Katedra Sztucznej Inteligencji i Matematyki Stosowanej
WI ZUT Szczecin

Sztuczna inteligencja i maszynowe uczenie w systemach interaktywnych

Joanna Kolodziejczyk
jkolodziejczyk@zut.edu.pl
pokój nr 27 WI1
konsultacje: czwartki 12:00 - 14:00

October 20, 2021

Plan wykładu

Czym są drzewa decyzyjne

Jak budować drzewo?
- Działanie podstawowe
- Tworzenie drzewa - przykład

Algorytm ID3
- Ważne cechy

Algorytm C4.5

CART (Classification and Regression Tree)

Definicje i zastosowanie

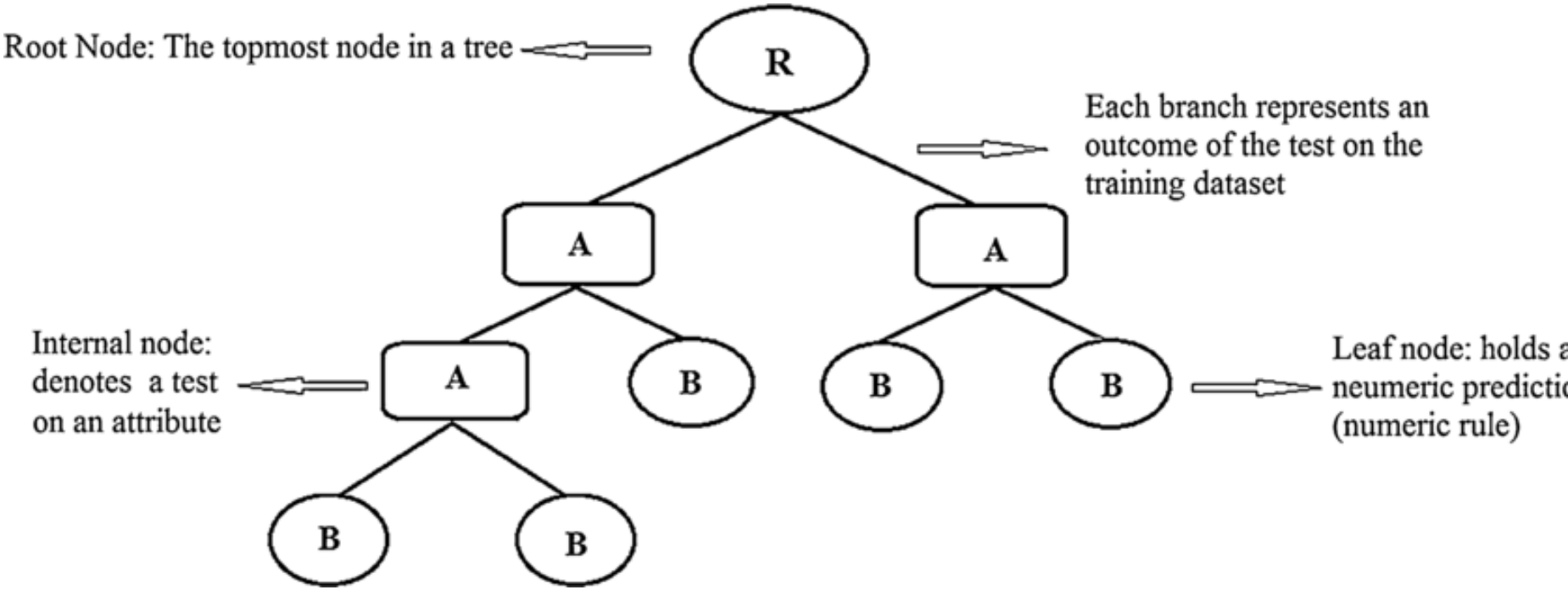
Drzewa decyzyjne

Decision Trees wykorzystują logikę i matematykę do generowania reguł, zamiast tworzyć je na podstawie intuicji eksperta. Są wykorzystywane do rozwiązywania problemu klasyfikacji.

