

VORLESUNG MACHINE LEARNING

Aufgaben und Hinweise zur Projektarbeit

Prof. Dr. Pedram Nazari
Informatik, Software Engineering

Fachbereich Elektro- und
Informationstechnik

Hochschule Düsseldorf

Wichtig:

Lesen Sie das gesamte Dokument sorgfältig durch und klären Sie Unklarheiten frühzeitig.

ÜBERSICHT DER AUFGABEN

Jedes Team muss **alle drei Aufgaben** vollständig bearbeiten (siehe detaillierte Aufgabenstellungen in Moodle):

- **Aufgabe 1:** Bildklassifikation mit CNN und CIFAR-10 Datensatz
- **Aufgabe 2:** Zeitreihenanalyse mit RNN und Jena Climate Datensatz
- **Aufgabe 3:** Automatisierung einer Machine-Learning-Pipeline

WICHTIGE TERMINE

- **bis 19.01.2026** Abgabe der:
 - finalen Präsentationsfolien
 - Code (in GitHub, spätere Änderungen werden nicht berücksichtigt)
 - Die Frist **gilt für ALLE Teams**, unabhängig vom Präsentationstermin
- **am [TODO: wird noch festgelegt]** Präsentation der Ergebnisse sowie Fragen dazu und zu den Vorlesungsthemen
 - Die konkreten Terminslots für jedes Team werden noch mitgeteilt.

PRÄSENTATION

- Vorstellung der Ergebnisse
- 10 Minuten pro Teammitglied
- Gefolgt von einer kurzen Fragerunde (< 10 Minuten pro Teammitglied)
- Inhalte: Präsentieren Sie alle wesentlichen Aspekte des Projekts (siehe Aufgaben)
- **Jedes Teammitglied muss mindestens zwei Themen aktiv (mit-)präsentieren**

- **Vorbereitung der Präsentation**
 - Eigene Laptops sind zur Prüfung mitzubringen
 - Die Geräte müssen vorab vollständig eingerichtet sein
 - Die Präsentation muss bereits vorbereitet sein und direkt startbereit sein
 - Falls eine Demo geplant ist, muss vorab sichergestellt werden, dass alles installiert, konfiguriert und lauffähig ist

BEWERTUNGSGRUNDLAGE

- Die Bewertung setzt sich zusammen aus:
 - Code (20%)
 - Präsentation/Vortrag (30%)
 - Fragen zum Vortrag und Vorlesungsthemen (50%)
 - Bonuspunkte

PROJEKTVORGEHEN UND ZUSAMMENARBEIT

- Kommunikation und Transparenz:
 - Stimmen Sie ab, über welche Kanäle und Tools Sie im Team kommunizieren
 - Stimmen Sie ab, wie Sie den Status und Fortschritt Ihrer Aufgaben tracken (z.B. über Kanban-Boards, Task-Manager)
- Feedbackschleifen:
 - Halten Sie regelmäßige Feedback-Meetings (z.B. Retrospektiven) im Team ab. Reflektieren Sie folgende Fragen:
 - Was lief gut?
 - Was lief nicht gut?
 - Welche Action Items definieren Sie, um Verbesserungen zu erreichen?
 - Dokumentieren Sie, wie Sie diese Action Items in Ihre Projektplanung eingebunden haben.
 - Es finden wöchentliche Feedbackrunde im Plenum statt
- Stimmen Sie sich über eine sinnvolle Aufgabenverteilung im Team ab. Begründen Sie Ihre Entscheidung
 - **Beachten Sie: Jedes Teammitglied muss mindestens zwei Themen aktiv mitpräsentieren.**

TECHNOLOGY-STACK UND TOOLS

- Verwenden Sie eine Plattform zur Versionsverwaltung (z.B. GitHub oder Azure DevOps)
- Wählen Sie einen geeigneten Technologie-Stack für Ihr Thema aus und begründen Sie Ihre Wahl
 - z.B. Python, TensorFlow, PyTorch etc.
- Verwenden Sie Tools zur Automatisierung und zum (Team-)Management Ihrer Aufgaben
 - z.B. Automatisierte Modelltraining-Workflows (z. B. GitHub Actions für Training-Pipelines), Kanban Boards

QUELLEN

- Verwenden Sie eine Vielfalt an Quellen (Vorlesungen, Bücher, Artikel, Websites, Videos, KI-Tools etc.), um sich fachlich zu informieren und unterschiedliche Perspektiven kennenzulernen.

WICHTIGE HINWEISE

- Es dürfen **keine sensiblen oder persönlichen Daten** in Commits, Dokumentationen, Quellcode, Konfigurationsdateien, Datenbanken etc. gespeichert oder verwendet werden.
- Wenn es nicht gewünscht ist, dass GitHub oder andere verwendete Tools/Services den echten Namen oder die echte E-Mail-Adresse kennen, kann eine neue, neutrale E-Mail-Adresse erstellt und verwendet werden.
- **Zur Erfüllung der Aufgaben reichen kostenfreie Tools aus.** Ihr müsst keine kostenpflichtigen Tools nutzen. Falls ihr dennoch kostenpflichtige Tools verwenden möchtet oder Kosten durch die Überschreitung des Free-Tiers entstehen könnten (insbesondere bei Cloud-Diensten), achtet eigenverantwortlich darauf, dass keine zusätzlichen Kosten anfallen.
- **Umgang mit Lizenzen:** Verwendet nur Libraries und Frameworks mit geeigneter Lizenz (z. B. MIT, Apache 2.0, BSD). Baut keinen Code ein, der nicht zur Wiederverwendung freigegeben ist. Fragt im Zweifel nach, wenn Lizenzbedingungen unklar sind.
- **Sicherheit und Zugangsdaten:** Speichert keine API-Keys, Passwörter oder sonstige Zugangsdaten im Quellcode, in Commits oder in öffentlichen Repositories. Nutzt geeignete Mechanismen wie Umgebungsvariablen oder Secret-Management, falls Zugangsdaten erforderlich sind.
- **Bei Fragen oder Unklarheiten wendet euch bitte an den Professor!**