



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Narzędzia uczenia maszynowego [S2SI1E>NUM]

### Przedmiot

Kierunek studiów

Sztuczna inteligencja/Artificial Intelligence

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obieralny

### Liczba godzin

Wykład

15

Laboratorium

15

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

### Liczba punktów ECTS

3,00

### Koordynatorzy

dr hab. inż. Mikołaj Morzy prof. PP  
mikolaj.morzy@put.poznan.pl

### Wykładowcy

### Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę w zakresie uczenia maszynowego i przetwarzania danych. Powinien posiadać umiejętność rozwiązywania podstawowych problemów w zakresie projektowania systemów informatycznych i ich realizacji (wersjonowanie, testowanie, integracja kodu komputerowego). Wymagana jest podstawowa znajomość języka programowania Python. Student powinien posiadać umiejętność korzystania z zewnętrznych API programistycznych. Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu. W zakresie kompetencji społecznych student musi prezentować takie postawy jak odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi, umiejętność pracy grupowej.

## Cel przedmiotu

Głównym celem przedmiotu jest zaprezentowanie bogatego zestawu narzędzi koniecznych do praktycznego wdrażania rozwiązań informatycznych wykorzystujących techniki uczenia maszynowego. W trakcie wykładów studenci zapoznają się z technikami i narzędziami produktyzacji modeli statystycznych, w szczególności z narzędziami wykorzystywanymi do włączania opracowywanych modeli do istniejącej infrastruktury informatycznej. W trakcie laboratoriów studenci zapoznają się praktycznie z omawianymi narzędziami. Tematyka poruszana podczas zajęć obejmuje: • wersjonowanie danych i modeli statystycznych • narzędzia do zarządzania przepływem pracy (ang. workflow) dla uczenia maszynowego • narzędzia do adnotacji danych na potrzeby zbiorów uczących • narzędzia do monitorowania procesu trenowania modeli • praktyczne aspekty produktyzacji modeli statystycznych • narzędzia do zarządzania projektem uczenia maszynowego

## Przedmiotowe efekty uczenia się

### Wiedza

Student posiada uporządkowaną i pogłębioną wiedzę w zakresie uczenia maszynowego i praktycznych aspektów wdrażania rozwiązań wykorzystujących uczenie maszynowe [K2st\_W1]

Student posiada wiedzę na temat dobrych praktyk związanych z rozwojem i praktycznym wdrażaniem rozwiązań uczenia maszynowego w systemach informatycznych, w szczególności, na temat potrzeby testowania i weryfikacji modeli statystycznych [K2st\_W2].

Student posiada szczegółową wiedzę na temat procesu kolekcji, adnotacji i wersjonowania danych na potrzeby uczenia modeli statystycznych [K2st\_W3].

Student jest świadomy najnowszych narzędzi, bibliotek i architektur szkieletowych dostępnych na rynku w obszarze wdrażania i pielęgnacji systemów wykorzystujących uczenie maszynowe [K2st\_W4].

Student rozumie pełny cykl życia systemu informatycznego wykorzystującego narzędzia uczenia maszynowego, potrafi ocenić poprawność cyklu i zidentyfikować nietrywialne związki jakie występują między poszczególnymi etapami cyklu (zbieranie i adnotacja danych, trenowanie modeli statystycznych, walidacja i optymalizacja modeli statystycznych, monitorowanie modeli statystycznych) [K2st\_W5].

Student zna podstawowe narzędzia inżynierskie stosowane przy projektowaniu złożonych systemów informatycznych i potrafi zastosować te narzędzia do specyfikacji projektowania systemów wykorzystujących uczenie maszynowe [K2st\_W6].

### Umiejętności

Student potrafi korzystać z różnorodnych API i dokumentacji złożonych systemów informatycznych [K2st\_U1].

W trakcie projektowania systemu informatycznego wykorzystującego narzędzia uczenia maszynowego student potrafi zaprojektować i przeprowadzić metodologicznie poprawny eksperyment pomiarowy i zinterpretować jego wyniki [K2st\_U3].

Umie wykorzystać narzędzia wizualizacji do monitorowania procesu uczenia modeli statystycznych [K2st\_U4].

Student potrafi zastosować zwinne metodyki programowania do zarządzania projektem, którego głównym członem są modele statystyczne. Rozumie konieczność włączenia wiedzy dziedzinowej do procesu projektowania rozwiązań wykorzystujących uczenie maszynowe [K2st\_U5].

Student potrafi budować systemy informatyczne wykorzystujące narzędzia uczenia maszynowego w oparciu o technikę konteneryzacji umożliwiającą włączanie nowych narzędzi i rozwiązań w sposób płynny [K2st\_U6].

Umie oszacować czasochłonność poszczególnych etapów budowy systemu informatycznego opartego na narzędziach uczenia maszynowego [K2st\_U7].

Student rozumie naturę długu technologicznego generowanego przez techniki uczenia maszynowego i potrafi wykorzystać narzędzia informatyczne do przeciwdziałania negatywnym skutkom, jakie błędy w projekcie procesu uczenia modelu statystycznego mogą wywierać na resztę systemu [K2st\_U8].

W trakcie projektowania systemu wykorzystującego narzędzia uczenia maszynowego student potrafi przeanalizować dostępne dane pod kątem ukrytych obciążeń i korelacji. Umie przeanalizować cykl życia danych i dostrzec zagrożenia dla procesu uczenia modelu [K2st\_U9].

Student potrafi zaprojektować i zaimplementować rozwiązanie konkretnego problemu ekonomicznego, technologicznego lub społecznego przy użyciu narzędzi uczenia maszynowego [K2st\_U10].

