UNIWERSYTET ŚLĄSKI
W KATOWICACH

Wydział Nauk Ścisłych i Technicznych

Konrad Gaik
328133

Globalna optymalizacja reguł decyzyjnych względem długości

PRACA DYPLOMOWA
MAGISTERSKA

Promotor:
dr hab. Beata Zielosko, prof. UŚ

Sosnowiec, 2024

Spis treści

1 Wstęp

Wraz z gwałtownym rozwojem technologicznym oraz wzrostem ilości dostępnych danych [1], coraz większą rolę odgrywa umiejętność wyboru, analizy i efektywnego przetwarzania informacji. Postęp w cyfrowej transformacji społeczeństwa, których dane stanowią jeden z najcenniejszych zasobów wymusza ekstrakcję wiedzy z tych danych, gdyż samo ich gromadzenie jest niewystarczające w procesach decyzyjnych.

Jednym z obiecujących podejść w analizie danych jest wykorzystanie reguł decyzyjnych. Ze względu na swoją postać zapisu, prostote i intuicyjnośc interpretacji stanowia one popularna formę reprezentacji wiedzy. Jednakże każdy z algorytmów indukcji reguł decyzyjnych ma swoje własne mocne strony oraz ograniczenia, co podkreśla potrzebę elastycznego podejścia w wyborze odpowiedniego algorytmu w zależności od charakterystyki danych oraz oczekiwanej interpretowalności i precyzji modelu.

Celem niniejszej pracy było stworzenie modelu opartego o reguły decyzyjne poprzez opracowanie metody globalnej optymalizacji reguł decyzyjnych, ze szczególnym uwzględnieniem ich długości. Tablice decyzyjne danych wygenerowane na podstwie reduktów zostały zaprezentowane przez zbiory reguł decyzyjnych indukowanych z drzewa decyzyjnego, wygenerowanego za pomocą algorytmu CART. Badania przeprowadzone na różnorodnych zbiorach danych z UCI ML Repository oraz Kaggle Repository po odpowiednim ich przygotowaniu, miały na celu analizę skuteczności zaproponowanej metody optymalizacji.

W ramach pracy zostały omówione zagadnienia związane z drzewami decyzyjnymi oraz regułami decyzyjnymi jako metodami reprezentacji wiedzy, miarami wyboru atrybutów, tworzących węzły w drzewie oraz miarami jakości reguł decyzyjnych. Skoncentrowano się na znanym algorytmie CART [2], który jest popularną techniką indukcji drzew decyzyjnych.

Spodziewano się, że zaproponowana metoda globalnej optymalizacji reguł decyzyjnych przyniesie korzyści poprzez wykazanie jej skuteczności w procesie klasyfikacji danych. Przeprowadzone eksperymenty pozwoliły na analizę i porównanie wyników w celu lepszego zrozumienia potencjalnych zalet i ograniczeń tej metody.

2 Eksploracja danych i proces KDD

W rozdziale zostaną omówione podstawowe pojęcia dotyczące eksploracji danych i procesu KDD (ang. Knowledge Data Discovery) czyli odkrywania wiedzy z danych.

Choć często używane zamiennie, eksploracja danych i odkrywanie wiedzy w bazach danych (KDD) to różne pojęcia. Eksploracja danych jest istotnym, ale nie jedynym etapem tego procesu. KDD obejmuje przygotowanie danych, ich analizę i interpretację wyników, podczas gdy eksploracja danych skupia się na wydobywaniu wzorców z danych [3].

2.1 Wprowadzenie do procesu KDD

Proces KDD (ang. Knowledge Discovery in Databases) to proces polegający na wydobyciu użytecznych, wcześniej nieznanych i potencjalnie wartościowych informacji z dużych zbiorów danych. Eksploracja danych jest jednym z etapów procesu odkrywania wiedzy z baz danych [4]. Proces KDD jest procesem iteracyjnym, wymaga wielokrotnych iteracji poniższych kroków:

- **Czyszczenie danych**
  Jest realizowane przy użyciu narzędzi do wykrywania niezgodnści danych i narzędzi do transformacji danych i obemuje m.in:
  - Czyszczenie danych polegjące na usuwaniu niepełnych, niepoprawnych i nieistotnych danych ze zbiorów.
  - Czyszczenie przy użyciu narzędzi do wykrywania niezgodności danych i narzędzi do transformacji danych.
- **Integracja danych**
  Jest realizowana przy uzyciu narzędzi migracji danych, narzędzi synchronizacji danych i narzedzi ETL (Extraction - Transformation - Loading) i polega na łączeniu heterogenicznych i rozproszonych danych z wielu źródeł w jedno wspólne źródło (ang. Data Warehouse).

- **Wybieranie danych**

  Selekcja danych wykorzystuje wieloetapowy proces, w którym określane są i pobierane dane istotne dla analizy. W tym celu można użyć sieci neuronowych, drzew decyzyjnych, naiwnych klasyfikatorów Bayesa, grupowania i metody regresji.

- **Przetwarzanie danych**

  Transformacja danych wymaga przekształcania ich w odpowiednią formę, wyznaczoną przez wymagania eksploracji danych. Proces ten obejmuje przetwarzanie oraz łączenie danych w sposób odpowiedni dla eksploracji, wykonując operacje takie jak podsumowanie lub agregacja [3].

- **Eksploracja danych**

  Eksploracja danych to etap procesu odkrywania wiedzy, którego celem jest wyodrębnienie potencjalnie użytecznych wzorców ze zbioru wyselekcjonowanych danych [5]. Obejmuje techniki stosowane do przekształcania istotnych danych we wzorce, które mogą być wykorzystane do klasyfikacji lub charakterystyki badanego zjawiska. Proces ten umożliwia odkrywanie ukrytych zależności i informacji w dużych zbiorach danych, wspierając podejmowanie decyzji w różnych dziedzinach [3].

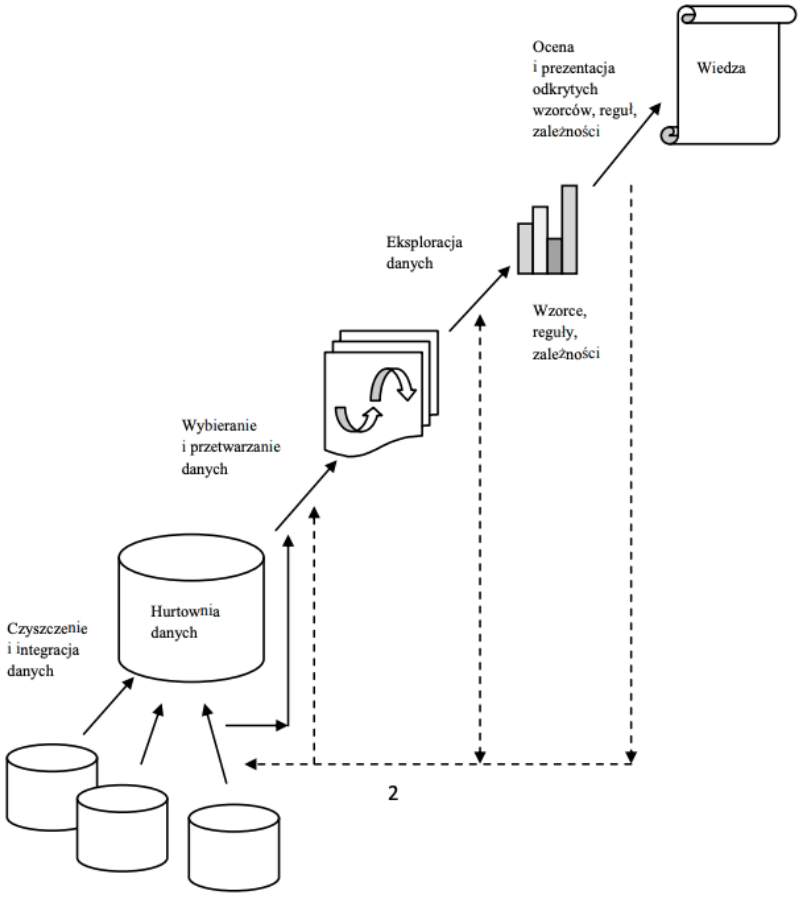
- **Ocena odkrytych wzorców, reguł, zależności**

