

## Didaktische Überlegungen, Überblick und Infrastruktur

An den Round-Tables in diesem Zoom-Raum wollen wir Ihnen den aktuellen Arbeitsstand im Hinblick auf den Einsatz von Jupyter Notebooks zur Interaktiven, Integrierten Programmierausbildung (Projekt IIP) präsentieren.

### Didaktische Überlegungen:

Der Fokus des Projektes IIP lag bei folgenden Elementen:

- (1) Interaktion zwischen Studierenden und Lehrenden
- (2) Integration der eingesetzten Techniken zur Minimierung der Medienbrüche

Dabei sollte folgender Ansatz verfolgt werden:

- Die Präsentation des Materials in der Vorlesung soll in einer integrierten Form geschehen, in der Begleitmaterial (erklärender Text und Grafiken) wie Code angezeigt, der Code editiert, ausgeführt und animiert werden kann. Durch die einfache Integration und den schnellen Wechsel ist zu erwarten, dass Studierende stärker in die Veranstaltung mit eingebunden werden können.
- Die gleiche Umgebung soll für den Übungsbetrieb eingesetzt werden. Idealerweise soll hier eine Webgestützte Umgebung zum Einsatz kommen, die den Einstieg in das Programmieren möglich macht, ohne sekundäre Tätigkeiten wie Installation von Software vorauszusetzen.
- Zusätzlich soll diese Übungsumgebung auch für Präsenz- und Hausübungsblätter geeignet sein; auch hier ist ein Medienbruch so weit wie möglich zu vermeiden.
- Für die Hausübungsblätter soll ein automatisches Testen der abgegebenen Lösung möglich sein, um die Tutoren von Routinetätigkeit zu entlasten und ihnen zu erlauben, sich auf didaktische Arbeiten zu konzentrieren.



### INTERAKTIVE INTEGRIERTE PROGRAMMIERAUSBILDUNG INTERVIEWS MIT LEHRENDEN

Befragte Personen	<ul style="list-style-type: none"> <li>• Lehrende der Universität Paderborn</li> <li>• Bereich Informatik</li> </ul>
Schwerpunkte	<ul style="list-style-type: none"> <li>• Vor- und Nachteile beim Einsatz</li> <li>• Aufwand, Umsetzbarkeit, Didaktischer Mehrwert</li> </ul>

Vorteile	Nachteile
Browserbasiert	Hoher Einrichtungsaufwand für Lehrende (JupyterHub)
Papierlose Abgabe und automatisierte Bewertung	Erstellen von Vorlesungsmaterialien zeitaufwändig
Literate programming	Probleme mit der Accountverwaltung
Interaktion mit Studierenden	Keine integrierte GIT-Anbindung
Interaktives Skript	Übersichtlichkeit bei großen Programmen geht verloren

Abbildung 1: Ergebnisse der Interviews mit Lehrenden der Universität Paderborn zum Einsatz von Jupyter Notebooks

Um die Projektergebnisse vorzustellen, zeigen wir hier exemplarisch sechs Jupyter Notebooks, die sich in dieser oder leicht abgewandelter Form auch in der Schule einsetzen lassen. Teilweise ist dies in ersten Erprobungen auch bereits geschehen. Sollten Sie genauere Fragen zu einem der Notebooks oder auch im Allgemeinen haben, sprechen Sie uns gerne an.

## Überblick:

Folgende Module möchten wir Ihnen gerne an unserem Round-Table präsentieren:

- **Was Künstliche Intelligenz aus unserer Mediennutzung lernt**
  - Sie können mit diesem Notebook automatisiert Entscheidungsbäume erstellen, die zur Vorhersage von Mediennutzungsverhalten genutzt werden können. Sie können den Erstellungsprozess von Entscheidungsbäumen explorieren, sowie die „gelernten“ Entscheidungsregeln auf ihre Plausibilität prüfen.
- **Visualisierung von Coronadaten**
  - In diesem Jupyter Notebook soll exemplarisch gezeigt werden, wie eine Datenanalyse und Visualisierung aktueller Daten durchgeführt werden kann. Dabei wird der Fokus auf eine aktuelle Thematik, die Coronakrise, gesetzt.
- **Was sagen meine Handydaten über meinen Charakter aus?**
  - In diesem Notebook können Standortdaten auf einer Karte visualisiert und dort exploriert werden. Dazu können verschiedene Filter verwendet werden, um die Standortdaten etwa nach einem Wochentag oder einem Monat zu filtern.
- **Analyse von Umweltdaten:**
  - In diesem Notebook können Sie eigenständig Umweltdaten wie Feinstaubwerte, Temperaturwerte und Luftfeuchtigkeitswerte auswerten und geeignete Visualisierungen erstellen.
- **Schrifterkennung mit Künstlicher Intelligenz**
  - Die Kernidee des Notebooks ist es, den Trainings- und Evaluationsprozess eines Machine Learning (ML) Modells zur Bilderkennung Schritt für Schritt abzubilden. Exemplarisch wird dazu eine Support Vector Machine (SVM) mit Bildern von handgeschrieben Ziffern trainiert, wodurch diese lernt, zwischen den verschiedenen Ziffern zu unterscheiden. Das Gelernte kann anschließend dazu genutzt werden, die Handschrift des Benutzers zu erkennen.

