ANALIZA DANYCH

Wykład 1: Wprowadzenie do uczenia maszynowego



dr inż. Bartosz Czech

Katedra Genetyki · Uniwersytet Przyrodniczy we Wrocławiu

bartosz.czech@upwr.edu.pl

Plan i cel

Zakres materiału

1. Wprowadzenie do uczenia maszynowego
2. Przygotowanie danych
3. Uczenie nienadzorowane
4. Uczenie nadzorowane
5. Ocena modelu
6. Modele złożone i cechy
7. Wiarygodność modelu
8. Zastosowania i podsumowanie

Zasady zaliczenia

Zaliczenie

Kurs kończy się projektem zespołowym — pełną analizą wybranego zbioru danych, od danych surowych aż po wnioski.

Nieobecności i konsultacje

Obecność

- dopuszczalne nieobecności: **do 20%** zajęć — przy 8 spotkaniach **najwyżej 1**
- przekroczenie limitu rozpatrywane zgodnie z regulaminem studiów
- usprawiedliwienie — niezwłocznie po ustaniu przyczyny

Konsultacje: poniedziałek 14:00–15:30, pokój 306 w Katedrze Genetyki
(proszę o wcześniejszy kontakt mailowy: `bartosz.czech@upwr.edu.pl`)

Literatura

- James i in., *An Introduction to Statistical Learning* (R / Python) — statistical learning, obciążenie–wariancja (rozdz. 2), podział danych i walidacja krzyżowa (rozdz. 5)
  https://www.statlearning.com/
- Boehmke & Greenwell, *Hands-On Machine Learning with R* — proces modelowania, przeuczenie, ocena (rozdz. 1–2)
  https://bradleyboehmke.github.io/HOML/
- Kuhn & Silge, *Tidy Modeling with R* — podział danych, resampling (ekosystem tidymodels, poza kursem)
  https://www.tmwr.org/
- Wickham i in., *R for Data Science* — fundament pracy z danymi w R
  https://r4ds.hadley.nz/
- VanderPlas, *Python Data Science Handbook* — NumPy, pandas, scikit-learn: fundament pracy z danymi w Pythonie
  https://jakevdp.github.io/PythonDataScienceHandbook/
- scikit-learn — *User Guide* — kanoniczna referencja metod ML w Pythonie
  https://scikit-learn.org/stable/user_guide.html

Plan wykładu

1. Uczenie maszynowe w analizie danych biologicznych
2. Zbiór danych: mikrobiom w raku jelita grubego
3. Rodzaje uczenia: nadzorowane i nienadzorowane
4. Język i dane: cechy, cel, macierz X i wektor y
5. Uczenie modelu, źródła błędów i podział danych
6. Proces analizy danych i zasada braku przecieku
7. Narzędzia i organizacja pracy

Coraz więcej danych w biologii

Jeden eksperyment sekwencjonowania — tysiące zmiennych na próbkę:

- transkryptom — ekspresja dziesiątek tysięcy genów
- metagenom — liczebności setek gatunków drobnoustrojów
- warianty genetyczne — miliony pozycji w genomie

Konsekwencja

Zmiennych często znacznie więcej niż próbek. Metody „jedna zmienna naraz” przestają wystarczać.

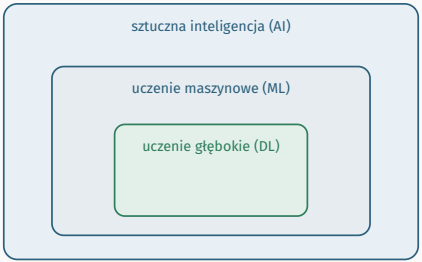
Uczenie maszynowe (*machine learning*)

Uczenie maszynowe to budowanie modeli, które **uczą się wzorców z danych**, zamiast być zaprogramowane regułą wprost — i wykorzystują je do predykcji dla nowych danych albo do opisu struktury danych.

Kluczowe słowa:

- **wzorzec** — regularność w danych; w uczeniu nadzorowanym: zależność między cechami a wielkością przewidywaną
- **nowe dane** — model ma działać poza zbiorem, na którym się uczył

Sztuczna inteligencja, uczenie maszynowe, uczenie głębokie



- **AI** — najszersze pojęcie: systemy realizujące zadania wymagające „inteligencji”
- **ML** — poddziedzina AI: reguły wyprowadzane z danych, nie programowane wprost
- **DL** — poddziedzina ML: sieci neuronowe o wielu warstwach

Programowanie klasyczne a uczenie maszynowe



Dwa różne pytania

Statystyka klasyczna i uczenie maszynowe odpowiadają na odmienne pytania:

Wnioskowanie

Czy dany czynnik wywiera wpływ i jak jest on silny?