  Ocena wzorca polega na identyfikacji ściśle rosnących wzorców reprezentujących wiedzę na podstawie określonych miar. Wyszukuje ona stopień przydatności każdego wzorca i stosuje sumaryzację i wizualizację, aby ułatwić użytkownikowi zrozumienie danych.

- **Prezentacja odkrytej wiedzy**

  Polega na przedstawieniu wyników w sposób jednoznaczny i możliwy do wykorzystania w procesie podejmowania decyzji.

Poniższy rysunek obrazuje podgląd etapów wydobywania wiedzy z danych: czyszczenie i integracje danych, selekcje i tansformacje danych, eksploracje danych oraz ocenę i prezentacje odkrytych wzorców, reguł, zależności.



Rys. 1: Eksploracja danych jako jeden z kroków w procesie odkrywania wiedzy
Źródło: [4]

2.2 Wprowadzenie do eksploracji danych

Wraz z rozwojem technologicznym gromadzenie ogromnych ilości danych stało się powszechne, jednak wiedza ukryta w tych zbiorach często pozostaje niewykorzystana. Eksploracja danych wydaje się doskonałym środkiem do realizacji tego celu poprzez przygotowanie i analize danych, pozwalając na odkrywanie nowej, wcześniej nieznanej wiedzy.

Eksploracja danych to obszar badawczy zajmujący się odkrywaniem wzorców, zależności, podobieństw oraz trendów w zbiorach danych [6]. Eksploracja danych oferuje narzędzia i techniki do odkrywania ukrytej wiedzy w tych zbiorach, co pozwala na podejmowanie lepszych decyzji biznesowych, naukowych i innych.

2.3 Etapy, definicje i istota eksploracji danych

Na przestrzeni lat znaczenie eksploracji danych znacząco wzrosło. W początkach XXI wieku, definicja "dużego zbióru danych"(ang. big data) różniła się znacznie od dzisiejszych standardów. Wówczas zbiory te mieściły się w zakresie od kilkudziesięciu gigabajtów do kilkuset gigabajtów, a głównym problemem było ich przechowywanie i zarządzanie tymi danymi. Obecnie, dzięki postępowi technologicznemu, swobodnie przechowujemy pliki wielkości kilku terabajtów, co jest uznawane za przeciętne pod względem rozmiaru zbioru danych. Głównym wyzwaniem stała się analiza i interpretacja ogromnej ilości danych.

Poniżej omówiono kwestie związane z eksploracją danych, takie jak etapy tego procesu, przygotowanie danych do analizy oraz radzenie sobie z problemem brakujących wartości. Omówiono kluczowe etapy eksploracji danych, obejmujące przekształcenie, oczyszczanie i formatowanie danych, identyfikację nietypowych obserwacji oraz dyskretyzację danych. Kolejne podrozdziały poświęcono analizie etapów eksploracji danych. Omówiono najważniejsze kroki związane z przygotowaniem danych do eksploracji, eliminacją obserwacji odstających oraz radzeniem sobie z brakującymi wartościami. Przedstawiono również techniki identyfikacji i analizy obserwacji odstających, a także metody uzupełniania brakujących danych.

Eksploracja danych wymaga odpowiedniego przygotowania danych źródłowych przed przystąpieniem do modelowania. Proces ten obejmuje kilka kluczowych etapów, z których każdy ma istotne znaczenie dla jakości wynikowego zbioru danych.

Szacuje się, że wstępna obróbka danych zajmuje około 60% czasu i wysiłku poświęconego na cały proces eksploracji danych [7].

- **Przekształcenie danych w celu ich oczyszczenia oraz sformatowania.**

  Na tym etapie dokonuje się wszelkich koniecznych przekształceń mających na celu usunięcie błędów. Czynności te obejmują standaryzację formatu, poprawę struktury oraz eliminację błędnych wpisów. Jest to kluczowe dla zapewnienia spójności i rzetelności analizy.

- **Obserwacje odstające**
  Proces usuwania obserwacji odstających jest istotnym krokiem w eliminacji potencjalnych zakłóceń. Obejmuje wykrycie i identyfikację odstających wartości przy użyciu zaawansowanych technik statystycznych, takich jak analiza skrzynkowa czy metody oparte na odległościach. Przed usunięciem obserwacji odstających zaleca się dokładne zrozumienie kontekstu danych i konsultację z ekspertem dziedzinowym [4].

- **Uzupełnienie brakujących informacji**
  Braki danych są powszechnym wyzwaniem podczas eksploracji danych. Brakujące wartości mogą zostać zastąpione lub usunięte zależnie od kontekstu danych oraz dostępnych metod uzupełniania.

- **Dyskretyzacja danych**
  W analizie danych, szczególnie w przypadku zmiennych o różnej dziedzinie wartości, konieczne jest przekształcanie danych w celu ujednolicenia wpływu każdej zmiennej na wyniki analizy. Dyskretyzacja polega na zamianie atrybutów o wartościach ciągłych na atrybuty o wartościach dyskretnych. Proces ten obejmuje zastąpienie każdego atrybutu ciągłego atrybutem o wartościach dyskretnych, odpowiadających pewnym przedziałom ciągłych wartości oryginalnego
  atrybutu [8, 9].

  Powoduje to, że otrzymuje się zamiast atrybutu ciągłego atrybut porządkowy o skończonej liczbie wartości, możliwe niewielkiej [10].
  Zaletą dyskretyzacji jest poprawa efektywności obliczeniowej klasyfikacji.

- **Normalizacja danych**
  Normalizacja danych jest kluczowa dla zastąpienia różnorodnych reprezentacji tych samych danych jednolitą wartością. Polega ona na modyfikacji struktury jej tabel w celu zlikwidowania nadmiarowości danych [11].

- **One-Hot Encoding**
  Technika ta stosowana jest do przekształcenia zmiennych kategorycznych na format zrozumiały dla algorytmów uczenia maszynowego operujących na danych numerycznych, takich jak CART. Polega na przekształceniu zmiennych kategorycznych na wektory binarne, gdzie każda kategoria jest reprezentowana przez wektor zer i jedynek. Dzięki temu zachowujemy informacje o kategoriach, unikając jednocześnie fałszywych interpretacji przez algorytmy [12].

