



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Inżynieria uczenia maszynowego

Sylabus zajęć

Informacje podstawowe

Kierunek studiów Informatyka - przedmioty do wyboru Specjalność - Jednostka organizacyjna Wydział Matematyki i Informatyki Poziom studiów studia drugiego stopnia poinżynierskie Forma studiów studia stacjonarne Profil studiów profil ogólnoakademicki		Cykl dydaktyczny 2023/24 Kod zajęć 06INFPWS.4200000S.01005.23 Języki wykładowe polski Obligatoryjność Fakultatywny Blok zajęciowy Przedmioty specjalnościowe
Koordynator zajęć	Tomasz Ziętkiewicz	
Prowadzący zajęcia	Tomasz Ziętkiewicz	
Okres Semestr letni	Forma zajęć / liczba godzin / forma zaliczenia • Laboratorium: 30, Zaliczenie z oceną	Liczba punktów ECTS 3

Cele kształcenia dla zajęć

Kod	Cel
C1	Poznanie specyfiki rozwoju modeli uczenia maszynowego.
C2	Poznanie specyfiki technik ciągłej integracji stosowanych do automatyzacji procesów uczenia maszynowego.
C3	Zrozumienie zasad działania konteneryzacji.
C4	Nabycie umiejętności korzystania z narzędzia konteneryzacji Docker.
C5	Poznanie różnych systemów ciągłej integracji.
C6	Nabycie umiejętności korzystania i konfiguracji środowiska ciągłej integracji Jenkins.
C7	Poznanie i praktyczne wykorzystanie narzędzi kontroli eksperymentów uczenia maszynowego na przykładzie narzędzi takich jak Sacred, DVC i MLFlow.

Wymagania wstępne

Umiejętność programowania na poziomie inżyniera informatyki. Znajomość podstaw inżynierii programowania. Znajomość podstaw systemu kontroli wersji Git. Umiejętność korzystania z systemu typu Linux z wiersza poleceń. Podstawowa znajomość języków programowania Python i Bash.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	Zna specyfikę rozwoju modeli uczenia maszynowego.		Projekt
W2	Zna i rozumie podstawowe pojęcia i koncepcje związane z procesem ciągłej integracji.		Projekt
W3	Zna popularne biblioteki uczenia maszynowego i różnice między nimi.		Projekt
W4	Rozumie podstawę działania kontenerów w systemie operacyjnym Linux i zna ich zastosowania.		Projekt
W5	Rozumie, na czym polega proces ciągłej integracji, potrafi podać przykład takiego procesu.		Projekt
W6	Zna popularne narzędzia do kontroli danych, parametrów i wyników eksperymentów uczenia maszynowego.		Projekt
Umiejętności - Student/ka:			
U1	Potrafi przeprowadzić trening i ewaluację prostego modelu uczenia maszynowego przy użyciu wybranej biblioteki.		Projekt
U2	Potrafi przygotować dane trenujące model uczenia maszynowego i podzielić je na podzbiory i rozumie na czym polega ewaluacja krzyżowa.		Projekt
U3	Potrafi korzystać z narzędzia konteneryzacji Docker.		Projekt
U4	Potrafi utworzyć i skonfigurować zadanie w systemie ciągłej integracji Jenkins.		Projekt

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
U5	Potrafi zintegrować system ciągłej integracji z systemem kontroli wersji.		Projekt
U6	Potrafi stworzyć proces ciągłej integracji modelu uczenia maszynowego w systemie ciągłej integracji Jenkins.		Projekt
U7	Potrafi stworzyć wykres wizualizujący wyniki działania modelu uczenia maszynowego.		Projekt
U8	Potrafi zintegrować proces wizualizacji wyników ewaluacji modelu uczenia maszynowego z procesem ciągłej integracji w środowisku Jenkins.		Projekt
U9	Potrafi zastosować wybrane narzędzie do kontroli danych, parametrów i wyników eksperymentów uczenia maszynowego.		Projekt
U10	Potrafi zintegrować wybrane narzędzie do kontroli danych, parametrów i wyników eksperymentów uczenia maszynowego z narzędziem do wizualizacji wyników. Potrafi zinterpretować wyniki przedstawione w tym narzędziu.		Projekt
U11	Potrafi zintegrować wszystkie elementy kompletnego systemu ciągłej integracji uczenia maszynowego i używać tego systemu do rozwoju modeli uczenia maszynowego.		Projekt

