Laboratorium 7

Afektywny interfejs i ewaluacja

Informatyka afektywna

dr inż. Mateusz Pomianek

23 września 2026

Część I
Część organizacyjna

1 Cel i rezultat

Laboratorium prowadzi od poprawnie zdefiniowanego problemu do wyniku, który można zweryfikować i zinterpretować. Rezultatem nie jest sam działający kod: student ma przedstawić dane wejściowe, konfigurację, wynik ilościowy, przypadek błędny oraz ograniczenia wnioskowania.

2 Zasady pracy

Obszar	Wymaganie
Raport	Uzupełniaj `evidence/lab07.md`; szablon jest osadzony w PDF.
Reprodukowalność	Zapisz wersję danych, Pythona i bibliotek, seed, parametry oraz sposób podziału danych.
Dane	Nie modyfikuj danych źródłowych; artefakty i wyniki zapisuj osobno.
Walidacja	Nie dobieraj parametrów na podstawie finalnego zbioru testowego.
Interpretacja	Rozdziel obserwację, cechę obliczoną, predykcję modelu i interpretację psychologiczną.
Niepewność	System powinien dopuszczać wynik niepewny lub `unknown`; nie wymuszaj klasyfikacji bez podstaw.

3 Wymagania wstępne

Python, działający pipeline z wcześniejszych laboratoriów i wybrany framework interfejsu. Przed rozpoczęciem sprawdź uruchomienie środowiska, dostęp do danych i możliwość zapisania artefaktów eksperymentu.

4 Punkty kontrolne

3.0	działający baseline, poprawne wejście/wyjście i udokumentowane środowisko
3.5	kompletna implementacja zadania głównego i poprawnie dobrana metryka
4.0	kontrolowany eksperyment, wynik ilościowy i właściwa wizualizacja
4.5	failure case oraz analiza odporności lub generalizacji
5.0	ablation, analiza niepewności i krytyczna interpretacja ograniczeń

Część II
Część teoretyczna

5 Co właściwie mierzymy?

System afektywny nie powinien reagować na każdą chwilową predykcję. Potrzebuje polityki decyzji uwzględniającej confidence, stan unknown, koszt FP/FN i stabilność w czasie. Ewaluacja obejmuje zarówno jakość modelu, jak i jakość zachowania całego systemu człowiek–komputer.

Zasada interpretacji

Pipeline dokumentuj jako: **dane** → **preprocessing** → **reprezentacja** → **model/algorytm** → **wynik** → **interpretacja**. Każde przejście w tym łańcuchu wprowadza założenia i potencjalne źródła błędu.

6 Walidacja i generalizacja

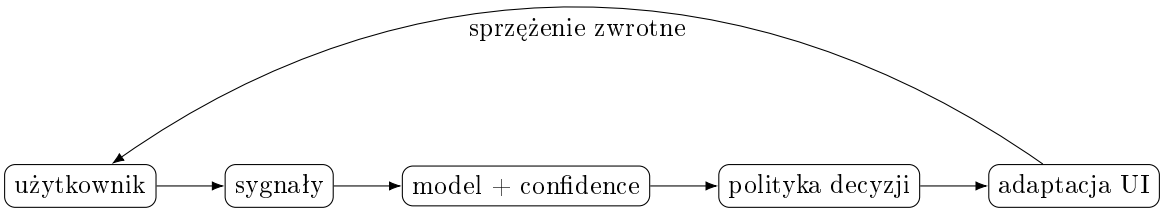
Jeżeli próbki pochodzą od wielu osób, rozróżnij generalizację na nowe próbki tej samej osoby od generalizacji na nową osobę. W drugim przypadku stosuj podział subject-independent, gdy identyfikatory uczestników są dostępne. Raportuj wyniki per klasa lub per warunek, a nie wyłącznie jedną wartość średnią. Zawsze sprawdź niezbalansowanie klas, brakujące dane i możliwość data leakage.

