

O interakcjach nowoczesnych technologii informacyjnych

z nauczaniem matematyki i statystyki

Andrzej Giniewicz

Andrzej.Giniewicz@pwr.edu.pl

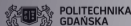
Wydział Matematyki
Politechnika Wrocławska

24 czerwca 2024



DIAM

DIDACTIC
INNOVATIONS IN
ACADEMIC
MATHEMATICS



POLITECHNIKA
GDAŃSKA

Agenda

1. Kontekst
2. Nasi studenci
3. Problemy do rozwiązania
4. Rozwiązanie
5. Jak to wszystko pomieścić w jednym kursie?
6. Podsumowanie

1

Kontekst

Na jakich kierunkach bazują moje doświadczenia? Głównie:

- Matematyka (I stopień, inż.)
mat. ogólna · uczenie maszynowe i inżynieria danych · mat. informatyczna
- Matematyka i Analiza Danych (I stopień, inż.)
- Matematyka Stosowana (I stopień, inż.)
- Matematyka (II stopień, mgr.)
Matematyka Finansowa i Ubezpieczeniowa · Matematyka Teoretyczna ·
Statystyka i Analiza Danych
- Applied Mathematics (II stopień, mgr., ang.)
Financial and Actuarial Mathematics · Mathematics for Industry and
Commerce · Data Engineering · Modelling · Simulation and Optimization

Technologie informacyjne dla matematyków–inżynierów.

2 Studenci na pierwszym semestrze

W większości oscylują po studiach pomiędzy matematyką a inżynierią oprogramowania.

Tymczasem, na pierwszym semestrze...

- niejednokrotnie nie wiążą swojej przyszłości z programowaniem lub technologiami informacyjnymi,
- mają bardzo różne doświadczenia z informatyką w szkołach średnich, od *graliśmy tylko w Counter Strike* do *programowaliśmy w Pythonie i C++*,
- mają bardzo różny poziom ogólnego alfabetyzmu cyfrowego, niektóre osoby wybierają wysłać kody programu zdjęciem ekranu zrobionym telefonem, zamiast załącznika z kodem, niektórzy mają profil na GitHubie przed studiami.

3.2

Problem 1 — motywacja

Podczas projektowania kursów informatycznych maksymalizuję interakcje.

- Już po pięciu tygodniach studenci mogą zacząć korzystać z narzędzi na innych przedmiotach:
 - studenci mogą rysować wykresy funkcji na analizie,
 - zobaczyć, jak działają macierze obrotu na algebrze,
 - przekonać się, skąd wynikają niektóre tożsamości rekurencyjne na matematyce dyskretnej;
- O wiele wcześniej studenci są przygotowani do pracy w grupie oraz przygotowywania raportów i sprawozdań, co zwiększa pulę możliwych zadań na innych przedmiotach;
- Na kursach matematycznych można zrobić większe przykłady i ciekawsze zadania, które bez komputera nie są praktyczne — na kursach informatycznych zyskuję zastrzyk motywacji u studentów oraz lepsze przećwiczenie materiału.

3.3

Problem 2 — wariacja na wejściu

Musimy zacząć szybko — najlepiej pierwszy, najpóźniej drugi semestr — dodatkowo nie możemy pozwolić sobie na duży próg wejścia.

Idealne środowisko to takie, w którym:

- można łączyć tekst, wzory i kod wykonywalny, w duchu programowania piśmiennego,
- student czuje się możliwie najmniej obco,
- student nie musi zmieniać systemu operacyjnego, aby zainstalować egzotyczne oprogramowanie,
- można zminimalizować liczbę kroków potrzebnych pomiędzy wywołaniem kodu a zobaczeniem wyniku.

3.4

Problem 3 — przykucie uwagi

Technologie informacyjne nie mogą kończyć się na wiedzy z lat 90-tych, dodają do kursu wykłady o nowych technologiach, przy okazji których studenci uczą się o nieznanym sobie jeszcze działach i kursach. Wbrew pozorom, można dużo zrobić w ciągu jednego wykładu, jeśli nie wchodzi się w detale, tylko w odpowiednich miejscach oddelegowujemy zainteresowanych na konkretne kursy wybieralne.

3.5

Problem 3 — przykłady

Blockchain w jeden wykład?

