SYLABUS PRZEDMIOTU

***Uczenie maszynowe***

1. **Informacje ogólne**

Nazwa przedmiotu ***Uczenie maszynowe***

Kod przedmiotu **(UCM)**

Rodzaj przedmiotu: Obowiązkowy

Kierunek studiów: Informatyka

Poziom kształcenia: II stopień

Profil kształcenia: Ogólnoakademicki

Rok studiów: pierwszy

Rodzaje zajęć i liczba godzin

Wykład **30**

Ćwiczenia **0**

Laboratoria **30**

Praktyki **0**

Liczba punktów ECTS **6**

Imię, nazwisko, tytuł/stopień naukowy, adres e-mail wykładowcy (wykładowców)/ prowadzących zajęcia

* dr Paweł Skórzewski [pawel.skorzewski@amu.edu.pl](mailto:pawel.skorzewski@amu.edu.pl)

Język wykładowy polski

Przedmiot prowadzony zdalnie (e-learning) **tak, częściowo**

1. **Informacje szczegółowe**
2. **Cele przedmiotu**

Przedmiot stawia następujące cele:

- zrozumienie koncepcji uczenia maszynowego

- poznanie najważniejszych algorytmów uczenia maszynowego

- nabycie umiejętności stosowania metod uczenia maszynowego w praktyce programistycznej

- nabycie umiejętności poprawnej ewaluacji rozwiązań programistycznych wykorzystujących metody uczenia maszynowego

1. **Wymagania wstępne w zakresie wiedzy, umiejętności oraz kompetencji społecznych**

Podstawowa umiejętność programowania.

Znajomość podstaw algebry liniowej.

1. **Efekty uczenia się (EU) dla zajęć i odniesienie do efektów uczenia się (EK) dla kierunku studiów**

