



Karta przedmiotu

Nazwa i kod przedmiotu	Języki programowania Python i R, PG_00045765						
Kierunek studiów	Fizyka Techniczna						
Data rozpoczęcia studiów	luty 2023 r.		Rok akademicki realizacji przedmiotu		2022/2023		
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 Fizyki Technicznej i Matematyki Stosowanej -> Katedra Fizyki Teoretycznej i Informatyki Kwantowej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Patryk Jasik				
	Prowadzący zajęcia z przedmiotu		dr inż. Patryk Jasik  dr inż. Paweł Syty				
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	0.0	0.0	30.0	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		5.0		15.0	50
Cel przedmiotu	Celem przedmiotu jest uczenie studentów praktycznego wykorzystywania języków Python i R w zakresie podstaw uczenia maszynowego.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu			Sposób weryfikacji i oceny efektu	
	[K7_K01] Zna ograniczenia własnej wiedzy. Rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych i osobistych.		Student rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych poprzez stosowanie stale udoskonalanych bibliotek stworzonych dla wybranego języka programowania.			[SK2] Ocena postępów pracy	
	[K7_U02] Posiada pogłębioną umiejętność programowania w wybranym języku oraz stosowania pakietów oprogramowania.		Student posiada umiejętność programowania w języku Python i R.			[SU1] Ocena realizacji zadania	

Treści przedmiotu	<p>Język Python</p> <ol style="list-style-type: none"> <li>1. Podstawowe typy danych i operacje na nich. Funkcja print(). Funkcja input(). Instrukcja warunkowa. Różne rodzaje pętli. Wyjątki. Listy, krotki, słowniki.</li> <li>2. Funkcje. Funkcja z argumentem opcjonalnym. Wykorzystanie zmiennej __main__, opisów funkcji i funkcji help(). Moduły. Stworzenie własnego modułu i wykorzystanie go. Zapis/odczyt danych do/z plików. Pliki YAML. Klasy i obiekty. Wstęp do wielowątkowości.</li> <li>3. Analiza danych pochodzących z wybranego zbioru. Wczytanie obserwacji dla wybranych zmiennych. Sprawdzenie podstawowych statystyk dla poszczególnych zmiennych. Wykreślenie histogramów. Identyfikacja zmiennych, w których występują potencjalnie błędne dane (obserwacje) lub braki danych. Naprawa danych. Obliczenie unormowanych korelacji pomiędzy poszczególnymi zmiennymi. Przeprowadzenie regresji liniowej dla wybranych zmiennych, wraz z wykresami.</li> <li>4. Stworzenie własnego modelu sieci neuronowej - perceptron prosty. Pakiety Tensorflow i Keras. Stworzenie prostego modelu sieci neuronowej z wykorzystaniem Tensorflow i Keras.</li> <li>5. Pakiet scikit-learn i model regresji liniowej. Współczynnik determinacji <math>R^2</math>, MSE, MAE. Podział zbioru danych na część treningową i testową. Predykcja wartości przy użyciu stworzonego modelu.</li> <li>6. Pakiet scikit-learn i preprocessing. Model wielomianowy. Generowanie nowych cech. Redukcja zmiennych modelu - kryterium Schwarza (BIC - Bayesian Information Criterion). Działanie modelu wielomianowego w praktyce.</li> <li>7. Pakiet scikit-learn, metoda k-najbliższych sąsiadów, drzewa decyzyjne i lasy losowe. Problem klasyfikacyjny. Wybór cech - predyktory i zmienna celu. Parametry modelu. Ocena jakości modelu - macierz pomyłek, czułość, swoistość, precyzja, dokładność, krzywa ROC, krzywa LIFT. Walidacja krzyżowa: k-krotna, n-krotna i Monte-Carlo (bootstrap).</li> <li>8. Pakiet scikit-learn i algorytm k-średnich jako przypadek uczenia się bez nadzoru. Analiza skupień - klasteryzacja. Parametry modelu. Indeks Fowlkesa-Mallowsa, czyli zgodność między dwoma podziałami zbioru na skupienia. Analiza głównych składowych - PCA.</li> <li>9. Pakiet scikit-learn i OpenCV. Maszyna wektorów nośnych - SVM. Problem rozpoznawania elementów na obrazkach. Wybór kernela w metodzie SVM. Wczytywanie archiwum z obrazkami. Histogram gradientów zorientowanych - HOG (histogram of oriented gradients), jako deskryptor cech. Tworzymy zbiór treningowy i testowy. Budowanie modelu, testowanie, boosting i sprawdzenie działania metody na wybranych zdjęciach.</li> </ol> <p>Język R</p> <ol style="list-style-type: none"> <li>1. Podstawy języka R. Zmienne i ich typy. Funkcje wejścia i wyjścia. Wbudowane stałe i funkcje. Sekwencje. Operacje na zbiorach. Listy. Funkcje. Warunki i pętle. Data i czas. Wyrażenia regularne i łańcuchy znaków.</li> <li>2. Ustawianie katalogu roboczego. Instalowanie pakietów i ładowanie bibliotek. Uzyskiwanie pomocy. Wczytywanie plików z danymi. Uzyskiwanie informacji o ramce danych. Formatowanie daty. Usuwanie błędnych danych. Przygotowanie danych do analizy. Tworzenie nowych zmiennych. Wizualizacja danych.</li> <li>3. Grupowanie danych w sposób istotny dla wszystkich obserwacji. Mapowanie i redukcja wymiaru (map-reduce). Badanie szeregów czasowych. Badanie sezonowości występowania danego zjawiska w rozważanym zbiorze. Normalizacja i standaryzacja danych.</li> <li>4. Sieci neuronowe w R. Tworzenie architektury sieci neuronowej - perceptron wielowarstwowy. Dobór parametrów sieci. Budowanie modelu. Testowanie modelu.</li> <li>5. Przegląd wybranych metod uczenia maszynowego w R - metoda k-najbliższych sąsiadów, drzewa decyzyjne i lasy losowe. Dobór parametrów metod. Budowanie modeli. Testowanie model.</li> <li>6. Przetwarzanie języka naturalnego na przykładzie klasyfikatora wiadomości niechcianych (spam). Prawdopodobieństwo warunkowe. Naiwny klasyfikator bayesowski. Macierz wyrazów i dokumentów TDM (TermDocumentMatrix). Dobór parametrów modelu. Trening modelu i jego testowanie.</li> </ol>
-------------------	--