7 Niepewność i ground truth

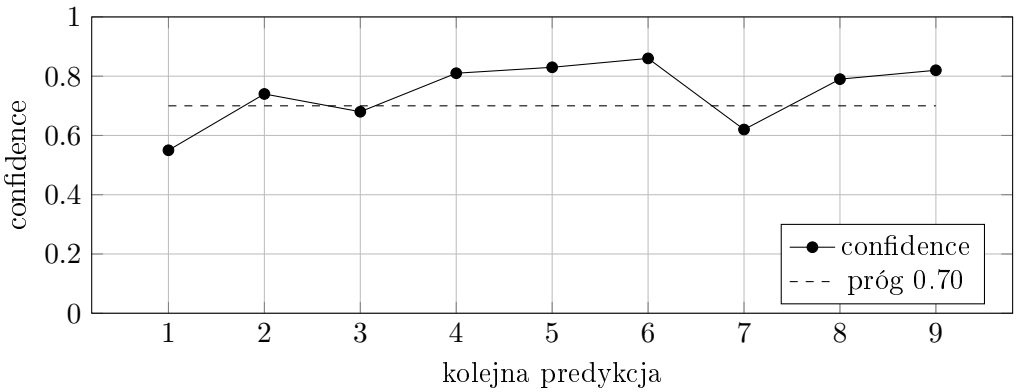
Ground truth w informatyce afektywnej może pochodzić z samoopisu, anotacji obserwatora, etykiety eksperymentalnej albo heurystyki. Te źródła nie są równoważne. W raporcie zapisz, skąd pochodzi etykieta referencyjna, jaka jest jej rozdzielczość i jakie ma ograniczenia. Confidence modelu opisuje zachowanie modelu, a nie pewność co do rzeczywistego stanu psychicznego osoby.

8 Diagramy wspomagające interpretację

8.1 Pętla adaptacyjnego interfejsu



8.2 Próg confidence



Pojedyncze przekroczenie progu nie musi wywoływać adaptacji. Stabilizacja ogranicza migotanie interfejsu i koszt fałszywych reakcji.

9 Wzory i miary

Niech c_t będzie confidence w chwili t, a τ progiem decyzji. Prosta bramka:

$$g_t = \mathbb{1}(c_t \geq \tau).$$

Dla stabilizacji wymagającej K zgodnych predykcji:

$$a_t = \begin{cases} \hat{y}_t, & g_t = 1 \ \wedge \ \hat{y}_{t-K+1} = \ldots = \hat{y}_t, \\ \text{no-action}, & \text{w przeciwnym razie.} \end{cases}$$

Koszt decyzji interfejsu można jawnie modelować:

$$J = C_{FP}N_{FP} + C_{FN}N_{FN} + C_{switch}N_{switch}.$$

Pozwala to dobierać politykę adaptacji według konsekwencji błędów, a nie wyłącznie accuracy modelu.

Część III
Część praktyczna

10 Dane wejściowe i oczekiwane artefakty

Wejście. Strumień predykcji zawierający czas, etykietę i confidence oraz zdefiniowane akcje interfejsu. Przygotuj również scenariusze, w których model chwilowo się myli.

Oczekiwany rezultat. Log ‘prediction → policy → UI action‘, liczba FP/FN adaptacji, liczba zmian stanu i porównanie polityk decyzyjnych.

Minimalny kontrakt danych

Przed implementacją sprawdź nazwy pól, jednostki, zakresy wartości, braki oraz identyfikator osoby/próbki. Jeżeli format danych różni się od przykładu, dostosuj kod jawnie i opisz zmianę w raporcie.

11 Przykład startowy w Pythonie

Poniższy listing jest punktem startowym, a nie gotowym rozwiązaniem eksperymentu. Student ma go dostosować do rzeczywistego formatu danych i udokumentować zmiany.

```python
THRESHOLD = 0.70
STABLE_SAMPLES = 3
history = []

def decide(label, confidence):
    history.append(label if confidence >= THRESHOLD else "unknown")
    recent = history[-STABLE_SAMPLES:]
    if len(recent) < STABLE_SAMPLES or len(set(recent)) != 1:
        return "no_action"
    return recent[0]

# UI reaguje na wynik polityki, a nie bezpośrednio na pojedynczą predykcję.
```

12 Jak czytać wynik

Najpierw sprawdź poprawność techniczną: zakres wartości, braki, liczbę próbek i zgodność wymiarów. Następnie oceń wynik ilościowo. Dopiero na końcu formułuj interpretację afektywną. Nie wyciągaj wniosku psychologicznego wyłącznie z pojedynczej predykcji lub atrakcyjnej wizualizacji.