Predykcja

Czy na podstawie danych można przewidzieć wynik dla nowej próbki?

Predykcja a wnioskowanie

	Statystyka klasyczna	Uczenie maszynowe
Pytanie	Czy efekt istnieje?	Czy potrafię przewidzieć?
Oceniamy	istotność, wielkość efektu	skuteczność na nowych danych
Typowy błąd	mylenie korelacji z przyczyną	model, który tylko *pamięta* dane treningowe

Kurs skupia się na **prawej kolumnie**: budowa modeli i ocena na nowych danych.

Przesłanki stosowania uczenia maszynowego

Dobre przesłanki

- wiele cech, złożone zależności
- cel: trafna predykcja
- dostatecznie dużo przykładów

Ostrożnie

- bardzo mało próbek
- potrzebne wyjaśnienie przyczyny
- dane niskiej jakości

Zasada

Uczenie maszynowe nie zastępuje myślenia o danych — je wspomaga. Model jest tak dobry, jak dane i pytanie, które mu postawimy.

Uczenie maszynowe w naukach biologicznych

Metody tego kursu działają dziś w praktyce badawczej i klinicznej:

- **przewidywanie struktury białka** z sekwencji (AlphaFold)
- **klasyfikacja wariantów** na patogenne / łagodne
- **histopatologia** — wykrywanie zmian nowotworowych na preparatach
- **mikrobiom jako biomarker** — sygnatury składu w chorobach jelita
- **wiek biologiczny** z profilu metylacji DNA (zegary epigenetyczne)

Wspólny mianownik

Wiele cech naraz, złożone zależności, cel = trafna predykcja dla nowej próbki. Dokładnie ten schemat przejdziemy na danych mikrobiomowych.

Ograniczenia uczenia maszynowego

Nie zastąpi

- dobrego pytania badawczego
- kontroli jakości danych
- wiedzy dziedzinowej o biologii

Nie udowodni

- przyczynowości (tylko zależność)
- mechanizmu biologicznego
- że sygnał nie jest artefaktem technicznym

Model znajdzie **jakiś** wzorzec niemal zawsze — także w danych czysto losowych. Zadaniem analityka jest odróżnić wzorzec rzeczywisty od przypadkowego.

Zbiór danych

Przykład przewodni: rak jelita grubego

> Wykłady opieramy na jednym zbiorze danych i jednym zadaniu: na podstawie **mikrobiomu jelitowego** odróżnić osobę z **rakiem jelita grubego** (CRC, *colorectal cancer*) od osoby zdrowej.

- **Dane** — publiczna kohorta `ZellerG_2014` z pakietu `curatedMetagenomicData` (Bioconductor)
- **Cechy (X)** — względne liczebności setek taksonów z sekwencjonowania metagenomicznego próbki kału
- **Cel (y)** — grupa kliniczna: zdrowy / gruczolak / CRC

Mikrobiom a rak jelita grubego

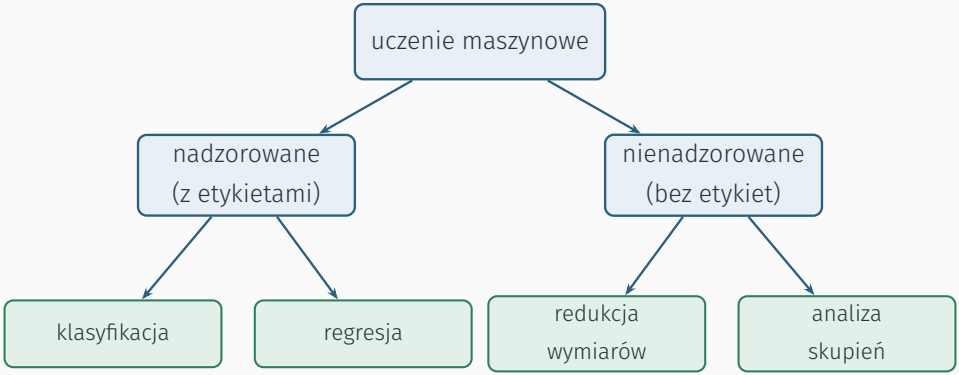
- jelito zasiedlają **setki gatunków** drobnoustrojów (mikrobiom); ich skład wiąże się ze zdrowiem gospodarza — metabolizmem, odpornością, stanem zapalnym — i **zmienia się** w wielu chorobach
- CRC to **jeden z najczęstszych nowotworów**; wczesne wykrycie znacząco poprawia rokowanie
- u pacjentów z CRC obserwuje się **powtarzalne zmiany** składu mikrobiomu — jest więc szansa, że sygnał w danych istnieje

Pytanie badawcze

Czy skład mikrobiomu niesie sygnał wystarczający, by odróżnić pacjentów z CRC od osób zdrowych?

