

Drzewa decyzyjne

Joanna Góral, Marta Kamińska, Anna Witkiewicz

Spis treści

Wprowadzenie	1
Definicja drzewa decyzyjnego	1
Budowa drzewa decyzyjnego	2
Rodzaje atrybutów	3
Podział drzew decyzyjnych	3
Algorytmy	4
Wstęp	4
<i>CART</i>	4
<i>CHAID</i>	6
<i>ID3</i> , <i>C4.5</i> , <i>C5.0</i>	7
Zastosowanie drzew decyzyjnych	9
Optymalizacja drzew decyzyjnych	9
Podsumowanie	12
Bibliografia	13

Wprowadzenie

Definicja drzewa decyzyjnego

Drzewo decyzyjne to narzędzie analityczne służące do graficznego przedstawienia procesu podejmowania decyzji, uwzględniające różne opcje działania oraz ich potencjalne konsekwencje. Metoda ta jest szczególnie przydatna w sytuacjach obarczonych niepewnością lub ryzykiem, ponieważ pozwala analizować alternatywne ścieżki, ich prawdopodobieństwa oraz związane z nimi koszty i korzyści.

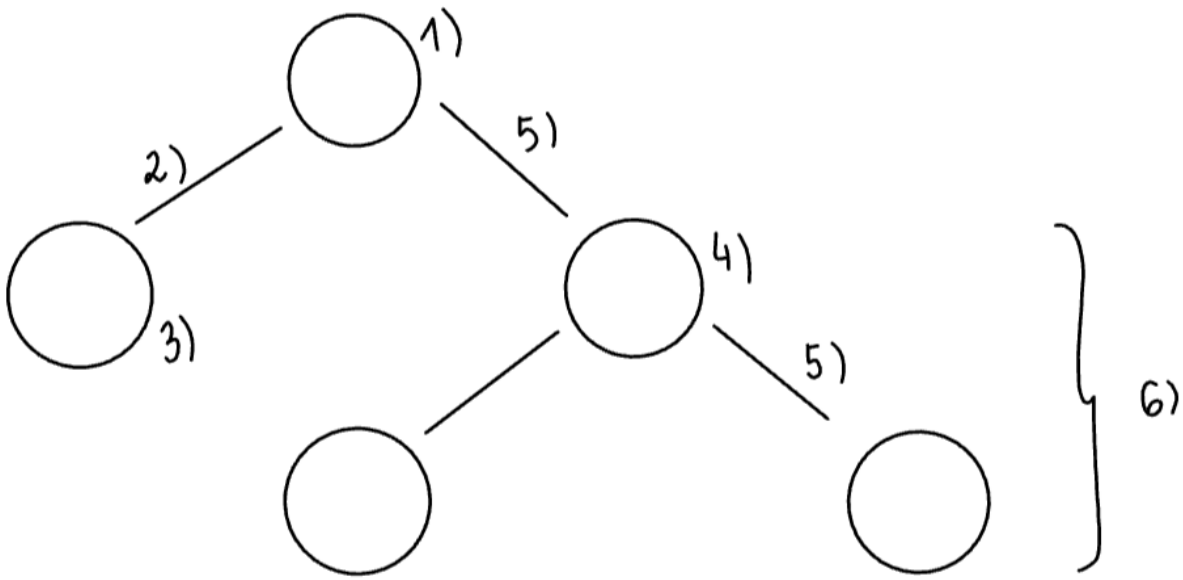


Figure 1: Drzewo

Budowa drzewa decyzyjnego

- 1) **Korzeń** - początkowy węzeł drzewa decyzyjnego, od którego rozpoczyna się proces podziału danych. Jest jedynym wierzchołkiem, z którego wychodzą krawędzie, ale do którego żadna krawędź nie prowadzi. W drzewie decyzyjnym występuje dokładnie jeden korzeń.
- 2) **Krawędź** - element łączący dwa węzły drzewa decyzyjnego, wyznaczający kierunek przepływu informacji między nimi. Krawędzie reprezentują wyniki warunków logicznych zdefiniowanych w węzłach wewnętrznych.
- 3) **Liść** - końcowy węzeł drzewa decyzyjnego, który nie ma dalszych podziałów. Zawiera przewidywaną wartość (w przypadku drzewa regresyjnego) lub klasę (w przypadku drzewa klasyfikacyjnego).
- 4) **Węzeł wewnętrzny** - węzeł, który nie jest liściem. W każdym węźle wewnętrznym sprawdzany jest pewien warunek dotyczący danej cechy obserwacji. Na podstawie jego wyniku wybierana jest jedna z gałęzi prowadząca do kolejnego węzła.
- 5) **Ścieżka** - ciąg krawędzi od korzenia do danego węzła lub liścia, reprezentujący kolejne decyzje podejmowane w modelu.

Przydatne definicje:

- 6) **Poddrzewo** - część drzewa decyzyjnego, która sama w sobie tworzy strukturę drzewa, zaczynającą się od dowolnego węzła wewnętrznego.

Reguła podziału - kryterium, według którego dane są dzielone na podstawie wartości określonej cechy. Reguły podziału decydują, do którego węzła potomnego trafi dana obserwacja.

Drzewo binarne - drzewo decyzyjne, w którym każdy węzeł wewnętrzny ma dokładnie dwie gałęzie (lewa i prawa), reprezentujące dwie możliwe ścieżki wynikające z warunku podziału.

Rodzaje atrybutów

Atrybuty to cechy lub właściwości obiektów, które znajdują się w węzłach drzewa decyzyjnego i stanowią podstawę do podejmowania dalszych decyzji na kolejnych poziomach drzewa. W procesie budowy drzewa decyzyjnego wybierane są kluczowe atrybuty do rozgałęzień - te, które najlepiej dzielą zbiór danych, maksymalizując przewidywalność i precyzję kolejnych decyzji.

Podczas tworzenia drzewa należy najpierw określić, jaki problem chcemy rozwiązać, co pozwala na dobór odpowiednich atrybutów na każdym poziomie drzewa.

Rodzaje atrybutów:

- **atrybuty kateryczne (nominalne)** - polegają na wyborze opcji pochodzących z zamkniętego zbioru możliwości. Nie mają naturalnego porządku, tzn. każda z opcji traktowana jest na równym poziomie.
- **atrybuty numeryczne (liczbowe)** - reprezentują wartości lub przedziały liczbowe, które można porównywać oraz porządkować.
- **atrybuty porządkowe** - mają określony porządek, jednak różnice między poszczególnymi wartościami nie są mierzalne.
- **atrybuty binarne** - przyjmują tylko dwie możliwe wartości, np. tak/nie, prawda/fałsz.

Warto zauważyć, że wiele atrybutów można zamienić na binarne, jeśli przyjmujemy odpowiednie zasady podziału, zwłaszcza gdy rozważamy tylko dwie gałęzie węzła. Przykład:

Czy temperatura jest poniżej zera?

$(-\infty, 0) \rightarrow \text{Prawda}$

$[0, +\infty) \rightarrow \text{Fałsz}$

W tym przypadku przedziały liczbowe zostały zamienione na odpowiadające im wartości logiczne - prawda/fałsz.

Podczas budowy drzewa decyzyjnego możemy wykorzystać różne typy atrybutów na różnych poziomach drzewa, dostosowując je do specyfiki problemu, który rozwiązuje.

Podział drzew decyzyjnych

Drzewa decyzyjne są przydatne do predykcji zarówno zmiennej zależnej, która ma charakter ciągły, jak i zmiennej zależnej o charakterze katerycznym. Na tej podstawie wyróżnia się dwa rodzaje drzew decyzyjnych:

- **Drzewa regresyjne** są używane, gdy zmienna zależna jest ciągła, czyli może przyjmować dowolne wartości liczbowe. Wówczas przewidywanymi wartościami znajdującymi się w liściach drzewa decyzyjnego są zazwyczaj średnie wartości zmiennej zależnej dla obserwacji w danych liściach.
- **Drzewa klasyfikacyjne** są stosowane, gdy zmienna zależna ma charakter kateryczny, czyli przyjmuje wartości z określonego zbioru klas. W takich drzewach każdy węzeł wewnętrzny reprezentuje test na jednej z cech, a gałęzie odpowiadają wynikom tego testu, prowadząc do kolejnych węzłów lub liści. Liście drzewa wskazują przewidywaną klasę dla obserwacji spełniających warunki określone na ścieżce od korzenia do danego liścia.

Algorytmy

Wstęp

Istnieje wiele algorytmów budowy drzew decyzyjnych. W tej części projektu omówimy najbardziej znane, takie jak *CART*, *CHAID* oraz grupa algorytmów *ID3*, *C4.5*, *C5.0*. Warto je omówić ze względu na ich popularność, szerokie zastosowanie oraz różnorodne podejście do konstrukcji drzewa decyzyjnego.

Wszystkie te algorytmy opierają się na schemacie rekurencyjnym, to znaczy wybierają takie zmienne, które najlepiej przewidują lub klasyfikują wynik, i powtarzają ten proces aż do momentu, kiedy spełni się jakiś określony warunek zatrzymujący działanie algorytmu. To co je różni, to właśnie kryteria podziału danych i warunki zatrzymania, ale także rodzaj drzewa, które budują lub typ obsługiwanych zmiennych.

Dla wybranych przez nas algorytmów, przedstawimy potrzebne założenia, opis ich mechanizmu działania oraz poprzemy teorię przykładowym kodem w języku *R*.

CART

Algorytm *CART* został opracowany przez amerykańskich statystyków z Uniwersytetu Stanforda: Leo Breimana, Jerome'a Friedmana, Richarda A. Olshena i Charlesa J. Stone'a w 1984 roku i po raz pierwszy opisany w książce *Classification and Regression Trees*.

CART tworzy wyłącznie drzewa binarne, co odróżnia go od innych metod. Jego główną zaletą jest to, że potrafi pracować zarówno na danych dyskretnych (kategorycznych) jak i ciągłych. Jest również w stanie, zgodnie ze swoją nazwą, budować zarówno drzewa klasyfikacji jak i regresji, używając dwóch różnych kryteriów podziału:

1. **Miara nieczystości (impurity) - współczynnik Giniego** - kryterium stosowane przy tworzeniu drzew klasyfikacyjnych; mierzy prawdopodobieństwo błędnej klasyfikacji losowo wybranego elementu zbioru, przy założeniu, że klasyfikacja jest losowa.

Niech c będzie liczbą klas, a p_i - prawdopodobieństwem wylosowania elementu należącego do klasy i . Wtedy nieczystością Giniego węzła S nazywamy wartość:

$$G(S) = 1 - \sum_{i=1}^c p_i^2.$$

Indeks Giniego to średnia ważona miar nieczystości węzłów potomnych, służąca do oceny jakości danego podziału.

Niech S_L i S_R to zbiory danych odpowiadające lewemu i prawemu dziecku węzła S oraz niech $|S_L|$ i $|S_R|$ są mocami tych zbiorów. Wtedy indeks Giniego dla podziału węzła S można wyrazić wzorem:

$$G_S = \frac{|S_L|}{|S|} G(S_L) + \frac{|S_R|}{|S|} G(S_R).$$

2. **Suma kwadratów wartości resztowych** - kryterium stosowane przy tworzeniu drzew regresji, czyli wtedy, gdy mamy do czynienia z danymi ciągłymi; określa jak bardzo nasze przewidywania różnią się od prawdziwych wartości.

Niech J będzie liczbą rozłącznych obszarów, na które dzielimy przestrzeń predyktorów $X_1, \dots, X_p$ i niech R_j będzie j -tym obszarem dla $j = 1, 2, \dots, J$. Ponadto, niech y_i oznacza wartość zmiennej odpowiedzi dla i -tej obserwacji, a $\hat{y}_{R_j}$ oznacza średnią wartość zmiennej odpowiedzi dla obserwacji treningowych znajdujących się w obszarze R_j . Wówczas sumę kwadratów wartości resztowych wyraża się wzorem:

$$RSS = \sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2.$$

Znając już kryteria podziału, jakich używa algorytm *CART*, możemy krótko opisać jego działanie.

CART buduje drzewo decyzyjne, zaczynając od korzenia i dzieląc zbiór danych w sposób rekurencyjny na dokładnie dwa podzbiory. Na każdym etapie wybiera taki podział, aby indeks Ginię (dla drzew klasyfikacyjnych) lub suma kwadratów wartości resztowych (dla drzew regresji) były jak najmniejsze. Ten proces powtarza się aż do spełnienia warunków zatrzymania, którymi mogą być np. maksymalna głębokość drzewa lub minimalna liczba obserwacji w liściu, lub zbyt mała poprawa nieczystości Ginię po kolejnym podziale.

Na koniec pokażemy przykład działania tego algorytmu, korzystając z biblioteki *rpart*.

```
# wczytujemy biblioteki
library(rpart)
library(rpart.plot)

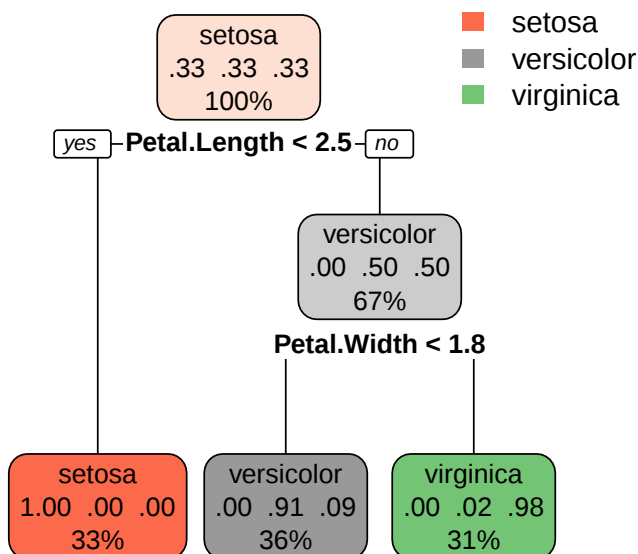
# Przykład 1: Klasyfikacja na zbiorze iris
model_class <- rpart(Species ~ ., data = iris, method = "class")

# Przykład 2: Regresja na zbiorze mtcars
model_reg <- rpart(mpg ~ ., data = mtcars, method = "anova")
```

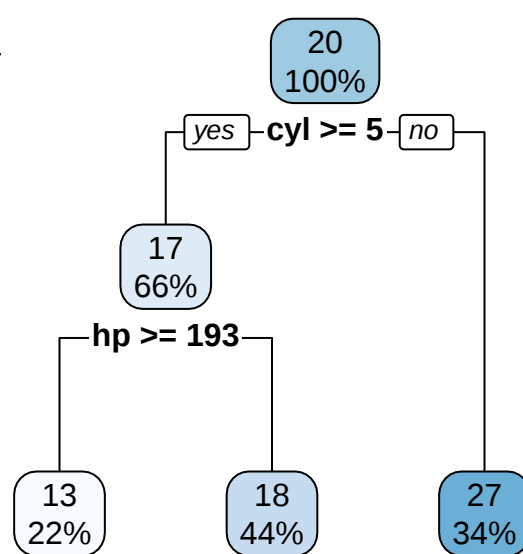
W pierwszym przykładzie budujemy drzewo klasyfikacyjne (*method* = "class"), które przewiduje gatunek kwiatu na podstawie jego cech (długość i szerokość działki kielicha oraz płatk). W drugim przykładzie tworzymy drzewo regresyjne (*method* = "anova"), które przewiduje zużycie paliwa (mpg) na podstawie różnych zmiennych opisujących samochód, takich jak masa, liczba cylindrów czy moc silnika.

Poniżej przedstawiono oba drzewa obok siebie, co pozwala na łatwe porównanie struktury drzewa klasyfikacyjnego i regresyjnego.

Drzewo klasyfikacyjne (iris)



Drzewo regresyjne (mtcars)



CHAID

Algorytm *CHAID* jest następcą innych, prostszych algorytmów, które powstawały w latach 60. i 70., takich jak *AID* (Automatic Interaction Detector) czy *THAID* (THeta Automatic Interaction Detector). Powstawały one w odpowiedzi na rosnące zapotrzebowanie analizy danych ankietowych, przede wszystkim w dziedzinach nauk społecznych. *CHAID*, czyli Chi-square Automatic Interaction Detector, został opracowany w RPA w 1975 roku, a po raz pierwszy został szczegółowo opisany w pracy doktorskiej Gordona V. Kassa pt. *Applied Statistics* w 1980 roku. Współczesne metody bazują właśnie na tym algorytmie, dlatego warto go omówić, aby zrozumieć jego działanie.

CHAID jest wykorzystywany do tworzenia drzew decyzyjnych wyłącznie na podstawie danych kategorycznych. Jeśli dysponujemy danymi ciągłymi, należy je najpierw zdyskretyzować - w przeciwnym razie algorytm nie będzie działał. W odróżnieniu od *CART*, *CHAID* pozwala na tworzenie drzew niebinarnych, czyli takich o wielu rozgałęzieniach.

Podstawą działania algorytmu jest znalezienie najlepszego podziału danych za pomocą testu chi-kwadrat.

Niech k będzie liczbą kategorii w zbiorze danych. Wtedy A_i to rzeczywista liczba obserwacji występujących w kategorii i , E_i to oczekiwana liczba obserwacji w kategorii i , gdzie $i = 1, \dots, k$. Statystykę testową chi kwadrat możemy zdefiniować następującym wzorem:

$$\chi^2 = \sum_{i=1}^k \frac{(A_i - E_i)^2}{E_i}.$$

Dla każdej statystyki testowej obliczana jest p-wartość (często stosuje się korektę Bonferroniego, aby zmniejszyć ryzyko popełnienia błędu I rodzaju). Przy każdym podziale wybierany jest predyktor o najmniejszej p-wartości. Jeśli jest ona mniejsza niż ustalony poziom istotności α , np. $\alpha = 0.05$, to używamy tego predyktora do podziału. Jeśli nie - nie dokonuje się dalszych podziałów i węzeł zostaje oznaczony jako końcowy.

Istnieją również inne warunki zatrzymania działania algorytmu *CHAID*, takie jak: wszystkie przypadki w węźle mają taką samą wartość zmiennej zależnej, drzewo osiągnęło ustaloną głębokość maksymalną lub liczba obserwacji w węźle jest mniejsza niż zdefiniowane minimum.

Możemy zademonstrować działanie tego algorytmu za pomocą kodu w języku *R*. W tym celu użyjemy biblioteki *CHAID*, która nie jest już dostępna bezpośrednio w *RStudio*, ponieważ jest nieco przestarzała, jednak można pobrać ją ze strony: <http://R-Forge.R-project.org>. Wykorzystamy ramkę danych *Titanic* i skonstruujemy drzewo, które pozwoli przewidzieć, kto ma największe szanse na przetrwanie katastrofy, w oparciu o płeć, wiek oraz klasę podróży.

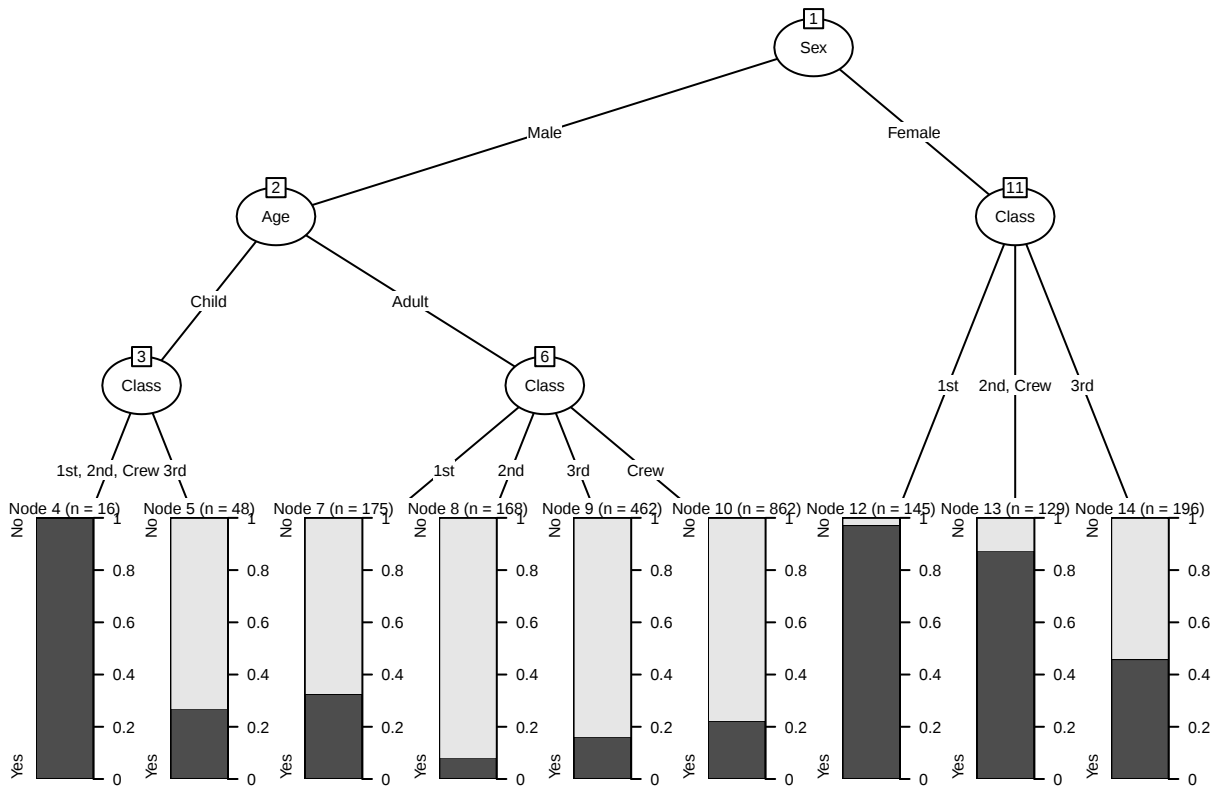
```
library(CHAIID)
library(grid)

dane <- as.data.frame(Titanic)

# Powielamy wiersze zgodnie z częstotliwością (Freq), żeby mieć klasyczny data.frame
rozwiniete_dane <- dane[rep(1:nrow(dane), dane$Freq), 1:4]

# Budujemy model CHAID - przewidujemy Survived na podstawie Class, Sex i Age
model <- chaid(Survived ~ Class + Sex + Age, data = rozwiniete_dane)

# Wykres
plot(model, gp = gpar(fontsize = 6))
```



ID3, C4.5, C5.0

Ostatnimi z przykładów algorytmów budowania drzew decyzyjnych jest grupa *ID3*, *C4.5*, *C5.0*. Powstawały one dokładnie w takiej kolejności, a ich twórcą jest Ross Quinlan. *ID3* został opracowany w podobnym czasie co *CART* i *CHAID* - dokładnie w 1986 roku w pracy *Induction of Decision Trees* - jest więc jednym z pierwszych tego typu algorytmów.

ID3

Algorytm *ID3* został zaprojektowany z myślą o zbiorach danych o wielu atrybutach. Buduje on drzewo w oparciu o zbiór uczący, tzn. na podstawie zbioru danych treningowych, które wykorzystywane są do klasyfikacji nowych danych. Ta część danych, która zostanie źle sklasyfikowana, zostaje dodana do danych treningowych, a sam proces powtarza się, aż wszystkie dane zostaną poprawnie zaklasyfikowane.

Podstawą działania *ID3* jest entropia, czyli miara niepewności. Ustalamy zbiór danych S . Niech c będzie liczbą klas w zbiorze, a p_i prawdopodobieństwem wystąpienia klasy i w zbiorze S . Wtedy entropię możemy wyrazić wzorem:

$$Entropy(S) = - \sum_{i=1}^c p_i \log_2(p_i).$$

Podczas budowy drzewa, wybierany jest taki atrybut, dla którego zysk informacji jest największy. Mierzy on, o ile zmniejszy się entropia po dokonaniu podziału danych na podstawie wybranego atrybutu.

Ustalamy zbiór danych S i atrybut T . Niech $Values(T)$ oznacza zbiór wszystkich możliwych wartości atrybutu T oraz niech S_v będzie podzbiorem S , w którym atrybut T przyjmuje wartość v . Zysk informacyjny obliczamy według wzoru:

$$Gain(S, T) = Entropy(S) - \sum_{v \in Values(T)} \frac{|S_v|}{|S|} Entropy(S_v).$$

Dzięki takiemu podejściu nie trzeba przeszukiwać całego drzewa naraz, co pozwala zmniejszyć koszt obliczeniowy. Z tego względu metoda ta jest stosowana przede wszystkim na dużych zbiorach danych. Warto jednak pamiętać, że metoda nie gwarantuje znalezienia najlepszego możliwego drzewa decyzyjnego.

C4.5

Algorytm C4.5 bazuje na *ID3*, jednak zawiera kilka istotnych ulepszeń:

1. oprócz zmiennych kategorycznych radzi sobie również z ciągłymi (dzięki dyskretyzacji);
2. tworzy drzewo decyzyjne rozpoczynając od małego zbioru danych, nie wykorzystuje całego zbioru, tak jak *ID3*;
3. radzi sobie z brakami w danych a także stosuje przycinanie drzewa w celu redukcji nadmiernego dopasowania modelu;
4. stosuje współczynnik zysku informacyjnego, zdefiniowany jako:

$$GainRatio(S, T) = \frac{Gain(S, T)}{SplitEntropy(S, T)}, \quad \text{gdzie}$$

$$SplitEntropy(S, T) = - \sum_{v \in Values(T)} \frac{|S_v|}{|S|} \log_2 \left(\frac{|S_v|}{|S|} \right).$$

C5.0

Algorytm C5.0 to dalsze rozwinięcie metod *ID3* i C4.5. Obsługuje więcej typów danych, na przykład daty czy znaczniki czasu (stemple czasowe). Najważniejszą różnicą jest to, że potrafi uwzględniać różne koszty błędów - rozpoznaje, które pomyłki są poważniejsze i minimalizuje ich łączny koszt.

Dodatkowo, automatycznie usuwa atrybuty nieistotne dla klasyfikacji, co zwiększa dokładność modelu i zmniejsza czas obliczeń.

Ponieważ te algorytmy najlepiej działają na dużych zbiorach danych, nie będziemy prezentować przykładu tworzenia drzewa w języku *R*. Nie ma sensu rozważać danych z kilkoma atrybutami w przypadku algorytmów Rossa Quinlana. Warto wspomnieć, że w *RStudio* dostępna jest biblioteka *C50*, która obsługuje właśnie te metody.

Opisane przez nas algorytmy stanowią fundament do współczesnych metod budowania drzew regresji, takich jak lasy losowe lub gradient boosting. Dzięki nim rozumiemy podstawowe zasady działania i ograniczenia drzew decyzyjnych, co pozwala na efektywne wykorzystanie tych bardziej zaawansowanych technik.

Zastosowanie drzew decyzyjnych

Drzewa decyzyjne znajdują zastosowanie w analizie danych pochodzących z wielu różnych dziedzin. Dzieje się tak zapewne dlatego, że są one proste w odczytywaniu, przez co proces podejmowania decyzji jest przejrzysty. Na podstawie przeglądu zastosowań zawartego w artykule *A Survey of Decision Trees: Concepts, Algorithms, and Applications* [13] możemy wskazać przykładowo następujące zastosowania:

- **Medycyna:** w dziedzinie tej wykorzystuje się je do przewidywania wystąpienia chorób, oceniania ryzyka powikłań oraz wspomagania diagnozy na podstawie danych klinicznych i demograficznych pacjenta. Metody te umożliwiają identyfikację istotnych czynników zdrowotnych, co wspiera personalizację leczenia i profilaktykę. W licznych badaniach osiągały wysoką skuteczność m.in. w diagnozowaniu chorób serca, nowotworów, COVID-19 czy przewlekłych chorób nerek. Często łączy się je także z innymi technikami uczenia maszynowego, tworząc modele hybrydowe o jeszcze większej precyzji.
- **Finanse:** w sektorze tym drzewa decyzyjne wykorzystuje się zwłaszcza w obszarach związanych z oceną ryzyka i wykrywaniem oszustw. Są wykorzystywane do przewidywania zdolności kredytowej klientów na podstawie danych historycznych, co pozwala instytucjom finansowym lepiej zarządzać ryzykiem udzielania pożyczek. Modele te sprawdzają się również w wykrywaniu nadużyć finansowych, identyfikując nietypowe wzorce w danych transakcyjnych. W wielu badaniach wykazano, że zespołowe wersje drzew decyzyjnych - takie jak na przykład lasy losowe (ang. random forests) - osiągają bardzo wysoką skuteczność klasyfikacji, często przewyższając inne metody.

Oprócz powyższych zastosowań jest też wiele innych. Drzewa decyzyjne używane są również do wyceny cen nieruchomości na podstawie poszczególnych cech. W tej tematyce powstał nawet artykuł Urszuli Litwin i Alicji Malczewskiej pt. *Zastosowanie drzew decyzyjnych do oceny wpływu cech niezabudowanych nieruchomości gruntowych na ich wartość* [14].

Modele te znajdują zastosowanie również w sektorze przemysłu i logistyki, a nawet mogą pomóc w podjęciu decyzji w sytuacji kryzysowej. Na podobny temat dostępny jest artykuł Marzeny Małyszko i Mirosława Wielgosza pt. *Wykorzystanie metody drzew decyzyjnych w systemie wspomagania decyzji kapitana statku w sytuacjach kryzysowych* [15].

Optymalizacja drzew decyzyjnych

Aby poprawić efektywność predykcyjną modelu, a przy tym ograniczyć jego złożoność, warto go zoptymalizować. W tym celu stosuje się przycinanie drzewa.

Przycinanie drzew to technika redukująca złożoność modelu i zapobiegająca przeuczeniu (overfittingowi), czyli sytuacji, w której model za bardzo dopasowuje się do danych treningowych. Dzięki przycinaniu drzewo lepiej radzi sobie z nowymi, nieznanymi danymi, co poprawia ogólną jakość prognoz. Polega na usunięciu nieistotnych poddrzew i zastąpieniu ich pojedynczymi liśćmi, które reprezentują kategorię większości w usuwanym poddrzewie. Docelowo bowiem chcemy, aby drzewo uwzględniało tylko istotne i ogólne cechy.

Sposoby przycinania drzew

- **Przycinanie wstępne** to technika, w której przed zbudowaniem drzewa ustalamy warunki początkowe, które zapobiegają rozrośnięciu drzewa. Możemy ustalić na przykład maksymalną głębokość drzewa, maksymalną liczbę liści lub minimalną liczbę próbek do podziału. Przy ustalaniu tych wartości trzeba być jednak ostrożnym, bo metoda ta radzi sobie z nadmiernym dopasowaniem danych, ale z drugiej strony może doprowadzić do niedouczenia modelu.
- **Przycinanie wsteczne** to metoda, w której mamy już utworzone całe drzewo decyzyjne i chcemy je zoptymalizować, analizując po kolei wpływ każdego poddrzewa na jakość modelu i usuwając od końców nieistotne liście i węzły. W tym celu warto zastosować walidację krzyżową, czyli podzielić dane

na podzbiory i po kolei na każdym z nich testować model, podczas gdy pozostałe będą wtedy pełnić rolę zbioru uczącego, a na końcu wybrać najlepsze rezultaty, np. chcąc zminimalizować liczbę błędnych klasyfikacji, co pokażemy w poniższym kodzie.

Kod w R

Poniżej zaprezentujemy, jak w *R* można w łatwy sposób zbudować drzewo decyzyjne i zastosować na nim przycinanie wsteczne. Pracować będziemy na zbiorze danych *iris* dostępnym w *R*. Zbiór ten zawiera informacje o 150 kwiatach irysa: gatunek, długość i szerokość działki kielicha oraz długość i szerokość płatk. Stworzymy drzewo, które na podstawie cech irysa będzie mogło decydować, z jakim gatunkiem kwiata (z trzech możliwych: *setosa*, *versicolor* i *virginica*) mamy do czynienia.

Na początku wczytujemy bibliotekę *tree*, a następnie dane i przygotowujemy je pod budowę drzewa, tzn. dzielimy na dane treningowe i testowe. W naszym przypadku bierzemy 100 danych na trening, a 50 na test.