|  |  |  |
| --- | --- | --- |
| **Symbol EU dla przedmiotu** | **Symbol EK dla kierunku studiów** | **Po zakończeniu modułu i potwierdzeniu osiągnięcia EU student/ka:** |
| UCM\_01 | KINF2\_W02 | Rozumie rolę i znaczenie uczenia maszynowego we współczesnej informatyce, potrafi wskazać przykłady zastosowań uczenia maszynowego. |
| UCM\_02 | KINF2\_W02 | Potrafi wyróżnić podstawowe typy zadań uczenia maszynowego i wskazać ich przykłady. |
| UCM\_03 | KINF2\_U02 | Umie korzystać z podstawowych narzędzi bibliotek NumPy i PyTorch oraz elementów języka Python przydatnych do implementowania rozwiązań z dziedziny uczenia maszynowego. |
| UCM\_04 | KINF2\_U02 | Umie przetwarzać dane przechowywane w tekstowych formatach tabelarycznych (CSV/TSV). |
| UCM\_05 | KINF2\_U02 | Umie wizualizować dane, korzystając z bibliotek Matplotlib i Seaborn. |
| UCM\_06 | KINF2\_W01 | Rozumie zagadnienie regresji liniowej jednej i wielu zmiennych. |
| UCM\_07 | KINF2\_W04 | Rozumie metodę gradientu prostego. |
| UCM\_08 | KINF2\_U02 | Umie zaimplementować algorytm gradientu prostego do znalezienia rozwiązania problemu regresji liniowej. |
| UCM\_09 | KINF2\_W01 | Rozumie zagadnienie regresji logistycznej. |
| UCM\_10 | KINF2\_U02 | Umie zaimplementować algorytm gradientu prostego do znalezienia rozwiązania problemu regresji logistycznej. |
| UCM\_11 | KINF2\_W04 | Rozumie znaczenie ewaluacji algorytmów uczenia maszynowego i zna jej podstawowe metody. |
| UCM\_12 | KINF2\_W04 | Rozumie rolę zbiorów danych: uczącego, walidacyjnego i testowego, i potrafi z nich korzystać. |
| UCM\_13 | KINF2\_W04 | Zna podstawowe miary jakości stosowane przy ewaluacji algorytmów uczenia maszynowego. |
| UCM\_14 | KINF2\_U02 | Potrafi korzystać z modułów pakietu Scikit-Learn do implementacji rozwiązań uczenia maszynowego. |
| UCM\_15 | KINF2\_U05 | Potrafi dokonać ewaluacji zaimplementowanego rozwiązania. |
| UCM\_16 | KINF2\_W01 | Rozumie zjawiska nadmiernego i niedostatecznego dopasowania. |
| UCM\_17 | KINF2\_W04 | Zna metody regularyzacji. |
| UCM\_18 | KINF2\_U02 | Umie zapobiegać nadmiernemu i niedostatecznemu dopasowaniu w implementowanych przez siebie rozwiązaniach. |
| UCM\_19 | KINF2\_U02 | Umie poprawnie reprezentować dane różnych typów i korzystać z nich do rozwiązywania problemów metodami uczenia maszynowego. |
| UCM\_20 | KINF2\_W04 | Rozumie znaczenie optymalizacji i zna jej podstawowe metody. |
| UCM\_21 | KINF2\_U02 | Umie stosować metody optymalizacji uczenia maszynowego. |
| UCM\_22 | KINF2\_W04 | Rozumie ideę uczenia nienadzorowanego i zna najważniejsze algorytmy uczenia nienadzorowanego. |
| UCM\_23 | KINF2\_U02 | Potrafi zaimplementować przykładowy algorytm uczenia nienadzorowanego. |
| UCM\_24 | KINF2\_W04 | Rozumie zasadę działania naiwnego klasyfikatora bayesowskiego. |
| UCM\_25 | KINF2\_W04 | Rozumie zasadę działania algorytmu k najbliższych sąsiadów. |
| UCM\_26 | KINF2\_W04 | Rozumie zasadę działania drzew decyzyjnych. |
| UCM\_27 | KINF2\_W04 | Rozumie zasadę działania sztucznych sieci neuronowych, w tym wielowarstwowych. |
| UCM\_28 | KINF2\_U02 | Potrafi wykorzystywać metodę propagacji wstecznej do uczenia wielowarstwowych sieci neuronowych. |
| UCM\_29 | KINF2\_U02 | Potrafi implementować sieci neuronowe z wykorzystaniem biblioteki PyTorch. |
| UCM\_30 | KINF2\_U02 | Potrafi implementować sieci neuronowe z wykorzystaniem biblioteki Keras. |
| UCM\_31 | KINF2\_W04 | Rozumie zasadę działania i potrafi wskazać zastosowania rekurencyjnych sieci neuronowych. |
| UCM\_32 | KINF2\_W04 | Rozumie zasadę działania i potrafi wskazać zastosowania splotowych sieci neuronowych. |
| UCM\_33 | KINF2\_W04 | Rozumie zasadę działania i potrafi wskazać zastosowania modeli typu encoder-decoder, w szczególności do tworzenia modeli języka i tłumaczenia maszynowego. |
| UCM\_34 | KINF2\_W04 | Rozumie mechanizm atencji i zasadę działania architektury typu transformer, potrafi wskazać ich zastosowania. |
| UCM\_35 | KINF2\_W04 | Rozumie ideę uczenia przez wzmacnianie i zna podstawowe paradygmaty uczenia przez wzmacnianie. |
| UCM\_36 | KINF2\_U05 | Potrafi zaprojektować, zaimplementować i zewaluować system wykorzystujący uczenie maszynowe. |

