Lab 10: Dataset Frankenstein

Budujemy dane do predykcji defektów

Jarosław Hryszko
jaroslaw.hryszko@uj.edu.pl

Instytut Informatyki i Matematyki Komputerowej UJ

Otwarte repozytoria kodu i pomiary oprogramowania

Wielkie pytanie

Czy metryki kodu mogą predykować bugi?

Przez ostatnie laby zbieraliście metryki: LOC, CC, metryki OO, churn, ownership. Dziś łączymy je w **dataset do machine learningu**.

- Predykcja defektów (defect prediction) — aktywny obszar badań od lat 2000
- Cel: dla każdego pliku przewidzieć, czy zawiera bugi
- Wynik: priorytetyzacja testów, code review, refactoringu

Efekty kształcenia: W1, U1, U2

Schemat datasetu

filename	loc	avg_cc	...	churn	is_buggy
models.py	342	4.2	...	8234	1
sessions.py	218	3.8	...	5621	1
adapters.py	156	2.1	...	3102	0

Cechy produktowe LOC, avg_cc, max_cc, num_functions (z radona)

Cechy procesowe churn, num_commits, num_authors, age_days (z git log)

Etykieta is_buggy: 1 = plik zmieniany w bug-fixing commicie

Etykietowanie: buggy vs clean

Heurystyka

Plik jest **buggy**, jeśli był zmieniany w commicie z wiadomością zawierającą: `fix`, `bug`, `error`, `fault`, `defect`, `patch`, `crash`

Zalety

- Proste i szybkie
- Działa bez issue trackera
- Wystarczające na start

Wady

- „fix typo” to nie bug fix
- Nie łapie bugów bez „fix” w commit message
- Brak powiązania z issue trackerem

W badaniach naukowych: SZZ algorithm, linkowanie commitów z JIRA/GitHub Issues.

Eksploracja danych

Po zbudowaniu datasetu — zanim trenujemy model — **eksplorujemy dane**:

1. **Rozkłady cech** — histogramy (czy dane są skośne?)
2. **Balans klas** — ile buggy vs clean? (zwykle niezbalansowane!)
3. **Macierz korelacji** — które cechy są ze sobą powiązane?
4. **Boxploty buggy vs clean** — które cechy rozróżniają klasy?

Garbage in, garbage out

Jakość modelu ML zależy przede wszystkim od **jakości danych**. Dlatego eksploracja jest ważniejsza niż wybór algorytmu.

Plan zajęć

1. **Zadanie 1 — Jupyter setup** (30 min)
   - Uruchomienie Jupyter, zapoznanie z szablonem notebooka
2. **Zadanie 2 — Budowanie datasetu** (75 min)
   - Metryki produktowe (LOC, CC) + procesowe (churn, autorzy)
   - Etykietowanie buggy/clean
   - Eksploracja: histogramy, korelacje, boxploty
3. **Zadanie 3 — Feature engineering** (30 min, opcjonalne)
   - Dodatkowe cechy: metryki OO, ownership, wiek modyfikacji

Co oddajecie?

W branchu `lab10_nazwisko1_nazwisko2`:

1. `dataset_builder.ipynb` — wypełniony notebook
2. `dataset.csv` — wygenerowany dataset
3. *(opcjonalnie)* rozszerzony dataset z dodatkowymi cechami

Do roboty!

Instrukcja: `README.md` w repozytorium

"Zbieramy części z różnych miejsc i ożywiamy potwora... to znaczy dataset." — dr Frankenstein (wersja data science)