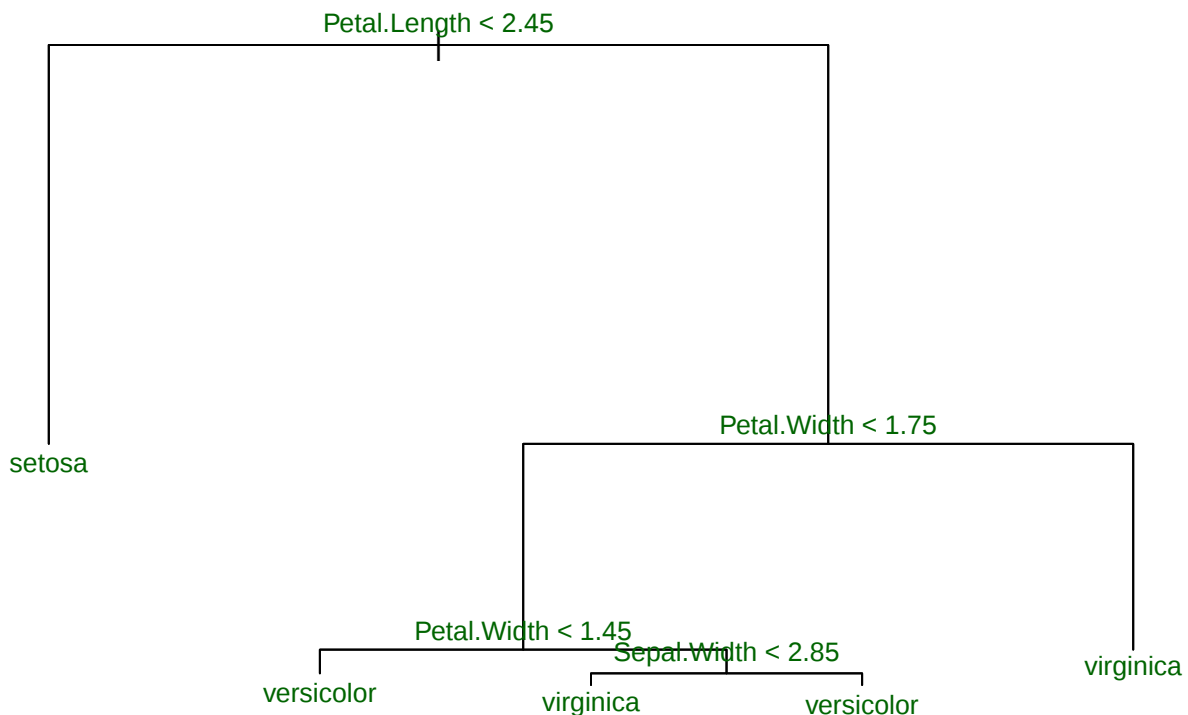
```
library(tree)
set.seed(123)
iris$Species <- as.factor(iris$Species)

trening <- sample(1:nrow(iris), 100)
iris_trening <- iris[trening, ]
iris_test <- iris[-trening, ]
```

Następnie budujemy drzewo i rysujemy jego wykres.

```
drzewo <- tree(Species ~ ., data=iris_trening)

par(mar = c(1, 1, 1, 1))
plot(drzewo)
text(drzewo, pretty = 0, cex = 0.8, col = "darkgreen")
```



Na zbiorze testowym dokonujemy teraz predykcji i sprawdzamy, czy drzewo decyzyjne ma podstawie poszczególnych cech kwiata wybrało rzeczywiście ten gatunek, jaki w rzeczywistości ten kwiat ma. Wyniki prezentujemy w tabeli poniżej.

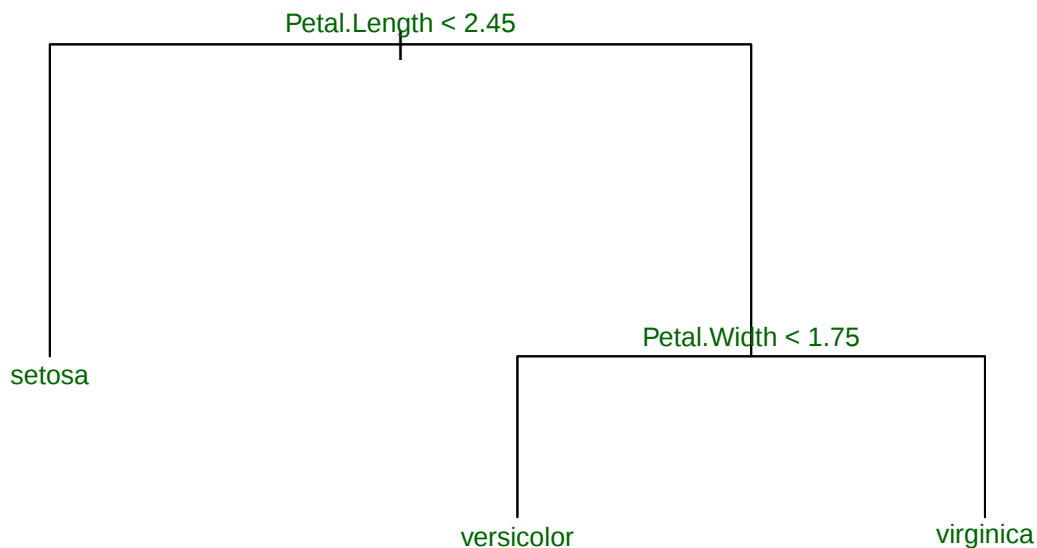
```
drzewo_pred <- predict(drzewo, iris_test, type="class")
czy_prawdziwa <- table(Predicted=drzewo_pred, Actual=iris_test$Species)
knitr::kable(czy_prawdziwa)
```

	setosa	versicolor	virginica
setosa	16	0	0
versicolor	0	18	1
virginica	0	3	12

Dokładność tego drzewa wynosi więc 0.92. Spróbujmy teraz dokonać przycinania wstecznego tego drzewa na podstawie walidacji krzyżowej i minimalizując liczbę błędnych klasyfikacji, a na koniec rysujemy drzewo po przycięciu.

```
set.seed(14)
cv <- cv.tree(drzewo, FUN=prune.misclass) # walidacja krzyżowa
najlepszy_rozmiar <- cv$size[which.min(cv$dev)]
# dev - liczba błędnych klasyfikacji przy danej wielkości drzewa w CV

przyciete_drzewo <- prune.misclass(drzewo, best=najlepszy_rozmiar)
plot(przyciete_drzewo)
text(przyciete_drzewo, pretty = 0, cex = 0.8, col = "darkgreen")
```



Widzimy, że po przycięciu drzewo uprościło się. Sprawdźmy więc jeszcze, czy poprawiło się również przewidywanie gatunku irysa (kodu nie umieszczamy, bo wygląda on identycznie jak przy predykcji na drzewie przed przycięciem). W tym celu umieszczamy analogiczną tabelę jak poprzednio:

	setosa	versicolor	virginica
setosa	16	0	0
versicolor	0	20	1
virginica	0	1	12

Dokładność po przycięciu wynosi zatem 0.96, co jest lepszym wynikiem niż przed przycięciem (mimo że teraz drzewo jest prostsze). To nam pokazuje, że nie zawsze bardziej szczegółowe drzewo jest lepsze. Przy budowaniu drzew decyzyjnych chcemy mieć jak najprostsze drzewo przy jednoczesnym zachowaniu jak najwięcej istotnych informacji i właśnie do tego dążymy, wykonując przycinanie drzewa.

Podsumowanie

Jak widzimy, drzewa decyzyjne są przydatnym narzędziem w analizie danych. Do ich głównych zalet możemy zaliczyć między innymi łatwość interpretacji, przejrzystość i zdolność do pracy zarówno z danymi liczbowymi, jak i kategorycznymi. Z drugiej strony jest to narzędzie podatne na przeuczenie, jednak przy zastosowaniu odpowiednich ograniczeń i metod optymalizacyjnych można tego uniknąć. Obiektem niniejszej pracy były pojedyncze drzewa decyzyjne, jednak warto wspomnieć, że istnieją również lasy losowe (ang. random forests), które łączą wiele drzew decyzyjnych oraz ich wyniki, co pozwala zwiększyć dokładność predykcji.

Bibliografia

- [1] Grafika przedstawiająca budowę drzewa decyzyjnego – wykonana samodzielnie.
- [2] G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, 2021.
- [3] Mirosław Mamczur, *Czym jest i jak działa drzewo decyzyjne*, 2024.
- [4] Grzegorz Dudek, *Uczenie maszynowe. Drzewa decyzyjne – klasyfikacja*, Politechnika Częstochowska. Dostęp online: https://gdudek.el.pcz.pl/images/Dydaktyka/Wyklad3_UM_DDkl.pdf
- [5] A. Singhal, G. Shukla, *Decision Trees: Classification and Regression Trees (CART)*, National Institute of Science Education and Research (NISER), Bhubaneswar, 2023.
- [6] J. Morgan, *Classification and Regression Tree Analysis*, Boston University School of Public Health, 8 maja 2014.
- [7] G. Ritschard, *CHAID and Earlier Supervised Tree Methods*, w: J.J. McArdle & G. Ritschard (red.), *Contemporary Issues in Exploratory Data Mining in Behavioral Sciences*, Routledge, New York, 2013, s. 48–74.
- [8] Y. Yang, F. Yi, C. Deng, G. Sun, *Performance Analysis of the CHAID Algorithm for Accuracy*, Mathematics, 2023, t. 11, nr 11, art. 2558.
- [9] B. Hssina, A. Merbouha, H. Ezzikouri, M. Erritali, *A comparative study of decision tree ID3 and C4.5*, International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Advances in Vehicular Ad Hoc Networking and Applications, 6(4), 2015, s. 93–99.
- [10] S. Miertschin, *C4.5 Decision Tree Algorithm*, University of Houston, 2007. Dostęp online: https://uh.edu/~smiertsc/4397cis/C4.5_Decision_Tree_Algorithm.pdf
- [11] M. Arif, *Decision Tree Algorithms C4.5 and C5.0 in Data Mining: A Review*, International Journal of Database Theory and Application, 11(1), 2018, s. 1–8.
- [12] V. Lavrenko, *IAML7.9 Information Gain Ratio* [Wideo], YouTube, 2016. Dostęp online: <https://www.youtube.com/watch?v=rb1jdBPKzDk>
- [13] D.I. Mienye, N.R. Jere, *A Survey of Decision Trees: Concepts, Algorithms, and Applications*, 2024.
- [14] U. Litwin, A. Malczewska, *Zastosowanie drzew decyzyjnych do oceny wpływu cech niezabudowanych nieruchomości gruntowych na ich wartość*, 2004.
- [15] M. Małyszko, M. Wielgosz, *Wykorzystanie metody drzew decyzyjnych w systemie wspomaganie decyzji kapitana statku w sytuacjach kryzysowych*, 2016.
- [16] N. Arya, *Decision Tree Pruning: The Hows and Whys*, 2022.