- wykład nie tylko o kryptowalutach, ale o pojęciu własności cyfrowych dóbr, o kontraktach i rozproszonym programowaniu na blockchain, o opłacalności i bezpieczeństwie tych systemów, o aspektach ekologicznych,
- szybko przeradza się w interakcję pomiędzy informatyką a matematyką dyskretną,
- pokazuję wersję mini — pobieżne omówienie algorytmów RSA i MD4 nie wykracza poza możliwości studentów pod koniec pierwszego semestru, w gratisie dowiadują się, jak działa podpis cyfrowy,
- stanowi doskonałą motywację do zgłębiania kryptografii w późniejszym czasie.

3.6

Problem 3 — przykłady

Duże modele językowe w jeden wykład?

- wykład o tym, czym są, i czym nie są modele językowe, na co trzeba uważać oraz pobieżnie jak działa architektura transformerów,
- interakcje z algebrą i analizą wręcz oczywiste dla studentów,
- jak działa sztuczny neuron z identycznościową funkcją aktywacji, niemal zawsze z reakcją grupy — „ooo! to kombinacja liniowa?”,
- studenci są w stanie zrozumieć, jak działają algorytmy zagnieżdżeń słów, na przykład skip-gram,
- stanowi doskonałą motywację do zgłębiania uczenia maszynowego na kolejnych kursach.

3.7

Problem 4 — spektrum ról

Narzędzia zależą od roli, ale przez rozmyty zakres kompetencji przy tych samych nazwach stanowisk u różnych pracodawców, trudno zidentyfikować te role.

Patrzę na role u absolwentów jak na spektrum rozciągające się pomiędzy matematyką a inżynierią oprogramowania. Różne firmy potrzebują różnego zakresu specjalistów z tego spektrum i dzielą je na różną liczbę stanowisk — bardzo trudno wydzielić jednoznacznie zdefiniowane stanowiska na podstawie tylko ofert pracy. Oderwijmy się więc na chwilę od myślenia o ofertach i pomyślmy abstrakcyjnie o innych znanych nam spektrach.

Problem 4 — tęcza możliwości

Jeśli chodzi o spektrum światła widzialnego, większość kultur wydziela siedem kolorów tęczy — w przypadku światła to jednocześnie wystarczająco różnorodny podział, abyśmy mogli zidentyfikować różne kolory, jak i wystarczająco mało liczny, aby był praktyczny.

Podział ról naszych przyszłych absolwentów:

- Matematyka
- Modelowanie matematyczne
- Analiza danych
- Data science
- Inżynieria modeli
- Inżynieria danych
- Inżynieria oprogramowania

Problem 4 — dalszy podział

W każdej z siedmiu ról można wydzielić trzy poziomy stanowisk twardej ze względu na używane metody i doświadczenie oraz jeden poziom miękkiej:

1. praktyk — realizuje zadania, ale nie podejmuje decyzji, jakich narzędzi używać,
2. metodolog — podejmuje decyzje, jakich narzędzi używać, ale nie tworzy nowych narzędzi,
3. naukowiec — tworzy nowe narzędzia,
4. menadżer — zajmuje się zarządzaniem zespołem złożonym z osób na pozostałych poziomach.

Problem 4 — w stronę rozwiązania

Przy takim określeniu 7 ról, 3 poziomów zaawansowania oraz jednego zarządczego, niemal każde ze stanowisk dostępnych na portalach agregujących oferty pracy w branży, na przykład <https://ai-jobs.net/>, może być zakwalifikowane dość jasno do jednej z 28 ról.

Jeśli łatwiej przyporządkować oferty pracy do ról, to również łatwiej przypisać narzędzia do ról.

Problem 4 — przykładowe narzędzia

- Matematyka — język lub pakiet matematyczny symboliczny (np.: Sage)
- Modelowanie matematyczne — język lub pakiet matematyczny numeryczny (np.: Julia)
- Analiza danych — język lub pakiet statystyczny (np.: R)
- Data science — język programowania interpretowany (np.: Python)
- Inżynieria modeli — język programowania kompilowany do bytecode (np.: Scala)
- Inżynieria danych — język programowania kompilowany z GC (np.: Go)
- Inżynieria oprogramowania — język programowania kompilowany bez GC (np.: Rust)

Rozpoczęcie od Pythona maksymalizuje nasz zasięg, ponieważ jest w samym środku spektrum, jakie rozważam dla naszych absolwentów.

Co robimy na pierwszym semestrze?

