



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Narzędzia uczenia maszynowego [S2Inf1-SzInt>NUM]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/1

Studia w zakresie (specjalność)
Sztuczna inteligencja

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
16

Laboratorium
16

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

2,00

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 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 wykorzystywać 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:

Studenci otrzymują łączną ocenę z wykładu i laboratorium. Kryteria oceniania pracy laboratoryjnej są następujące:

Realizacja 4 mini-projektów w trakcie semestru:

- Pytorch Lightning; Model monitoring; Hyper-parameter optimization (1 pkt, 1 pkt bonus)
- Model serving and deployment (0.5 pkt)
- Serverless deployment on AWS platform as an example (1 pkt, 1 pkt bonus)
- Inference Optimization (1.5 pkt)
- Basics of RAGs (0.5 pkt)
- 5 minut prezentacji nt. narzędzia które nie było prezentowane na zajęciach (1 pkt)

Wymagany próg zaliczenia przedmiotu to 3 punkty.

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.:

- cykl wytwarzania aplikacji wykorzystujących modele statystyczne
- pojęcie długu technologicznego
- antywzorce MLOps
- najczęściej używane narzędzia MLOps (Pytorch, Pytorch Lightning, Docker, Optuna, format ONNX, AWS SageMaker, AWS Lambda, AWS EC2, Google Compute Engine, pakiet transformers, pakiet sentence-transformers, pakiet timm)
- standaryzacja pętli treningowej i monitorowanie modeli
- optymalizacja hiperparametrów w uczeniu maszynowym i głębokim uczeniu
- serwowanie i wdrażanie modeli
- wykorzystanie środowisk chmurowych
- podstawy RAG

Tematyka zajęć

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, prezentacja

Literatura

Literatura podstawowa

1. Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., ... & Heidmann, L. (2020). Introducing MLOps. O'Reilly Media.
2. Gift, Noah, and Alfredo Deza. Practical MLOps. O'Reilly Media, Inc., 2021.
3. Lakshmanan, Valliappa, Sara Robinson, and Michael Munn. Machine learning design patterns. O'Reilly Media, 2020.
4. Raj, Emmanuel. Engineering MLOps. Packt Publishing, 2021.

Literatura uzupełniająca

1. Burkov, Andriy. Machine learning engineering. Vol. 1. True Positive Incorporated, 2020.
2. Hapke, Hannes, and Catherine Nelson. Building machine learning pipelines. O'Reilly Media, 2020.
3. Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc., 2019.
4. Kleppmann, Martin. Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media, Inc., 2017.
5. McMahon, Andrew, Machine Learning Engineering with Python: Manage the production life cycle of machine learning models using MLOps with practical examples, Packt Publishing Ltd, 2021

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	32	1,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwίων/egzaminu, wykonanie projektu)	18	1,00