

# Digitale Lernplattform Physik Didaktik - Hin zur Analyse des Nutzerverhaltens

Stand der Ansätze und Ideen 04. April 2022

# Die Herausforderungen

## Strukturieren

- 1) Datenquellen:
  - a) Website
  - b) Log-Files
  - c) Tests/Umfragen
- 2) Programme:
  - a) Wordpress
  - b) Excel/Word
  - c) Python/Pandas

## Automatisieren

- 1) Optimierung Input:
  - a) Reduktion
  - b) Flexibilität
- 2) Optimierung Prozess:
  - a) Manuelle Eingriffe
  - b) Anpassung
- 3) Optimierung Output:
  - a) Umfangreich
  - b) Übersichtlich

## Analysieren

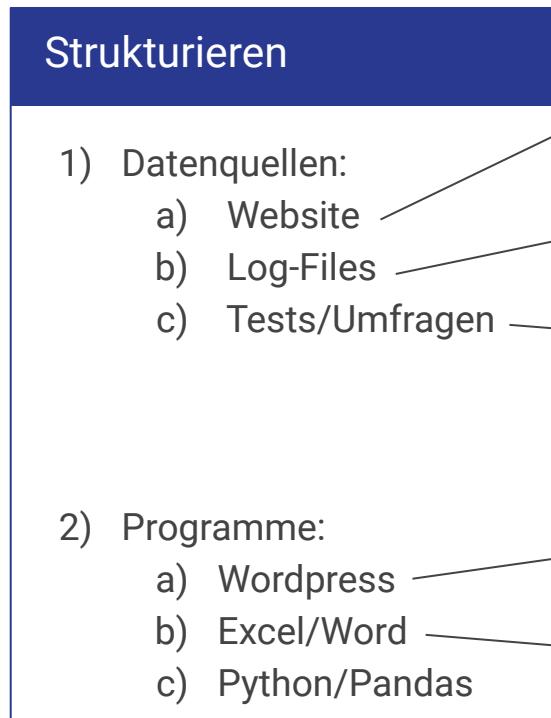
- 1) Mikroprozesse:
  - a) Seitenaufrufe
  - b) User-Gruppierung
  - c) Mustererkennung
- 2) Meso/Makroprozesse:
  - a) User-Verhalten
  - b) Korrelationen
  - c) User-Clustering

# Zielformulierung

Wir wollen herausfinden wie sich **Nutzer** der digitalen Lernplattform **verhalten**, um auf der Grundlage von **Tests**, Hinweise für den **Einfluss unterschiedlicher Lernverhalten auf den Lernerfolg** zu finden. Hierzu sollen typische **Verhaltensmuster** identifiziert werden, um **Nutzer gruppieren** zu können. Das kann sowohl auf der **Mikroebene** (einzelne Folgen von Aktivitäten), als auch auf der **Meso- und Makroebene** (gesamte Nutzeraktivitäten) geschehen. Dazu werden **Daten über die Aktivitäten** der Nutzer analysiert, die bei der Verwendung der Plattform angefallen ist. Anhand zukünftiger Daten sollen die Ergebnisse validiert werden. Der Fokus liegt dabei auf:

1. **Schneller und automatisierter Auswertung**, insbesondere auch von neuen Daten
2. Einer möglichst **vollständigen Nutzung aller vorhandenen Daten** (data bottleneck)
3. **Aussagekräftige Ergebnisse**, die **konkrete Schlussfolgerungen** zulassen (Transparenz)

# Die Bereiche im Detail



- Online high-level Daten (Header, Links,Urls, ...)
- Online Text-Daten (Beispiele, Erklärungen, ...)
- Online Bilddaten (Abbildungen, Fotos, ...)
- Offline Daten (image/Kopie der Plattform)
- Primär: Benutzer, Datum/Zeit, Link, (Referrer)
- Sekundär: Code, Kennziffer, Thema, Lernform
- Tests: Benutzer, Testversion, Ergebnisse
- Umfragen: Fragen, Antworten, Kommentare
- Ermittlung Wissensstand vorher und nachher
- Zugriff/Synchronisation online/offline Daten
- Log-Files, Zuordnungen, offline Sicherungen
- Datenstrukturierung -> Sortieren, zuordnen, ...
- Generieren von Sekundärdaten (ableiten)
- Datenauswertung -> Korrelationen, Muster, ...

# Verwendung von Python

## Pro

- Standard in Data Science (neben R)
- Direkte Anwendung von Algorithmen für Clustering, Klassifizierung, Maschinelles Lernen, uvm.
- All-in-one Lösung für Datenmanagement, Verarbeitung, Automatisierung, Datenanalyse

## Kontra

- Grundkenntnisse im Programmieren erforderlich
- Manchmal eine Art black-box
- (Performance kann Problem sein)

# Wie geht Datenanalyse mit Python?

- In Python wird alles was man tut als Befehl in der Programmiersprache formuliert
- Python hat unendlich viele Bibliotheken (packages, modules) für fast alle Anwendungen
- Übliche Befehle zur Datenauswertung:
  - Einlesen von Daten aus Dateien, Datenbanken, etc
  - Sortieren, Gruppieren, Transformieren der Daten
  - Bestimmung von Korrelationen, Muster, Cluster, etc
  - Anwendung von komplexeren Algorithmen/Methoden
  - Darstellung der Daten (Tabellen, Plots, Animationen, ...)
- Wie werden die Befehle formuliert?
  - Schritt für Schritt -> Jupyter-Notebooks
  - In Dateien (Skripte) -> IDEs (PyCharm, VSCode)
  - Als Anwendungen -> Verpacken in Programm

# Jupyter Notebooks

The screenshot shows a Jupyter Notebook interface running in a browser window. The title bar indicates the file is 'csv\_Verarbeitungs-Modul.ipynb' and the URL is 'localhost:8888/notebooks/Documents/DataScience/Tasks/Python/Abgaben/Blatt\_7/csv\_Verarbeitungs-Modul.ipynb'. The notebook has three cells:

- In [1]:**

```
1 import pathlib
2 import pickle
3
4 from test_classes import TestModule
5
6 with open('test_csv_verarbeiten.pickle', 'rb') as file:
7     test = pickle.load(file)
8
9 module_file = 'csv_verarbeiten.py'
```

executed in 127ms, finished 11:27:38 2021-12-07
- In [ ]:**

```
1
```
- In [2]:**

```
1 spec = test.read_submission(pathlib.Path(module_file))
```

executed in 3ms, finished 11:27:38 2021-12-07

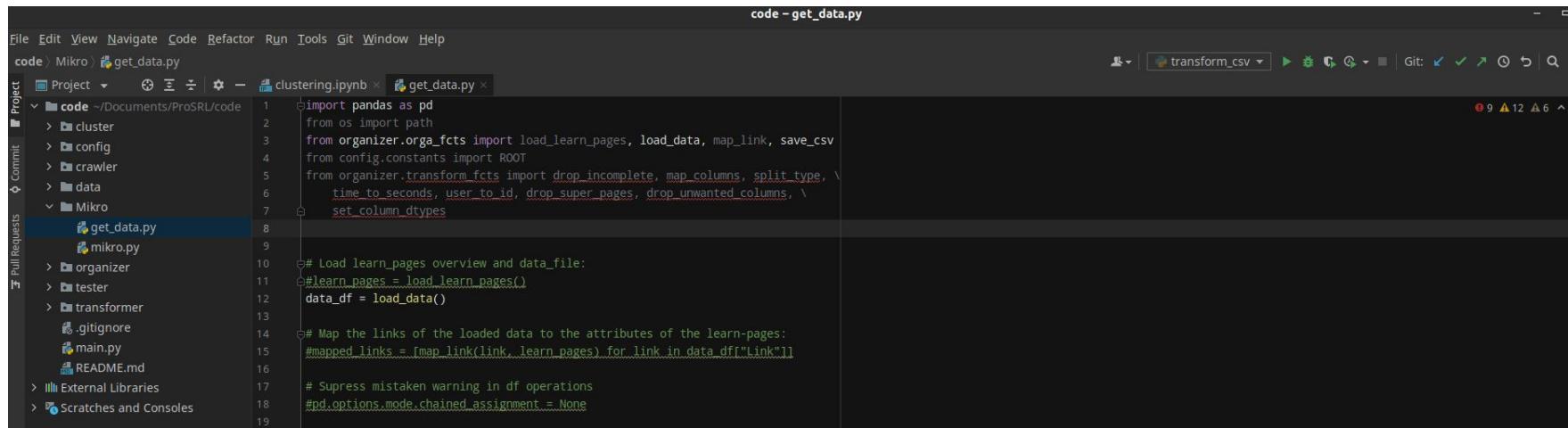
Pro:

- + Befehle Schritt für Schritt ausführen
- + Einfache und flexible Handhabung
- + Daten bleiben im Arbeitsspeicher

Kontra:

- + Wird leicht unübersichtlich
- + Praktisch nicht modularisierbar
- + Debugging schwierig

# Skript in IDE



```

code - get_data.py
File Edit View Navigate Code Refactor Run Tools Git Window Help
code > Mikro > get_data.py
code -> /Documents/ProSRL/code
Project ▾ code ~/Documents/ProSRL/code
  > cluster
  > config
  > crawler
  > data
  > Mikro
    > get_data.py
    > mikro.py
  > organizer
  > tester
  > transformer
  .gitignore
  main.py
  README.md
> External Libraries
> Scratches and Consoles
code - get_data.py
1 import pandas as pd
2 from os import path
3 from organizer.orga_fcts import load_learn_pages, load_data, map_link, save_csv
4 from config.constants import ROOT
5 from organizer.transform_fcts import drop_incomplete, map_columns, split_type,
6   time_to_seconds, user_to_id, drop_super_pages, drop_unwanted_columns,
7   set_column_dtypes
8
9
10 # Load learn_pages overview and data_file:
11 #learn_pages = load_learn_pages()
12 data_df = load_data()
13
14 # Map the links of the loaded data to the attributes of the learn-pages:
15 #mapped_links = [map_link(link, learn_pages) for link in data_df["Link"]]
16
17 # Suppress mistaken warning in df operations
18 #pd.options.mode.chained_assignment = None
19