Rodzaje uczenia

Rodzaje uczenia maszynowego



Jedno kryterium podziału: **czy dane mają etykiety**.

Rodzaje pośrednie uczenia

Oprócz uczenia nadzorowanego i nienadzorowanego spotykają Państwo warianty pośrednie:

Rodzaj	Na czym polega
półnadzorowane	część próbek z etykietą, większość bez — etykietowanie bywa kosztowne
samonadzorowane	etykiety tworzone automatycznie z samych danych (np. fragment sekwencji)
ze wzmocnieniem	agent uczy się decyzji przez nagrody — rzadkie w analizie danych *omics*

Na tym kursie

Skupiamy się na dwóch głównych: **nadzorowanym** (trzon kursu) i **nienadzorowanym** (eksploracja). Pozostałe — dla orientacji.

Uczenie nadzorowane

Dysponujemy parami: cechy $\longrightarrow$ znana odpowiedź. Model uczy się odwzorowania z cech na odpowiedź.

Przykłady biologiczne

- czy próbka pochodzi od chorego, czy zdrowego (klasyfikacja)
- jaki jest wiek biologiczny na podstawie metylacji (regresja)
- czy wariant jest patogenny (klasyfikacja)

Klasyfikacja a regresja

	Klasyfikacja	Regresja
Cel	kategoria	liczba
Przykład	CRC / zdrowy	stężenie metabolitu
Wynik	etykieta lub prawdopodobieństwo	wartość ciągła
Miara	dokładność, ROC-AUC, F1	błąd średniokwadratowy

Różnica: **typ zmiennej celu** (a więc i sposób oceny). Nasz cel — kategoryczny, więc **klasyfikacja**.

Klasyfikacja binarna i wieloklasowa

Liczba kategorii zmienia trudność problemu i dobór miar:

- **binarna** — dwie klasy (chory / zdrowy); najprostsza do oceny
- **wieloklasowa** — trzy lub więcej kategorii (np. stadia choroby)

Przykład z realnej kohorty CRC

Metadane dzielą próbki na **trzy** stany: zdrowy, gruczolak (zmiana przedrakowa), rak. To naturalny problem **wieloklasowy**; można go sprowadzić do binarnego: rak / kontrola (gruczolaki pominięte — tak w dalszych wykładach) albo rak / nie-rak.

```
table(y)   # surowy study_condition w kohorcie ZellerG_2014
```

```
y
adenoma control     CRC
     42      61      53
```

Uczenie nienadzorowane

Brak etykiet — poszukujemy **struktury** ukrytej w samych cechach.

Redukcja wymiarów

Sprowadzenie setek cech do kilku, przy zachowaniu głównej zmienności (np. PCA). Służy wizualizacji i odszumianiu.

Analiza skupień

Grupowanie próbek podobnych do siebie bez znajomości etykiet — np. odkrycie podtypów w kohorcie.

Nienadzorowane w praktyce biologicznej

Odkrywanie podtypów

Analiza skupień na profilach ekspresji ujawniła podtypy molekularne raka piersi — bez wcześniejszych etykiet.

Mapa próbek

Redukcja wymiarów (PCA, UMAP) pozwala zobaczyć na płaszczyźnie, czy próbki układają się wg grup, partii pomiarowych czy ośrodków.

Skupienia to **hipotezy**, nie wnioski. Metoda zawsze zwróci jakiś podział — także dla danych bez realnej struktury. Trzeba go dopiero zweryfikować.

Problem

Zadanie główne

Uczenie **nadzorowane**, **klasyfikacyjne**: na podstawie składu mikrobiomu jelitowego przewidzieć, czy próbka pochodzi od pacjenta z rakiem jelita grubego (CRC), czy od osoby zdrowej.

Język i dane

Wspólny słownik

Pojęcie	Znaczenie
próbka / obserwacja	pojedynczy pacjent (wiersz danych)
cecha (*feature*)	jedna zmienna objaśniająca, np. liczebność taksonu (kolumna)
zmienna celu (*target*)	wielkość przewidywana
macierz cech X	tabela: próbki $\times$ cechy
wektor celu y	kolumna etykiet
etykieta	wartość celu dla jednej próbki

Konwencja: X — macierz (wiele cech), y — pojedynczy wektor.

Macierz cech i wektor celu

Model uczy się odwzorowania z cech (X) na cel (y).