1. **Treści programowe zapewniające uzyskanie efektów uczenia się (EU) z odniesieniem do odpowiednich efektów uczenia się (EU) dla przedmiotu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lp.** | **Symbol EU dla przedmiotu** | **Godzin Wykład** | **Godzin  ĆW/ LAB/ SEM** | **Godzin pracy własnej** | **Opis treści kształcenia modułu zajęć/przedmiotu** |
| **Suma** | | **30** | **30** | **90** |  |
| 1. | UCM\_01 UCM\_02 | 2 |  | 2 | Wprowadzenie do uczenia maszynowego. Czym jest uczenie maszynowe? Rola i miejsce uczenia maszynowego we współczesnej informatyce. Przegląd zastosowań i metod uczenia maszynowego. Podstawowe pojęcia związane z uczeniem maszynowym. |
| 2. | UCM\_03 |  | 2 | 2 | Podstawowe narzędzia uczenia maszynowego. Elementy języka Python przydatne przy implementowaniu algorytmów uczenia maszynowego. Biblioteki NumPy i PyTorch. |
| 3. | UCM\_04 UCM\_05 |  | 2 | 2 | Narzędzia przetwarzania i wizualizacji danych w języku Python. Format CSV/TSV. Biblioteki Matplotlib i Seaborn. |
| 4. | UCM\_06 UCM\_07 | 2 |  | 2 | Regresja liniowa jednej zmiennej. Funkcja kosztu. Metoda gradientu prostego. Regresja liniowa wielu zmiennych. |
| 5. | UCM\_08 |  | 2 | 2 | Implementacja regresji liniowej jednej zmiennej w języku Python. |
| 6. | UCM\_09 | 2 |  | 2 | Regresja logistyczna. Metoda gradientu prostego dla regresji logistycznej. |
| 7. | UCM\_10 |  | 2 | 2 | Implementacja regresji logistycznej w języku Python. |
| 8. | UCM\_11 UCM\_12 UCM\_13 | 2 |  | 2 | Ewaluacja algorytmów uczenia maszynowego. Podział na zbiory: uczący, testowy i walidacyjny. Walidacja krzyżowa. Miary jakości. |
| 9. | UCM\_14 UCM\_15 |  | 2 | 2 | Pakiet Scikit-Learn. Implementacja regresji liniowej i regresji logistycznej z wykorzystaniem gotowych modułów. Implementacja wybranych metod ewaluacji. |
| 10. | UCM\_16 UCM\_17 | 2 |  | 2 | Nadmierne i niedostateczne dopasowanie. Obciążenie i wariancja. Ilustracja problemu nadmiernego dopasowania na przykładzie regresji wielomianowej. Metody regularyzacji. |
| 11. | UCM\_18 |  | 2 | 2 | Nadmierne i niedostateczne dopasowanie w praktyce. Implementacja metod zapobiegających nadmiernemu dopasowaniu. |
| 12. | UCM\_19 |  | 2 | 2 | Sposoby reprezentacji danych. Implementacja algorytmów regresji dla danych różnych typów, w tym dla danych nieliczbowych, oraz dla danych niepełnych. |
| 13. | UCM\_20 | 2 |  | 2 | Stochastic Gradient Descent. Przegląd metod optymalizacji. |
| 14. | UCM\_21 |  | 2 | 2 | Porównanie różnych metod optymalizacji na przykładach. |
| 15. | UCM\_22 | 2 |  | 2 | Uczenie nienadzorowane. Algorytm k średnich. Algorytm analizy głównych składowych. |
| 16. | UCM\_23 |  | 2 | 2 | Implementacja metod uczenia nienadzorowanego na przykładzie algorytmu k średnich. |
| 17. | UCM\_24  UCM\_25 UCM\_26 | 2 |  | 2 | Przegląd metod uczenia nadzorowanego. Naiwny klasyfikator bayesowski. Algorytm k najbliższych sąsiadów. Drzewa decyzyjne. |
| 18. | UCM\_27 | 2 |  | 2 | Wprowadzenie do sztucznych sieci neuronowych. Prosty perceptron. Funkcje aktywacji. Głębokie uczenie maszynowe. Wielowarstwowe sieci neuronowe. |
| 19. | UCM\_28 | 2 |  | 2 | Propagacja wsteczna. Uczenie wielowarstwowych sieci neuronowych. |
| 20. | UCM\_29 |  | 2 | 2 | Implementacja sieci neuronowych z wykorzystaniem biblioteki PyTorch. |
| 21. | UCM\_30 |  | 2 | 2 | Implementacja sieci neuronowych z wykorzystaniem biblioteki Keras. |
| 22. | UCM\_31 | 2 |  | 2 | Rekurencyjne sieci neuronowe – idea, przegląd najpopularniejszych architektur, przegląd zastosowań. |
| 23. | UCM\_32 | 2 |  | 2 | Splotowe sieci neuronowe – idea, przegląd najpopularniejszych architektur, przegląd zastosowań. |
| 24. | UCM\_33 | 2 |  | 2 | Modele typu encoder-decoder. Neuronowe tłumaczenie maszynowe. Autoencoder. Word embeddings. |
| 25. | UCM\_34 | 2 |  | 2 | Mechanizm atencji. Modele typu Transformer. |
| 26. | UCM\_35 | 2 |  | 2 | Uczenie przez wzmacnianie. |
| 27. | UCM\_36 |  | 8 | 38 | Indywidualny projekt programistyczny – implementacja wybranych metod uczenia maszynowego. |