```

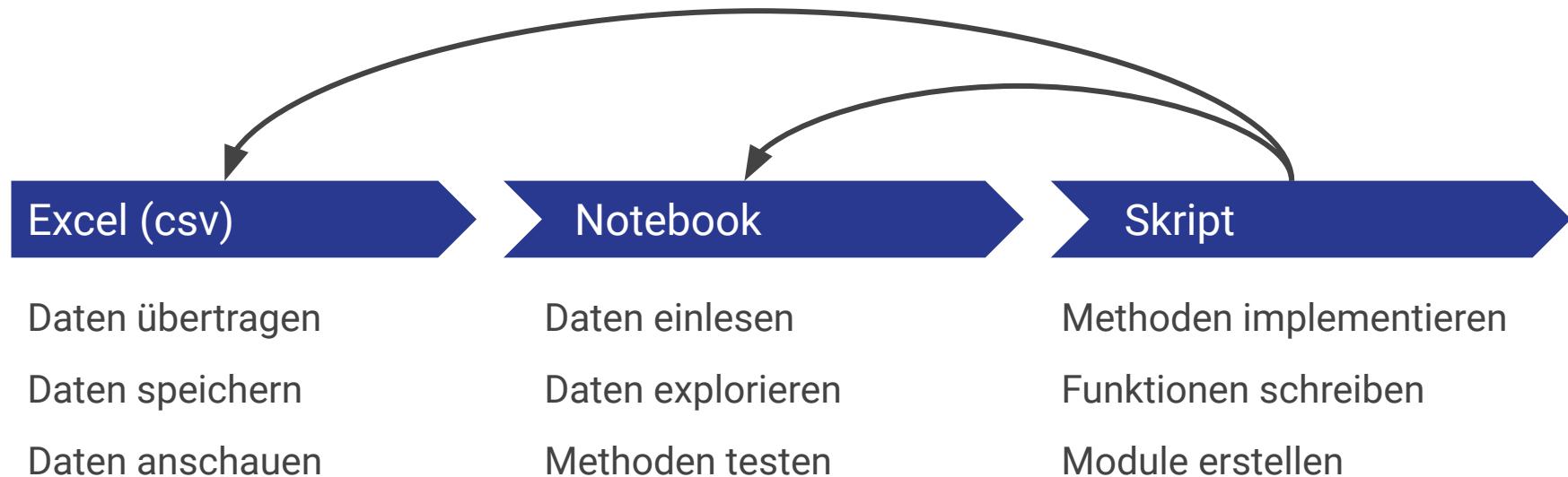
Pro:

- + Übersichtlich, viele Hilfestellungen
- + Optimiert für modulare Projekte
- + Debugging etwas leichter

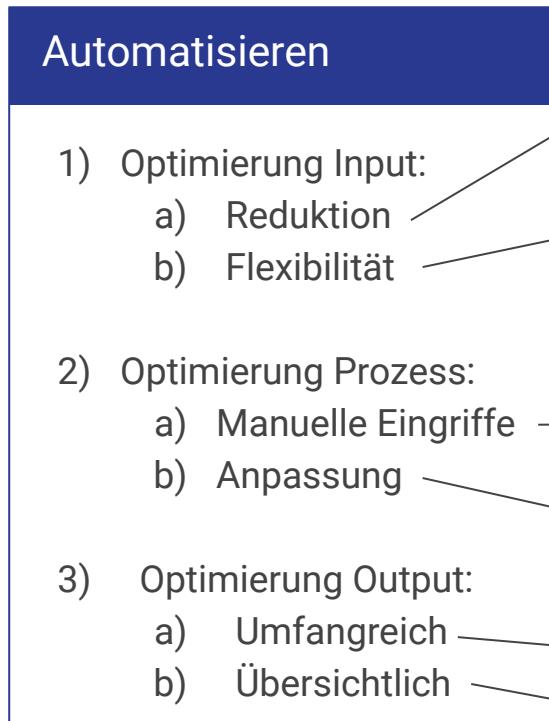
Kontra:

- + Skripte nur komplett ausführbar
- + Benötigt etwas mehr Gewöhnung
- + Daten müssen gespeichert werden

# Eine mögliche Herangehensweise



# Zurück zu den Bereichen



- Möglichst wenige Daten/Dateien als Input
  - > Übersichtlichkeit/Flexibilität
  - > Weniger manuelle Arbeit
  - > Keine redundante Datenspeicherung
- Code muss bei neuen Daten nicht (nur minimal) angepasst werden
  - > Schnelle neue Auswertungen
  - > Spart manuelle Arbeit
- Im Idealfall läuft alles automatisch
- Mögliche Eingriffe: Optimierung von Parametern, Performance, Bugs
- Neue Methoden leicht zu implementieren
- Alle möglichen Zusammenhänge erkennen
- Möglichst leicht verständlich darstellen

# Möglicher Aufbau der Module für input Verarbeitung

## crawler

Holt Daten direkt von der Website der Lernplattform

- + Schnell
- + Flexibel
- + Unabhängig

Art/Speicherung der Daten

- + Metadaten zu jeder einzelnen Lernseite (Label, Titel, Thema, Bereich, Link, Typ)
- + yaml file (Wörterbuch)

## organizer

Liest log files und yaml file ein und verknüpft beide

- + Schnell
- + Flexibel
- + Sicher

Was passiert hier genau?

- + Korrektur von Fehlern
- + Mappen von Metadaten mittels Link aus csv
- + Speichern der relevanten Daten in einer csv Datei

## transformer

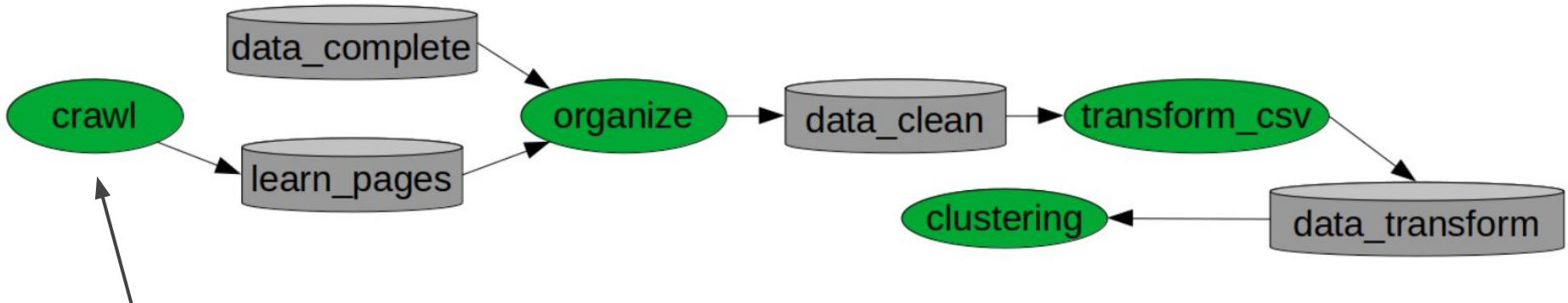
Liest csv von organizer ein und transformiert Daten adäquat

- + Schnell
- + Flexibel
- + Angepasst

Was ist hier möglich?

- + Sekundärdaten generieren
- + Unnötige/unvollständige Daten entfernen
- + Numerische Daten erzeugen
- + Datentypen festlegen

# Datenfluss



Dynamik I

- Grundgrößen zur Kraft
- Mehrkraftsysteme
- Besondere Kräfte

Dynamik II

- Kraft und Translationsbewegungen
- Kraft und Rotationsbewegungen
- Reibungskräfte

Dynamik I

 Grundgrößen zur Kraft

 Erarbeiten Grundwissen

 Üben Grundwissen

 Testen Grundwissen



Dynamik II

 Kraft und Translationsbewegungen

 Erarbeiten Grundwissen

 Üben Grundwissen

 Testen Grundwissen



## learn\_pages

ksla1b1\_1a:

Title: kraft identifizieren anhand von serienbildern

Topic: Kraft und Wirkung

Type: Erarbeiten Grundwissen

Category: Grundgrößen zur Kraft

Link: [https://lenvi.l3hrit.de/online-lernen/kraft-2/ksla1b1\\_1a](https://lenvi.l3hrit.de/online-lernen/kraft-2/ksla1b1_1a)

ksla1b2\_1a:

Title: keine aussage ueber kraft anhand serienbild

Topic: Kraft und Wirkung

Type: Erarbeiten Grundwissen

Category: Grundgrößen zur Kraft

Link: [https://lenvi.l3hrit.de/online-lernen/kraft-2/ksla1b2\\_1a](https://lenvi.l3hrit.de/online-lernen/kraft-2/ksla1b2_1a)

ksla1b3\_1a:

Title: identifizieren durch verformung

Topic: Kraft und Wirkung

Type: Erarbeiten Grundwissen

Category: Grundgrößen zur Kraft

Link: [https://lenvi.l3hrit.de/online-lernen/kraft-2/ksla1b3\\_1a](https://lenvi.l3hrit.de/online-lernen/kraft-2/ksla1b3_1a)

## data\_complete

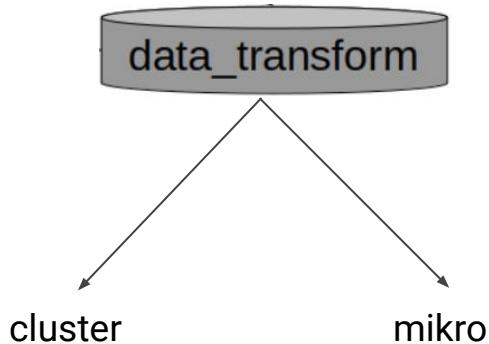
	A	B	C	D	E	F	G	H
1	Date/Time	P	K	A	O	User	C	R
2	04/10/2021 09:28:59	S	Start	s	t	LBenutzer HE1021-900	H	ht
3	04/10/2021 09:29:13	E	Einf	e	i	LBenutzer HE1021-900	H	ht
4	04/10/2021 09:29:17	L	Log	o	g	LBenutzer HE1021-900	H	ht
5	04/10/2021 11:54:43	S	Star	s	t	LBenutzer HE1021-900	H	ht
6	04/10/2021 11:55:01	E	Einf	e	i	LBenutzer HE1021-900	H	ht
7	04/10/2021 11:55:05	K	Kraf	k	r	LBenutzer HE1021-900	H	ht
8	04/10/2021 11:56:30	K	K	K	R	LBenutzer HE1021-900	H	ht
9	04/10/2021 11:56:34	K	K	K	R	LBenutzer HE1021-900	H	ht
10	04/10/2021 11:54:12	S	Star	s	t	LBenutzer HE1021-901	H	ht
11	04/10/2021 11:54:31	E	Einf	e	i	LBenutzer HE1021-901	H	ht
12	04/10/2021 11:54:33	K	Kraf	k	r	LBenutzer HE1021-901	H	ht
13	04/10/2021 11:56:22	K	K	K	R	LBenutzer HE1021-901	H	ht
14	04/10/2021 11:56:29	K	K	K	R	LBenutzer HE1021-901	H	ht
15	04/10/2021 11:56:38	S	Star	s	t	LBenutzer HE1021-901	H	ht
16	04/10/2021 11:56:59	E	Einf	e	i	LBenutzer HE1021-901	H	ht
17	04/10/2021 11:57:06	K	Kraf	k	r	LBenutzer HE1021-901	H	ht