- X ma wymiar $n \times p$: n próbek (wierszy), p cech (kolumn)
- y ma długość n: po jednej etykiecie na próbkę

Sytuacja $p \gg n$

W danych biologicznych cech (p) bywa znacznie więcej niż próbek (n).

Typy cech

Typ	Opis	Przykład
ciągła	wartość liczbowa	liczebność taksonu
zliczeniowa	liczba całkowita nieujemna	liczba odczytów
kategoryczna	jedna z kilku kategorii	płeć, ośrodek
binarna	dwie kategorie	obecność/brak

Typ cechy decyduje o jej przygotowaniu — kategorie trzeba zakodować liczbowo.

Struktura danych kursu

	takson_1	takson_2	$\cdots$	takson_p	**grupa**
pacjent_1	3,2	0,0	$\cdots$	0,1	CRC
pacjent_2	0,8	4,1	$\cdots$	0,0	zdrowy
pacjent_3	0,0	0,0	$\cdots$	1,2	CRC

macierz cech X (udziały taksonów w %) | wektor celu y (kolumna *grupa*)

Liczne zera — typowe dla mikrobiomu, nie błąd.

Pierwszy przegląd realnych danych

Zanim cokolwiek policzymy — poznajemy kształt zbioru (kohorta CRC):

```
dim(X)                      # cała kohorta: wiersze = próbki, kolumny = taksony
dim(X[y != "adenoma", ])    # zadanie: kontrola i rak (bez gruczolaków)
```

```
[1] 156 652
[1] 114 652
```

Odczyt

- cała kohorta (z gruczolakami): 156 próbek
- zadanie kontrola–rak: **114 próbek**, **652 cechy** $\Rightarrow p/n \approx 5{,}7$
- więcej cech niż próbek ($p > n$) — główne źródło trudności w kursie

Dane kompozycyjne

Względne liczebności jednej próbki sumują się do (prawie) 100%:

```
round(rowSums(X)[1:3], 1)   # suma udziałów na próbkę (%)
```

```
CCIS00146684ST-4-0 CCIS00281083ST-3-0 CCIS02124300ST-4-0
              99.1               100.0               99.9
```

Drobne odchylenia od 100 to efekt zaokrągleń i frakcji niesklasyfikowanej.

Konsekwencja kompozycyjności

Wzrost udziału jednego taksonu wymusza spadek udziałów pozostałych — nawet gdy ich bezwzględna liczba się nie zmienia. Surowe udziały dają **korelacje pozorne**; stąd potrzeba przekształceń (do nich wrócimy w dalszej części kursu).

Zera w danych mikrobiomowych

```
mean(as.matrix(X) == 0) * 100    # odsetek zer w całej macierzy
```

```
[1] 81.8
```

- ponad **cztery piąte** wartości to zera — macierz jest **rzadka**
- zero zwykle znaczy „taksonu nie wykryto u tego pacjenta”, nie „pomiar nieudany”
- to wynik pomiaru, a nie brak danych (`NA`)

Dlaczego to ważne

Interpretacja zer wpływa na dobór przekształceń i miar odległości. Potraktowanie ich jak braków danych byłoby błędem.

Uczenie modelu

Model jako funkcja

Model to funkcja, która cechom próbki przypisuje przewidywanie (predykcję):

- x — wektor cech jednej próbki
- $\hat{y}$ — przewidywanie (etykieta lub prawdopodobieństwo)
- f — reguła wyuczona z danych treningowych

Uczenie = **dobór** f tak, by $\hat{y}$ było najbliższe y na treningu.

Uczenie jako minimalizacja błędu

Funkcja straty (*loss function*)

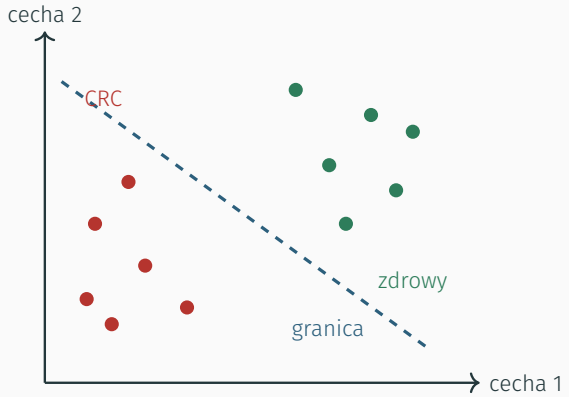
Miara rozbieżności między przewidywaniem $\hat{y}$ a prawdą y. Uczenie to poszukiwanie takich parametrów modelu, które tę stratę minimalizują.