### Kompetencje społeczne

Student rozumie niezwykle dynamiczny charakter obszaru uczenia maszynowego i jest świadomy mnogości dostępnych narzędzi [K2st\_K1].

Rozumie konieczność kształcenia się w obszarze narzędzi ze względu na szybki cykl ich wymiany [K2st\_K2].

Potrafi komunikować się w grupie i współpracować w ramach zwinnych metodyk rozwoju oprogramowania [K2st\_K3].

## Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Metoda zaliczenia kursu opiera się na ocenie ciągłej, realizowanej w postaci pięciu mini-projektów (oceny binarne: zaliczone / niezaliczone), w których studenci pogłębiają wiedzę w wybranych obszarach MLOps. Studenci mają pełną swobodę w doborze technologii w danym obszarze, na przykład w wyborze dostawcy chmurowego (np. AWS vs. GCP), i są gorąco zachęceni do włączenia tych mini-projektów w swoje prace magisterskie lub większe projekty realizowane w ramach innych przedmiotów. Ostateczna ocena końcowa kursu jest wspólna zarówno dla części wykładowej, jak i laboratoryjnej, i jest ustalana na podstawie pomyślnego ukończenia tych mini-projektów.

## Treści programowe

Ogólne implikacje i zastosowania narzędzi uczenia maszynowego są przedstawiane podczas wykładów. Laboratoria prezentują szczegółowe studia przypadków użycia narzędzi na danym etapie rozwoju systemu opartego na uczeniu maszynowym.

Tematyka poruszana na wykładach obejmuje:

- dług techniczny i jego konsekwencje
- przepływy pracy (workflow) w ML
- uniwersalne platformy dla ML
- przygotowanie, wdrażanie i monitorowanie rozwiązań ML
- najlepsze i najgorsze praktyki MLOps
- LLMOps

Tematyka poruszana na laboratoriach obejmuje:

- krótkie przypomnienie Pytorch
- Pytorch Lightning; monitorowanie modeli; optymalizacja hiperparametrów
- udostępnianie i wdrażanie modeli oraz dockerizacja
- wdrażanie modeli w środowiskach chmurowych
- bezserwerowe wdrażanie w chmurze
- optymalizacja predykcji
- duże modele językowe (LLM) i systemy RAG

## Tematyka zajęć

Kurs wprowadza w praktyczne i teoretyczne aspekty rozwoju nowoczesnych systemów uczenia maszynowego, koncentrując się na integracji współczesnych narzędzi i frameworków. Wykłady dostarczają ogólnego przeglądu implikacji i metodyk narzędzi uczenia maszynowego, natomiast sesje laboratoryjne obejmują studia przypadków ilustrujące ich zastosowanie na różnych etapach rozwoju.

Wykłady wprowadzają w teoretyczne i strategiczne aspekty budowania i zarządzania nowoczesnymi systemami uczenia maszynowego (ML) w środowiskach produkcyjnych, koncentrując się na metodykach służących efektywnej i skalowalnej integracji modeli ML z szerszymi systemami IT. Kluczowe tematy rozpoczynają się od koncepcji długu technicznego specyficznego dla ML, obejmującego ukryte koszty i strategię minimalizowania obciążenia związanego z utrzymaniem złożonych systemów. Omówimy przepływy pracy ML, szczegółowo przedstawiając cykl życia modelu – od gromadzenia danych po ciągłe monitorowanie – z naciskiem na CI/CD/CT i narzędzia do orkiestracji przepływów. Omówione zostaną praktyczne aspekty przygotowania, wdrażania i monitorowania, w tym bezpieczne strategie wdrażania (Canary, testy A/B) oraz rozróżnienie między metrykami monitorowania operacyjnego a metrykami specyficznymi dla ML, takimi jak dryf danych i dryf modelu. Na koniec omawiamy najlepsze i najgorsze praktyki operacji ML, kładąc nacisk na odtwarzalność i odpowiedzialną AI, a także podsumowujemy przeglądem LLMOps, obejmujących unikalne wyzwania skalowania, monitorowania i adaptacji LLM-ów w środowisku produkcyjnym za pomocą technik takich jak fine-tuning i Prompt Engineering.

Część laboratoryjna kursu obejmuje przypomnienie podstaw PyTorch i jego ustrukturyzowanego

rozszerzenia, PyTorch Lightning, wraz z metodami monitorowania modeli i optymalizacji hiperparametrów. Porusza kwestie udostępniania i wdrażania modeli, w tym konteneryzacji za pomocą Dockera oraz wdrożenia w infrastrukturze chmurowej, takiej jak AWS EC2 i AWS Lambda. Dalsze tematy obejmują optymalizację wnioskowania (inference optimization), wykorzystanie dużych modeli językowych oraz RAG.

## Metody dydaktyczne

Wykład: prezentacja multimedialna, seminarium z prezentacjami studentów, wyszukiwanie informacji

Laboratorium: przykłady programistyczne, nieduże ćwiczenia samodzielne

Projekt: rozwiązanie praktycznego problemu, praca zespołowa, design thinking, dokumentowanie

## Literatura

### Podstawowa

1. Treveil, Mark, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, and Lynn Heidmann. *Introducing MLOps*. O'Reilly Media, 2020.
2. Gift, N., & Deza, A. (2021). *Practical MLOps*. O'Reilly Media, Inc.
3. Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.
4. Raschka, Sebastian, and Vahid Mirjalili. *Python machine learning*. No. 1. Packt Publishing,, 2019.

### Uzupełniająca

1. Pustejovsky, James, and Amber Stubbs. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc., 2012.
2. Zheng, Alice, and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018.

## Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	45	1,50