaften Dummy Daten MOCK_DATA.csv ein und speichern Sie diese in einem Dataframe. Das sind z.T. sehr viele NaN Werte. Wenn alle Daten ignoriert werden würden, würden nicht mehr viele übrig bleiben. Wir verfolgen zunächst folgende Strategie:

- Für numerische Werte wird für einen fehlenden Wert der Mittelwert der restlichen Werte eingesetzt.
- Für nicht-numerische Werte wird für einen fehlenden Wert der häufigsten Wert (=Modus) der restlichen Werte eingesetzt.

Zum Auffüllen der Werte stellt scikit-learn die Klasse SimpleImputer bereit.

Im weiteren Sollen Sie nun die Häufigkeitsverteilungen (Methode: value_counts()) für das Attribut gender ermittelt und Grafisch darstellt werden. Beachten Sie das ein Säulen und Liniendiagram in einem Subplot ausgegeben werden soll.

Tim Ehmann
Fabian Juny
Raphaele Salvatore Licciardo, B.Sc.

Matthias Mruzek-Vering, M.Sc.

Aufgabenteil 2: Time Series Analysis mit Pandas

Sie haben verschiedene Temperatursensoren installiert, die die Temperatur im 15-Minuten-Takt messen. Jeder Sensor speichert dabei zwei Werte in einer CSV-Datei ab. Sie finden diese Sensordaten in "sensor1.csv", "sensor2.csv" und "sensor3.csv".

Hinweis: Sollten Sie nicht weiterkommen, bzw. wissen Sie nicht, welche Methoden Sie verwenden können, schauen Sie sich die Hinweise am Ende der Aufgabe an.

1. Einlesen der Daten

- Ihre erste Aufgabe ist es jede CSV-Datei als ein pandas DataFrame einzulesen und anschließend zu einem DataFrame zu kombinieren.
- Passen Sie den Datentyp des Zeitstempels an und setzen Sie die Indizes des DataFrames auf die Werte der Zeitstempel

2. Zugriff auf Daten der Time Series

- Geben Sie alle Sensordaten um 7 Uhr aus
- Geben Sie alle Sensordaten zwischen 8 und 10 Uhr aus

3. Berechnungen

- Berechnen Sie den Tagesdurchschnitt jedes Sensors
- Berechnen Sie den Durchschnitt aller Sensoren und speichern Sie die Werte in einer extra Spalte
- Berechnen Sie den gleitenden Durchschnitt 5. Ordnung und speichern Sie die Werte in einer extra Spalte
- Berechnen Sie die Temperaturänderung des einfachen Durchschnitts pro Uhrzeit (Änderung zum vorherigen Durchschnitt)

4. Visualisieren

- Stellen Sie den Verlauf des Durchschnitts, sowie des gleitenden Durchschnitts jeweils als Liniendiagramm dar.

Empfehlung: Nutzen Sie aus Gründen der Übersichtlichkeit zwei verschiedene Diagramme

Hinweise:

1) read_csv(), merge(), to_datetime(), set_index()
2) loc[value], loc[value:value2]
3) mean() -> axis-Attribut betrachten, rolling(), diff(), shift()

Tim Ehmann
Fabian Juny
Raphaele Salvatore Licciardo, B.Sc.

Aufgabenteil 3:

Zu einem Wettbewerb haben sich mehrere Personen angemeldet, aber nicht alle haben teilgenommen, weshalb sich in den Ergebnisdaten NaN Werte gebildet haben. Ersetzten Sie diese durch den Wert 0.

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anna', 'Dima', 'Katherina', 'James',
'Emily', 'Michael', 'Florian', 'Laura', 'Kevin', 'Jonas'],
        'score': [12, 9, 16, np.nan, 10, 20, 14, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes',
'no', 'no', 'yes']}
df = pd.DataFrame(exam_data)
```

Florians Punktezahl soll im Nachhinein auf 15 erhöht werden und aufgrund eines Täuschungsversuches wird Jonas disqualifiziert und der Teilnehmer mit der nächst höchsten Punktzahl soll nachnominiert werden.
Aktualisieren Sie das Dataframe mit den neuen Informationen.

In der Hauptrunde des Wettbewerbs hat jeder Teilnehmer 5 Versuche, um eine höchstmögliche Punktezahl aufzustellen. Überprüfen Sie mithilfe der Datei scores.csv, welche Daten von Teilnehmern des Wettbewerbs über mehrere Jahre gesammelt hat, ob die erreichte Punktzahl mit der Nummer des Versuches zusammenhängt.
Verwenden Sie hierfür die Aggregatfunktionen mean(), median() oder auch std()
Stellen Sie diesen mithilfe eines Diagramms dar, indem die Durchschnittliche Punktzahl der Versuche angezeigt werden.
Hinweis: der csv Datei fehlen noch die Spaltennamen. Fügen Sie diese hinzu (Participant_ID, Attempt_1, Attempt_2, ...)