- mała strata na treningu — model dobrze pamięta dane, które widział
- to jednak **nie** jest cel: chcemy skuteczności na *nowych* danych

Zbiór treningowy — próbki, na których model się uczy. **Zbiór testowy** — próbki odłożone na bok, których model nie widzi podczas uczenia; służą do sprawdzenia działania na nowych danych.

Granica decyzyjna

Klasyfikator dzieli przestrzeń cech na obszary przypisane do klas:



Cel: granica najlepiej oddzielająca klasy — działająca też dla nowych punktów.

Niedouczenie i przeuczenie

Niedouczenie

Model zbyt prosty — nie ujmuje wzorca. Słaby i na treningu, i na teście.

Dobre dopasowanie

Model ujmuje wzorzec, pomija szum. Dobry na treningu *i* na teście.

Przeuczenie

Model zbyt złożony — uczy się szumu. Znakomity na treningu, słaby na teście.

Cel: znaleźć złożoność **pośrednią** — wystarczającą, lecz nie nadmierną.

Przeuczenie — demonstracja

Dopasowujemy wielomiany rosnącego stopnia do tych samych danych; błąd liczymy na treningu i na niezależnym teście jako **błąd średniokwadratowy** (MSE) — średnią kwadratów różnic między wartością przewidzianą a rzeczywistą (im mniejszy, tym lepiej).

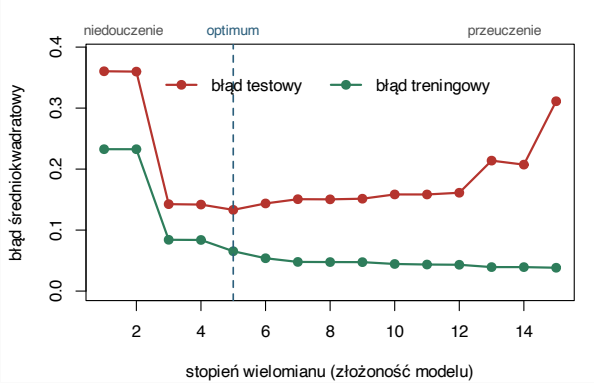
```
set.seed(7)                          # y = sin(2*pi*x) + szum
x_tr <- sort(runif(50));   y_tr <- sin(2*pi*x_tr) + rnorm(50, 0, .35)
x_te <- sort(runif(2000)); y_te <- sin(2*pi*x_te) + rnorm(2000, 0, .35)
tab <- data.frame(stopien = c(1, 2, 3, 5, 9, 15))
for (i in seq_len(nrow(tab))) {
  m <- lm(y_tr ~ poly(x_tr, tab$stopien[i]))
  tab$blad_treningowy[i] <- mean((y_tr - predict(m))^2)
  tab$blad_testowy[i]    <- mean((y_te - predict(m, data.frame(x_tr = x_te)))^2)
}
print(round(tab, 3), row.names = FALSE)
```

Przeuczenie — wynik

stopien	blad_treningowy	blad_testowy
1	0.233	0.360
2	0.233	0.360
3	0.084	0.143
5	0.065	0.133
9	0.047	0.151
15	0.038	0.311

Błąd treningowy **nie rośnie** (od 0,233 do 0,038); testowy osiąga minimum przy stopniu **5**, a następnie znów rośnie.

Przeuczenie — ujęcie na wykresie



- błąd treningowy — nie rośnie
- błąd testowy — krzywa **U**: spada, osiąga minimum, znów rośnie
- optimum = **pośrednia** złożoność

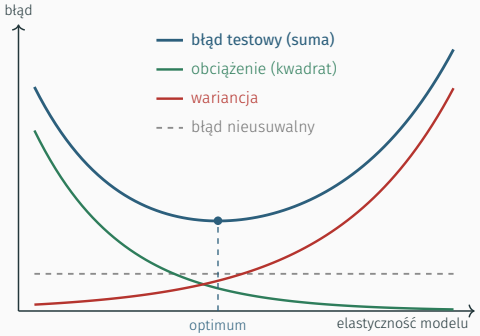
Niski błąd treningowy sam w sobie nic nie mówi o jakości modelu.