## organize

	A	B	C	D	E	F	G	H
1	Label	Title	Topic	Type	Category	Link	Date/Time	
2	0 Startseite							04/10/2021 09:28:59
3	1 Tour							04/10/2021 09:29:13
4	2 Logout							04/10/2021 09:29:17
5	3 Startseite							04/10/2021 11:54:43
6	4 Tour							04/10/2021 11:55:01
7	5 Übersicht							04/10/2021 11:55:05
8	6 ks1a			Erarbeiten Grundwissen	Grundgrößen zur Kraft	<a href="https://lenvi.l3hrit.de/online-lernen/kraft-2/k">https://lenvi.l3hrit.de/online-lernen/kraft-2/k</a>		04/10/2021 11:56:30
9	7 ka1a1	Kraft und Wirkung	Erarbeiten Grundwissen	Erarbeiten Grundwissen	Grundgrößen zur Kraft	<a href="https://lenvi.l3hrit.de/online-lernen/kraft-2/">https://lenvi.l3hrit.de/online-lernen/kraft-2/</a>		04/10/2021 11:56:34

transform

	A	B	C	D	E	F	G
1	User	Title	Category	0	TotSec	LearnType	Level
2	6	0	0	0	8851	2	0
3	7	0	0	0	8855	4	0
4	11	1	0	0	8843	0	0
5	12	1	1	0	8850	0	0
6	16	1	0	0	8891	2	0
7	17	1	0	0	8895	4	0
8	21	5	0	0	8819	1	0
9	22	5	2	0	8851	1	0
10	26	4	0	0	8775	2	0



## Zurück zu den Bereichen

### Automatisieren

- 1) Optimierung Input:
  - a) Reduktion
  - b) Flexibilität
- 2) Optimierung Prozess:
  - a) Manuelle Eingriffe
  - b) Anpassung
- 3) Optimierung Output:
  - a) Umfangreich
  - b) Übersichtlich

### Analysieren

- 1) Mikroprozesse:
  - a) Seitenaufrufe
  - b) User-Gruppierung
  - c) Mustererkennung
- 2) Meso/Makroprozesse:
  - a) User-Verhalten
  - b) Korrelationen
  - c) User-Clustering

# Möglicher Aufbau des Clustering Moduls - Teil I

cluster\_prep\_fcts

Präpariert die transformierten Daten für die Anwendung von Clustering - Methoden

- + Erzeugt Dummies
- + Gruppiert Daten nach einzelnen Usern
- + Generiert zusätzliche Feature über Muster im Lernverhalten -> Welche Formate werden genutzt, wie wird gewechselt

cluster\_fcts

Verwendet Cluster Algorithmen zum Auffinden von User Clustern

- + Skaliert die Daten
- + Extrahiert relevante Feature
- + Wendet unterschiedliche Algorithmen darauf an
- + Anzupassende Parameter: Skalierung, Verwendete Feature, Dim, Algorithmus (kMeans, Meanshift, spectral, agglomerative)

cluster\_plot\_fcts

Stellt die Daten und Ergebnisse in graphischer Form dar

- + Heatmaps für die Korrelationen von Features
- + Pairplots zum Vergleich von einzelnen Features
- + Lineplots für Visualisierung von Lernmustern
- + Scatterplots zur Darstellung der Ergebnisse der Cluster Algorithmen

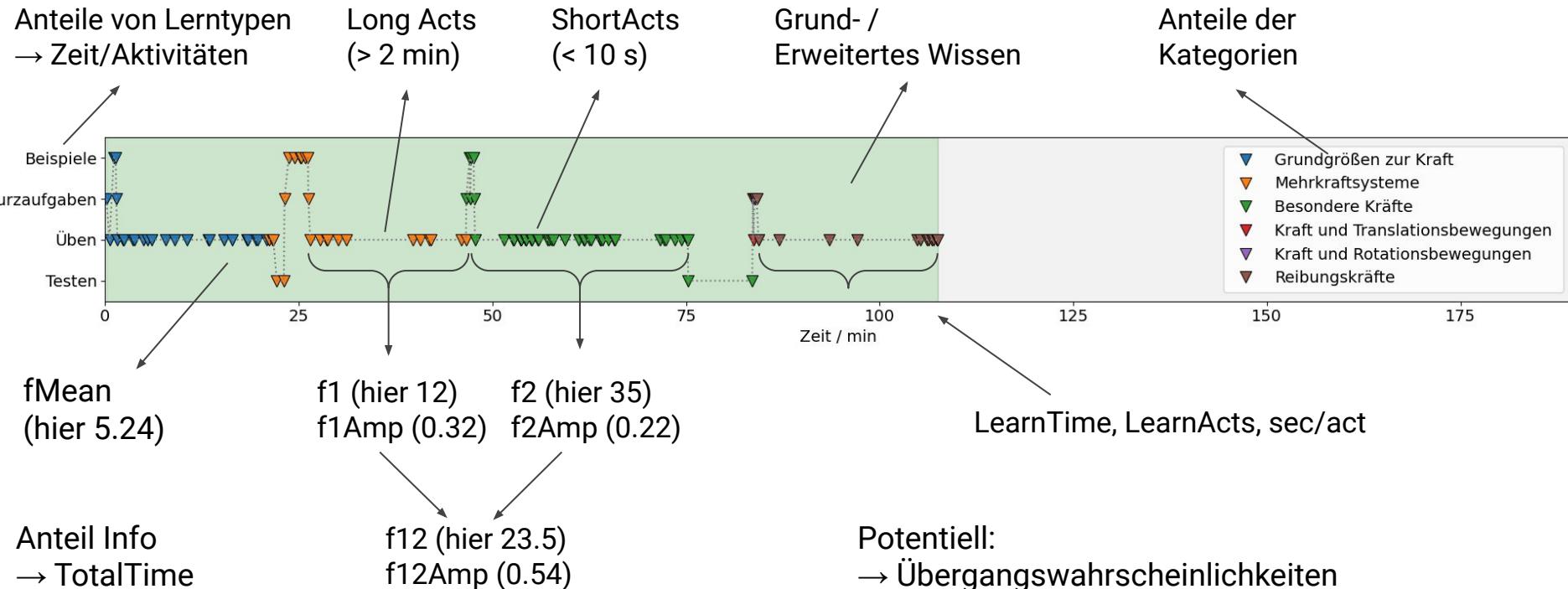
## cluster\_prep\_fcts

	A	B	C	D	E	F	G
1	User	Title	Category	0	TotSec	LearnType	Level
2	6	0	0	0	8851	2	0
3	7	0	0	0	8855	4	0
4	11	1	0	0	8843	0	0
5	12	1	1	0	8850	0	0
6	16	1	0	0	8891	2	0
7	17	1	0	0	8895	4	0
8	21	5	0	0	8819	1	0
9	22	5	2	0	8851	1	0
10	26	4	0	0	8775	2	0

User	Title	TotSec	Tests	Übungen	Kurzaufgaben	Beispiele	Info	Grund	Erweitert	Cat_0	Cat_1	Cat_2	Cat_3	Cat_4	Cat_5
0	0	8851	0	0	1	0	0	1	0	1	0	0	0	0	0
0	0	8855	0	0	0	0	1	1	0	1	0	0	0	0	0
1	0	8843	1	0	0	0	0	1	0	1	0	0	0	0	0
1	1	8850	1	0	0	0	0	1	0	1	0	0	0	0	0
1	0	8891	0	0	1	0	0	1	0	1	0	0	0	0	0
User	Tests	Übungen	Kurzaufgaben	Beispiele	Info	Grund	Erweitert	Cat_0	Cat_1	Cat_2	Cat_3	Cat_4	Cat_5		
0	11	10	18	19	41	99	0	49	50	0	0	0	0	0	0
1	8	0	43	39	82	172	0	53	71	9	0	39	53	71	9
2	0	7	54	0	21	82	0	34	23	6	6	34	23	6	11
4	0	0	23	11	19	53	0	1	0	0	6	1	0	6	46
5	7	19	21	5	18	62	8	15	5	17	4	15	5	17	4

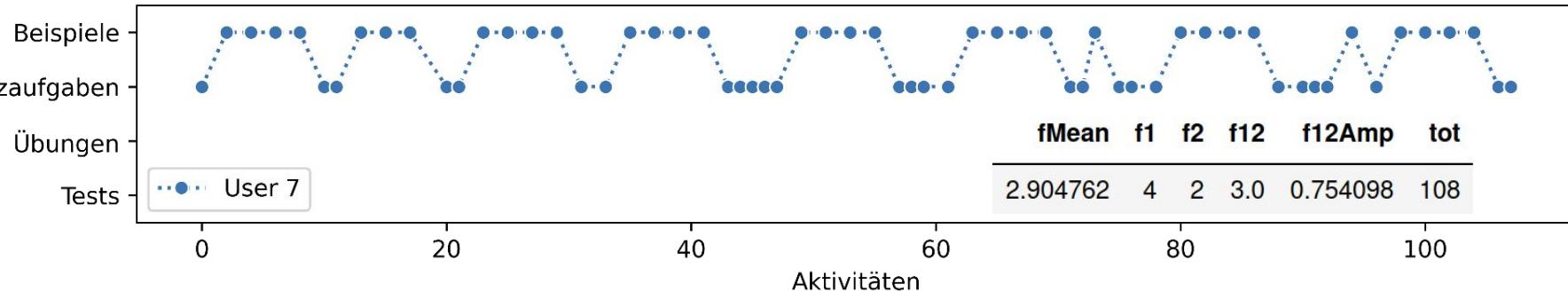
User	Tests	Übungen	Kurzaufgaben	Beispiele	Info	Grund	Erweitert	Cat_0	Cat_1	Cat_2	Cat_3	Cat_4	Cat_5	fMean	f1	f2	f12	f12Amp	tot	
0	0	11	10	18	19	41	99	0	49	50	0	0	0	2.521739	4	1	2.5	0.396552	99	
1	1	8	0	43	39	82	172	0	53	71	9	0	39	4.090909	4	21	12.5	0.544444	172	
2	2	0	7	54	0	21	82	0	34	23	6	6	11	3.058824	10	2	6.0	0.384615	82	
3	4	0	0	23	11	19	53	0	1	0	0	6	46	2.615385	4	2	3.0	0.705882	53	
4	5	7	19	21	5	18	62	8	15	5	17	4	19	10	2.300000	2	1	1.5	0.507246	70
5	6	2	3	28	36	68	113	24	7	34	33	7	30	26	2.904762	4	2	3.0	0.754098	137