3 Drzewa decyzyjne

W tym rozdziale omówione zostały podstawowe definicje, sposoby indukcji drzew decyzyjnych oraz zastosowania drzew decyzyjnych. Omówiono proces indukcji oraz miary wyborów atrybutów, które są istotne dla skutecznej konstrukcji drzew decyzyjnych.

Drzewo decyzyjne τ jest zdefiniowane jako trójka $\tau = (T, \psi, \lambda)$, gdzie:

- T jest skończonym zbiorem węzłów, takim że:
  - Każdy węzeł $j \in T$ ma dokładnie jednego rodzica, z wyjątkiem korzenia r, który nie ma rodzica.
  - Każdy węzeł $j \in T$ jest rodzicem dokładnie zero lub dwóch węzłów potomnych.
- $\psi : T \setminus l(T) \to \{1, \ldots, d\}$ jest funkcją przypisującą każdemu węzłowi niebędącemu liściem wymiar podziału, gdzie d jest liczbą cech, a $l(T)$ oznacza zbiór liści drzewa T.
- $\lambda : T \setminus l(T) \to R$ jest funkcją przypisującą każdemu węzłowi niebędącemu liściem wartość progową podziału.

Każdy węzeł $j \in T$ jest związany z regionem $P_j \subset R^d$ przestrzeni wejściowej, takim że:

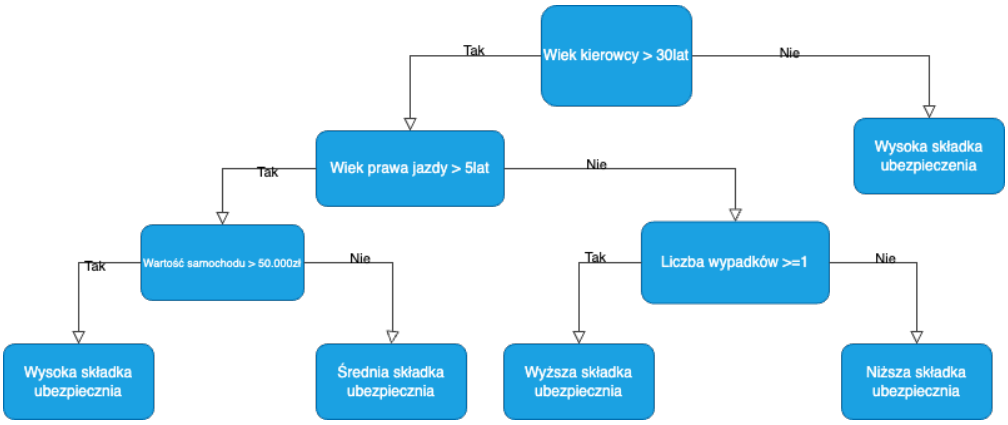
- Dla korzenia r: $P_r = R^d$
- Dla każdego węzła wewnętrznego $j \in T \setminus l(T)$ z potomkami $\{0, 1\}$:

$$P_0(j) = \{x \in P_j : x_{\psi_j} \leq \lambda_j\}$$
$$P_1(j) = \{x \in P_j : x_{\psi_j} > \lambda_j\}$$

gdzie x_{ψ_j} oznacza ψ_j-tą współrzędną wektora cech x [12].

Drzewa decyzyjne są jedną z metod klasyfikacji. Celem drzewa decyzyjnego jest podzielenie danych na mniejsze, bardziej jednorodne grupy. Jednorodność oznacza, że większość próbek w każdym węźle pochodzi z jednej klasy [12].

Przykładowo drzewa decyzyjne pozwalają uprościć decyzję o ulgach dla właścieli samochodów, którzy chcą ubezpieczyć samochód. Sposób dedukcji przedstawia poniższy rysunek.



Rys. 2: Drzewo decyzyjne klasyfikacji kierowców w odniesieniu do składki na ubezpieczenie samochodu
Źródło: Opracowanie własne

Drzewo decyzyjne możemy użyć w sytuacji, gdy chcemy określić wysokość składki dla kierowcy. Zaczynamy od pytania, czy kierowca ma więcej niż 30 lat. Jeśli tak, przechodzimy w lewo i pytamy, czy samochód ma więcej niż 5 lat. Jeśli odpowiedź brzmi "tak", składka jest wysoka. Jeśli odpowiedź brzmi "nie", przechodzimy do pytania, czy samochód jest droższy niż 50 tysięcy złotych. Jeśli tak, składka jest wysoka, w przeciwnym razie składka jest średnia. Jeśli jednak kierowca ma mniej niż 30 lat, składka jest wysoka. Jeśli staż kierowcy wynosi mniej niż 5 lat, sprawdzamy, czy były jakiekolwiek wypadki. Jeśli liczba wypadków jest większa niż jeden, cena jest wysoka, w przeciwnym razie cena jest niska [8].

Na powyższej ilustacji każdy węzeł w drzewie przedstawia pytanie lub jest węzłem końcowym (zwanym także liściem), który zawiera odpowiedź. Krawędzie łączą odpowiedzi na pytanie z następnym pytaniem, które zostałoby zadane [8].

Algorithm 1 Buduj_drzewo [13]

Require: S - zbiór przykładów wejściowych, A - zbiór atrybutów opisujących przykłady

Ensure: drzewo decyzyjne

1: Utwórz węzeł n {przy pierwszym wywołaniu korzeń drzewa}
2: **if** wszystkie przykłady w S należą do tej samej klasy K **then**
3: **return** n jako liść z etykietą klasy K
4: **end if**
5: **if** A jest pusty **then**
6: **return** n jako liść z etykietą klasy, do której należy większość przykładów w S
7: **end if**
8: W przeciwnym razie:
9: Wybierz atrybut $a \in A$, który najlepiej klasyfikuje przykłady z S zgodnie z przyjętą miarą wyboru atrybutu
10: Przypisz węzłowi n test wykorzystujący a
11: **for** każdą wartość v_i atrybutu a **do**
12: Dodaj do węzła n gałąź odpowiadającą warunkowi $(a = v_i)$
13: Niech S_i będzie podzbiorem przykładów z S, które posiadają wartość v_i dla atrybutu a
14: **if** S_i jest pusty **then**
15: Dodaj do gałęzi liść z etykietą klasy, do której należy większość przykładów w S
16: **else**
17: Indukuj poddrzewo wywołując Buduj_drzewo($S_i, A \setminus \{a\}$)
18: **end if**
19: **end for**
20: **return** drzewo o korzeniu w n