Drzewo konstrukcja

- Korzeń
- Węzły - zmienne
- Krawędzie - wartości zmiennych
- Liście - klasy

Ogólny schemat drzewa decyzyjnego
Schemat



[1] https://datascience.eu/pl/matematyka-i-statystyka/drzewo-decyzyjne/

Definicje i zastosowanie

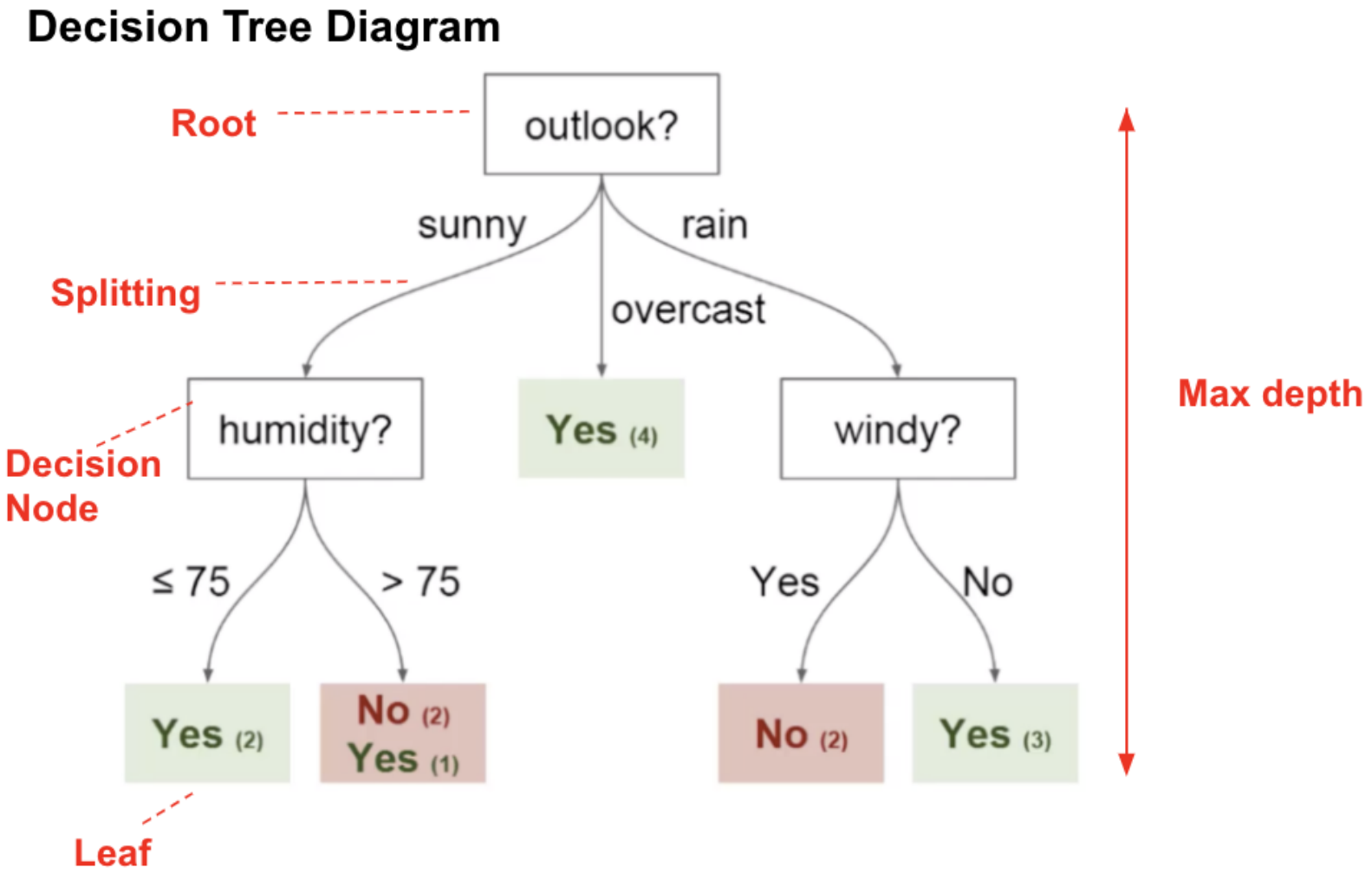
Definicje

- Klasyfikacja - wyznaczenie reguł decyzyjnych opisujących sposoby przypisywania obiektów do wyróżnionych klas.
- Dzięki drzewu decyzyjnemu, zbudowanemu na podstawie danych empirycznych, można sklasyfikować nowe obiekty, które nie brały udziału w procesie tworzenia drzewa.
- Przejście do korzenia do liścia tworzy regułę klasyfikacyjną. Pomiędzy węzłami (na krawędzi) jest łącznik logiczny „AND”.

Przykład zbioru

a1	a2	a3	a4	dec
outlook	temp.	humid.	windy	play
sunny	mild	80	False	yes
sunny	hot	75	True	no
overcast	hot	77	False	yes
rain	cool	70	False	yes
overcast	cool	72	True	yes
sunny	mild	77	False	no
sunny	cool	70	False	yes
rain	mild	69	False	yes
sunny	mild	65	True	yes
overcast	mild	77	True	yes
overcast	hot	74	False	yes
rain	mild	77	True	no
rain	cool	73	True	no
rain	mild	78	False	yes

Schemat drzewa decyzyjnego do zbioru danych
Przykład



[2]https://www.kdnuggets.com/2020/02/decision-tree-intuition.html

Inny przykład

Nr	zwiedzanie zamku	kupiony kilt	whiskey (ile butelek)	Turysta
1	Yes	Yes	4	Yes
2	Yes	No	1	No
3	No	No	3	Yes
4	No	Yes	4	Yes
5	No	Yes	2	No

Jak mogłoby wyglądać drzewo dla tych danych?

Drzewo - omówienie

Struktura

- Korzeń to „Zwiedzanie zamku”
- Liście są wynikiem klasyfikacji, w tym przypadku, czy ktoś jest turystą; tak czy nie.
- Drzewo ma głębokość 3, co oznacza, że najdłuższa gałąź ma 3 poziomy podziałów.
- Wielkość drzewa jest określana na podstawie liczby węzłów. 6 liści - na 5 obserwacji.
- „kupiony kilt” występuje dwa razy.
- Reguła: IF „zwiedzanie zamku” YES AND „kupiony kilt” YES THEN turysta YES.

Opisane drzewo ma więcej wyników niż obserwacji, co oznacza, że prawdopodobnie zawiera zbędne informacje.

Reguła dla rekordu - brak generalizacji

Jeśli każdy rekord w zbiorze danych jest unikalny, zawsze można wytworzyć tyle reguł, ile jest rekordów, w oparciu o wartości zmiennych.

Przetrenowanie

Takie drzewo nazywa się przetrenowanym. Powinno się unikać takich drzew decyzyjnych. Ważne jest, aby pracować z zestawem treningowo-walidacyjno-testowym.

Generalizacja

Dążenie do opracowania drzew, które mogą być wykorzystane do innych danych niż ze zbioru trenującego.

Proste drzewo dla przykładu

Inne rozwiązanie

- Jeden poziom - głębokość drzewa 1
- W korzeniu jest „whiskey (ile butelek)”
- Jakie wartości na krawędziach?

Uwaga

Drzewo niekoniecznie może odwzorowywać wiedzę o innych przypadkach. Używa tylko jednej zmiennej i może nie być wystarczająco dyskryminacyjne dla innych zbiorów danych.

Zalety

- Drzewo decyzyjne buduje się w sposób rekurencyjny od korzenia do liścia
- Algorytm ID3 (Iterative Dichotomiser 3) jest algorytmem zaproponowanym przez Rossa Quinlana używanym do generowania drzewa decyzyjnego ze zbioru danych. ID3 jest prekursorem algorytmu C4.5.
- Algorytm C4.5 - autor:Ross Quinlan. Drzewa decyzyjne generowane przez C4.5 mogą być wykorzystywane do klasyfikacji i z tego powodu C4.5 jest często określany jako klasyfikator statystyczny.

Algorytm ogólny

1. Zacznij od pojedynczego węzła - reprezentującego cały zbiór treningowy.
2. Jeżeli wszystkie przykłady należą do jednej klasy decyzyjnej, to zbadany węzeł staje się liściem i jest on etykietowany tą decyzją.
3. W przeciwnym przypadku wykorzysta heurystykę do wyboru atrybutu, który najlepiej dzieli zbiór przykładów treningowych.
4. Dla każdego wyniku testu tworzy się jedno odgałęzienie i przykłady treningowe są odpowiednio rozdzielone do nowych węzłów (poddrzew).
5. Rekurencyjnie buduj drzewo dla zbiorów przykładów przydzielonych do poddrzew.
6. Koniec, gdy kryterium stopu jest spełnione.

Algorytm
Pseudokod

Algorithm 1: Buduj_Drzewo (P, d, T)

wejście: P - Zbiór przykładów, dla których ma być zbudowane drzewo.
d – funkcja decyzyjna
TEST - zbiór możliwych testów.
R_t - dziedzina testu $t \in \mathbf{TEST}$.
wyjście: T - zbudowane drzewo.
begin
  if *kryterium_stopu*(P, d) **then**
    T.etykieta = kategoria(P, d);
    return;
  t = wybierz_test(P, TEST);
  T.test = t;
  for $(v \in R_t)$ **do**
    $P_v = \{x \in P | t(x) = v\}$;
    utwórz_nowe_poddrzewo T';
    T.gałąź(v) = T';
    buduj_drzewo (P_v, d, T');
end

Warunki stopu

1. Wszystkie przykłady przydzielone do danego węzła należą do jednej klasy decyzyjne.
2. Nie istnieje atrybut, który może dalej podzielić zbiór przykładów. W tym przypadku, liść jest etykietowany nie jedną wartością decyzji, lecz wektorem wartości zwanym rozkładem decyzji.
3. Wszystkie liście mają założoną z góry przewagę jednej klasy decyzyjnej (np. w żadnym nie ma mniej, niż 1% obiektów z innych klas, niż dominująca).

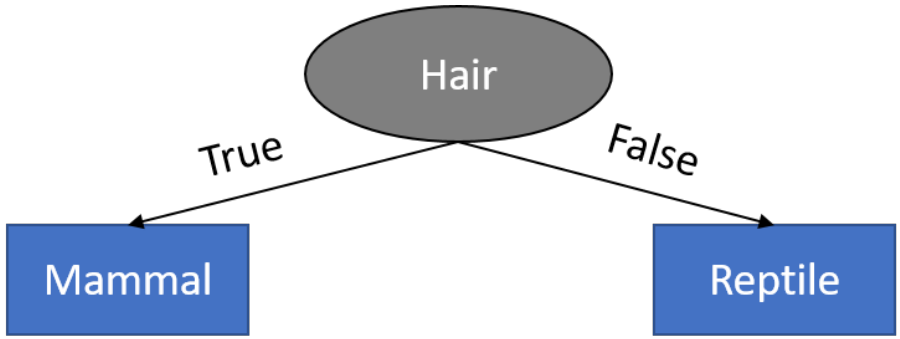
Zbiór treningowy
Klasyfikacja dyskretna

	toothed (ma zęby)	hair (ma włosy)	breathes (oddychają)	legs (ma nogi)	species (gatunek)
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
2	True	False	True	False	Reptile
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

Główne zadanie

- Każdy liść powinien (w najlepszym przypadku) zawierać tylko „Ssaki/Mammal” lub „Gady/Reptile”.
- Cel — znalezienie najlepszego „sposobu” na podzielenie zbioru danych.
- Istnieje atrybut, który dokona właściwego podziału.

Drzewo - wynik
Schemat



	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
8	True	True	True	True	Mammal

	toothed	hair	breathes	legs	species
2	True	False	True	False	Reptile
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
9	False	False	True	True	Reptile

Dyskusja

Kompletność zestawu reguł

- podział jest bardzo prosty, 1 atrybut rozdziela dane, a zbiór treningowy jest całkowicie rozdzielny według wartości tego atrybutu.
- Większość zbiorów danych nie jest tak łatwo rozdzielna i musimy podzielić zbiór danych więcej niż jeden raz.
- Od którego atrybutu powinniśmy zacząć?
- W jakiej kolejności powinniśmy wybierać atrybuty?

Warto zmierzyć „informatywność” atrybutu i użyć go do podziału danych.

Zagadnienie

Jak oblicza się przyrost informacji oraz jak na tej podstawie można zbudować model drzewa.

Entropia
Przyrost informacji

Zastosowanie

Entropia zbioru danych jest wykorzystywana do pomiaru stopnia zanieczyszczenia zbioru danych. Autor: Claude E. Shannon.

Analiza entropia 0

- Koło loterii, zawiera 100 zielonych kulek.
- Zestaw kulek w kole loterii jest całkowicie czysty - ma entropię 0 (możemy powiedzieć, że ma zero zanieczyszczeń).

Analiza entropia różna od 0

- 30 z tych kulek zostało zastąpionych przez czerwone, a 20 przez niebieskie.
- Prawdopodobieństwo otrzymania zielonej piłki spadło z 1,0 do 0,5.
- Zanieczyszczenie wzrosło, czystość spadła, stąd wzrosła entropia.

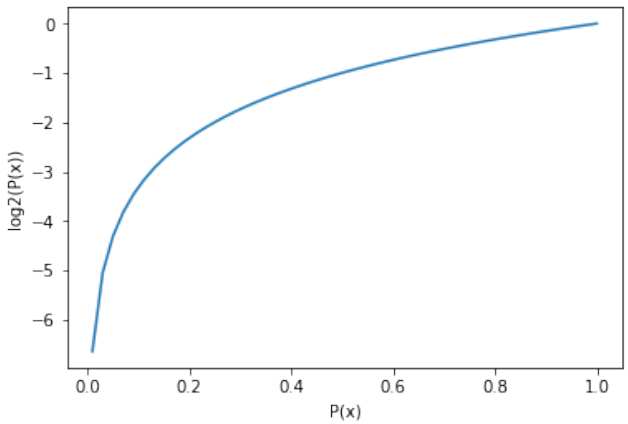
Entropia definicja

Definicja

$$H(x) = -\sum_{k \in target} (P(x = k) * \log_2(P(x = k)))$$

gdzie $P(x = k)$ jest prawdopodobieństwem, że klasa przyjmuje określoną wartość k.

Entropia wykorzystuje $log_2(P(x))$, który ma wartość bliską 0 dla $P(x) = 1$.



Analiza entropii

Dyskusja

Gdy zbiór danych zawiera więcej niż jeden rodzaj elementów, a konkretnie więcej niż jedną wartość klasy decyzyjnej, zanieczyszczenie będzie większe od zera. Tym samym również entropia zbioru danych będzie większa od zera.

Dlatego warto zsumować entropie każdej możliwej wartości docelowej i zważyć ją prawdopodobieństwem (zakładając, że losowo wylosujemy wartości z przestrzeni wartości docelowej)

Przykład z kulkami

- Zielone kule: $H(x = green) = 0.5 * \log_2(0.5) = -0.5$
- Niebieskie kule: $H(x = blue) = 0.2 * \log_2(0.2) = -0.464$
- Czerwone kule: $H(x = red) = 0.3 * \log_2(0.3) = -0.521$

$$H(x) : H(x) = -((-0.5) + (-0.464) + (-0.521)) = 1.485$$

Zbiór treningowy
Klasyfikacja dyskretna

	toothed (ma zęby)	breathes (oddychają)	legs (ma nogi)	species (gatunek)
0	True	True	True	Mammal
1	True	True	True	Mammal
2	True	True	False	Reptile
3	False	True	True	Mammal
4	True	True	True	Mammal
5	True	True	True	Mammal
6	True	False	False	Reptile
7	True	True	False	Reptile
8	True	True	True	Mammal
9	False	True	True	Reptile

Entropia dla zbioru treningowego

Animals

- $P(D = Mammal) = 0.6$
- $P(D = Reptile) = 0.4$
- Entropia dla klasy decyzyjnej:

$$H(D) = -((0.6 * \log_2(0.6)) + (0.4 * \log_2(0.4))) = 0.971$$

Jak obliczyć przyrost informacji?
Dyskusja

Całkowitą czystość/nieczystość - (entropia) zbioru treningowego wynosi około 0,971.

Cel

Znalezienie najlepszego atrybutu pod względem pozyskiwania informacji, który należy wykorzystać do kolejnego podziału danych.

- Używamy każdego atrybutu i dzielimy zbiór danych według ich wartości, a następnie obliczamy entropię po podziale danych.
- Odejmujemy tę wartość od pierwotnie obliczonej entropii, aby zobaczyć jak bardzo podział zmniejsza entropię decyzji.

Przyrost informacji
Information gain

Definicja

$$IG(atr_d, D) = H(D) - H(D|atr_d)$$

Podzielić zbiór danych według wartości każdego z atrybutów, a następnie traktować te podzbiory tak, jakby były „oryginalnym” zbiorem danych pod względem obliczeń entropii. Wzór na obliczanie przyrostu informacji dla każdego atrybutu:

$$IG(atr_d, D) = H(D) - \sum_{t \in atr_d} \left(\frac{|atr_d = t|}{|D|} * H(atr_d = t) \right)$$

Podsumowując, dla każdej cechy/atrybutu sumujemy entropię dla podziału zbioru treningowego według wartości atrybutu i dodatkowo ważymy entropie prawdopodobieństwem ich wystąpienia (proporcja liczby elementów $atr_d = t$ do liczby elementów w zbiorze D).

Działanie - przyrost informacji

	toothed	breathes	legs	species
0	True	True	True	Mammal
1	True	True	True	Mammal
2	True	True	False	Reptile
3	False	True	True	Mammal
4	True	True	True	Mammal
5	True	True	True	Mammal
6	True	False	False	Reptile
7	True	True	False	Reptile
8	True	True	True	Mammal
9	False	True	True	Reptile

toothed == True

toothed == False

After computing the IG of feature *toothed* do this for features *breathes* and *legs*

	toothed	breathes	legs	species
0	True	True	True	Mammal
1	True	True	True	Mammal
2	True	True	False	Reptile
4	True	True	True	Mammal
5	True	True	True	Mammal
6	True	False	False	Reptile
7	True	True	False	Reptile
8	True	True	True	Mammal

	toothed	breathes	legs	species
3	False	True	True	Mammal
9	False	True	True	Reptile

1. Calculate the entrop for toothed == True

↓

2. Calculate the entrop for toothed == False

↓

3. Sum up the entropie of 1. and 2.

↓

4. Subtract this sum from the whole dataset entropy → InfoGain

Przyrost informacji
Przykład dla toothed

$$H(toothed) = (\frac{8}{10}(-((\underbrace{\frac{5}{8} * log2(\frac{5}{8}))}_{\text{toothed = True ; Mammal}} + (\underbrace{\frac{3}{8} * log2(\frac{3}{8})}_{\text{toothed = True ; Reptile}}))) + \frac{2}{10}(-((\underbrace{\frac{1}{2} * log2(\frac{1}{2}))}_{\text{toothed = False ; Mammal}} + (\underbrace{\frac{1}{2} * log2(\frac{1}{2})}_{\text{toothed = False ; Reptile}})))$$

$$\underbrace{\qquad\qquad}_{\text{toothed = True}} \qquad\qquad \underbrace{\qquad\qquad}_{\text{toothed = False}}$$

$$= 0.963547$$

$$IG(toothed, species) = H(species) - H(toothed) =$$

$$= 0.971 - 0.963547 = 0.00745$$

Przyrost informacji
Przykład dla breathes i legs

Breathes

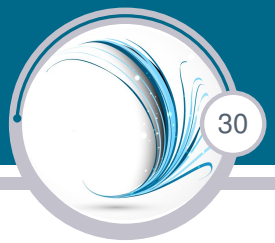
$$H(breathes) = \left(\frac{9}{10} * -\left(\left(\frac{6}{9} * \log_2\left(\frac{6}{9}\right)\right) + \left(\frac{3}{9} * \log_2\left(\frac{3}{9}\right)\right)\right) + \frac{1}{10} * -\left((0) + (1 * \log_2(1))\right)\right) = 0.82647$$

$$IG(breathes, species) = 0.971 - 0.82647 = 0.1445$$

Legs

$$H(legs) = \frac{7}{10} * -\left(\left(\frac{6}{7} * \log_2\left(\frac{6}{7}\right)\right) + \left(\frac{1}{7} * \log_2\left(\frac{1}{7}\right)\right)\right) + \frac{3}{10} * -\left((0) + (1 * \log_2(1))\right) = 0.41417$$

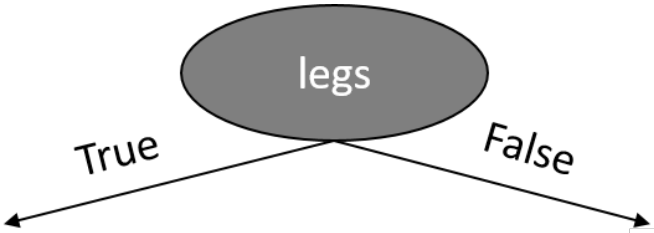
$$IG(legs, species) = 0.971 - 0.41417 = 0.5568$$



Wniosek

Podział zbioru wg wartości atrybutu „legs” skutkuje największym przyrostem informacji i ta cecha powinna być lokalnie węzłem głównym.

Przyrost informacji
Przykład dla toothed

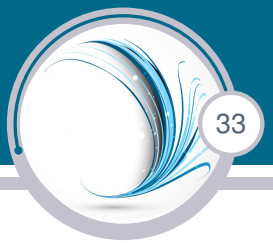


	toothed	breathes	species
0	True	True	Mammal
1	True	True	Mammal
3	False	True	Mammal
4	True	True	Mammal
5	True	True	Mammal
8	True	True	Mammal
9	False	True	Reptile

	toothed	breathes	species
2	True	True	Reptile
6	True	False	Reptile
7	True	True	Reptile

Definicja

- IF Legs == False THEN species = Reptile
- Wartości klasy decyzyjnej dla „Legs == False” to tylko Reptiles
- Powstaje liść, ponieważ uzyskano czysty zbiór (dalsze dzielenie zbioru danych wg pozostałych atrybutów nie doprowadzi do uzyskania innego lub dokładniejszego wyniku, ponieważ klasa zawsze będzie należeć do „Reptile”).
- Atrybut legs nie jest już zawarty w kolejnych podzbiorach, ponieważ został użyty do podziału i nie można go dalej używać.



Kolejny krok

Wykonać te same kroki obliczający przyrost informacji dla podzbioru „legs=True”, gdyż nadal mamy mieszankę różnych wartości klas. Obliczenie przyrostu informacji dla atrybutów toothed i breathes w podzbiorze „legs=True”:

$$H(D) = -\left(\left(\frac{6}{7} * \log_2\left(\frac{6}{7}\right)\right) + \left(\frac{1}{7} * \log_2\left(\frac{1}{7}\right)\right)\right) = 0.5917$$

Entropia klasy dla uzyskanego podzbioru to 0.5917

Dzielenie kolejny atrybut cd.
Algorytm

- toothed:

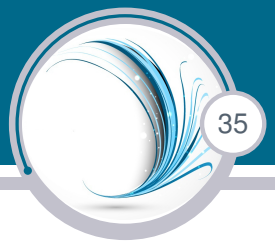
$$\begin{aligned} H(\textit{toothed}) = & \frac{5}{7} * -((1 * \log_2(1)) + (0)) + \frac{2}{7} * \\ & -\left(\left(\frac{1}{2} * \log_2\left(\frac{1}{2}\right)\right) + \left(\frac{1}{2} * \log_2\left(\frac{1}{2}\right)\right)\right) \\ = & \ 0.285 \end{aligned}$$

$$IG(\textit{toothed}, \textit{species}) = 0.5917 - 0.285 = 0.3067$$

- breathes:

$$\begin{aligned} H(\textit{breathes}) = & \frac{7}{7} * -\left(\left(\frac{6}{7} * \log_2\left(\frac{6}{7}\right)\right) + \left(\frac{1}{7} * \log_2\left(\frac{1}{7}\right)\right)\right) + 0 \\ = & \ 0.5917 \end{aligned}$$

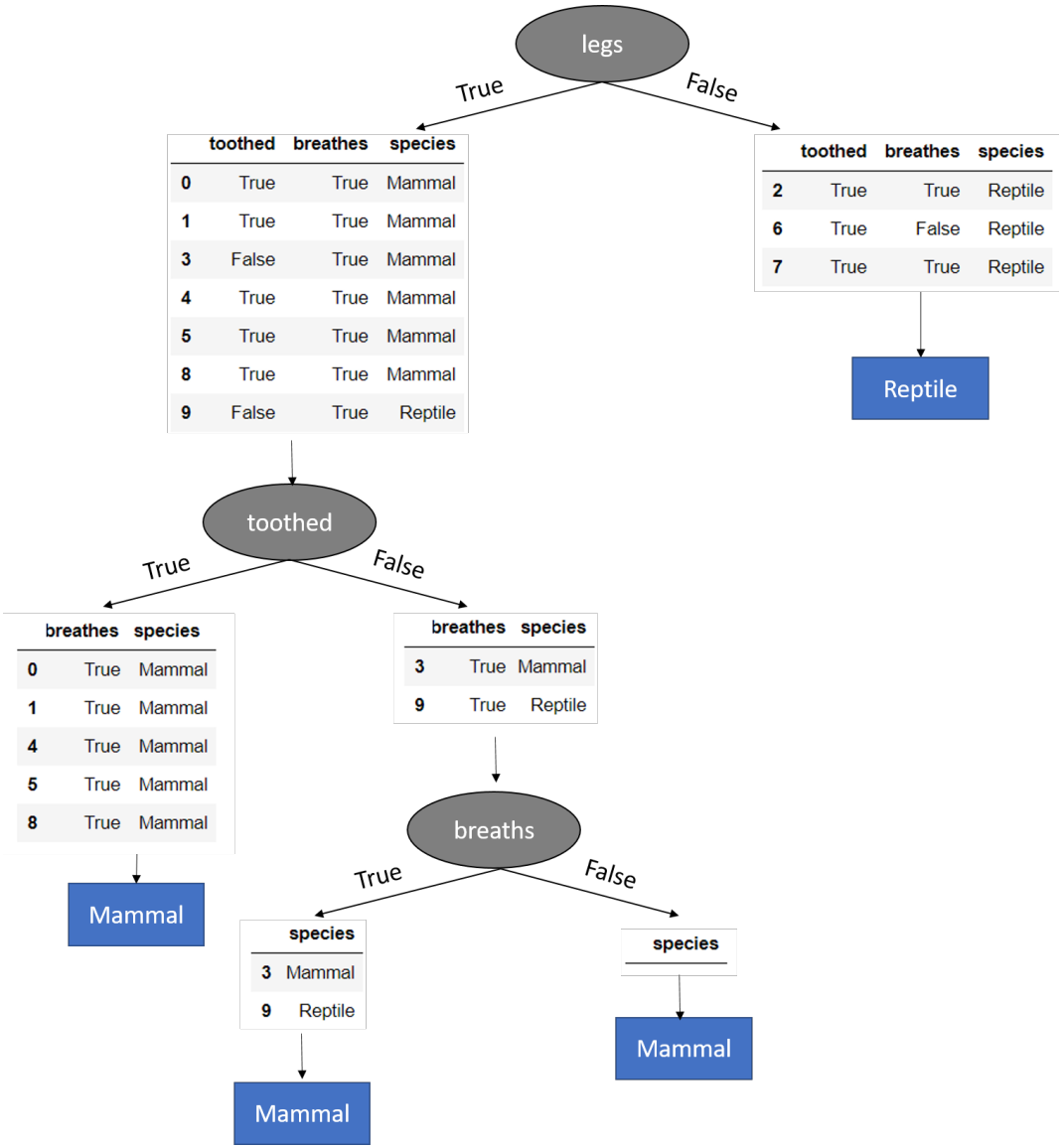
$$IG(\textit{toothed}, \textit{species}) = 0.5917 - 0.5917 = 0$$



Wynik

Zbiór „toothed == False” wciąż zawiera mieszankę wartości, dlatego dokonamy podziału ze względu na atrybut breathes

Drzewo wynikowe



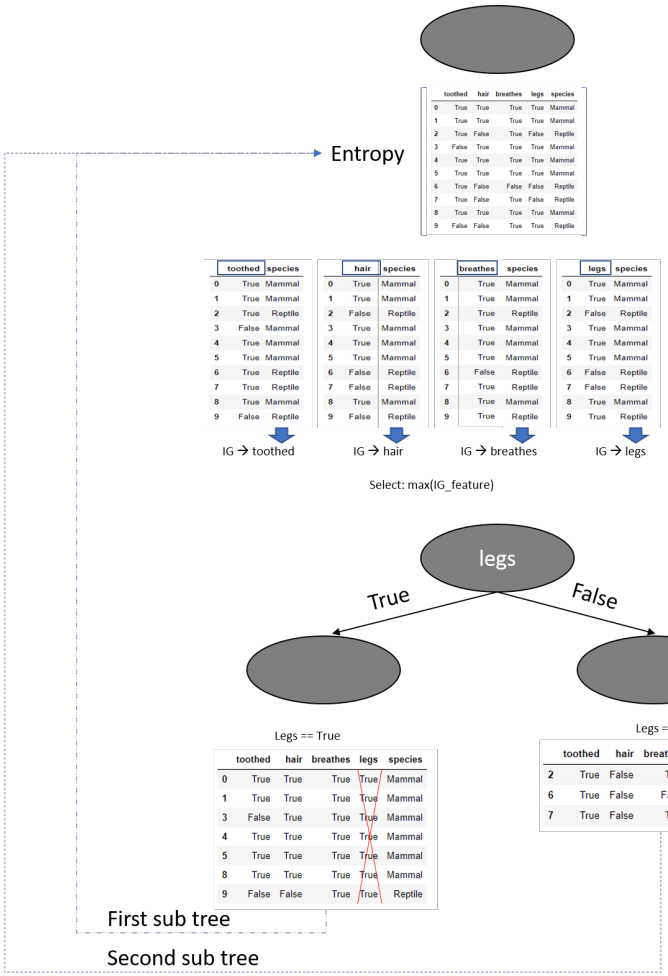
Dyskusja węzła breaths

- Breaths zawiera wyłącznie dane, gdzie *breaths* == *True*.
- breaths == False nie ma żadnych instancji w zbiorze treningowym W tym przypadku zwracamy najczęściej występującą wartość klasy decyzyjnej ze zbioru pierwotnego czyli Mammal.
- Jest to przykład generalizacji na zbiorze treningowym.
- Jeśli weźmiemy pod uwagę drugą gałąź, *breaths* == *True*, wiemy, iż po podzieleniu zbioru danych wg wartości (breaths {True,False}) atrybut breaths musi zostać usunięty.
- Prowadzi to do powstania podzbioru, w którym nie są już dostępne do dalszego podziału żadne atrybuty.
- Przestajemy więc rozwijać drzewo i zwracamy wartość węzła rodzicielskiego (toothed), jaką jest „Mammal”.

Algorytm
pseudokod


```
ID3(D,Feature_Attributes,Target_Attributes)

    Create a root node r

    Set r to the mode target feature value in D

    If all target feature values are the same:
        return r

    Else:
        pass

    If Feature_Attributes is empty:
        return r

    Else:
        Att = Attribute from Feature_Attributes with the largest information gain value
        r = Att

        For values in Att:
            Add a new node below r where node_values = (Att == values)
            Sub_D_values = (Att == values)

            If Sub_D_values == empty:
                Add a leaf node l where l equals the mode target value in D
            Else:
                add Sub_Tree with ID3(Sub_D_values,Feature_Attributes = Feature_Attributes without Att, Target_At
```

Algorytm
działanie



1. Create a root node

2. Calculate the entropy of the whole (sub) dataset

3. Calculate the Information gain of each single feature and pick that feature with the larges Information gain

4. Assign the (root) node the label of the feature with the maximum information gain. Grow for each feature value an outgoing branch and add unlabelled nodes at the end

5. Split the dataset along the values of the maximum information gain feature and remove this feature from the dataset

6. For each of the sub_datasets, repeat steps 2 to 5 until a stopping criteria is satisfied →Here the recursion kicks in

Kryterium wyboru testu
Różne atrybuty

Atrybuty nominalne (symboliczne)

- Test tożsamościowy $a(x) == ?$. Zbiór możliwych wyników testu jest zbiorem wartości testowanego atrybutu a czyli $t(x) = a(x)$.
- Test równościowy: czy $(a(x) = v)?$, gdzie $v \in V_a$. Zbiór wyników testu jest binarny
- Test przynależnościowy: czy $(a(x) \in V)$, gdzie $V \in V_a$. Zbiór wyników testu jest binarny

Atrybuty rzeczywiste (ciągłe)

- Test nierównościowy: czy $(a(x) \geq c)?$, gdzie $c \in R$ jest wartością progową. Zbiór wyników testu jest binarny

Kryterium informacyjne

- Przy wyborze testu zależy nam na ogół na tym, aby uzyskać proste drzewo.
- Najczęściej stosowane kryterium wyboru testu ma charakter teorioinformacyjny.
- Zgodnie z nim wybiera się test, którego zastosowanie dla aktualnego zbioru przykładów daje największy przyrost informacji.
- Nieformalnie, oznacza to preferowanie takiego testu, który wyznacza podział zbioru przykładów P na podzbiory jak najbardziej jednolite pod względem wartości decyzji.