# Mögliche Feature für einzelne User

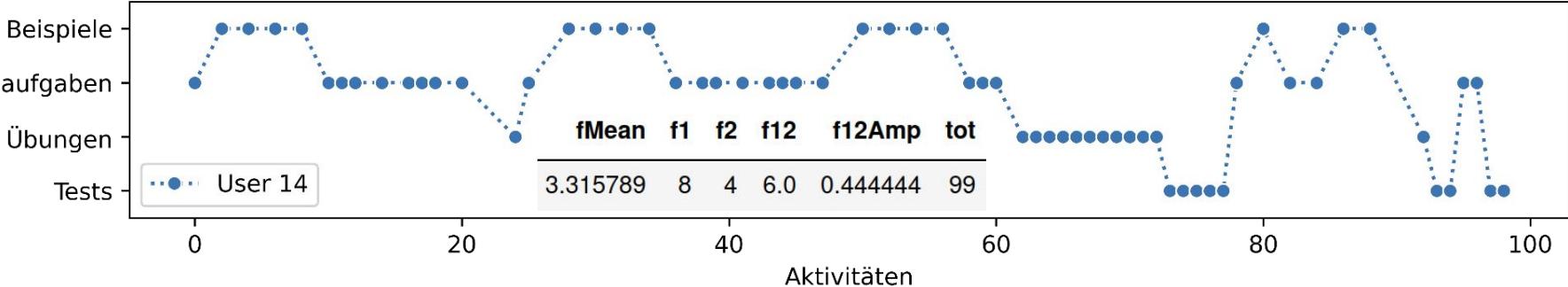


# Analyse der Run-Längen

LearnType



LearnType



# Analyse der Run-Längen

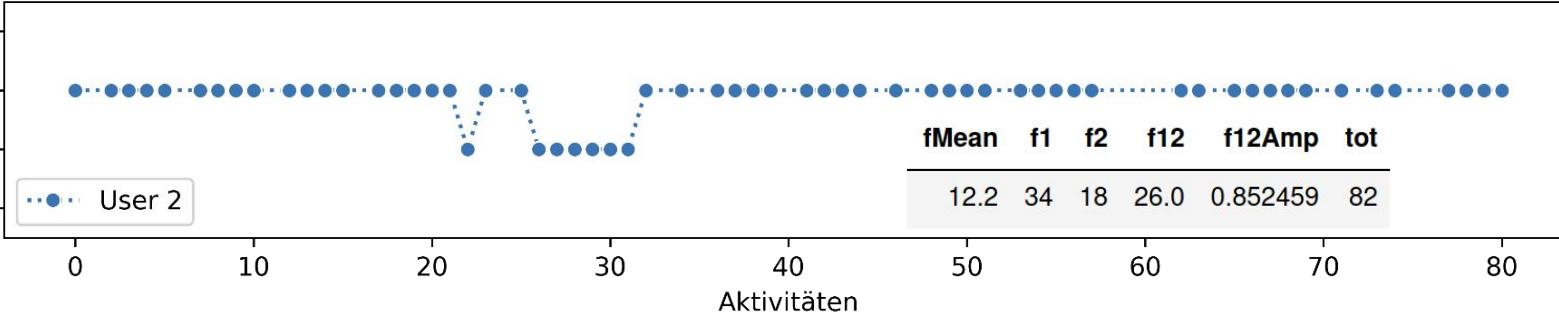
LearnType

Beispiele

Kurzaufgaben

Übungen

Tests  
User 2



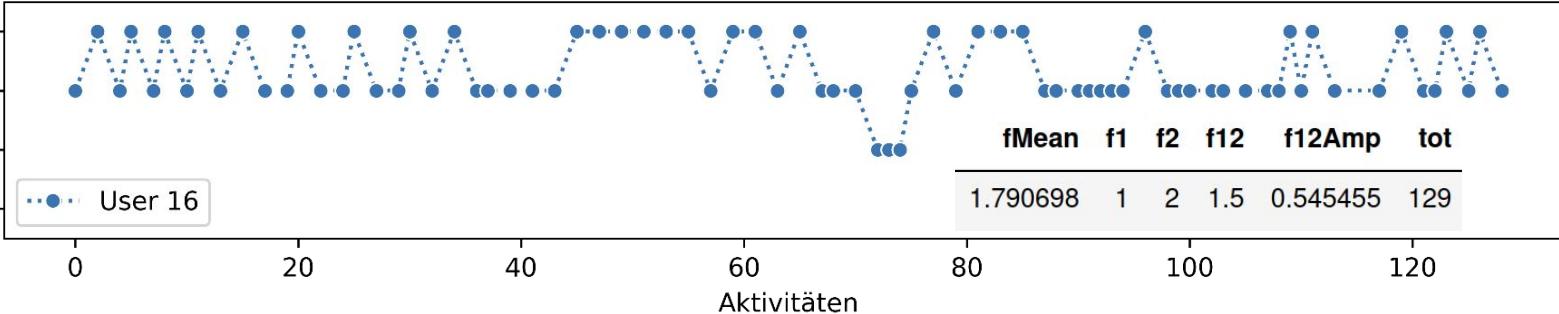
LearnType

Beispiele

Kurzaufgaben

Übungen

Tests  
User 16



## cluster\_fcts

Feature selektieren -> Scaler anwenden -> Model fitten -> Label, Zentren, SSE zurückgeben

- 1) MinMax (Alle Daten auf Intervall [0, 1])
- 2) Standard (Daten so, dass mean=0, std=1)

### 1) k-means:

The k-means algorithm divides a set of  $N$  samples  $X$  into  $K$  disjoint clusters  $C$ , each described by the mean  $\mu_j$  of the samples in the cluster. The means are commonly called the cluster “centroids”; note that they are not, in general, points from  $X$ , although they live in the same space.

The K-means algorithm aims to choose centroids that minimise the **inertia**, or **within-cluster sum-of-squares criterion**:

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2)$$

## 2) MeanShift

Given a candidate centroid  $x_i$  for iteration  $t$ , the candidate is updated according to the following equation:

$$x_i^{t+1} = m(x_i^t)$$

Where  $N(x_i)$  is the neighborhood of samples within a given distance around  $x_i$  and  $m$  is the *mean shift* vector that is computed for each centroid that points towards a region of the maximum increase in the density of points. This is computed using the following equation, effectively updating a centroid to be the mean of the samples within its neighborhood:

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$$

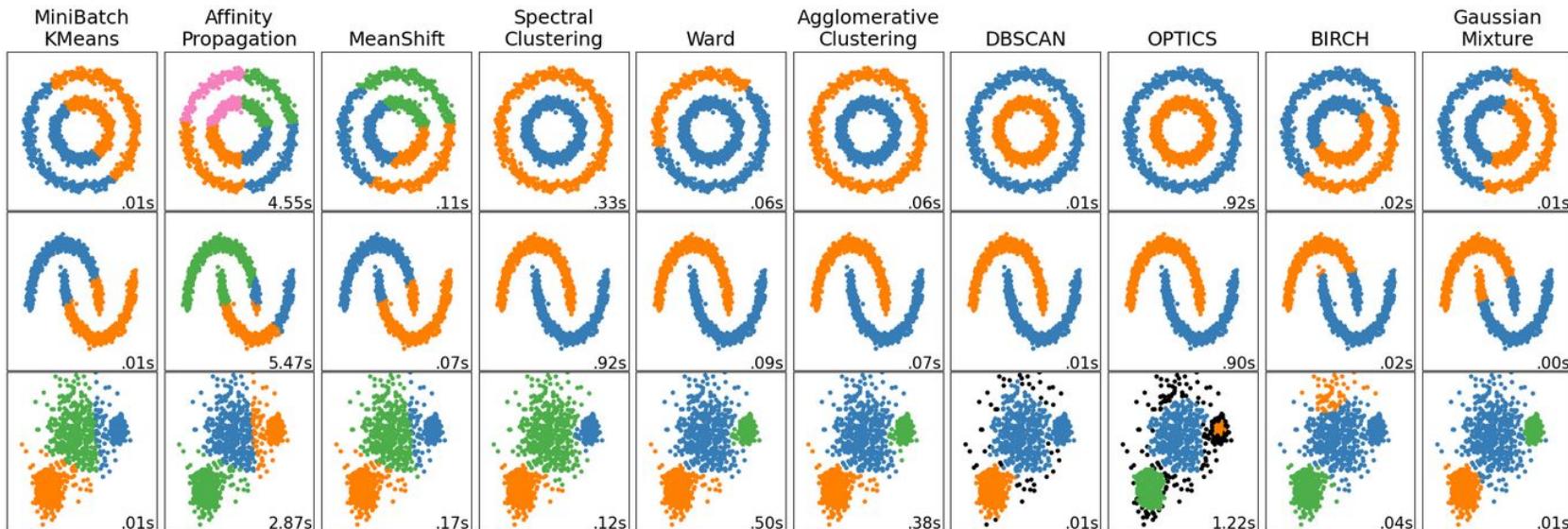
## 3) Spectral Clustering

[SpectralClustering](#) performs a low-dimension embedding of the affinity matrix between samples, followed by clustering, e.g., by KMeans, of the components of the eigenvectors in the low dimensional space. It is especially computationally efficient if the affinity matrix is sparse and the `amg` solver is used for the eigenvalue problem (Note, the `amg` solver requires that the [pyamg](#) module is installed.)

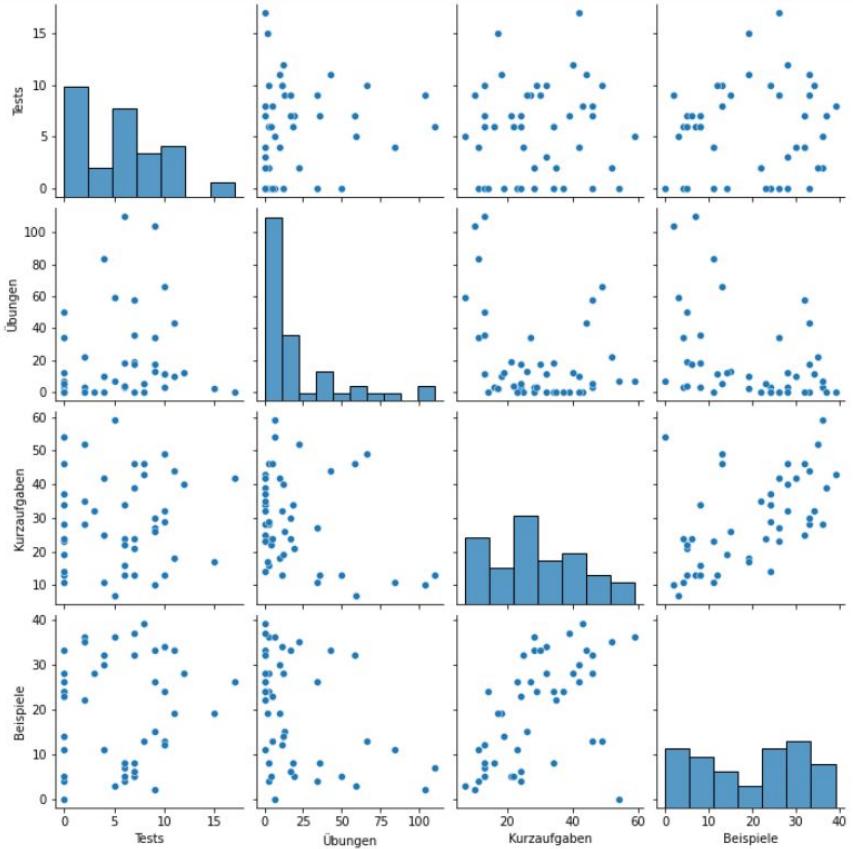
## 4) Agglomerative Clustering

The [AgglomerativeClustering](#) object performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy:

- **Ward** minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.