1. **Zalecana literatura**
2. S. Raschka, Python Machine Learning, Packt, Birmingham 2015
3. S. Marsland, Machine Learning: An Algorithmic Perspective, CRC, Boca Raton 2015
4. W. Richert, L.P. Coelho, Building Machine Learning Systems with Python, Packt, Birmingham 2013
5. G. Moncecchi, R. Garreta, Learning scikit-learn: Machine Learning in Python, Packt, Birmingham 2013
6. K. Krawiec, J. Stefanowski, Uczenie maszynowe i sieci neuronowe, WPP, Poznań 2004
7. M. Krzyśko, W. Wołyński, T. Górecki, M. Skorzybut, Systemy uczące się, WNT, Warszawa 2008
8. W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz, Sieci neuronowe, Exit, Warszawa 2000
9. K.P. Murphy, Machine Learning: a Probabilistic Perspective, 2015
10. M. Nielsen, Neural Networks and Deep Learning, neuralnetworksanddeeplearning.com (odczyt: 2020-11-03)
11. J. Brownlee, Machine Learning Mastery, machinelearningmastery.com com (odczyt: 2020-11-03)
12. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, www.deeplearningbook.org com (odczyt: 2020-11-03)
13. A. Ng, Machine Learning, Coursera – kurs online, https://www.coursera.org/learn/machine-learning (odczyt: 2020-11-03)
14. G. Hinton, Neural Networks for Machine Learning, Coursera – kurs online, https://www.coursera.org/learn/neural-networks (odczyt: 2020-11-03)

1. **Informacje dodatkowe**
2. **Metody i formy prowadzenia zajęć umożliwiające osiągnięcie założonych EU (proszę wskazać z proponowanych metod właściwe dla opisywanych zajęć lub/i zaproponować inne)**

|  |  |
| --- | --- |
| **Realizacja** | **Metody i formy prowadzenia zajęć** |
| ✔ | Wykład z prezentacją multimedialną wybranych zagadnień |
|  | Wykład konwersatoryjny |
|  | Wykład problemowy |
|  | Dyskusja |
|  | Praca z tekstem |
|  | Metoda analizy przypadków |
|  | Uczenie problemowe (Problem-based learning) |
|  | Gra dydaktyczna/symulacyjna |
|  | Rozwiązywanie zadań (np.: obliczeniowych, artystycznych, praktycznych) |
|  | Metoda ćwiczeniowa |
| ✔ | Metoda laboratoryjna |
|  | Metoda badawcza (dociekania naukowego) |
|  | Metoda warsztatowa |
| ✔ | Metoda projektu |
|  | Pokaz i obserwacja |
|  | Demonstracje dźwiękowe i/lub video |
| ✔ | Metody aktywizujące (np.: „burza mózgów”, technika analizy SWOT, technika drzewka decyzyjnego, metoda „kuli śniegowej”, konstruowanie „map myśli”) |
|  | Praca w grupach |
| ✔ | Wykład zdalny w czasie rzeczywistym |
|  | Wykład zdalny asynchroniczny uzupełniony spotkaniem w czasie rzeczywistym |
|  | Wykład zdalny asynchroniczny z aktywnością studenta uzupełniony spotkaniem w czasie rzeczywistym |
| ✔ | Ćwiczenia/laboratoria/konwersatoria zdalne w czasie rzeczywistym |
| ✔ | Ćwiczenia zdalne asynchroniczne z pracą indywidualną studenta uzupełnione spotkaniem w czasie rzeczywistym |
|  | Ćwiczenia zdalne asynchroniczne z pracą grupową studentów uzupełnione spotkaniem w czasie rzeczywistym |
| ✔ | Laboratorium cyfrowe zdalne uzupełnione spotkaniem w czasie rzeczywistym |
|  | Konwersatorium asynchroniczne zdalne uzupełnione spotkaniem w czasie rzeczywistym |
|  | Seminarium zdalne w czasie rzeczywistym |
|  | Seminarium asynchroniczne zdalne ze spotkaniem w czasie rzeczywistym |