13 1. Przygotowanie eksperymentu

Utwórz katalog wyników i zanotuj wersje środowiska. Ustal pytanie badawcze w formie, na którą można odpowiedzieć pomiarem. Zapisz hipotezę, zmienną niezależną, zmienną zależną oraz kryterium poprawności. Przed analizą sprawdź strukturę danych, braki i etykiety.

14 2. Baseline

Uruchom najprostszy wariant rozwiązania, który stanowi punkt odniesienia. Zapisz wejście, parametry i wynik do `evidence/lab07.md` → **Baseline**. Baseline powinien być prosty i możliwy do ponownego uruchomienia bez ręcznych kroków.

Punkt kontrolny 3.0

Pokaż działający baseline oraz wskaż dokładnie dane wejściowe, wynik i konfigurację używaną do jego uzyskania.

15 3. Zadanie główne

Zbuduj interfejs reagujący na stan modelu dopiero po spełnieniu jawnego warunku decyzyjnego. Dodaj stan unknown/no-action, histerezę lub minimalny czas stabilnego stanu. Zarejestruj zdarzenia: predykcję, confidence, decyzję UI i znacznik czasu.

Punkt kontrolny 3.5

Pokaż kompletny przepływ danych. Wyjaśnij znaczenie parametrów oraz wskaż, które elementy są pomiarem, które cechą, a które inferencją.

16 4. Eksperyment ilościowy

Wyznacz liczba poprawnych i błędnych adaptacji, FP/FN decyzji interfejsu, liczba niepotrzebnych zmian stanu i – jeśli ma znaczenie – latency reakcji. Zmieniaj jeden kontrolowany czynnik naraz. Dla procedur losowych zachowaj seed i wykonaj powtórzenia pozwalające ocenić zmienność. Wynik przedstaw w tabeli oraz na wykresie z podpisanymi osiami, jednostkami i opisem warunków.

Punkt kontrolny 4.0

Wynik ma odpowiadać na pytanie badawcze. Sama wizualizacja bez metryki i warunków eksperymentu nie wystarcza.

17 5. Failure case

Znajdź przypadek, w którym pipeline daje wynik błędny, niestabilny lub niepewny. Nie ograniczaj się do pokazania błędu. Ustal, na którym etapie powstaje: akwizycja, preprocessing, reprezentacja, model, próg decyzyjny czy interpretacja. Zaproponuj test, który pozwala odróżnić co najmniej dwie możliwe przyczyny.

Punkt kontrolny 4.5

Raport zawiera failure case, diagnozę źródła błędu oraz kontrolowany test odporności albo generalizacji.

18 6. Ablation i analiza niepewności

porównaj zachowanie systemu bez progu confidence, z progiem oraz z progiem i stabilizacją; omów kompromis między czułością a błędną adaptacją. Użyj tej samej metryki i tego samego zbioru ewaluacyjnego dla obu wariantów. Następnie wskaż warunek, w którym system powinien zwrócić `unknown` lub powstrzymać się od działania.

Punkt kontrolny 5.0

Wniosek z ablation jest ilościowy i uwzględnia alternatywne wyjaśnienia, confoundery oraz granice generalizacji.

19 7. Raport i artefakty

Raport musi zawierać: pytanie badawcze i hipotezę; pochodzenie danych i ground truth; środowisko i parametry; baseline; opis pipeline'u; metryki; tabelę lub wykres; przypadek poprawny; failure case; ablation; niepewność; ograniczenia i krótki wniosek. Zachowaj kod i artefakty potrzebne do odtworzenia wyniku.

20 Pytania kontrolne

1. Co w tym laboratorium jest obserwowane bezpośrednio, a co wywnioskowane?
2. Jak powstał ground truth i jaki błąd może zawierać?
3. Jaki confounder może zmienić wynik bez zmiany badanego konstruktu?
4. Czy zastosowana walidacja pozwala wnioskować o nowej osobie?

5. Jak rozpoznasz data leakage?
6. Kiedy system powinien zwrócić `unknown`?
7. Które parametry są niezbędne do reprodukcji wyniku?

21 Warunek zaliczenia

Pipeline musi działać, wynik musi być mierzalny i odtwarzalny, a interpretacja nie może wykraczać poza informacje dostępne w danych. Student powinien umieć wyjaśnić zarówno poprawny wynik, jak i co najmniej jeden przypadek niepowodzenia.