Obciążenie i wariancja

Obciążenie i wariancja (*bias*, *variance*)

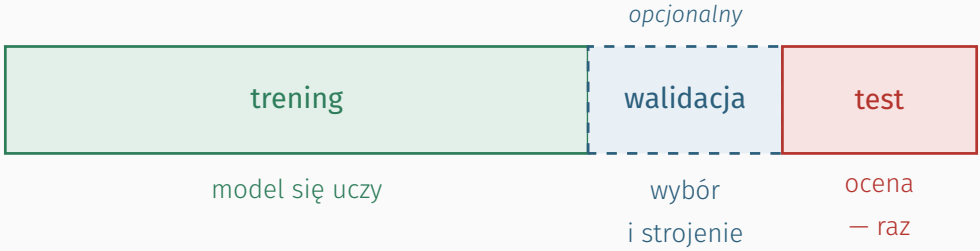
Dla błędu średniokwadratowego (regresja) oczekiwany błąd testowy to suma trzech składników; w klasyfikacji ten sam kompromis obowiązuje jakościowo:

- **obciążenie** — błąd systematyczny: model zbyt prosty, by uchwycić zależność
- **wariancja** — czułość modelu na konkretną próbkę treningową
- **błąd nieusuwalny** — szum w danych, którego żaden model nie usunie



Proste modele: duże obciążenie, mała wariancja (niedouczenie). Złożone: małe obciążenie, duża wariancja (przeuczenie).

Podział na zbiory danych



- podział **przed** jakimkolwiek liczeniem
- zawsze zbiór **treningowy** i **testowy**
- **walidacyjny** (ramka przerywana) — tylko czasem, do wyboru i strojenia
- test — używany **raz**, na samym końcu

Uzasadnienie trzech zbiorów

Problem

Gdy wybieramy model i stroimy go, patrząc na zbiór testowy, zaczynamy się do niego **dopasowywać** — i test przestaje być uczciwy. Dzieje się to nawet bez złej woli, przez wielokrotne zaglądanie.

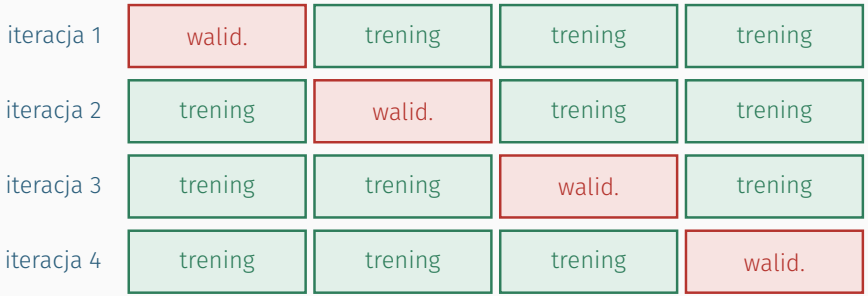
- trening — model dobiera **parametry** (wielkości wyznaczane podczas uczenia, np. współczynniki)
- walidacja — porównanie modeli i wybór **hiperparametrów** (ustawień wybieranych przed uczeniem, np. siła kary, stopień wielomianu)
- test — **jedno** końcowe spojrzenie, nietykany do końca

Przy małej liczbie próbek rolę walidacji przejmuje **walidacja krzyżowa**.

Walidacja krzyżowa — intuicja

Walidacja krzyżowa (*cross-validation*)

Zbiór treningowy dzielony na k części; każda kolejno służy do walidacji, pozostałe — do uczenia. Przy małej liczbie próbek zastępuje jednorazowy podział.



Skuteczność = **średnia** z k przebiegów. Każda próbka raz w walidacji.

Przekleństwo wymiarowości

Przekleństwo wymiarowości (*curse of dimensionality*)

Zjawiska pojawiające się przy bardzo wielu cechach: przestrzeń jest słabo wypełniona danymi, a odległości między punktami stają się do siebie podobne.

```r
set.seed(1)
# 200 losowych punktów; stosunek min/max odległości
ratio <- function(p, n = 200) {
  D <- dist(matrix(runif(n * p), n, p))
  min(D) / max(D)
}
dims <- c(2, 10, 100, 1000)
res <- round(sapply(dims, ratio), 3)
data.frame(wymiary = dims, min_do_max = res)
```

wymiary	min_do_max
2	0.003
10	0.126
100	0.639
1000	0.849

Im więcej wymiarów, tym bardziej wszystkie punkty są **tak samo** odległe.

Przekleństwo wymiarowości — konsekwencje

Co się psuje

- „najbliższy sąsiad” przestaje być bliski
- punkty daleko od siebie — przestrzeń słabo wypełniona
- łatwo o przypadkowy wzorzec

Co pomaga

- wybór i filtrowanie cech
- redukcja wymiarów (np. PCA)
- regularyzacja modelu

Gdy cech jest więcej niż próbek (u nas 652 cechy i 114 próbek w zadaniu kontrola–rak), przekleństwo wymiarowości to nie ciekawostka, lecz codzienny problem analizy.