1. **Sposoby oceniania stopnia osiągnięcia EU (proszę wskazać z proponowanych sposobów właściwe dla danego EU lub/i zaproponować inne**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sposoby oceniania** | **Symbole EU**  **dla modułu** **zajęć/przedmiotu** | | |
| **UCM\_01, UCM\_02, UCM\_06, UCM\_07, UCM\_09, UCM\_11, UCM\_12, UCM\_13, UCM\_16, UCM\_17, UCM\_20, UCM\_22, UCM\_24, UCM\_25, UCM\_26, UCM\_27, UCM\_28, UCM\_31, UCM\_32, UCM\_33, UCM\_34, UCM\_35** | **UCM\_03, UCM\_04, UCM\_05, UCM\_08, UCM\_10, UCM\_14, UCM\_15, UCM\_18, UCM\_19, UCM\_21, UCM\_23, UCM\_29, UCM\_30** | **UCM\_36** |
| Egzamin pisemny | ✔ |  |  |
| Egzamin ustny |  |  |  |
| Egzamin z „otwartą książką” |  |  |  |
| Kolokwium pisemne |  |  |  |
| Kolokwium ustne |  |  |  |
| Test | ✔ |  |  |
| Projekt |  |  | ✔ |
| Esej |  |  |  |
| Raport |  |  |  |
| Prezentacja multimedialna |  |  |  |
| Egzamin praktyczny (obserwacja wykonawstwa) |  |  |  |
| Portfolio |  |  |  |
| Zadania cząstkowe na wykładzie |  |  |  |
| Zadania cząstkowe na laboratoriach |  | ✔ |  |

1. **Nakład pracy studenta i punkty ECTS**

|  |  |  |
| --- | --- | --- |
| **Forma aktywności** | | **Średnia liczba godzin na zrealizowanie aktywności** |
| Godziny zajęć (wg planu studiów) z nauczycielem | | 60 |
| Praca własna studenta\* | Przygotowanie do zajęć | 10 |
| Czytanie wskazanej literatury | 5 |
| Przygotowanie pracy pisemnej, raportu, prezentacji, itp. | 0 |
| Przygotowanie projektu | 38 |
| Przygotowanie pracy semestralnej | 0 |
| Przygotowanie do egzaminu/zaliczenia | 0 |
| Praca z materiałem do samokształcenia (np. Jupyter Notebook) | 15 |
| Praca z laboratorium cyfrowym (np. Code Runner) | 22 |
| Inne (jakie?) | 0 |
| **SUCM GODZIN** | | 150 |
| LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU | | **6** |

\* proszę wskazać z proponowanych przykładów pracy własnej studenta właściwe dla opisywanego modułu lub/i zaproponować inne

1. **Kryteria oceniania wg skali stosowanej w UAM**

|  |  |
| --- | --- |
| **Ocena** | **Kryterium** |
| bardzo dobry (bdb; 5,0) | od 90% punktów |
| dobry plus (+db; 4,5) | od 80% punktów |
| dobry (db; 4,0) | od 70% punktów |
| dostateczny plus (+dst; 3,5) | od 60% punktów |
| dostateczny (dst; 3,0) | od 50% punktów |
| niedostateczny (ndst; 2,0) | poniżej 50% punktów |