Klasyfikacja przy użyciu drzewa decyzyjnego

Daria Bartkowiak

26 listopada 2025

1 Przygotowanie danych

1.1 Dyskretyzacja zmiennych ciągłych

Ponieważ algorytm ID3 operuje na zmiennych kategorycznych, wszystkie zmienne ciągłe zdyskretyzowałam przy użyciu funkcji `pd.cut`. Ustaliłam następujące kategorie:

- wiek: 10 przedziałów (0–10, 10–20, . . . , 90–100 lat),
- wzrost: 7 przedziałów (0–140, 140–150, 150–160 . . . , 190–250),
- masa ciała: 8 przedziałów (0–40, 40–50, 50–60 . . . , 100–300),
- ciśnienie skurczowe: 4 przedziały (0–120, 120–140, 140–160, 160–300),
- ciśnienie rozkurczowe: 4 przedziały (0–80, 80–90, 90–100, 100–200).

Kolumny źródłowe usunięto po utworzeniu ich wersji kategorycznych.

1.2 One-hot encoding

Na kolumny kategoryczne (z więcej niż dwoma kategoriami):

- `age_sec`,
- `height_sec`,
- `weight_sec`,
- `ap_hi_sec`,
- `ap_lo_sec`,
- `cholesterol`,
- `gluc`,

zastosowałam one-hot encoding przy użyciu funkcji **get_dummies**:

```
pd.get_dummies(..., drop_first=False, dtype=int)
```

Ostateczna macierz cech `X_final` zawierała wyłącznie zmienne binarne.

2 Implementacja drzewa decyzyjnego

2.1 Struktura węzła

Węzeł (`Node`) zawiera:

- `leaf` — czy węzeł jest liściem,
- `attribute` — indeks cechy wykorzystywanej do podziału,
- `label` — najczęściej występującą klasę w danym podzbiorze,
- `children` — słownik: wartość atrybutu → poddrzewo.

Etykieta większościowa pełni także rolę predykcji domyślnej w przypadku, gdy podczas predykcji pojawi się wartość niewystępująca w zbiorze treningowym.

2.2 Funkcje jakości

Zaimplementowałam klasyczną entropię przedtsawioną na wykładzie:

$$I(U) = -\sum_i f_i \ln(f_i),$$

oraz zysk informacji:

$$\text{InfGain}(d, U) = I(U) - \text{Inf}(d, U),$$

gdzie $\text{Inf}(d, U)$ to średnia entropia podzbiorów po podziale według atrybutu d.

2.3 Budowa drzewa (ID3)

Rekurencyjną metodę `ID3` zaimplementowałam, zgodnie z pseudokodem przedstawionym na wykładzie:

Indukcja drzew decyzyjnych — ID3

Algorithm 1: ID3

Input: Y: zbiór klas, D: zbiór atrybutów wejściowych, $U \neq \emptyset$: zbiór par uczących

if $\forall_{\{x_i, y_i\} \in U}\ y_i == y$ **then**
  return *Liść zawierający klasę* y
if $|D| == 0$ **then**
  return *Liść zawierający najczęstszą klasę w* U
$d = \arg\max_{d \in D} InfGain(d, U)$
$U_j = \{x_i, y_i\} \in U\ :\ x_i[d] = d_j$, gdzie d_j - j-ta wartość atrybutu d
return *Drzewo z korzeniem* d *oraz krawędziami* $d_j, j = 1, 2, \ldots$ *prowadzącymi do drzew: ID3(*Y, $D - \{d\}$, U_1*), ID3(*Y, $D - \{d\}$, U_2*),...*

Rysunek 1: Algorytm ID3 z wykładu.

2.4 Uczenie i predykcja

- Metoda `fit` buduje drzewo od korzenia, na podstawie pełnej listy atrybutów.
- Metoda `predict` przechodzi w dół drzewa zgodnie z wartościami atrybutów danej obserwacji.
- Jeśli brak odpowiedniej gałęzi, zwracana jest etykieta węzła, co zapewnia stabilność działania modelu.