Regularyzacja — kara za złożoność

Regularyzacja (*regularization*)

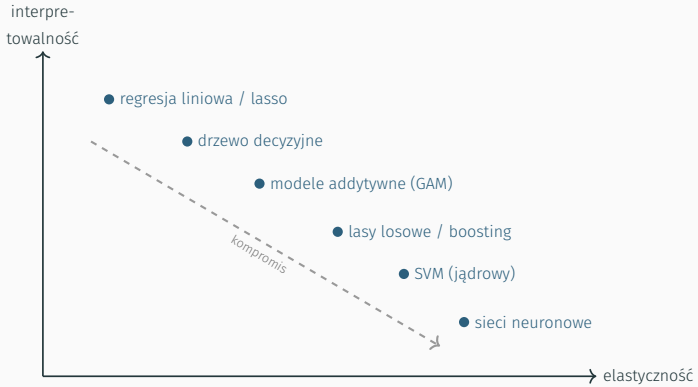
Dodanie do funkcji straty **kary** za nadmierną złożoność modelu — za zbyt duże lub zbyt liczne współczynniki. Każda cecha musi poprawiać dopasowanie na tyle, by przeważyć karę.

- efekt: prostsze, stabilniejsze modele, mniej podatne na przeuczenie
- część metod potrafi **wyzerować** nieistotne cechy (automatyczny wybór cech)
- siła kary to **hiperparametr**: wybiera się ją na danych walidacyjnych lub w walidacji krzyżowej, nigdy na teście

To jedno z głównych narzędzi przeciw przekleństwu wymiarowości (szczegóły — dalej w kursie).

Elastyczność a interpretowalność

Modele różnią się **elastycznością** (jak złożone zależności oddają) i **interpretowalnością** (jak łatwo wyjaśnić ich decyzje) — zwykle jedno rośnie kosztem drugiego:



Interpretowalność w bioinformatyce — równie ważna jak trafność.

Uogólnienie

Uogólnienie

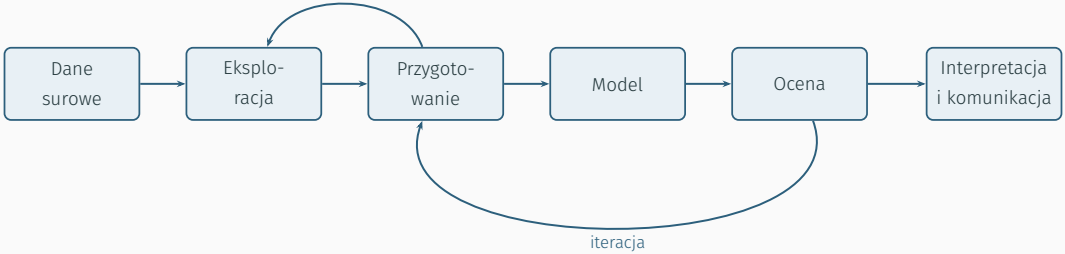
Wartość modelu mierzymy jego skutecznością na danych, których **nie widział** podczas uczenia.

Skuteczność na treningu niczego nie dowodzi.

Podział na zbiory i walidacja krzyżowa służą **uczciwej ocenie** uogólnienia; regularyzacja — jego **poprawie**.

Proces analizy

Cykl analizy danych



Proces jest **iteracyjny** — eksploracja i przygotowanie przeplatają się, a ocena odsyła do przygotowania danych lub wyboru modelu.

Etapy w skrócie

Etap	Co się dzieje
Dane surowe	wczytanie, poznanie struktury, kontrola jakości, podział na zbiory
Eksploracja	rozkłady, zależności, wizualizacja (na zbiorze treningowym)
Przygotowanie	braki, przekształcenia, filtrowanie, kodowanie
Model	wybór metody, uczenie, strojenie
Ocena	skuteczność na danych niewidzianych
Interpretacja	wnioski, ograniczenia, komunikacja wyników

Zasada nadrzędna: bez przecieku

Przeciek danych (*data leakage*)

Wykorzystanie podczas uczenia lub wyboru modelu informacji, która nie byłaby dostępna przy predykcji dla nowych danych — w szczególności informacji ze zbioru testowego. Skutek: zawyżona ocena modelu. Wniosek: model uczy się **wyłącznie** ze zbioru treningowego.

Druga postać — przeciek celu (*target leakage*): wśród cech znajduje się zmienna, która pośrednio zawiera odpowiedź (jest skutkiem albo zapisem rozpoznania), bo jej wartość staje się znana dopiero *po* ustaleniu wyniku.

Przykład: metformina to lek na cukrzycę i zmienia skład mikrobiomu jelitowego. Model przewidujący cukrzycę z tego śladu osiągnie świetny wynik, lecz bezużyteczny — rozpoznaje nie chorobę, a fakt, że pacjent *już* ją leczy.