# cluster\_plot\_fcts



	User	Tests	Übungen	Kurzaufgaben	Beispiele	Info	Grund	Erweitert	Cat_0	Cat_1	Cat_2	Cat_3	Cat_4	Cat_5	fMean	f1	f2	f12	f12Amp	tot
User	-	-0.032	0.14	0.067	-0.0058	0.0055	0.11	-0.021	0.26	0.23	-0.075	-0.031	-0.27	-0.21	-0.039	0.036	0.13	0.081	-0.12	0.11
Tests	-0.032	-	0.14	0.06	0.056	0.15	0.28	0.14	0.1	0.27	0.092	0.037	-0.021	0.18	0.039	0.058	-0.032	0.026	0.017	0.31
Übungen	0.14	0.14	-	-0.3	-0.39	-0.3	0.33	-0.17	0.21	0.38	0.38	-0.25	-0.33	-0.072	0.041	-0.024	-0.18	-0.096	-0.034	0.31
Kurzaufgaben	0.067	0.06	-0.3	1	0.54	0.68	0.61	-0.067	0.49	0.43	-0.072	0.12	0.035	0.079	0.17	0.14	0.12	0.15	-0.1	0.61
Beispiele	-0.0058	0.056	-0.39	0.54	1	0.88	0.61	0.094	0.35	0.39	0.12	0.1	0.11	0.3	0.13	0.23	0.2	0.25	-0.0027	0.65
Info	0.0055	0.15	-0.3	0.68	0.88	1	0.77	-0.042	0.43	0.49	0.21	0.15	0.047	0.32	0.19	0.23	0.16	0.23	0.045	0.78
Grund	0.11	0.28	0.33	0.61	0.61	0.77	1	-0.27	0.63	0.76	0.36	-0.017	-0.17	0.21	0.19	0.19	0.058	0.15	-0.051	0.99
Erweitert	-0.021	0.14	-0.17	-0.067	0.094	-0.042	-0.27	1	-0.22	-0.09	-0.0095	0.017	0.13	0.14	0.099	0.13	-0.012	0.082	0.14	-0.11
Cat_0	0.26	0.1	0.21	0.49	0.35	0.43	0.63	-0.22	1	0.62	-0.098	-0.48	-0.63	-0.22	0.22	0.3	0.24	0.31	-0.19	0.61
Cat_1	0.23	0.27	0.38	0.43	0.39	0.49	0.76	-0.09	0.62	1	0.083	-0.43	-0.5	0.0094	0.093	0.047	-0.051	0.0097	-0.045	0.77
Cat_2	-0.075	0.092	0.38	-0.072	0.12	0.21	0.36	-0.0095	-0.098	0.083	1	0.0034	0.014	0.054	0.007	-0.019	-0.11	-0.052	0.37	
Cat_3	-0.031	0.037	-0.25	0.12	0.1	0.15	-0.017	0.017	-0.48	-0.43	0.0034	1	0.74	0.4	0.082	0.082	0.11	0.1	0.11	-0.015
Cat_4	-0.27	-0.021	-0.33	0.035	0.11	0.047	-0.17	0.13	-0.63	-0.5	0.014	0.74	1	0.13	-0.0048	0.023	-0.058	-0.01	0.25	-0.15
Cat_5	-0.21	0.18	-0.072	0.079	0.3	0.32	0.21	0.14	-0.22	0.0094	0.054	0.4	0.13	1	-0.013	-0.13	0.065	-0.06	0.1	0.24
fMean	-0.039	0.039	0.041	0.17	0.13	0.19	0.19	0.099	0.22	0.093	0.007	0.082	-0.0048	-0.013	1	0.74	0.58	0.77	0.3	0.21
f1	0.036	0.058	-0.024	0.14	0.23	0.23	0.19	0.13	0.3	0.047	-0.019	0.082	0.023	-0.13	0.74	1	0.54	0.93	0.11	0.21
f2	0.13	-0.032	-0.18	0.12	0.2	0.16	0.058	-0.012	0.24	-0.051	-0.23	0.11	-0.058	0.065	0.58	0.54	1	0.82	-0.06	0.058
f12	0.081	0.026	-0.096	0.15	0.25	0.23	0.15	0.082	0.31	0.0097	-0.11	0.1	-0.01	-0.06	0.77	0.93	0.82	1	0.048	0.17
f12Amp	-0.12	0.017	-0.034	-0.1	-0.0027	0.045	-0.051	0.14	-0.19	-0.045	-0.052	0.11	0.25	0.1	0.3	0.11	-0.06	0.048	1	-0.029
tot	0.11	0.31	0.31	0.61	0.65	0.78	0.99	-0.11	0.61	0.77	0.37	-0.015	-0.15	0.24	0.21	0.21	0.058	0.17	-0.029	1

# Möglicher Aufbau des Clustering Moduls - Teil II

## clustering

Wendet die cluster Funktionen aus cluster\_fcts.py an

- + Lade transformierte Daten
- + Generiere sekund. Feature
- + Wähle Algorithmus und ggbf. Anzahl Cluster / Dim
- + Wende Funktionen an

-> 1D Histogram oder 2D Scatterplot der Ergebnisse  
-> Speichere Plots als png

## cluster\_eval\_fcts

Evaluiert die Ergebnisse des Clusters mit 3 Methoden:

- + Elbow Plot (über SSE)
  - + Cross Validation (auch gemittelt über n runs)
  - + Silhouette analysis (Distanz der Cluster)
- > Separation der Cluster  
-> "Beste" Anzahl an Cluster  
-> "Bester" Algorithmus

## cluster\_evaluation

Wendet Evaluierungsmethoden aus cluster\_eval\_fcts.py an

- + Wähle Algorithmus, dim, features, ggbf Anzahl Cluster
- + Wähle Eval. - Methode(n)
- + Wende Funktionen an

-> Elbow Plots, Cross-Validation lineplots/heatmaps, Silhouette score barplot/heatmap  
-> Speichere Plots als png

# clustering

## 1) Auswahl der Feature

- a) "learntype" (Tests, Übungen, Kurzaufgaben, Beispiele)
- b) "frequency" (f1, f2, f12, f12Amp)
- c) "level" (Grund, Erweitert)
- d) "tot" (tot)
- e) "Cat" (Cat\_0, Cat\_1, Cat\_2, Cat\_3, Cat\_4, Cat\_5)
- f) Beliebige Kombinationen (selected=...)

## 2) Auswahl der Parameter

- a) n\_cluster (Anzahl der erwarteten Cluster)
- b) dim (Dimension -> Anzahl Spalten gegeneinander)

## 3) Auswahl des Algorithmus

- a) k\_means (einfach, erlaubt elbow\_plot und cross\_valid)
- b) meanshift (bestimmt Anzahl Cluster von selber)
- c) spectral (komplexer, erlaubt kein elbow\_plot)
- d) agglomerative (bestimmt Anzahl Cluster von selber)

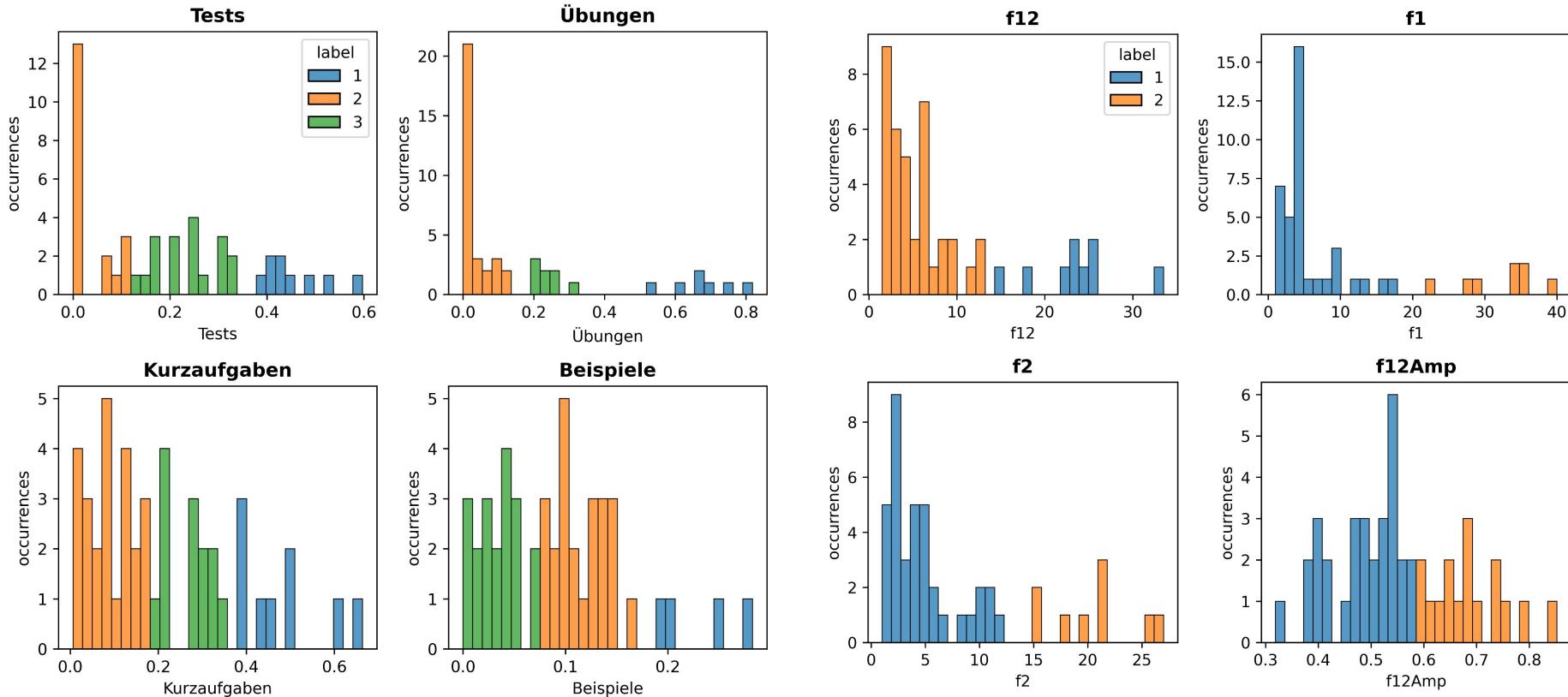
```
from cluster_prep_fcts import load_transformed_data, \
                                add_secondary_features
from cluster_fcts import do_cluster

df = load_transformed_data()
df = add_secondary_features(df)

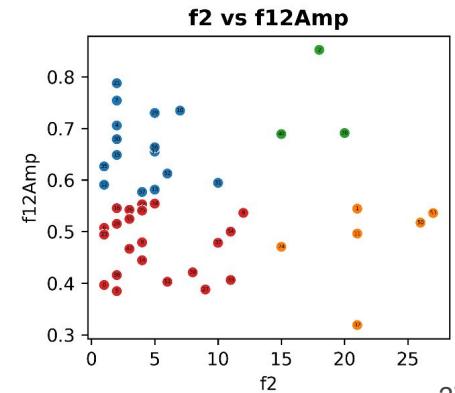
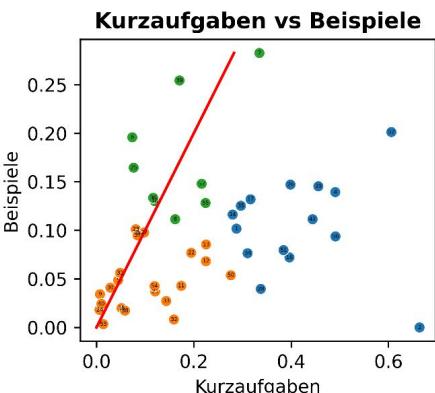
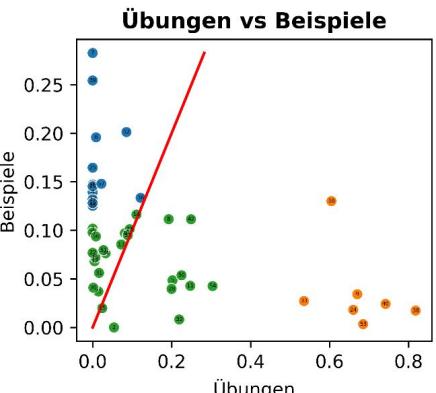
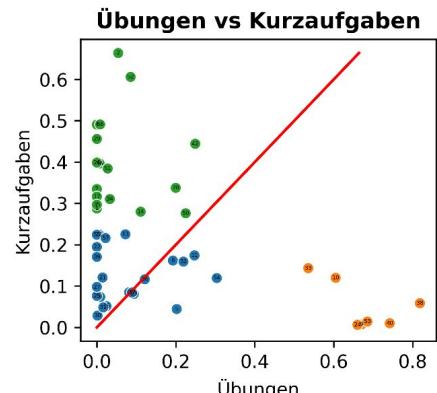
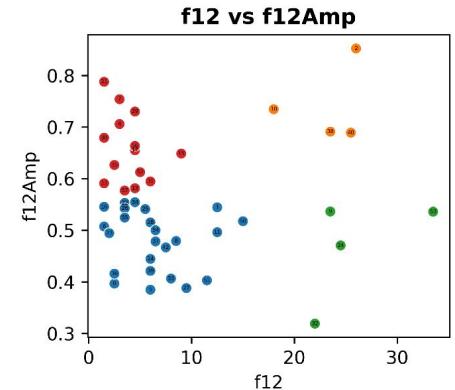
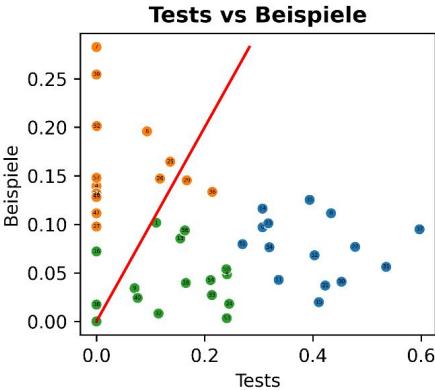
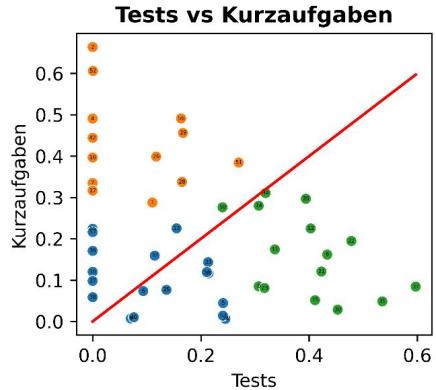
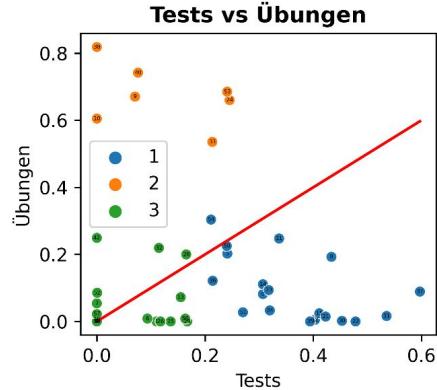
alg = 'agglomerative'

do_cluster(df, "learntype", n_cluster=3, dim=1, alg=alg)
do_cluster(df, "learntype", n_cluster=3, dim=2, alg=alg)
do_cluster(df, "frequency", n_cluster=2, dim=1, alg=alg)
do_cluster(df, "frequency", n_cluster=4, dim=2, alg=alg)
do_cluster(df, "level", n_cluster=3, dim=1, alg=alg)
do_cluster(df, "level", n_cluster=3, dim=2, alg=alg)
do_cluster(df, "tot", n_cluster=3, dim=1, alg=alg)
do_cluster(df, "Cat", n_cluster=3, dim=1, alg=alg)
do_cluster(df, "Cat", n_cluster=3, dim=2, alg=alg)
```