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Wprowadzenie podstawowych koncepcji ciągłej integracji, konteneryzacji i uczenia maszynowego. Przedstawienie specyfiki procesów rozwoju, testowania i integracji modeli uczenia maszynowego.	W1, W2	Laboratorium
2.	Przegląd najpopularniejszych bibliotek używanych w uczeniu maszynowym.	W3	Laboratorium
3.	Sposoby podziału danych trenujących i ewaluacji modeli uczenia maszynowego. Przygotowanie danych i bazowego rozwiązania dla wybranego problemu uczenia maszynowego	U1, U2	Laboratorium
4.	Konteneryzacja przy użyciu narzędzia Docker. Stworzenie obrazu Docker zawierającego środowisko potrzebne do przeprowadzania prostego eksperymentu uczenia maszynowego dla wybranego wcześniej problemu.	W4, U3	Laboratorium
5.	Wprowadzenie do ciągłej integracji. Przedstawienie istniejących środowisk ciągłej integracji. Zapoznanie z możliwościami i zastosowaniami ciągłej integracji na przykładzie systemu Jenkins. Stworzenie pierwszego zadania w systemie Jenkins uruchamiające prosty eksperyment uczenia maszynowego.	W5, U4, U5	Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
6.	Połączenie kilku zadań Jenkins ze sobą w całość automatyzującą wielostopniowy proces uczenia maszynowego.	U6	Laboratorium
7.	Wizualizacja wyników eksperymentów za pomocą biblioteki do tworzenia wykresów, np. Matplotlib, stworzenie zadania Jenkins realizującego wizualizację.	U7, U8	Laboratorium
8.	Przegląd narzędzi do kontroli danych, parametrów i wyników eksperymentów: DVC, Sacred, MLFlow.	W6	Laboratorium
9.	Kontrola parametrów i wyników eksperymentów uczenia maszynowego przy pomocy wybranego narzędzia.	U9	Laboratorium
10.	Wizualizacja wyników eksperymentów uczenia maszynowego przy pomocy wybranego narzędzia.	U10	Laboratorium
11.	Integracja kompletnego procesu trenowania i ewaluacji modelu uczenia maszynowego przy użyciu poznanych wcześniej narzędzi.	U11	Laboratorium
12.	Podsumowanie wyników pracy podczas kursu.	W1, U11	Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Laboratorium	Metoda laboratoryjna, Metoda warsztatowa, Metoda projektu

Forma zajęć	Warunki zaliczenia zajęć
Laboratorium	Na końcową ocenę składa się wynik uzyskany z projektu. Skala ocen: 1. bardzo dobry (bdb; 5,0) – od 90% punktów, 2. dobry plus (db plus; 4,5) – od 80% punktów, 3. dobry (db; 4,0) – od 70% punktów, 4. dostateczny plus (dst plus; 3,5) – od 60% punktów, 5. dostateczny (dst; 3,0) – od 50% punktów, 6. niedostateczny (ndst; 2,0) – poniżej 50% punktów.

Literatura

Obowiązkowa

1. Dokumentacja narzędzia konteneryzacji Docker: <https://docs.docker.com/getstarted>, dostęp 08.11.2020
2. Dokumentacja systemu ciągłej integracji Jenkins: <https://jenkins.io/doc>, dostęp 08.11.2020
3. Dokumentacja narzędzia do zarządzania eksperymentami Sacred: <https://github.com/IDSIA/sacred>, dostęp 08.11.2020
4. Dokumentacja narzędzia do zarządzania danymi i eksperymentami DVC: <https://dvc.org/>, dostęp 08.11.2020
5. Dokumentacja narzędzia do zarządzania danymi i eksperymentami MLFlow: <https://mlflow.org/>, dostęp 08.11.2020
6. Dokumentacja narzędzia do rysowania wykresów Matplotlib: <https://matplotlib.org>, dostęp 08.11.2020

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Laboratorium	30
Czytanie wskazanej literatury	15
Przygotowanie projektu	30
Łączny nakład pracy studenta	Liczba godzin 75
Liczba punktów ECTS	ECTS 3

* godzina (lekcyjna) oznacza 45 minut