Powyższy algorytm przedstawia ogólny schemat konstrukcji drzewa decyzyjnego. Proces rozpoczyna się od utworzenia węzła n, który przy pierwszym wywołaniu staje się korzeniem drzewa. Algorytm sprawdza, czy wszystkie przykłady w zbiorze S należą do tej samej klasy. Jeśli tak, tworzy liść z etykietą tej klasy. W przypadku pustego zbioru atrybutów A, również tworzy liść, ale z etykietą klasy większościowej w S. Jeżeli żaden z powyższych warunków nie jest spełniony, algorytm wybiera atrybut a ze zbioru A, który najlepiej klasyfikuje przykłady. Miara oceny zależy od konkretnego algorytmu: dla ID3 jest to zysk informacji (information gain), dla C4.5 - stosunek zysku (gain ratio), a dla CART - indeks Gini. Wybrany atrybut a staje się testem przypisanym do węzła n. Następnie, dla każdej możliwej wartości v_i atrybutu a, tworzona jest nowa gałąź. Dla każdej gałęzi tworzone jest S_i - podzbiór przykładów z S, które posiadają wartość v_i dla atrybutu a. Jeśli S_i jest pusty, do gałęzi dodawany jest liść z etykietą klasy większościowej w S. W przeciwnym razie, algorytm rekurencyjnie wywołuje funkcję Buduj_drzewo dla podzbioru S_i i zbioru atrybutów pomniejszonego o a. Proces ten powtarza się rekurencyjnie, aż wszystkie przykłady zostaną sklasyfikowane lub wyczerpią się atrybuty. Na końcu algorytm zwraca kompletne drzewo decyzyjne z korzeniem w węźle n [3, 13].

3.1 Podstawowe pojęcia

Z metodą drzew decyzyjnych wykorzystywanych w zadaniach klasyfikacji związane są pojęcia tablica decyzyjna, liść, węzeł, korzeń, kryterium stopu, głębokośc drzewa, kryterium podziału i redukt.

Tablica decyzyjna posiada wiersze, reprezentujące obiekty, opisane wartościami atrybutów. Pozwala w sposób tabelaryczny przedstawiać zjawiska występujące w rzeczywistych problemach [14].

W tablicach decyzyjnych w zbiorze atrybutów wyodrębnia się dwa podzbiory: podzbiór atrybutów warunkowych C i podzbiór atrybutów decyzyjnych D.
Zbiór $A = C \cup D$ oznacza zbiór wszystkich atrybutów systemu informacyjnego, na bazie którego powstała tablica decyzyjna TD [14, 8].

$$TD = (U, C, D, V, f)$$

gdzie:

- $C, D \subset A$ oraz $C \neq \emptyset, C \cup D = A, C \cap D = \emptyset$,
- C jest niepustym, skończonym zbiorem atrybutów warunkowych,
- D jest niepustym, skończonym zbiorem atrybutów decyzyjnych,
- f jest funkcją decyzyjną, zdefiniowaną w sposób analogiczny do funkcji informacji systemu informacyjnego,

Tabela 1: Przykładowa tablica decyzyjna
Źródło: Opracowanie na podstawie [15]

a1	a2	a3	a4	dec
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

Powyższa tablica decyzyjna (Tabela 1) przedstawia zbiór obserwacji dotyczących warunków atmosferycznych oraz decyzji o grze na zewnątrz.
Gdzie:

$C = \{a1, a2, a3, a4\}$

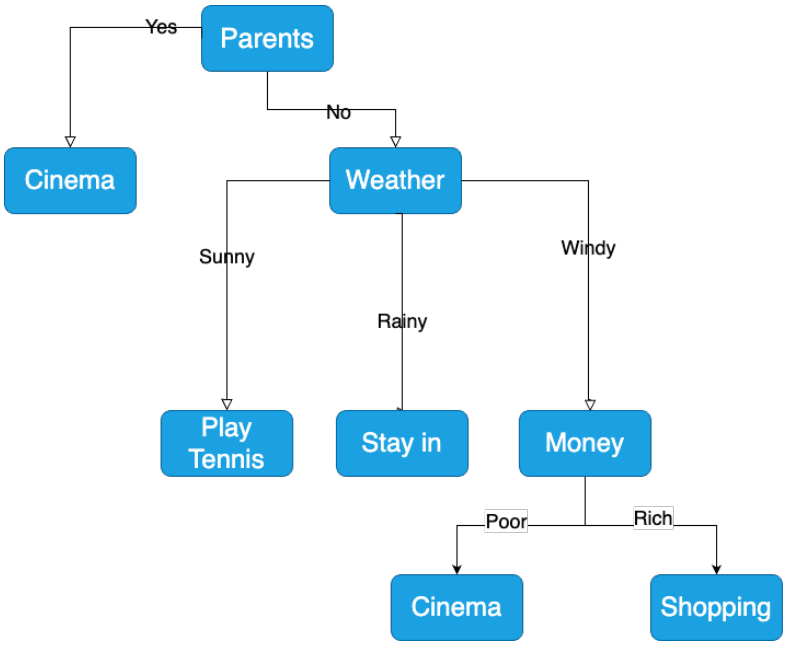
$D = \{dec\}$

Każdy wiersz reprezentuje konkretne warunki, takie jak pogoda, temperatura, wilgotność i wiatr, a także zawiera informację decyzyjną - czy grać na zewnątrz ("Yes") czy nie ("No"). Te dane stanowią podstawę do konstrukcji drzewa decyzyjnego, które mogą być wykorzystane do prognozowania decyzji na podstawie aktualnych warunków atmosferycznych.

- **Outlook(a1):** Reprezentuje warunki pogodowe, przyjmujące wartości "Sunny, ", "Overcast", i "Rainy", opisujące, czy jest słonecznie, zachmurzenie, czy pada deszcz.
- **Temperature(a2):** Określa temperaturę w stopniach Fahrenheit.
- **Humidity(a3):** Określa wilgotność w procentach.

- **Windy(a4):** Jest atrybutem binarnym, który informuje o występowaniu wiatru lub jego braku. ("TRUE"lub "FALSE").
- **(dec):** Atrybut decyzyjny, oznaczający pozytywną decyzję ("Yes") lub negatywną decyzję ("No") podjętą na podstawie analizy warunków atmosferycznych.

Schemat drzewa decyzyjnego powstałego w oparciu o dane zawarte w Tabeli 6.



Rys. 3: Przykładowe drzewo decyzyjne
Źródło: [15]

Liść w drzewie decyzyjnym to węzeł końcowy, który nie posiada krawędzi wychodzących, co oznacza, że nie ma dalszych podziałów. Liść jest zwykle etykietowany wartością klasy lub przewidywaną wartością dla danego przypadku [16]. Na rysunku 3 występują 4 liście; dwa razy "wysoka składka ubezpiecznia", "średnia składka ubezpieczenia"i "niska składka ubezpieczenia".

Węzeł w drzewie decyzyjnym to punkt podziału, który posiada krawędzie wychodzące i wchodzące. Węzeł jest etykietowany atrybutem warunkowym. Każda krawędź wychodząca z węzła reprezentuje wartość atrybutu podziałowego lub wynik testu, prowadząc do dalszego podziału drzewa [17].