Przeciek danych — typowe źródła

Najczęstsze źródła przecieku — wszystkie polegają na użyciu informacji ze zbioru testowego **przed** oceną:

Błąd	Dlaczego to przeciek
normalizacja (skalowanie cech) na całym zbiorze	parametry skalowania „widzą” test
wybór cech przed podziałem	cechy dobrane pod całość, w tym test
imputacja (uzupełnianie braków) na całości	brakujące wartości uzupełnione z testu
dostrajanie wg wyniku testowego	test przestaje być niezależny

Reguła: najpierw **podział**, potem obliczenia. Każde przekształcenie dopasowywać tylko na treningu i stosować do testu.

Cztery wyzwania na cały kurs

Wyzwanie	Konsekwencja
$p \gg n$	więcej cech niż próbek $\Rightarrow$ łatwe przeuczenie; kluczowy wybór cech i regularyzacja
kompozycyjność	udziały sumują się do 100% $\Rightarrow$ potrzebne przekształcenia
rzadkość	wiele zer (taksonu nie wykryto u części pacjentów) $\Rightarrow$ ostrożność przy logarytmowaniu i doborze miar odległości
efekt partii (*batch effect*)	próbki przetwarzane w różnych warunkach technicznych $\Rightarrow$ różnicę techniczną łatwo wziąć za biologiczną

Narzędzia i organizacja

Dwa języki: R i Python

Pracujemy **równolegle** w R i Pythonie.

R

Silna tradycja w statystyce i bioinformatyce; bezpośredni dostęp do Bioconductora.

Python

Dominujący w uczeniu maszynowym; spójny ekosystem `scikit-learn`.

Ekosystemy modelowania

Zadanie	R (w kursie)	Python (scikit-learn)
przygotowanie	parametry z treningu (`median`, `scale`), `clr`	transformatory (`preprocessing`, `impute`)
potok: przygotowanie + model	pętla po częściach, każdy krok na części uczącej	`Pipeline`, `ColumnTransformer`
podział danych	`sample` z `set.seed`	`train_test_split`, `StratifiedKFold`
modele	`glm`, `class::knn`, `glmnet`	estymatory
ocena	`table`, `pROC::roc`	`metrics`

Nazwy różne, **idee te same**: potok, podział, model, ocena.

> Inne możliwości (poza kursem): ekosystem `tidymodels` (`recipes`, `rsample`, `parsnip`, `yardstick`) i pakiet `caret` — ta sama logika w jednolitej składni.

Powtarzalność — warunek wiarygodności

Wynik, którego nie da się odtworzyć, nie jest wynikiem. Trzy nawyki od początku:

- **ziarno losowe** (`set.seed`) — ten sam podział i wynik przy każdym uruchomieniu
- **zapis wersji** pakietów i R (`sessionInfo()`, `renv` / środowisko)
- **raport generowany z kodu** (Quarto / R Markdown / notatnik) — nie ręcznie

```
set.seed(42)   # ustalamy ziarno PRZED losowym podziałem danych
```

Reguła praktyczna

Osoba trzecia, mając Państwa repozytorium, powinna odtworzyć wszystkie liczby z raportu. To praktyczny sprawdzian wiarygodności analizy.

Podsumowanie

Przykład ostrzegawczy

Typowa pułapka

Przykład: model o skuteczności **100% na danych widzianych** i zaledwie **55% na nowych pacjentach**. Nie jest to model dobry — to przeuczenie.

Miarą sukcesu w tym kursie jest:

- skuteczność na danych, których model **nie widział**;
- **uczciwe** raportowanie tego wyniku.

Do zapamiętania

- uczenie maszynowe = wyprowadzanie reguł z danych, aby przewidywać lub opisywać strukturę danych
- nadzorowane (z etykietami) vs nienadzorowane (bez); nasz problem to klasyfikacja
- dane: macierz cech X i wektor celu y; u nas więcej cech niż próbek
- główne zagrożenie: **przeuczenie**; zabezpieczenie: uczciwa ocena
- zasada nadrzędna: **model uczy się tylko z treningu**

Na następny wykład

Wykład 2: Przygotowanie danych

- Podział i eksploracja danych przed modelowaniem
- Wartości brakujące i odstające, skale cech, kodowanie
- Specyfika danych mikrobiomowych: zera, kompozycyjność, filtrowanie cech, CLR
- Efekt partii (*batch effect*), przeciek i potok przygotowania danych

Dziękuję za uwagę

dr inż. Bartosz Czech · bartosz.czech@upwr.edu.pl