



Karta przedmiotu

Nazwa i kod przedmiotu	Języki programowania dla sztucznej inteligencji , PG_00053334						
Kierunek studiów	Inżynieria biomedyczna, Inżynieria biomedyczna, Inżynieria biomedyczna						
Data rozpoczęcia studiów	luty 2022 r.	Rok akademicki realizacji przedmiotu			2021/2022		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	1	Liczba punktów ECTS			2.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Elektroniki -> Telekomunikacji i Informatyki -> Katedra Inżynierii Biomedycznej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Paweł Syty					
	Prowadzący zajęcia z przedmiotu	mgr inż. Natalia Kowalczyk dr inż. Paweł Syty					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
W tym liczba godzin zajęć na odległość: 0.0							
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	30	3.0		17.0		50
Cel przedmiotu	Wprowadzenie do zagadnień implementacji algorytmów sztucznej inteligencji w wybranych językach programowania i z wykorzystaniem wybranych narzędzi i bibliotek.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów, dokonując oceny i krytycznej analizy wykonanego oprogramowania, a także syntezy i twórczej interpretacji prezentowanych za jego pomocą informacji	Student potrafi skonfigurować środowisko pracy i dobrać właściwe narzędzia i metody programistyczne do rozwiązania postawionego problemu.	[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU2] Ocena umiejętności analizy informacji
	[K7_W04] zna i rozumie w zaawansowanym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo inne elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	Student potrafi wykorzystać wybrany język programowania do zaimplementowania wybranych algorytmów sztucznej inteligencji.	[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym
	[K7_U01] potrafi wykorzystywać posiadaną wiedzę matematyczną przy formułowaniu i rozwiązywaniu złożonych i nietypowych problemów związanych z kierunkiem studiów, poprzez: – właściwy dobór informacji źródłowych oraz dokonywanie ich krytycznej analizy, syntezy oraz twórczej interpretacji i prezentacji tych informacji, – zastosowanie właściwych metod i narzędzi	Student potrafi wykorzystać posiadaną wiedzę matematyczną do rozwiązywania podstawowych problemów sztucznej inteligencji,	[SU1] Ocena realizacji zadania
	[K7_K02] jest gotów do krytycznej oceny odbieranych treści, uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych	Student potrafi w sposób krytyczny odnieść się do praktycznych zagadnień które pojawiają się w tematyce przedmiotu.	[SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce

Treści przedmiotu	<p>Wykład</p> <ul style="list-style-type: none"> Wprowadzenie do zagadnień implementacji algorytmów sztucznej inteligencji. Ogólne omówienie języków programowania najczęściej wykorzystywanych do programowania sztucznej inteligencji (np. Python, Prolog, R, Julia, MTT). Konfiguracja środowiska programistycznego i podstawowych narzędzi, w tym do zarządzania konfiguracją. Przygotowanie danych (np. z wykorzystaniem pakietu Pandas). Wizualizacja danych (np. z wykorzystaniem pakietu Matplotlib). Badanie podstawowych statystyk (np. z wykorzystaniem pakietu NumPy). Implementacja wybranych algorytmów uczenia nadzorowanego i nienadzorowanego oraz klasyfikatorów uczenia maszynowego np. za pomocą bibliotek scikit-learn, SciPy. Implementacja (od podstaw) perceptronu prostego wraz z uczeniem nadzorowanym metodą gradientową (np. z wykorzystaniem pakietu NumPy). Wykorzystanie wybranych bibliotek programistycznych (np. TensorFlow / Keras) do implementacji jednokierunkowej, wielowarstwowej sieci neuronowej. Narzędzia do podglądu procesu uczenia w czasie rzeczywistym (np. biblioteka TensorBoard). Techniki augmentacji danych i wykorzystanie generatorów. Ewaluacja modeli. Elementy programowania równoległego oraz wykorzystanie GPU w zagadnieniach uczenia maszynowego. Optymalizacja programów i algorytmów. Dobre praktyki programistyczne. Kompletnie studia wybranych przypadków (np. związanych z przetwarzaniem danych biomedycznych). <p>Laboratorium</p> <ul style="list-style-type: none"> Przygotowanie i oczyszczanie danych z wykorzystaniem bibliotek programistycznych. Wizualizacja danych i wyników. (np. Pandas, NumPy, Matplotlib) Wykorzystanie bibliotek programistycznych w zadaniach uczenia maszynowego (np. SciKit) Tworzenie i wykorzystanie modeli sieci neuronowych w środowisku programistycznym (np. TensorFlow, Pytorch) - część 1. Tworzenie i wykorzystanie modeli sieci neuronowych w środowisku programistycznym (np. TensorFlow, Pytorch) - część 2. Wykorzystanie bibliotek programistycznych do ewaluacji modeli. Wykorzystanie bibliotek programistycznych w uczeniu maszynowym dla danych multimedialnych (obraz, sygnał/sekwencja) 											
Wymagania wstępne i dodatkowe												
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1"> <thead> <tr> <th data-bbox="456 981 794 1010">Sposób oceniania (składowe)</th> <th data-bbox="799 981 1139 1010">Próg zaliczeniowy</th> <th data-bbox="1144 981 1481 1010">Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td data-bbox="456 1016 794 1068">zaliczenie części laboratoryjnej (projekty)</td> <td data-bbox="799 1016 1139 1068">50.0%</td> <td data-bbox="1144 1016 1481 1068">60.0%</td> </tr> <tr> <td data-bbox="456 1075 794 1099">zaliczenie części wykładowej</td> <td data-bbox="799 1075 1139 1099">50.0%</td> <td data-bbox="1144 1075 1481 1099">40.0%</td> </tr> </tbody> </table>			Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	zaliczenie części laboratoryjnej (projekty)	50.0%	60.0%	zaliczenie części wykładowej	50.0%	40.0%
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej										
zaliczenie części laboratoryjnej (projekty)	50.0%	60.0%										
zaliczenie części wykładowej	50.0%	40.0%										
Zalecana lista lektur	<table border="1"> <tr> <td data-bbox="456 1115 794 1518">Podstawowa lista lektur</td> <td colspan="2" data-bbox="799 1115 1481 1518"> <p>M. Lutz, Python. Wprowadzenie. Wydanie V, O'Reilly, 2020</p> <p>R. Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion/Apress, 2021</p> <p>W. McKinney, Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython. Wydanie II, O'Reilly, 2018</p> <p>M. Gorelick, I. Ozsvald, Wysoko wydajny Python. Efektywne programowanie w praktyce. Wydanie II, O'Reilly, 2021</p> </td> </tr> <tr> <td data-bbox="456 1525 794 1576">Uzupełniająca lista lektur</td> <td colspan="2" data-bbox="799 1525 1481 1576">A. Géron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Wydanie II, O'Reilly, 2020</td> </tr> <tr> <td data-bbox="456 1583 794 1615">Adresy eZasobów</td> <td colspan="2" data-bbox="799 1583 1481 1615"></td> </tr> </table>			Podstawowa lista lektur	<p>M. Lutz, Python. Wprowadzenie. Wydanie V, O'Reilly, 2020</p> <p>R. Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion/Apress, 2021</p> <p>W. McKinney, Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython. Wydanie II, O'Reilly, 2018</p> <p>M. Gorelick, I. Ozsvald, Wysoko wydajny Python. Efektywne programowanie w praktyce. Wydanie II, O'Reilly, 2021</p>		Uzupełniająca lista lektur	A. Géron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Wydanie II, O'Reilly, 2020		Adresy eZasobów		
Podstawowa lista lektur	<p>M. Lutz, Python. Wprowadzenie. Wydanie V, O'Reilly, 2020</p> <p>R. Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion/Apress, 2021</p> <p>W. McKinney, Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython. Wydanie II, O'Reilly, 2018</p> <p>M. Gorelick, I. Ozsvald, Wysoko wydajny Python. Efektywne programowanie w praktyce. Wydanie II, O'Reilly, 2021</p>											
Uzupełniająca lista lektur	A. Géron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Wydanie II, O'Reilly, 2020											
Adresy eZasobów												
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<p>Wymienić metryki wykorzystywane do ewaluacji modeli maszynowych. Opisać w jednym zdaniu jedną z nich i metodę jej obliczania w języku Python.</p> <p>Zaimplementować perceptron prosty w języku Python z wykorzystaniem biblioteki Tensorflow i interfejsu Keras.</p>											
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy											