

Sprawozdanie

Temat: Zadanie 6

Przedmiot: Analiza Danych w Biznesie

Wykonał: Artur Kompała

Treść zadania

Zadanie 6

Dany jest następujący zbiór uczący. Atrybut ryzyko jest atrybutem decyzyjnym.

id	wiek	status	dochód	dzieci	ryzyko
1	25	kawaler	niski	0	wysokie
2	28	żonaty	niski	1	wysokie
3	29	kawaler	wysoki	0	niskie
4	31	kawaler	niski	0	wysokie
5	35	żonaty	średni	1	niskie
6	38	rozwiedziony	wysoki	2	niskie
7	38	rozwiedziony	niski	2	wysokie
8	39	rozwiedziony	wysoki	0	wysokie
9	41	żonaty	średni	1	niskie
10	42	rozwiedziony	średni	4	wysokie
11	45	żonaty	średni	2	niskie
12	48	żonaty	średni	1	niskie
13	56	żonaty	wysoki	2	niskie
14	56	rozwiedziony	wysoki	2	niskie

Należy przeprowadzić proces budowy całego drzewa algorytmem CART. Następnie wygenerować reguły decyzyjne.

Najpierw wykonać "ręcznie" np. w Excelu. Proszę wykonać realizację algorytmu budowania drzewa decyzyjnego do głębokości 3. Proszę wybrać kryterium podziału (w poleceniach jest napisane, że musi to być drzewo CART, ale nie trzeba się tego trzymać - może być CART, ID3 lub C4.5).

Następnie należy dane zaimportować do Python, wykonać analogiczny algorytm, narysować drzewo.

Ręczna realizacja algorytmu budowania drzewa CART

Kodowanie Danych

Przed rozpoczęciem obliczeń zakodowano zmienne kategoryczne:

- Status: kawaler=0, rozwiedziony=1, żonaty=2
- Dochód: niski=0, średni=1, wysoki=2
- Ryzyko: niskie=0, wysokie=1

Poziom 0 – Korzeń (14 próbek)

Rozkład klas: niskie=8, wysokie=6

Obliczenie Gini korzenia:

$$\text{Gini} = 1 - ((8/14)^2 + (6/14)^2) = 0.4898$$

Szukanie najlepszego podziału: Przetestowano wszystkie możliwe podziały dla 4 atrybutów.

Najlepszy wynik:

- Podział: dochod ≤ 0.5
- Gini po podziale: 0.2286
- Lewa gałąź: 4 próbki [0 niskie, 4 wysokie], Gini = 0.0000
- Prawa gałąź: 10 próbek [8 niskie, 2 wysokie], Gini = 0.3200

Poziom 1 – Lewa Gałąź (dochód ≤ 0.5)

Próbki: ID 1, 2, 4, 7 (4 próbki)

Rozkład: [0 niskie, 4 wysokie]

$$\text{Gini} = 1 - ((0/4)^2 + (4/4)^2) = 0$$

Węzeł czysty - wszystkie próbki należą do klasy wysokie. To jest liść.

Poziom 1 – Prawa Gałąź (dochód > 0.5)

Próbki: 10 (ID: 3, 5, 6, 8, 9, 10, 11, 12, 13, 14)

Rozkład: [8 niskie, 2 wysokie]

$$\text{Gini} = 1 - ((8/10)^2 + (2/10)^2) = 0.32$$

Najlepszy podział:

- Podział: dzieci ≤ 3.0
- Gini po podziale: 0.1778
- Lewa: 9 próbek [8 niskie, 1 wysokie]
- Prawa: 1 próbka [0 niskie, 1 wysokie]

Poziom 2 – Lewa Podgałąź (dzieci ≤ 3.0)

Próbki: 9 (ID: 3, 5, 6, 8, 9, 11, 12, 13, 14)

Rozkład: [8 niskie, 1 wysokie]

$$\text{Gini} = 1 - ((8/9)^2 + (1/9)^2) = 0.1975$$

Najlepszy podział:

- Podział: dzieci ≤ 0.5

- Gini po podziale: 0.1111
- Lewa: 2 próbki [1 niskie, 1 wysokie], Gini = 0.5000
- Prawa: 7 próbek [7 niskie, 0 wysokie], Gini = 0.0000

Poziom 2 – Prawa Podgałąź (dzieci >3.0)

Próbki: 1 (ID: 10)

Rozkład: [0 niskie, 1 wysokie]

Gini=0

Węzeł czysty - wszystkie próbki należą do klasy wysokie. To jest liść.

Poziom 3 – Lewa Podgałąź (dzieci <=0.5)

Próbki: 2 (ID: 3, 8)

Rozkład: [1 niskie, 1 wysokie]

Gini = 0.5

Węzeł liść - max_depth=3

Klasa dominująca: niskie

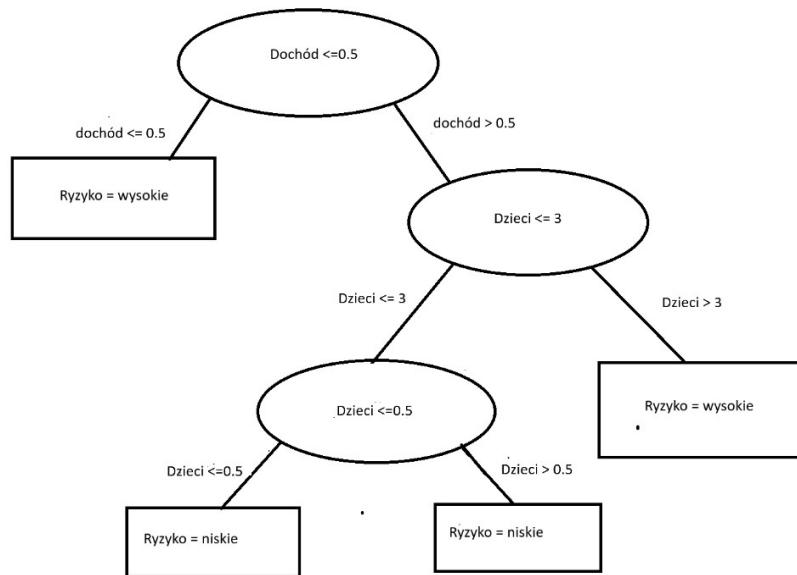
Poziom 3 – Prawa Podgałąź (dzieci > 0.5 i <= 3.0)

Próbki: 7 (ID: 5, 6, 9, 11, 12, 13, 14)

Rozkład: [7 niskie, 0 wysokie]

Gini = 0

Węzeł czysty - wszystkie próbki należą do klasy niskie. To jest liść.



Na podstawie zbudowanego w sprawozdaniu drzewa CART o głębokości 3 wyekstrahowano następujące reguły decyzyjne:

Reguła 1:

IF Dochód ≤ 0.5

THEN Ryzyko = WYSOKIE

Liczba próbek: 4 (wszystkie próbki z klasy wysokie)

Dokładność: 100%

Reguła 2:

IF Dochód > 0.5 AND Dzieci ≤ 0.5

THEN Ryzyko = NISKIE

Liczba próbek: 2 (1 niskie, 1 wysokie)

Dokładność: 50%

Dominująca klasa w węźle: NISKIE

Reguła 3:

IF Dochód > 0.5 AND Dzieci ≤ 3.0 AND Dzieci > 0.5

THEN Ryzyko = NISKIE

Liczba próbek: 7 (wszystkie próbki z klasy niskie)

Dokładność: 100%

Reguła 4:

IF Dochód > 0.5 AND Dzieci > 3.0

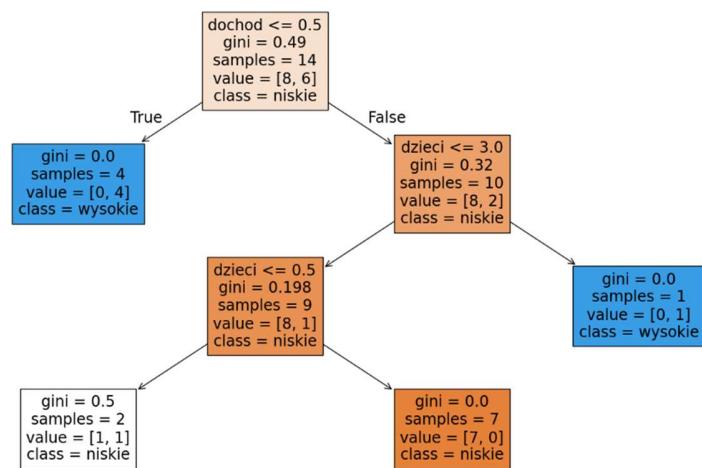
THEN Ryzyko = WYSOKIE

Liczba próbek: 1 (próbka z klasy wysokie)

Dokładność: 100%

Python realizacja algorytmu budowania drzewa CART

Poniżej przedstawiony jest rysunek wygenerowany przy użyciu języka Python.



Kod odpowiedzialny za realizację algorytmu budowania drzewa CART znajduje się w przesłanych plikach.

Wnioski

Drzewo wygenerowane automatycznie w Pythonie okazało się zgodne z drzewem obliczonym ręcznie zarówno struktura podziałów, jak i klasy końcowe w liściach były takie

same. Oznacza to, że przeprowadzone przeze mnie ręczne obliczenia są poprawne, a implementacja algorytmu CART w Pythonie działa zgodnie z oczekiwaniami.