# Darstellung einzelner Feature (Beispiele)



# Darstellung Feature-Paare (Beispiele)



# Möglicher Aufbau des Clustering Moduls - Teil III

## clustering\_hierarchy\_fcts

Funktionen zum Anwenden und Auswerten von hochdimensionalem hierarchischen Clustering:

- + hierarchical\_cluster\_map
  - Erzeugt cluster map ausgewählter Feature basierend auf seaborn builtin
- + cluster\_dendrogram
  - Erzeugt Dendrogram aus cluster map basierend auf scipy builtin
- + learn\_types\_lineplots
  - Plotted die detaillierten Runs aller User gemäß der Einordnung im Dendrogram

## clustering\_hierarchy

Wendet alle clustering\_hierarchy\_fcts an:

- + Feature die mit einbezogen werden
- + Ob runs nach Zeit oder Anzahl der Aktivitäten bewertet werden sollen
- + Erklärungen der Cluster (gezeigt in Plots)
- + Color threshold für das Dendrogram

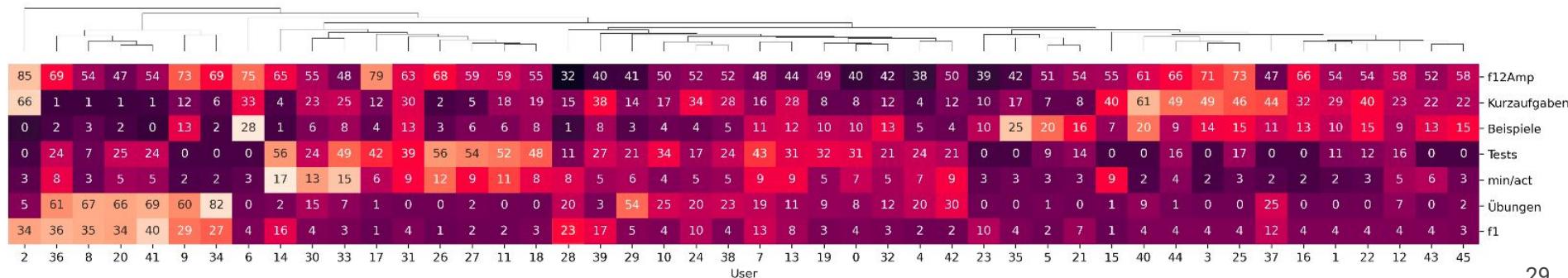
Erzeugt und speichert in results -> hierarchical:

- + Cluster map als jpg
- + Dendrogram als png
- + Learn type lineplots für alle user, eine png für jedes identifizierte Cluster

# Beispiele Ergebnisse Hierarchisches Clustering

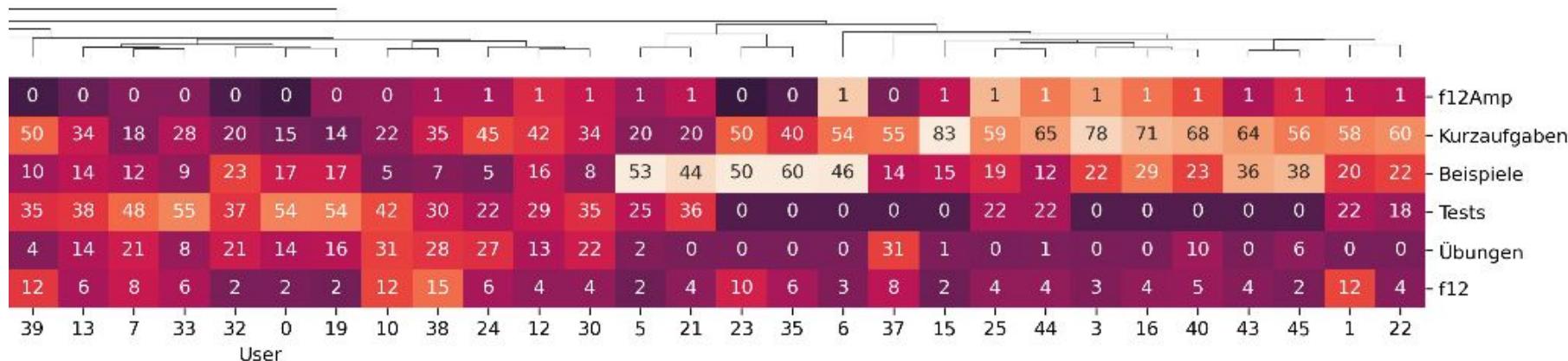
**Mehrere Feature** → In diesem Bsp: Tests, Übungen, Kurzaufgaben, Beispiele, f1, f12Amp, min/act

- In diesem Bsp: **Anzahl Aktivitäten** für Auswertung der **Runs**, nicht die Zeit
- Für **Anteile** der einzelnen **Lerntypen** aber **verbrachte Zeit**, nicht die Anzahl
- Transformation der Daten erfolgt vor Clustering mittels **z-Transformation**
- Es ergibt sich **in diesem Bsp** folgende Cluster Map:



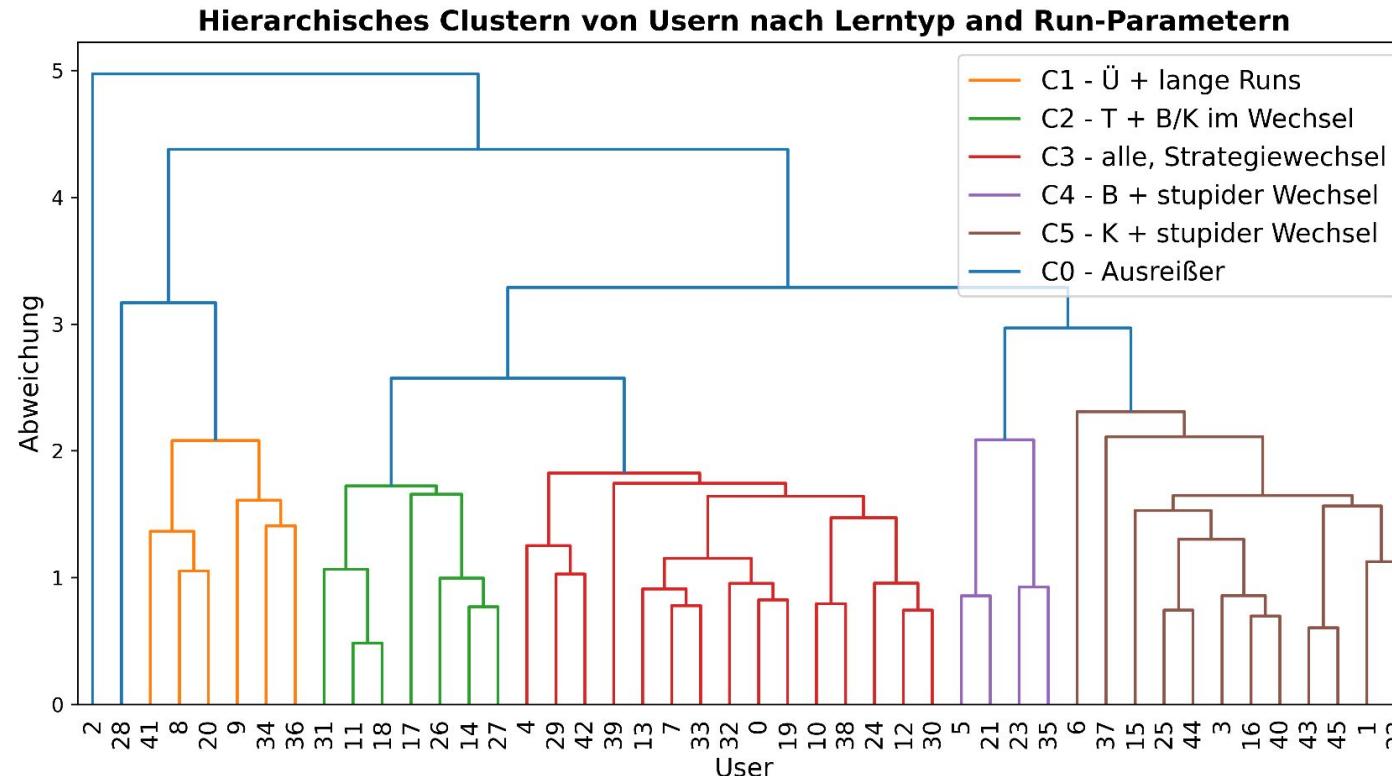
## Darstellung von mehr als zwei Features (Beispiele)