Korzeń drzewa decyzyjnego to początkowy węzeł, od którego rozpoczyna się konstrukcja drzewa. Reprezentuje on cały zbiór danych przed jakimkolwiek podziałem i zawiera pierwszy test, który będzie zastosowany do danych [16]. W procesie budowy drzewa, korzeń jest pierwszym punktem decyzyjnym, który dzieli dane na podgrupy na podstawie wybranego atrybutu. Na przykład, na rysunku 3 korzeń 'Wiek kierowcy > 30 lat' dzieli dane na dwie grupy: kierowców powyżej 30 lat i tych, którzy mają 30 lat lub mniej. Ten podział prowadzi do kolejnych węzłów drzewa, gdzie proces decyzyjny jest kontynuowany.

Kryterium stopu w drzewach decyzyjnych określa warunek, który musi być spełniony, aby zatrzymać proces konstruowania drzewa lub przycinania go. Jest to istotne narzędzie, które pomaga zapobiec nadmiernemu dopasowaniu (overfitting) i nadmiernemu rozgałęzieniu drzewa [18].

Głębokość drzewa określa maksymalną liczbę poziomów w drzewie decyzyjnym. Głębokość drzewa decyzyjnego określa maksymalną liczbę poziomów od korzenia do liścia w strukturze drzewa [12]. Im głębsze drzewo, tym bardziej złożone są reguły decyzyjne [19].

Kryterium podziału jest metryką używaną do oceny jakości podziału danych na węzły w drzewie decyzyjnym. Popularnymi kryteriami podziału są indeks Gini oraz entropia [20]. Podział zbioru danych polega na tworzeniu mniejszych, bardziej jednorodnych grup. Jednorodność oznacza, że większość próbek w każdym węźle pochodzi z jednej klasy [12].

Redukt jest najmniejszym zbiorem atrybutów, który zachowuje dotychczasową klasyfikację (rozróżnialność) obiektów. Podzbiór atrybutów $B \subseteq A$ jest nazywany reduktem zbioru atrybutów A, jeśli zbiór atrybutów B jest niezależny oraz

IND(B) = IND(A). Zbiór wszystkich reduktów oznaczamy przez RED(A) [21].
Redukt musi spełniać dwa kryteria:

- musi być niezależnym zbiorem atrybutów, czyli zawierać tylko atrybuty niezbędne
- musi zachowywać taką samą rozróżnialność obiektów jak zbiór redukowany

3.2 Miary wyboru atrybutów

Podstawowym krokiem w procesie konstruowania drzew decyzyjnych jest wybór odpowiednich atrybutów do podziału danych. Istnieje kilka różnych miar, które mogą być używane do oceny jakości podziału, z których najpopularniejszymi są entropia, zysk informacyjny (Information Gain) oraz indeks Gini [22]. W tym podrozdziale omówiono te miary oraz ich zastosowanie w kontekście algorytmów konstruujących drzewa decyzyjne.

Miary wyboru atrybutów są kluczowe w procesie konstrukcji drzew decyzyjnych, ponieważ determinują strukturę i efektywność drzewa. Ich rola polega na ocenie, który atrybut najlepiej dzieli dane na każdym etapie budowy drzewa. Wybór odpowiedniego atrybutu wpływa na zdolność drzewa do poprawnej klasyfikacji nowych przypadków, jego głębokość i złożoność.

Indeks Gini

Indeks Gini [7] jest miarą używaną w algorytmie CART do wyboru najlepszego podziału w drzewie decyzyjnym. Określa nieczystość zbioru poprzez obliczenie prawdopodobieństwa błędu, gdy losowo wybrana próbka zostanie źle sklasyfikowana. Wzór na indeks Gini dla zbioru D jest zdefiniowany jako:

$$Gini(D) = 1 - \sum_{i=1}^{c} (p_i)^2$$

gdzie:

- $Gini(D)$ - indeks Gini dla zbioru D,
- p_i - prawdopodobieństwo wystąpienia klasy i w zbiorze,
- c - liczba klas w zbiorze.

Niższa wartość indeksu Gini jest preferowana, ponieważ oznacza większą czystość (homogeniczność) podziału. Im niższy indeks Gini, tym bardziej jednolite są podgrupy utworzone przez podział.

Wykorzystanie powyższego wzoru do zbudowania drzewa decyzyjnego metodą CART ilustruje poniższy przykład. Dotyczy on wyboru formy spędzania weekendu w zależności od warunków pogodowych, finansowych i rodzinnych.

Tabela 2: Zbiór danych do analizy decyzji o spędzaniu wolnego czasu [7]

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Na podstawie danych zawartych w tablicy (Tabela 1) wyznaczono indeks Gini dla atrybutu decyzyjnego, który w analizie przyjmuje cztery wartości: Cinema, Tennis, Stay in i Shopping.

$$\begin{aligned} Gini(Decision) &= 1 - \left[\left(\frac{6}{10}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^2\right] \\ &= 0.58 \end{aligned} \tag{1}$$

Następnie obliczono wartość indeksu Gini dla pozostałych atrybutów: Weather, Parents i Money.

1. Weather:

$$Gini(Weather)_{Sunny} = 1 - \left[\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right] = 0.444 \tag{2}$$

$$Gini(Weather)_{Rainy} = 1 - \left[\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right] = 0.444 \tag{3}$$

$$Gini(Weather)_{Windy} = 1 - \left[\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right] = 0.375 \tag{4}$$

$$\begin{aligned} \tilde{x}_{Weather} &= 0.444 \times \frac{3}{10} + 0.444 \times \frac{3}{10} + 0.375 \times \frac{4}{10} \\ &= 0.416 \end{aligned} \tag{5}$$

2. Parents:

$$Gini(Parents)_{yes} = 1 - \left(\frac{5}{5}\right)^2 = 0 \tag{6}$$

$$Gini(Parents)_{no} = 1 - \left[\left(\frac{1}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^2\right] = 0.72 \tag{7}$$

$$\begin{aligned} \tilde{x}_{Parents} &= 0 \times \frac{5}{10} + 0.72 \times \frac{5}{10} \\ &= 0.36 \end{aligned} \tag{8}$$

3. Money:

$$Gini(Money)_{Poor} = 1 - \left(\frac{3}{3}\right)^2 = 0 \tag{9}$$

$$Gini(Money)_{Rich} = 1 - \left[\left(\frac{2}{7}\right)^2 + \left(\frac{3}{7}\right)^2 + \left(\frac{1}{7}\right)^2 + \left(\frac{1}{7}\right)^2\right] = 0.694 \tag{10}$$

$$\tilde{x}_{Money} = 0 \times \frac{3}{10} + 0.694 \times \frac{7}{10} \\ = 0.486 \tag{11}$$

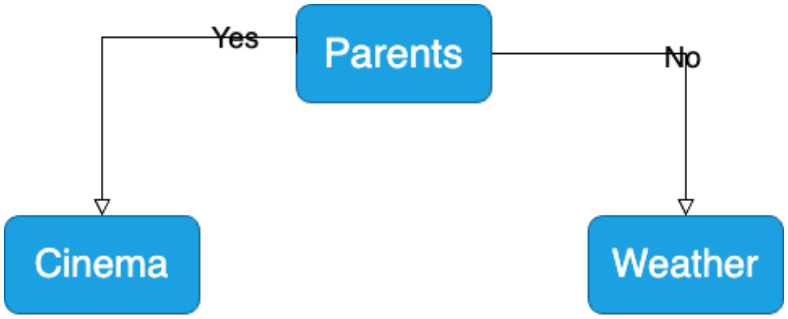
Otrzymano następujące wartości indeksu Gini:

- Weather = 0.416
- Parents = 0.36 (najniższa wartość indeksu Gini)
- Money = 0.486

Najniższą wartość indeksu Gini ma atrybut Parents, co oznacza, że dla tego atrybutu następuje najczystszy podział dla danego zbioru danych. W związku z tym, atrybut Parents zostaje wybrany jako korzeń drzewa decyzyjnego.