Wymagania wstępne i dodatkowe	Podstawowe umiejętności programowania w wybranym języku. Podstawowa wiedza z rachunku prawdopodobieństwa i statystyki.		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Raport z analizy i modelowania wybranego zbioru danych	50.5%	73.0%
	Wykonanie na zajęciach zadań cząstkowych	50.5%	27.0%
Zalecana lista lektur	Podstawowa lista lektur	1. Marek Gagolewski, Maciej Bartoszek, Anna Cena, "Przetwarzanie i analiza danych w języku Python", PWN 2019  2. Alberto Boschetti, Luca Massaron, "Python. Podstawy nauki o danych. Wydanie II", Helion 2017  3. Drew Conway, John Myles White, "Uczenie maszynowe dla programistów", Helion 2014	
	Uzupełniająca lista lektur	1. Sebastian Raschka, "Python. Uczenie maszynowe", Helion 2017  2. Hadley Wickham, Garrett Grolemund, "Język R. Kompletny zestaw narzędzi dla analityków danych", Helion 2017	
	Adresy eZasobów	Adresy na platformie eNauczanie: Języki programowania Python i R (2022/2023) - Moodle ID: 25655 <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=25655">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=25655</a>	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<p>Wytyczne do stworzenia raportu:</p> <p>1. Tytuł raportu</p> <p>2. Wstęp - motywacja, cele</p> <p>3. Opis danych - struktura zbiorów, opis zmiennych, pochodzenie</p> <p>4. Opis procesu przygotowywania danych do analizy - kolejne kroki</p> <p>5. Analiza danych - przyjęte założenia, krótki opis metod i obranej metodologii analizy</p> <p>6. Modelowanie danych - przyjęte założenia, krótki opis metod i obranej metodologii budowania modeli</p> <p>7. Rezultaty, wnioski i ich dyskusja</p> <p>Raport, wraz ze wszystkimi kodami, należy umieścić w wybranym repozytorium (np. GitLab, GitHub).</p> <p>Przykładowe zadania:</p> <p>1. Zrealizować automatyczne dodawanie "podejrzanych" zmiennych do słownika "zmienne_do_naprawienia", na podstawie analizy statystyk danej zmiennej.</p> <p>2. Znaleźć inną metodę wyznaczania progu anomalii tak, aby nie była to "hardkodowana" wartość 10000, ale liczba wyznaczana indywidualnie dla każdej zmiennej.</p>		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		