## → Cluster-Map (Seaborn)



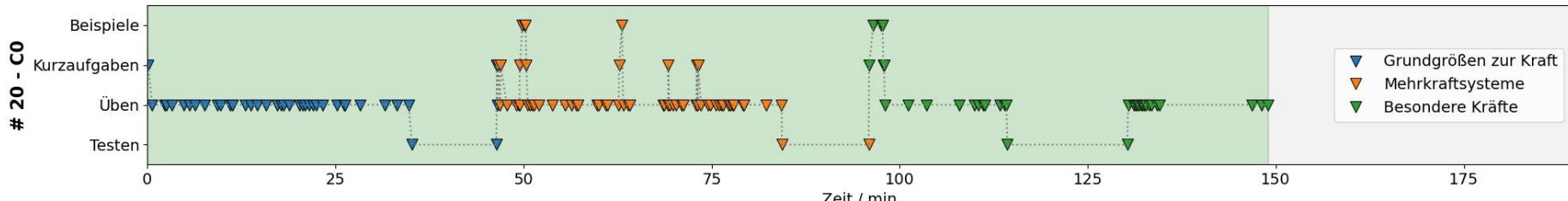
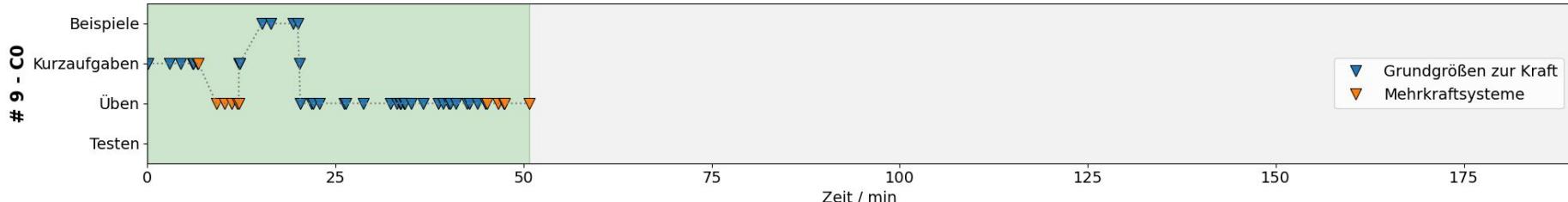
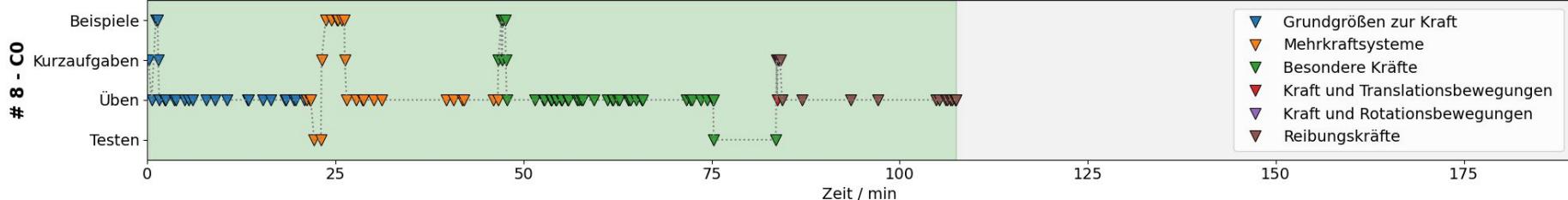
- Ähnliche User nebeneinander angeordnet
  - Farbe zeigt skalierte Ausprägung der Feature
  - Werte sind originale Werte der Feature

→ Zugehöriges Dendrogramm (SciPy)



## → Zugehörige User-Profile (Auszug)

C0



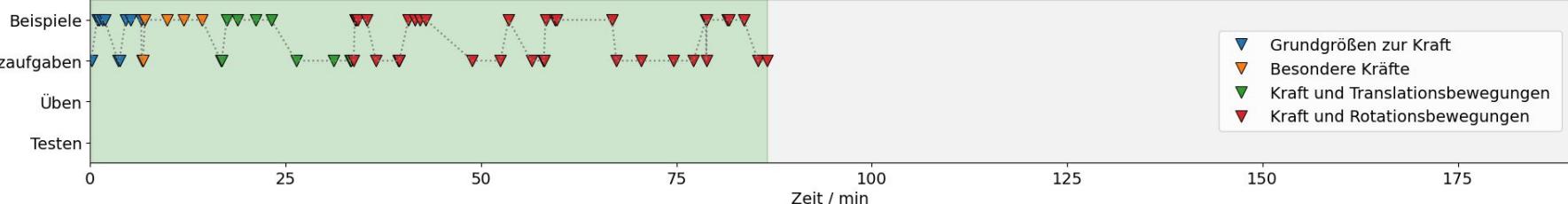
## → Zugehörige User-Profile (Auszug)

C5

# 5 - C5



# 6 - C5

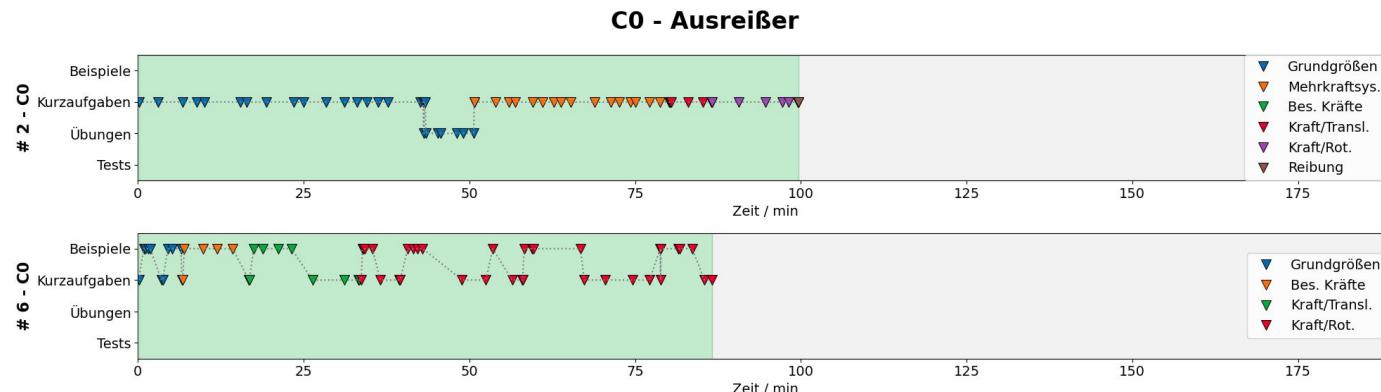


# 21 - C5

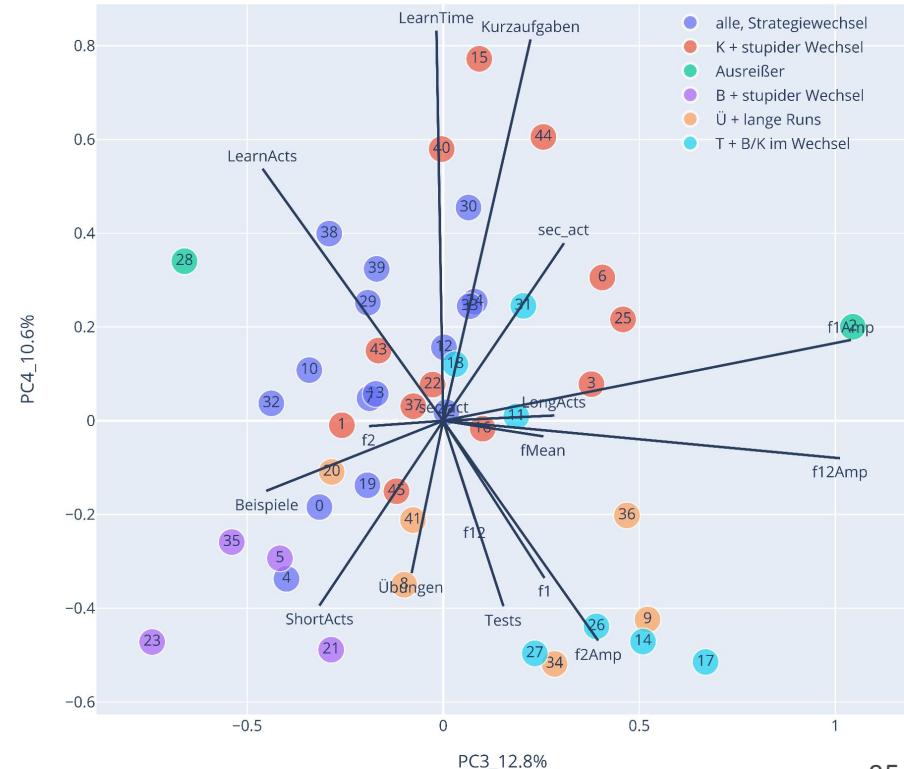
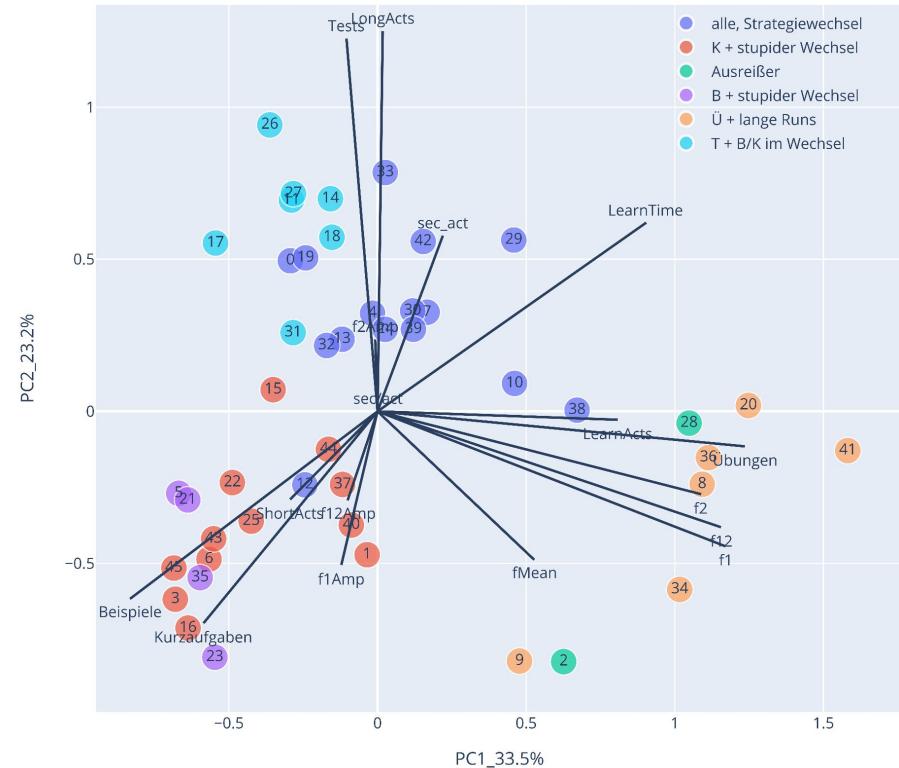


Zur Veranschaulichung werden zu jedem User die runs dargestellt, dabei werden...