Dla wartości 'yes' atrybutu Parents, wszystkie obiekty w tym podzbiorze mają wartość decyzyjną 'Cinema'. Ta gałąź kończy się więc liściem z decyzją 'Cinema'.

Dla wartości 'no' atrybutu Parents, potrzebny jest dalszy podział. W tym przypadku kolejnym atrybutem o najniższym indeksie Gini jest Weather, który staje się następnym węzłem decyzyjnym w tej gałęzi drzewa.



Rys. 4: Pierwsza iteracja indukcji drzewa decyzyjnego [7]

Tabela 3: Tablica decyzyjna dla atrybutu Parents = Yes

Weekend	Weather	Money	Decision
W1	Sunny	Rich	Cinema
W3	Windy	Rich	Cinema
W4	Rainy	Poor	Cinema
W6	Rainy	Poor	Cinema
W9	Windy	Rich	Cinema

Źródło: Opracowano na podstawie [7]

Tabela 4: Tablica decyzyjna dla atrybutu Parents = No

Weekend	Weather	Money	Decision
W2	Sunny	Rich	Tennis
W5	Rainy	Rich	Stay in
W7	Windy	Poor	Cinema
W8	Windy	Rich	Shopping
W10	Sunny	Rich	Tennis

Źródło: Opracowano na podstawie [7]

Powyższa tabela (Tabela 3) pozwala stwierdzić, że dla wartości 'No' w atrybucie 'Parents' są dwa obiekty z wartością atrybutu decyzyjnego Tennis. Drugi podział odbędzie się zatem w korzeniu dla atrybutu 'Weather', a następnie dla wartości Rich w korzeniu dla atrybutu 'Money'. W wyniku drugiego podziału wygenerowane zostanie drzewo decyzyjne:

Na podstawie danych z dalszych sporządzonych tablic (Tabela 5, Tabela 6, Tabela 7):

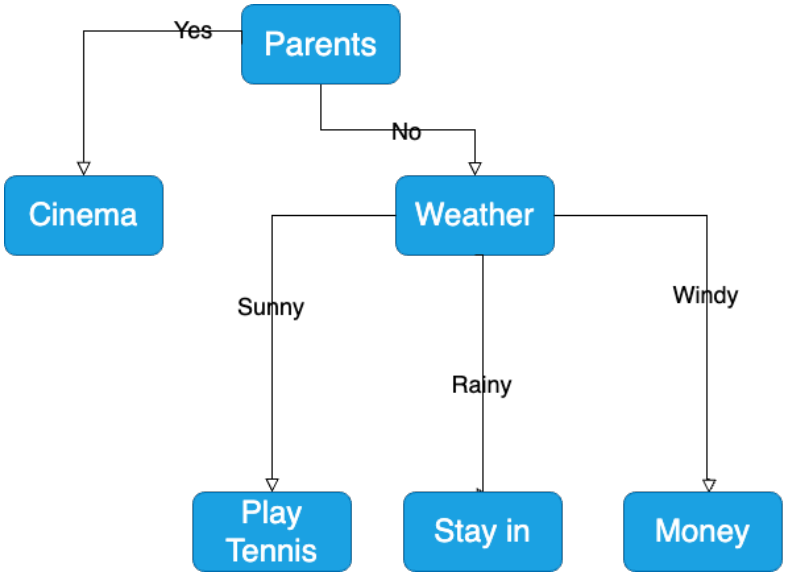
Tabela 5: Tablica decyzyjna dla atrybutu Weather = Sunny

Weekend	Weather	Money	Decision
W2	Sunny	Rich	Tennis
W10	Sunny	Rich	Tennis

Źródło: Opracowano na podstawie [7]

wyciągnięto wniosek, że dla atrybutu Weather = Sunny i Parents = No, atrybut decyzyjny przyjmuje wartość 'Tennis', a dla atrybutu Weather = Rainy i Parents = No, atrybut decyzyjny przyjmuje wartość 'Stay in'. Dla obiektów W7 i W8 atrybut 'Weather' przyjmuje wartość 'Windy', natomiast różnią je atrybuty decyzyjne (patrz tabela 7).

Na tej podstawie oraz obliczeń indeksu Gini dokonano trzeciego i ostatniego podziału z węzła 'Money'.



Rys. 5: Druga iteracja indukcji drzewa decyzyjnego

Tabela 6: Tablica decyzyjna dla atrybutu Weather = Rainy

Weekend	Weather	Money	Decision
W5	Rainy	Rich	Stay in

Tabela 7: Tablica decyzyjna dla atrybutu Weather = Windy

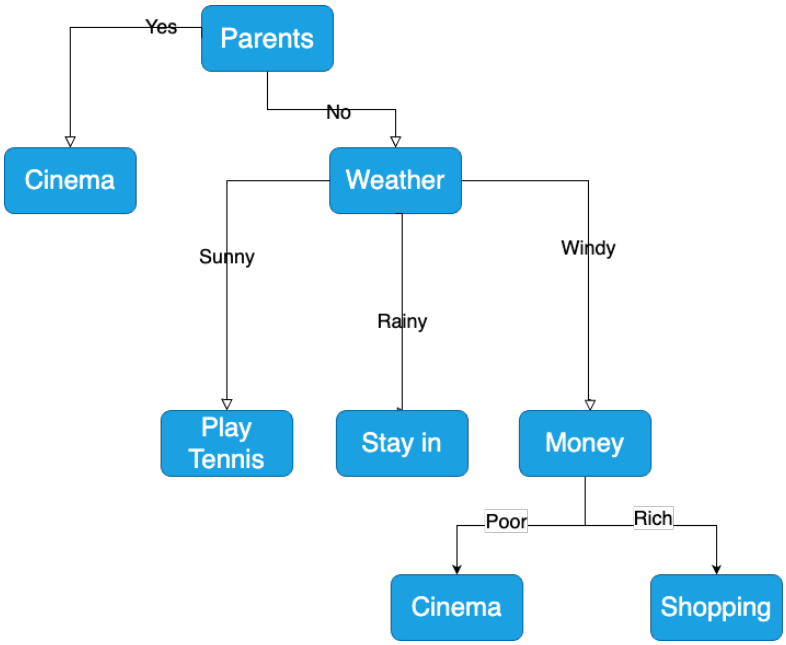
Weekend	Weather	Money	Decision
W7	Windy	Poor	Cinema
W8	Windy	Rich	Shopping

$$Gini(Money)_{Poor} = 1 - \left[\left(\frac{1}{1}\right)^2\right] = 0 \tag{12}$$

$$Gini(Money)_{Rich} = 1 - \left[\left(\frac{1}{1}\right)^2\right] = 0 \tag{13}$$

$$\tilde{x} = 0 \times \frac{1}{2} + 0 \times \frac{1}{2} = 0 \tag{14}$$

Zerowy wynik Indeksu Gini wskazuje na to, że otrzymaliśmy jedną czystą partycję. Na poniższym rysunku przedstawiono końcowy efekt dedukcji (rysunek 6).



