



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Narzędzia uczenia maszynowego

### Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Sztuczna inteligencja

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/1

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obieralny

### Liczba godzin

Wykład

15

Ćwiczenia

Laboratoria

15

Projekty/seminaria

Inne (np. online)

### Liczba punktów ECTS

3

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Mikołaj Morzy, prof. PP

email: [mikolaj.morzy@cs.put.poznan.pl](mailto:mikolaj.morzy@cs.put.poznan.pl)

tel. 61 665-3420

Instytut Informatyki, Wydział Informatyki

i Telekomunikacji

ul. Piotrowo 2, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

### 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 produktywizacji 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 produktywizacji 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 specyfiki 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:

Wiedza nabyta podczas wykładów jest weryfikowana na podstawie raportu przygotowanego przez studentów indywidualnie lub w parach. Raport dotyczy wybranego narzędzia z obszaru uczenia maszynowego i jest prezentowany w formie seminaryjnej podczas ostatniego wykładu.

Wiedza nabyta podczas laboratoriów jest weryfikowana na bieżąco poprzez realizację krótkich samodzielnych ćwiczeń programistycznych dotyczących prezentowanych zagadnień. Dodatkowo, na



koniec ćwiczeń studenci przygotowują jeden projekt grupowy w grupach od 2 do 4 osób. Projekt przedstawia cały cykl wytworzenia systemu informatycznego wykorzystującego narzędzia uczenia maszynowego.

### Treści programowe

W trakcie wykładu prezentowane są bardziej ogólne ramifikacje poszczególnych narzędzi. Laboratoria poświęcone są szczegółowemu praktycznemu przedstawieniu wybranego narzędzia w określonym etapie procesu wytwarzania oprogramowania z użyciem uczenia maszynowego. Tematyka przedmiotu obejmuje m.in.:

- wersjonowanie danych
- ręczna i programistyczna adnotacja danych, w tym danych tekstowych i graficznych
- generacja danych syntetycznych
- repozytoria cech
- narzędzia do wizualizacji procesu uczenia modeli statystycznych
- narzędzia do konteneryzacji
- architektury szkieletowe do zarządzania przepływami pracy (ang. ML workflows)
- uniwersalne platformy uczenia maszynowego

### 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. Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.
2. 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	3,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,5
Praca własna studenta (studia literaturowe, przygotowanie do seminarium, wykonanie projektu) <sup>1</sup>	45	1,5

<sup>1</sup> niepotrzebne skreślić lub dopisać inne czynności