SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2026/2027 - 2029/2030

Rok akademicki 2029/2030

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	*wdrażanie i utrzymanie modeli AI*
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	*Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych*
Nazwa jednostki realizującej przedmiot	*Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych*
Kierunek studiów	*sztuczna inteligencja*
Poziom studiów	*studia I stopnia*
Profil	*ogólnoakademicki*
Forma studiów	*stacjonarne*
Rok i semestr/y studiów	*rok IV, semestr 7*
Rodzaj przedmiotu	*przedmiot kierunkowy*
Język wykładowy	*polski*
Koordynator	*mgr inż. Jarosław Szkoła*
Imię i nazwisko osoby prowadzącej / osób prowadzących	*mgr inż. Jarosław Szkoła*

***** - *opcjonalnie, zgodnie z ustaleniami w Jednostce*

1.1.Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	**Liczba pkt. ECTS**
7	10			15		15			3

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

wykład – zaliczenie bez oceny, laboratoria – zaliczenie z oceną, zaj. projekt. – zaliczenie z oceną,

2.WYMAGANIA WSTĘPNE

Umiejętność obsługi systemów operacyjnych Linux / Windows na podstawowym poziomie, znajomość języków programowania C/C++, Python na podstawowym poziomie.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1 Cele przedmiotu

C1	Rozumie pełny cykl życia otwartych modeli językowych (LLM) – od uruchomienia i konfiguracji modelu, przez wdrożenie, aż po jego długoterminowe utrzymanie w środowisku produkcyjnym.
C1	Potrafi wdrażać i serwować modele językowe typu open-source z wykorzystaniem wyłącznie darmowego oprogramowania oraz lokalnej lub uczelnianej infrastruktury obliczeniowej.
C3	Umie projektować architekturę systemów opartych o LLM, w tym systemów z API oraz rozwiązań typu Retrieval-Augmented Generation (RAG).
C4	Potrafi utrzymywać i aktualizować wdrożone modele AI, realizując wymianę wersji modelu, testy regresji jakości oraz procedury rollbacku.
C5	Rozumie zagadnienia bezpieczeństwa i odpowiedzialnego wykorzystania modeli językowych, w szczególności ryzyka związane z prompt injection, ochroną danych oraz audytem systemów AI.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych [1]
EK_01	Posiada wiedzę w zakresie doboru odpowiednich narzędzi pozwalających na realizację wybranego projektu interdyscyplinarnego związanego z zastosowaniem sztucznej inteligencji.	K_W08
EK_02	Posiada wiedzę w zakresie utrzymania, wdrażania oraz zapewnienia bezpieczeństwa dla systemów informatycznych i modeli AI.	K_W09
EK_03	Potrafi dobrać odpowiednie parametry treningowe, metody walidacji, metodę oceny jakości modeli AI dla konkretnego problemu badawczego lub inżynieryjnego.	K_U05
EK_04	Potrafi zaprojektować i wdrożyć odpowiedni system wspomagania decyzji, sterowania lub diagnostyki z zastosowaniem AI.	K_U08
EK_05	Potrafi zabezpieczyć dane i systemy sztucznej inteligencji przed nieuprawnionym dostępem.	K_U14
EK_06	Jest gotów do przyjmowania nowej wiedzy, krytycznego oceniania swoich kompetencji, podejmowania konsultacji z ekspertami; przy wdrożeniu i utrzymaniu modeli AI działa przedsiębiorczo, szukając opłacalnych i społecznie odpowiedzialnych zastosowań.	K_K01, K_K03

3.3 Treści programowe

A. Problematyka wykładu

Wprowadzenie do open-source LLM i MLOps

[1] W przypadku ścieżki kształcenia prowadzącej do uzyskania kwalifikacji nauczycielskich uwzględnić również efekty uczenia się ze standardów kształcenia przygotowującego do wykonywania zawodu nauczyciela.

Architektura systemów opartych o LLM
Przygotowanie LLM do wdrożenia
Serwowanie modeli językowych
Konteneryzacja i automatyzacja
Monitoring i utrzymanie LLM
Bezpieczeństwo i odpowiedzialne użycie LLM

B. Problematyka laboratoriów

Uruchomienie open-source LLM lokalnie
API dla modelu językowego
RAG – połączenie LLM z wiedzą zewnętrzną
Konteneryzacja systemu LLM
Monitoring i logowanie
Utrzymanie i aktualizacja modelu

C. Problematyka zajęć projektowych

Asystent wiedzy dla organizacji (RAG)
Lokalny chatbot helpdesk (bez chmury)
System analizy i streszczania dokumentów
Asystent programisty oparty o lokalny LLM
System klasyfikacji i tagowania treści
Analiza zgłoszeń i ticketów (IT / support)
System wykrywania problemów jakościowych (feedback)
Asystent do przeglądu regulaminów i umów
System edukacyjny oparty o LLM
Monitor jakości odpowiedzi LLM

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną
Laboratorium: praca w grupie – dyskusja problemów, rozwiązywanie zadań; praca indywidualna – rozwiązywanie zadań
Zajęcia projektowe: zespoły 2-3 osobowe (opcjonalnie 1-osobowe), praca iteracyjna

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	kolokwium	w
EK_02	kolokwium	w
EK_03	obserwacja w trakcie zajęć, sprawozdania, projekt	lab, zp
EK_04	obserwacja w trakcie zajęć, sprawozdania, projekt	lab, zp
EK_05	obserwacja w trakcie zajęć, sprawozdania, projekt	lab, zp
EK_06	obserwacja w trakcie zajęć	lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Zaliczenie wykładu (EK_01 i EK_02) uzyskuje się na podstawie kolokwium, które realizowane jest w formie testu. Próg zaliczenia zostaje ustalony na 50%.