Rys. 6: Końcowe drzewo decyzyjne [23]

Przyrost informacji

Zysk informacyjny (Information Gain) jest kluczowym pojęciem w algorytmie C4.5. Jest to miara, która określa, jak bardzo atrybut przyczynia się do zwiększenia informacji o klasyfikacji danych [24]. Zysk informacyjny dla zbioru D po podziale ze względu na atrybut A jest zdefiniowany jako:

$$IG(D, A) = H(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} \cdot H(D_v)$$

gdzie:

- $IG(D, A)$ - zysk informacyjny dla zbioru D po podziale ze względu na atrybut A,
- $H(D)$ - entropia dla zbioru D,
- V - liczba podzbiorów utworzonych przez podział na atrybucie A,
- D_v - podzbiór D dla v-tego podziału,
- $|D|$ - liczność zbioru D,
- $|D_v|$ - liczność v-tego podzbioru.

Używając wcześniej obliczonej entropii, obliczono zysk informacyjny dla każdego atrybutu:

1. Weather:

$$\begin{aligned} IG(\text{Decision}, \text{Weather}) &= H(\text{Decision}) - (3/10 \cdot H(\text{Weather\_Sunny}) + 3/10 \cdot H(\text{Weather\_Rainy}) \\ &= 1.571 - (3/10 \cdot 0.918 + 3/10 \cdot 0.918 + 4/10 \cdot 0.811) \\ &\approx 0.693 \end{aligned} \tag{15}$$

2. Parents:

$$\begin{aligned} IG(\text{Decision}, \text{Parents}) &= H(\text{Decision}) - (5/10 \cdot H(\text{Parents\_Yes}) + 5/10 \cdot H(\text{Parents\_No})) \\ &= 1.571 - (5/10 \cdot 0 + 5/10 \cdot 1.922) \\ &\approx 0.610 \end{aligned} \tag{16}$$

3. Money:

$$\begin{aligned} IG(\text{Decision}, \text{Money}) &= H(\text{Decision}) - (7/10 \cdot H(\text{Money\_Rich}) + 3/10 \cdot H(\text{Money\_Poor})) \\ &= 1.571 - (7/10 \cdot 1.842 + 3/10 \cdot 0) \\ &\approx 0.282 \end{aligned} \tag{17}$$

Na podstawie obliczonego zysku informacyjnego, atrybut "Weather"ma najwyższą wartość (0.693), więc zostałby wybrany jako pierwszy atrybut do podziału w drzewie decyzyjnym według kryterium zysku informacyjnego.

Entropia

Entropia jest jedną z kluczowych miar wyboru atrybutów w procesie konstruowania drzew decyzyjnych. Jest to miara używana w algorytmie ID3. Entropia określa stopień nieuporządkowania danych w danym zbiorze [25]. Entropia jest miarą, którą staramy się minimalizować w węzłach drzewa decyzyjnego po każdym podziale. Atrybut, który przyczynia się do największego spadku entropii (największy zysk informacyjny), jest wybierany jako najlepszy do podziału w danym węźle [25]. Formalnie, entropia dla zbioru X jest definiowana jako:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

gdzie:

- $H(X)$ - entropia dla zbioru X,
- $p(x_i)$ - prawdopodobieństwo wystąpienia klasy x_i w zbiorze,
- n - liczba klas w zbiorze.

Najpierw obliczono entropię dla atrybutu decyzyjnego:

$$\begin{aligned} H(\text{Decision}) &= -((6/10)\log_2(6/10) + (2/10)\log_2(2/10) + (1/10)\log_2(1/10) + (1/10)\log_2(1/10)) \\ &\approx 1.571 \end{aligned} \tag{18}$$

Teraz obliczmy entropię dla każdego możliwego atrybutu:

1. Weather:

$$H(\text{Weather\_Sunny}) = -((1/3)\log_2(1/3) + (2/3)\log_2(2/3)) \approx 0.918 \tag{19}$$
$$H(\text{Weather\_Rainy}) = -((2/3)\log_2(2/3) + (1/3)\log_2(1/3)) \approx 0.918 \tag{20}$$
$$H(\text{Weather\_Windy}) = -((3/4)\log_2(3/4) + (1/4)\log_2(1/4)) \approx 0.811 \tag{21}$$

2. Parents:

$$H(\text{Parents\_Yes}) = -((5/5)\log_2(5/5)) = 0 \tag{22}$$

$$H(\text{Parents\_No}) = -((1/5)\log_2(1/5) + (2/5)\log_2(2/5) + (1/5)\log_2(1/5) + (1/5)\log_2(1/5)) \approx 1 \tag{23}$$

3. Money:

$$H(\text{Money\_Rich}) = -((3/7)\log_2(3/7) + (2/7)\log_2(2/7) + (1/7)\log_2(1/7) + (1/7)\log_2(1/7)) \approx \tag{24}$$

$$H(\text{Money\_Poor}) = -((3/3)\log_2(3/3)) = 0 \tag{25}$$

3.3 Indukcja drzew decyzyjnych

Proces konstruowania drzewa decyzyjnego można podzielić na dwie główne fazy: indukcję drzewa decyzyjnego oraz przycinanie. W pierwszym etapie, znanym jako etap indukcji, konstruowane jest drzewo decyzyjne na podstawie danych treningowych. W tej fazie algorytmy analizują cechy danych i podejmują decyzje dotyczące podziału, aby stworzyć hierarchię decyzji reprezentowaną przez drzewo [3].
W kroku drugim, zwany fazą przycinania, dokonywane są korekty w celu optymalizacji struktury drzewa. Przycinanie pozwala na usunięcie niepotrzebnych węzłów oraz krawędzi, co może poprawić skuteczność klasyfikacji oraz zmniejszyć złożoność modelu.

W niniejszym podrozdziale skupiono się na omówieniu trzech popularnych algorytmów konstruujących drzewa decyzyjne: CART, C4.5 oraz ID3. Każdy z tych algorytmów, potencjalnie mogących znaleść zastosowanie w mojej pracy, przeanalizowano pod kątem zalet i ograczniczeń, które omówiono poniżej.

CART

Algorytm CART (Classification and Regression Trees) jest szeroko stosowany w praktyce, zarówno w zadaniach klasyfikacyjnych, jak i regresyjnych. Wykorzystuje on Gini index jako kryterium do podziału węzłów podczas konstruowania drzew decyzyjnych. Gini index mierzy "czystość"węzła, czyli to, jak jednorodne są klasy w danym węźle. Im niższa wartość Gini index, tym bardziej jednorodne są klasy, co oznacza lepszy podział.