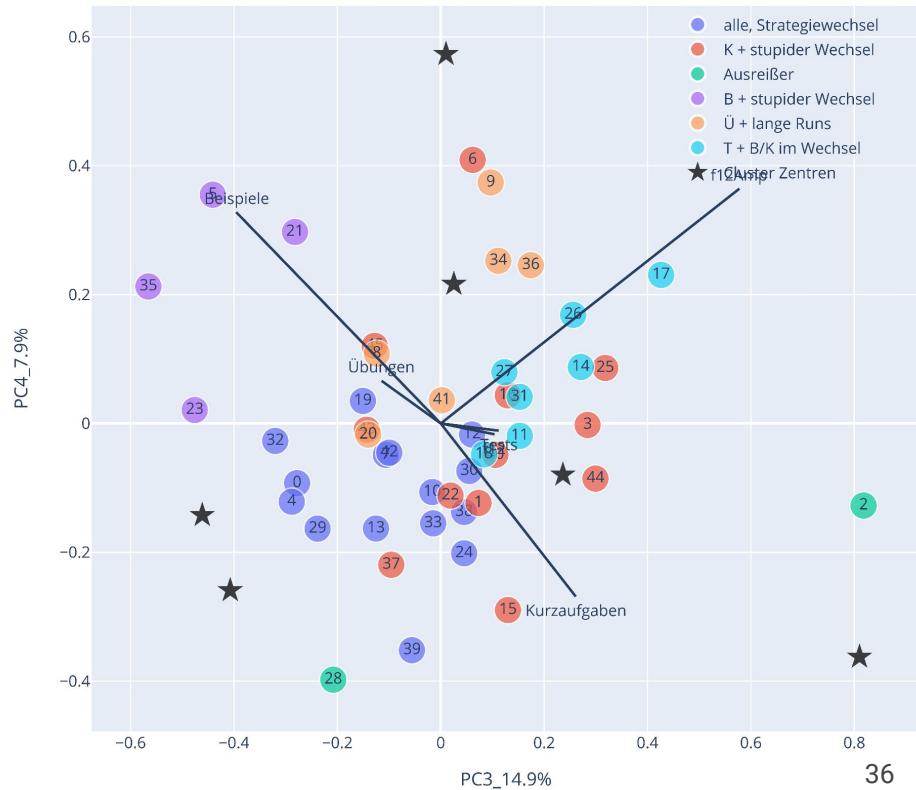
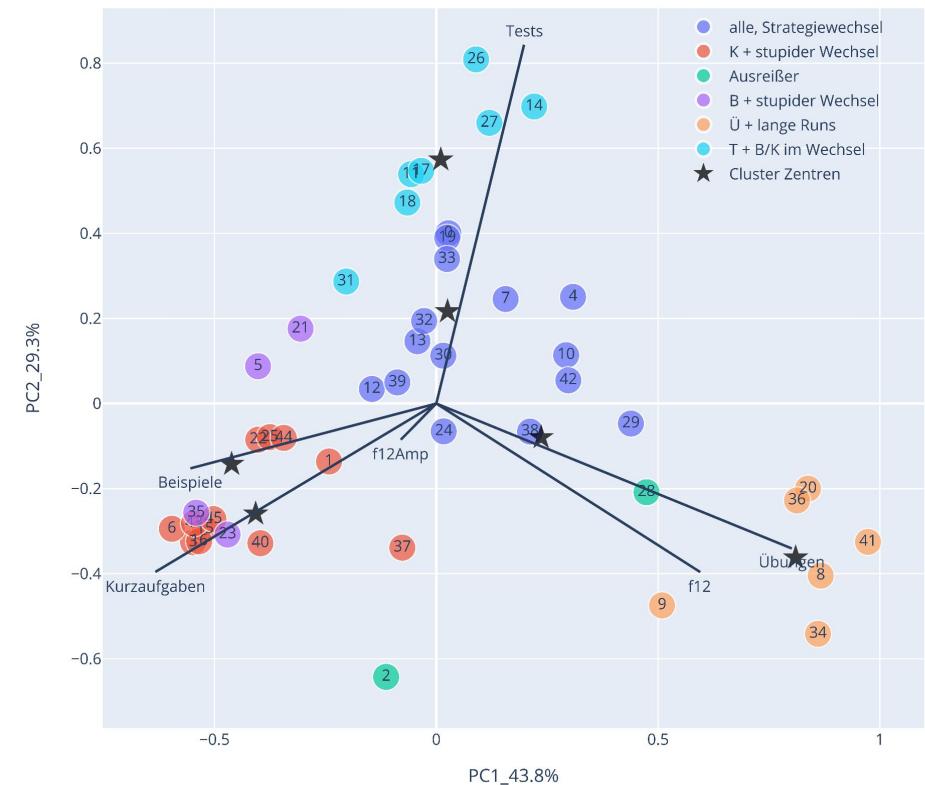
- ... die Plots in der **Reihenfolge** der User im **Dendrogram** erzeugt
- ... die **Cluster** jeweils in **einem Bild** mit Subplots für zugehörige User gezeigt
- ... die **Clusternummer** und zugehörige (manuelle) **Erklärung** als **Titel** gezeigt
- ... **links** neben dem Plot die **User ID** und nochmal die Kategorie angezeigt
- ... auf der **x-Achse** die vom User **verbrachte Zeit** in Lernaktivitäten dargestellt
- ... auf der **y-Achse** die **Lernaktivitäten** gemäß ihrer Einteilung angezeigt
- ... **Aktivitäten** durch einen **Marker** auf der Höhe des Lerntyps visualisiert
- ... die **Themenkategorien** der Aktivitäten durch die **Markerfarbe** codiert
- ... **Level** durch **Hintergrundfarben** markiert (grün → grund / rot → erweitert)



# Verifizieren mit Principal Component Analysis



# Verifizieren mit kMeans Clustering





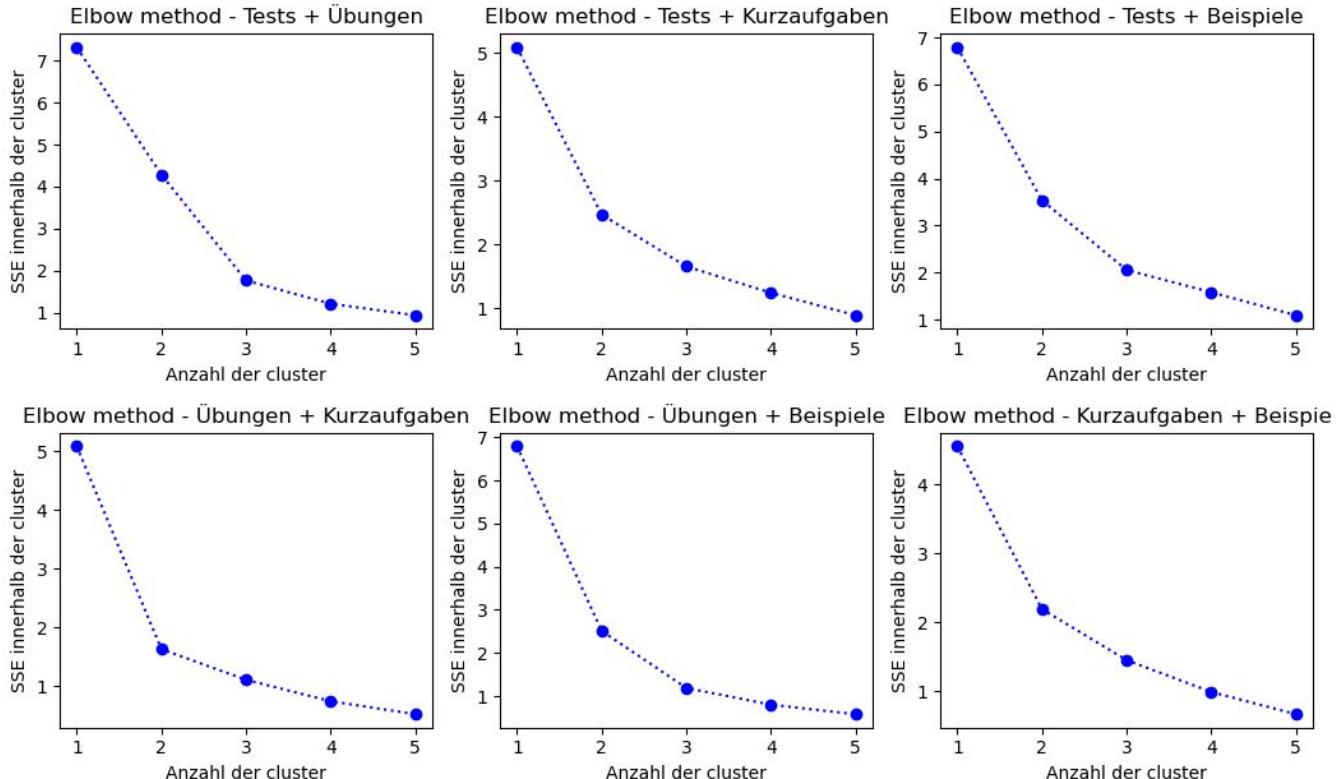
← Übersicht der Cluster-Zentren  
(Farbe → Abweichung in Std.Abw.  
Werte → Originale Feature Werte)



← Abweichung der Cluster voneinander  
(durchschnittliche Abweichungen in  
Std.Abw. pro Feature)

# Validierungsmethoden für Cluster

## Elbow Plot



# Validierungsmethoden für Cluster

→ Silhouette Analysis