Zaliczenie laboratorium (EK_03 - EK_05) odbędzie się na podstawie wykonanych sprawozdań. Wymagane jest wykonanie co najmniej 50% ćwiczeń w celu zaliczenia przedmiotu.

Progi zaliczenia laboratorium:
Ocena 3.0:

- potrafi uruchomić lokalnie wybrany model językowy
- potrafi zapewnić działającą końcówkę API do generowanie odpowiedzi,
- potrafi wykonywać podstawowe testy działania systemu

Ocena 3.5:

- spełnia wymogi oceny 3.0
- integruje prostą bazę wiedzy RAG lub mechanizm kontekstowy
- obsługuje błędy wejścia / wyjścia (walidacja promptu)
- potrafi opisać ograniczenia zastosowanego modelu

Ocena 4.0:

- spełnia wymogi oceny 3.5
- konteneryzuje rozwiązanie (Docker)
- implementuje logowanie zapytań i odpowiedzi
- mierzy czas odpowiedzi (latency)

Ocena 4.5:

- spełnia wymogi oceny 4.0
- implementuje monitorowanie zasobów (CPU/RAM/VRAM)
- porównuje co najmniej dwie konfiguracje lub modele
- potrafi wymienić model bez przestoju (hot-swap lub restart kontrolowany)

Ocena 5.0:

- spełnia wymogi oceny 4.5
- implementuje procedurę przywracania (rollback`u) modelu
- stosuje testy regresji jakości odpowiedzi
- identyfikuje i ogranicza ryzyka bezpieczeństwa (np. prompt injection)
- analizuje koszty obliczeniowe i skalowalność

Zaliczenie efektu EK_06 następuje na podstawie aktywności w trakcie zajęć.

Zaliczenie zajęć projektowych (EK_03 - EK_05) odbywa się na podstawie dostarczonego i zaprezentowanego w działaniu kompletnego systemu AI opartego o otwarty model językowy, działającego jako usługa (API), z monitoringiem i dokumentacją techniczną.

Progi zaliczenia projektu:

Ocena 3.0:

- realizuje definiowany problem projektowy
- uruchamia działający system oparty o otwarty model językowy LLM

- udostępnia API do obsługi zapytań
- projekt spełnia minimalne wymagania funkcjonalne
- repozytorium zawiera kod i podstawową instrukcję uruchomienia

Ocena 3.5:
- spełnia wymagania na ocenę 3.0
- implementuje RAG lub inny mechanizm rozszerzenia kontekstu
- poprawnie obsługuje błędy i przypadki brzegowe
- potrafi uzasadnić wybór modelu i narzędzi
- dokumentacja obejmuje opis architektury

Ocena 4.0:
- spełnia wymagania na ocenę 3.5
- dostarcza konteneryzację (Docker)
- implementuje logowanie i podstawowy monitoring
- projekt jest modularny i czytelnie zaprojektowany
- system można łatwo uruchomić w nowym środowisku

Ocena 4.5:
- spełnia wymagania na ocenę 4.0
- realizuje porównanie do co najmniej dwóch modeli lub konfiguracji
- implementuje procedurę aktualizacji modelu
- analizuje wydajność i zużycie zasobów
- dokumentuje decyzje projektowe i kompromisy

Ocena 5.0
- spełnia wymagania na ocenę 4.5
- implementuje procedurę przywracania (rollback`u) modelu
- przeprowadza testy regresji jakości odpowiedzi
- identyfikuje i ogranicza ryzyka bezpieczeństwa (np. prompt injection, data leakage)
- analizuje skalowalność i koszty obliczeniowe

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny z harmonogramu studiów	40
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	2
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	33
SUMA GODZIN	75
SUMARYCZNA LICZBA PUNKTÓW ECTS	**3**

** Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.*

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	–
zasady i formy odbywania praktyk	–

7. LITERATURA

Literatura podstawowa:

1. Chip Huyen, Designing Machine Learning Systems, O'Reilly Media, 2022
2. Mark Treveil, Nicolas Omont et al. Introducing MLOps, O'Reilly Media, 2020
3. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, Machine Learning with PyTorch and Scikit-Learn, Packt Publishing, 2022
4. Harrison Chase, LangChain for LLM Application Development (dostępne materiały online / książka) https://learn.deeplearning.ai/courses/langchain/lesson/u9olq/introduction
5. Dokumentacja projektu llama.cpp / Ollama https://github.com/ggml-org/llama.cpp, https://ollama.com/

Literatura uzupełniająca:

1. Eugene Yan, Applied Machine Learning Engineering 2023
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly Media
3. Shreya Shankar et al., Operationalizing Machine Learning (materiały online, Stanford)
4. Lewis Tunstall, Leandro von Werra, Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Media, 2022

Literatura techniczna:

1. FastAPI – dokumentacja
2. Docker – dokumentacja
3. FAISS / Chroma – dokumentacja
4. Prometheus & Grafana – dokumentacja
5. Git – dokumentacja