3 Eksperymenty z różnymi seedami

3.1 Schemat eksperymentu

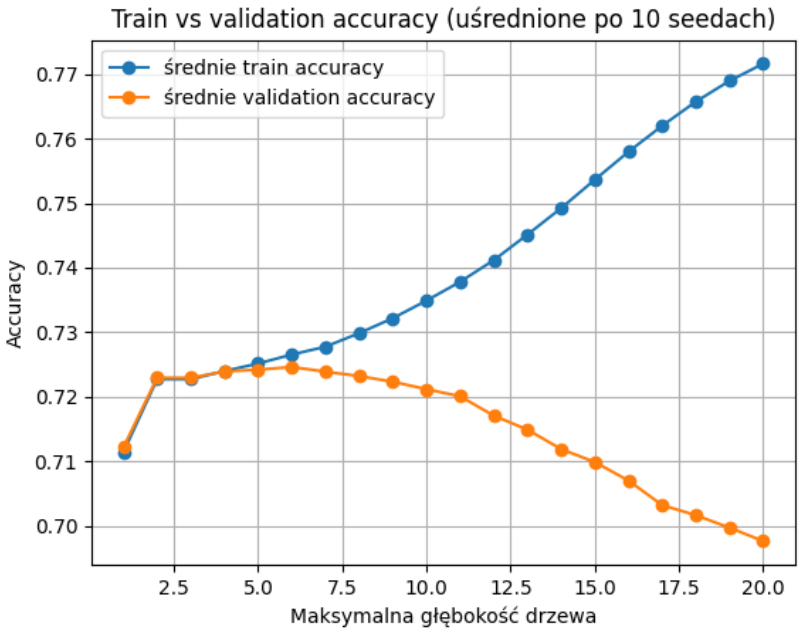
Dla 10 różnych seedów $(0 \ldots 9)$ wykonałam następujące kroki:

1. Stratyfikowany podział danych na:

$$70\% \text{ train}, \quad 15\% \text{ val}, \quad 15\% \text{ test}.$$

2. Dla każdej maksymalnej głębokości $d \in \{1, 2, \ldots, 20\}$:
   - wytrenowałam model na zbiorze treningowym,
   - obliczyłam accuracy na train i val.
3. Na podstawie wyników wyciągnęłam **najlepszą głębokość dla danego seeda** — tę, która dała najwyższe accuracy walidacyjne.
4. W efekcie dostałam listę 10 najlepszych głębokości (po jednej dla każdego seeda).

3.2 Średnie accuracy



Rysunek 2: Średnie accuracy (train i validation) w funkcji maksymalnej głębokości drzewa.

Można zauważyć, że:

- accuracy na treningu monotonicznie rośnie (oczekiwane przeuczenie),
- accuracy walidacyjne osiąga maksimum dla pewnej głębokości, a następnie spada.

3.3 Najczęściej optymalna głębokość

Z wcześniej uzyskanej listy najlepszych głębokości zliczyłam, która z nich pojawiała się najczęściej (moda):

$$\texttt{best\_depth\_mode} = \mathbf{5}, \quad \text{powtórzeń: } 3.$$

Tę głębokość przyjęłam jako **ostateczną optymalną wartość hiperparametru max_depth**.

4 Końcowa ewaluacja na zbiorze testowym

Dla każdej wartości seeda:

1. identycznie podzieliłam dane na train / val / test jak wcześniej,
2. przetreningowałam model z wybraną głębokością modową,
3. na koniec przeprowadziłam ewaluację na zbiorze testowym.

Otrzymałam 10 wyników accuracy, na podstawie których wyliczyłam:

$$\text{średnie accuracy} = \mathbf{0.7251}, \qquad \text{odchylenie standardowe} = \mathbf{0.0038}.$$

5 Wnioski

- Wraz ze wzrostem maksymalnej głębokości drzewa rośnie jego dokładność na zbiorze treningowym, co wskazuje na silną tendencję do przeuczenia.
- Dokładność walidacyjna osiąga najwyższe wartości dla niewielkich głębokości (około 5–7), po czym systematycznie spada, co oznacza pogorszenie zdolności generalizacji modelu.
- Wyniki trenowania i walidacji, uśrednione po 10 różnych losowych podziałach danych, pokazują stabilny i powtarzalny charakter tych obserwacji — niezależnie od wyboru ziarna losowości.
- Wynik na zbiorze testowym świadczy o umiarkowanej, lecz stabilnej jakości klasyfikatora.
- Rozbieżność między rosnącą dokładnością treningową a spadającą walidacyjną jednoznacznie wskazuje, że ID3 w wersji bez mechanizmów przycinania jest podatny na przeuczenie dla większych głębokości.