

Uczenie maszyn

Projekt: Zasady Zaliczenia

Prowadzący: Dr inż. Paweł Zyblewski

Semestr zimowy 2021/2022

1 Projekt – informacje i zasady zaliczenia

1.1 Cel projektu

Celem projektu jest nabycie wiedzy i praktycznych umiejętności z zakresu podstawowych metod projektowania systemów uczących się, oraz eksperymentalnej oceny jakości algorytmów uczenia maszynowego.

Wykonanie projektu polega nie tylko na implementacji środowiska eksperymentalnego oraz przeprowadzeniu badań, ale także (a właściwie przede wszystkim) na opracowaniu wyników i wyciągnięciu odpowiednich wniosków.

1.2 Zespoły projektowe

Zespoły projektowe powinny składać się z **dwóch** osób. Skład zespołu projektowego musi zostać zgłoszony prowadzącemu. Członkowie zespołu ponoszą **wspólną** odpowiedzialność za całość projektu.

1.3 Język programowania

Projekt powinien zostać wykonany w języku programowania **Python** z wykorzystaniem biblioteki uczenia maszynowego *scikit-learn*¹.

1.4 Warunki zaliczenia projektu

Projekt na ocenę dostateczną plus (max 3.5) W tym przypadku wystarczy zrealizowanie eksperymentu porównawczego.

- Jasno sformułowany cel eksperymentu.
- Plan eksperymentu.
- Kompletny kod środowiska eksperymentalnego.
- Wyniki wraz z analizą statystyczną otrzymanych rezultatów.
- Wnioski.

¹ <https://scikit-learn.org>

Projekt na ocenę od dobrej w górę (4.0 – 5.0) Tutaj wymagany jest dodatkowo przegląd literatury, a także podjęcie się realizacji bardziej ambitnego tematu projektowego.

- Przegląd dostępnej literatury oraz opracowanie teoretyczne na wybrany temat. Proszę dogłębnie przestudiować literaturę, nie ograniczać się jedynie do podręczników. Wymagam lektury co najmniej 6 artykułów w języku angielskim, w tym 3 z lat 2017-2021.
- Jasno sformułowany cel eksperymentu.
- Plan eksperymentu.
- Kompletny kod środowiska eksperymentalnego.
- Wyniki wraz z analizą statystyczną otrzymanych rezultatów.
- Wnioski.

Dodatkowo będą nagradzane projekty, które raport końcowy przekazany zostanie w języku angielskim w formie publikacji sformatowanej zgodnie z formatem IEEE² lub LNCS³

1.5 Harmonogram projektu według numeru zajęć

- 1 Przekazanie wymagań oraz zapoznanie ze sposobem oceny.
- 2 Zgłoszenie grupy projektowej oraz tematu projektu.
- 3-4 Przeprowadzenie studiów literaturowych. Przekazanie opracowanie na wybrany temat.
- 5-6 Przedstawienie założeń projektowych oraz planu eksperymentu + kod przygotowanego środowiska eksperymentalnego.
- 7-8 Oddanie finalnej wersji projektu.

1.6 Ocena końcowa – zasady

Ocenę końcową stanowi średnia ważona ocen w każdym z etapów, w której będą brane pod uwagę następujące elementy:

Projekt na ocenę dostateczną plus (max 3.5)

1. Prawidłowo sprecyzowane założenia projektowe oraz kod środowiska eksperymentalnego (50% oceny końcowej).
2. Projekt końcowy (50% oceny końcowej).

² http://www.ieee.org/conferences_events/conferences/publishing/templates.html

³ <http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0>

Projekt na ocenę od dobrej w górę (4.0 – 5.0)

1. Studia literaturowe (30% oceny końcowej).
2. Prawidłowo sprecyzowane założenia projektowe oraz kod środowiska eksperymentalnego (35% oceny końcowej).
3. Projekt końcowy (35% oceny końcowej).

Nieterminowe zgłoszenie tematu oznacza -0.5 od oceny końcowej. Oddanie opracowania któregoś z dalszych etapów po terminie wiąże się z otrzymaniem oceny niedostatecznej za dany etap. Ocena niedostateczna może być poprawiona wyłącznie na ocenę dostateczną.

Plagiaty Prace oddawane przez studentów są sprawdzane pod kątem wykrywania plagiatów. W przypadku stwierdzenia plagiatu grupy projektowe, których prace noszą znamiona plagiatu, otrzymują za projekt ocenę niedostateczną. Stwierdzenie plagiatu w jednej z części projektu jest jednoznaczne z uznaniem całego projektu za plagiat. Nie jest możliwa poprawa. Innymi słowy – popełnienie plagiatu skutkuje niezaliczeniem przedmiotu.

2 Lista przykładowych obszarów i tematów projektowych

Oczywiście w wielu przypadkach obszary będą się ze sobą przeplatać.

- Zespół klasyfikatorów (*classifier ensemble*):
 - Porównanie wpływu strategii uczenia klasyfikatorów złożonych na jakość klasyfikacji.
 - Modyfikacje metod uczenia klasyfikatorów złożonych (np. bazujących na koncepcji bagging, boosting, clustering and selection, random subspace itp.).
- Dynamiczna selekcja klasyfikatorów (*dynamic classifier selection*):
 - Porównanie algorytmów dynamicznej selekcji klasyfikatorów dla klasyfikacji binarnej.
 - Modyfikacje metod dynamicznej selekcji zespołów klasyfikatorów (np. KNORA-E, KNORA-U itp.).
- Klasyfikacja danych niezbalansowanych (*imbalanced data*):
 - Porównanie metod preprocessingu dla zadania klasyfikacji danych niezbalansowanych.
 - Propozycje nowych metod preprocessingu danych niezbalansowanych.
- Klasyfikacja strumieni danych (*data stream*):
 - Analiza porównawcza metod detekcji dryfu koncepcji w danych strumieniowych.
 - Połączenie preprocessingu oraz metod zespołowych do klasyfikacji niezbalansowanych strumieni danych.
- Uczenie aktywne (*active learning*):
 - Wpływ strategii etykietowania danych na jakość klasyfikacji.
 - Uczenie aktywne w klasyfikacji strumieni danych z dryfem koncepcji.

Grupa powinna zgłosić własny temat, posiadający odpowiedni poziom skomplikowania.

2.1 Dane do projektów:

Badania eksperymentalne należy przeprowadzić na przynajmniej **20** zbiorach danych, które można znaleźć w poniższych repozytoriach:

- *UCI Machine Learning repository* ⁴
- KEEL ⁵

⁴ <https://archive.ics.uci.edu/ml/datasets.php>

⁵ <https://sci2s.ugr.es/keel/datasets.php>