Co odróżnia C4.5 od ID3
Dyskusja

Cechy C4.5

C4.5 wprowadził szereg ulepszeń do ID3. Niektóre z nich to:

- Obsługa atrybutów zarówno ciągłych, jak i dyskretnych - W celu obsługi atrybutów ciągłych C4.5 tworzy próg, a następnie dzieli listę na te, których wartość atrybutu jest powyżej progu oraz te, których wartość jest mniejsza lub równa temu progowi.
- Obsługa danych treningowych z brakującymi wartościami atrybutów - C4.5 pozwala na oznaczenie wartości atrybutów jako ? dla brakujących. Brakujące wartości atrybutów po prostu nie są wykorzystywane w obliczeniach wzmocnienia i entropii.
- Obsługa atrybutów o różnych kosztach.
- Przycinanie drzew po utworzeniu - C4.5 wraca do drzewa po jego utworzeniu i próbuje usunąć gałęzie, które nie pomagają, zastępując je węzłami liści.

CART

Cechy CART

- CART wykorzystuje indeks Gini do tworzenia punktów podziału.
- Służy do generowania zarówno drzew decyzyjnych klasyfikacyjnych, jak i regresyjnych.
- Do rozwiązywania problemów klasyfikacji wieloklasowej (dla klasyfikacji binarnej generuje drzewo binarne).
- Celem jest minimalizacja funkcji kosztu (współczynnika Gini) w każdym węźle. Wybór zmiennych wejściowych/cech, które decydują o konkretnym podziale dla każdego węzła jest zachłanny minimalizujący funkcję kosztu. Czyli rozważana jest pewna liczba punktów podziału z różnymi zestawami zmiennych/cech i wybierany jest ten podział, który skutkuje minimalną wartością wskaźnika Gini (tj. bardziej jednorodnymi podziałami) w danym węźle. Proces ten jest przeprowadzany rekursywnie dla wszystkich węzłów podrzędnych w drzewie.

Gini Index

Definicja

Mierzy prawdopodobieństwo, że dana zmienna zostanie błędnie sklasyfikowana, gdy zostanie wybrana losowo.

$$GI = 1 - \sum_{i=1}^{D} p_i^2$$

Gdzie D - liczba klas, p - prawdopodobieństwo

- Wartość prawdopodobieństwa równa 0 oznacza, że zmienna nie może zostać błędnie sklasyfikowana i jest to możliwe tylko wtedy, gdy mamy tylko jedną klasę wyjściową, tzn. dane są w 100% czyste.
- Wraz ze wzrostem wartości indeksu wzrasta prawdopodobieństwo błędnej klasyfikacji danej zmiennej, ponieważ zwiększa się jej nieczystość.

Gini Index - obliczenia

		Yes	No	Total
Feature 2: Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	3	2	5
	Total	10	4	

Gini (PlayTennis, Outlook=Sunny)
= 1-(⅖)^2 - (⅗)^2 =0.48

Gini (PlayTennis, Outlook=Overcast)
= 1-(4/4)^2 - (0/4)^2=0

Gini (PlayTennis, Outlook=Rainy)
= 1-(⅗)^2 - (⅖)^2 =0.48

The Gigi Index of Outlook (children node)
= 5/14 x 0.48 + 4/14 x 0 + 5/14 x 0.48 = 0.3429

Gini Gain = Gini (parent node) - Gini (children node)

= [1- (10/14)^2 -(4/14)^2] - 0.3429
= 0.4082 - 0.3429
= 0.065

Gini Index - obliczenia

Po obliczeniu wzrostu indeksu Gini dla każdego atrybutu, wybierany zostanie atrybut z największym wzrostem Gini jako węzeł główny. Gałąź z Gini równym 0 jest liściem, podczas gdy gałąź z Gini większym niż 0 wymaga dalszego podziału. Węzły są rozwijane rekurencyjnie, aż wszystkie dane zostaną sklasyfikowane.

Przycinanie drzew decyzyjnych

Przycinanie drzew decyzyjnych jest techniką pozwalającą na zmniejszenie rozmiaru drzewa decyzyjnego poprzez usunięcie jego fragmentów, które w niewielkim stopniu przyczyniają się do osiągnięcia celu klasyfikacji. Dwie popularne metody przycinania to:

- Reduced Error Pruning (bottom-up)
- Cost complexity pruning (top-down)

Wady

Chociaż drzewa decyzyjne są dość intuicyjne i proste w tworzeniu, mają następujące wady.

- Są niestabilne, co oznacza, że niewielka zmiana w danych może prowadzić do dużej zmiany w strukturze optymalnego drzewa decyzyjnego.
- Obliczenia mogą stać się bardzo złożone, szczególnie jeśli wiele wartości jest niepewnych i/lub jeśli wiele wyników jest ze sobą powiązanych.
- Drzewa decyzyjne mają tendencję do zbytniego dopasowywania się do danych treningowych i mogą stać się niedokładne.

Źródła

- https://www.python-course.eu/Decision_Trees.php
- http://edu.pjwstk.edu.pl/wyklady/adn/scb/wyklad10/w10.htm
- EdX course „Predictive Analytics using Machine Learning”
- https://www.kdnuggets.com/2020/02/decision-tree-intuition.html

Dziękuję za uwagę
Czas na pytania ????