Zajęcia podzielone na cztery bloki.

1. Python od zera do wykresów — 7 tygodni:

- Python jako kalkulator + środowisko Jupyter Notebook + skład najprostszych wzorów w $\text{T}_{\text{E}}\text{X}$;
- Napisy i operacje na napisach, operacje wejścia/wyjścia;
- Wbudowane struktury danych, listy składane;
- Funkcje i instrukcje sterujące + interaktywne interfejsy w Notebook;
- NumPy/SymPy/Matplotlib — **studenci mogą robić wykresy, liczyć granice i operacje macierzowe po 5 tygodniu**;
- Wyjątki i testowanie, debugowanie kodu;
- Obsługa plików, Pandas i dane tabelaryczne.

Co robimy na pierwszym semestrze?

2 Terminal i git — 3 tygodnie:

- Praca z terminala (BASH), podstawy systemów operacyjnych;
- Procesy i zmienne środowiskowe, skrypty w BASH i Python;
- System kontroli wersji git.

3 \LaTeX i alfabetyzm naukowy — 3 tygodnie:

- Wprowadzenie do typografii cyfrowej i podstawy \LaTeX a oraz więcej składu wzorów w \LaTeX u;
- Skład ilustracji, materiałów pomocniczych i bibliografii, automatyczne generowanie raportów z Pythona;
- Cytowania, plagiaty i parafraza, jak działają czasopisma naukowe, czemu artykuł naukowy jest lepszym cytowaniem niż Wikipedia, co to liczba cytowań, Impact Factor i indeks Hirscha — demonstracja menadżera bibliografii (Zotero), integracja z narzędziami AI (Research Rabbit) i wyszukiwarkami w bazach naukowych i patentów (Lens).

4 Nowe technologie — 2 tygodnie, blockchain i duże modele językowe.

5.1

Jeszcze jeden problem — jak to wszystko pomieścić?

Wydaje się, że to mało czasu na każdy z tematów, ale dbam o ich zazębianie i wracanie do wcześniejszych bloków:

- \LaTeX jest dodatkowo na zajęciach z Jupyter Notebooka, studenci wcześniej uczą się pisać wzory,
- git jest używany w zadaniu domowym z \LaTeX a, więc jest ćwiczony na dużym projekcie — cały rocznik pracuje na jednym repozytorium, więc muszą się nauczyć rozwiązywać konflikty,
- Python jest pokazywany na zajęciach z terminala (skrypty) i \LaTeX a (automatyczne generowanie raportu z Pythona),
- przy dużych modelach językowych, wracam do rozmowy o plagiatach i parafrazie.

5.2

Jeszcze jeden problem — jak to wszystko pomieścić?

To wciąż duże tempo i nie zmieścilibyśmy wszystkiego, gdyby nie wariant odwróconej klasy:

- studenci dostają kompletne notatki kilka dni przed wykładem, zwykle kilkanaście stron,
- przychodzą na wykład po ich przeczytaniu, zaczynam od pytań do notatek, zwykle 30 minut,
- co tydzień kartkówka, 3 pytania wielokrotnego wyboru — zamiast kolokwium na koniec,
- zaraz po kartkówce robię omówienie, wyświetlam procent prawidłowych odpowiedzi na każde pytanie i wariant odpowiedzi,
- dyskusja pogłębiająca temat, zwykle zajmuje resztę wykładu,
- według ankiet po kursie, studenci lubią tę metodę.

Rozwój programu kursu z technologii informacyjnych to niekończąca się opowieść — cały czas pojawiają się nowe wersje oprogramowania, nowe narzędzia oraz nowe technologie, nowe wymagania od rynku pracy i kursów matematycznych i statystycznych na kolejnych semestrach.

Kurs w zaprezentowanej wersji odbył się w obecnym roku akademickim i wierzę, że dzięki interakcjom z kursami matematycznymi i statystycznymi, zauważonymi na etapie projektowania i pielęgnowanymi w trakcie realizacji, zebrał pozytywne oceny studentów — obecnie pracuję nad poszerzeniem jego interakcji z innymi przedmiotami.

Skończmy z nudnym TI!

6.2

Dziękuję za uwagę!

Pytania?

Chętnie — zarazem teraz, jak i po sesji!

Andrzej Giniewicz

Gdańsk, 24 czerwca 2024

Andrzej.Giniewicz@pwr.edu.pl