- Zalety:
  - uniwersalność
    Algorytm CART może być stosowany zarówno w zadaniach klasyfikacji, jak i regresji, co sprawia, że jest wszechstronny i znajduje zastosowanie w różnych dziedzinach.
  - prostota interpretacji
    Drzewa decyzyjne generowane przez algorytm CART są łatwe do zrozumienia i interpretacji nawet dla osób bez specjalistycznej wiedzy matematycznej czy informatycznej.

- Wady:
  - nadmierny podział
    Algorytm CART ma tendencję do tworzenia zbyt skomplikowanych drzew decyzyjnych, co może prowadzić do nadmiernego dopasowania do danych treningowych (overfitting).
  - brak stabilności
    Mała zmiana w danych treningowych może prowadzić do znaczących zmian w strukturze drzewa decyzyjnego, co oznacza brak stabilności algorytmu.

ID3

ID3 to jeden z pierwszych algorytmów do budowy drzew decyzyjnych, opracowany przez Rossa Quinlana [26]. Jego główną cechą jest zdolność do obsługi danych kategorycznych. ID3 stosuje podejście iteracyjne, dzieląc dane na podgrupy, aż do utworzenia kompletnego drzewa decyzyjnego.

- Zalety:
  - prostota
    Algorytm ID3 jest prosty w implementacji i zrozumieniu, co czyni go atrakcyjnym dla osób nieposiadających specjalistycznej wiedzy.
  - łatwa czytelność modelu
    Drzewa decyzyjne generowane przez ID3 są łatwe do interpretacji, co ułatwia zrozumienie podejmowanych decyzji.
- Wady:
  - tendencja do przeuczenia dla małych zbiorów treningowych
    ID3 może wykazywać tendencję do nadmiernego dopasowania do danych treningowych, co prowadzi do niskiej skuteczności klasyfikacji dla nowych danych.
  - brak obsługi danych ciągłych
    Algorytm ID3 zakłada, że wszystkie atrybuty są dyskretne, co może być ograniczeniem w przypadku danych ciągłych.

C4.5

Algorytm C4.5, następca ID3, jest również jednym z popularnych podejść do budowy drzew decyzyjnych. Algorytm ten jest często wybierany ze względu na jego elastyczność i zdolność obsługi różnorodnych danych.

- Zalety:
  - obsługa atrybutów ciągłych i dyskretnych
    Algorytm C4.5 umożliwia obsługę zarówno atrybutów ciągłych, jak i dyskretnych, co czyni go bardziej wszechstronnym w porównaniu do niektórych innych metod.
  - automatyczna obsługa brakujących danych
    C4.5 radzi sobie z brakującymi danymi poprzez przypisywanie im prawdopodobieństw na podstawie znanych wartości. W przypadku atrybutu z brakującą wartością, algorytm rozdziela instancję na wszystkie możliwe wartości tego atrybutu, ważąc każdą gałąź proporcjonalnie do częstości występowania danej wartości w zbiorze treningowym [27, 28].
- Wady:
  - niedokładność w przypadku danych nierównomiernie rozkładanych
    W przypadku danych, w których klasy są nierównomiernie rozkładane, C4.5 może wykazywać tendencję do preferowania większościowej klasy.
  - tendencja do przeuczenia
    W przypadku dużych zbiorów danych z wieloma atrybutami, C4.5 może tworzyć bardzo głębokie i skomplikowane drzewa, co zwiększa ryzyko przeuczenia (overfitting) [27].

4 Reguły decyzyjne

Niniejszy rozdział poświęcono kluczowemu elementowi analizy danych - regułom decyzyjnym. Stanowią one niezwykle użyteczne narzędzie w modelowaniu zależności między atrybutami danych a ich klasami decyzyjnymi. Jedną z najbardziej atrakcyjnych cech reguł decyzyjnych jest ich przejrzystość, co umożliwia łatwe zrozumienie zależności zachodzących w danych.

4.1 Wprowadzenie do reguł decyzyjnych

Reguły decyzyjne są narzędziami reprezentacji wiedzy, które opisują związki między atrybutami warunkowymi a atrybutem decyzyjnym za pomocą implikacji.

Regułą decyzyjną nazwiemy formułę postaci: *Jeśli* ***warunki*** *to* ***decyzja*** zwykle zapisywane jako implikacja: ***warunki*** -> ***decyzja*** [2]

$$a_1 = v_1 \wedge a_4 = v_4 \wedge ... \wedge a_9 = v_9 \implies Dec = 1$$

Powyżej przedstawiono schemat reguł decyzyjnych składających się z deskryptorów warunkowych (np. $a_1 = v_1$, $a_2 = v_2$, ..., $a_9 = v_9$), implikacji ($\Rightarrow$) oraz decyzji (np. $Dec = 1$). Reguły decyzyjne opisują związki między atrybutami warunkowymi, a atrybutem decyzyjnym, za pomocą implikacji: po lewej stronie znajdują się warunki wyrażone pewną formułą logiczną, po prawej - wartość atrybutu decyzyjnego [7].

4.2 Podstawowe pojęcia

Z regułami decyzyjnymi związane są następujące pojęcia: poprzednik, następnik, pokrycie, długość oraz wsparcie reguły.

Poprzednik to część reguły zawierająca warunki.

Następnik to część reguły określająca decyzję lub klasę.

Długość reguły to liczba warunków zawartych w regule decyzyjnej. Wskazuje, ile warunków musi być spełnionych, aby reguła była spełniona. Krótsze reguły są łatwiejsze do zrozumienia, podczas gdy dłuższe mogą być bardziej złożone [29].

Pokrycie reguły to liczba przypadków w zbiorze danych, które spełniają warunki określone przez regułę decyzyjną. Przykładowo, jeśli reguła mówi "Jeżeli wiek jest powyżej 30 lat, to decyzja jest pozytywna", pokrycie to liczba osób powyżej 30 lat w zbiorze danych [30].

Wsparcie reguły to liczba lub odsetek przypadków w zbiorze danych, które spełniają warunki reguły. Reguły krótkie, złożone z niewielkiej liczby deskryptorów, zazwyczaj mają większe wsparcie.

Wykorzystanie powyższych pojęć w praktyce ilustruje poniższy przykład. Analizując zamieszczoną poniżej tabelę reguł decyzyjnych (Tabela 5), można zauważyć, że dla różnych kombinacji atrybutów "Parents", "Weather"i "Money"uzyskujemy różne decyzje dotyczące formy spędzania wolnego czasu.

Tabela 8: Reguły decyzyjne
Źródło: Opracowano na podstawie [7]

Poprzednik	Następnik	Pokr.	Dł.	Wsp.
Jeżeli Parents = No i Weather = Rainy	Stay In	1	2	0.1
Jeżeli Parents = Yes	Cinema	5	1	0.5
Jeżeli Parents = No i Weather = Sunny	Tennis	2	2	0.2
Jeżeli Parents = No i Weather = Windy i Money = Rich	Shopping	0.1	3	0.1
Jeżeli Parents = No i Weather = Windy i Money = Poor	Cinnema	1	3	0.1

Tabela 5 przedstawia reguły decyzyjne zbudowane na podstawie zbioru danych zawartego w Tabeli 6. Każda reguła ma formę "Poprzednik"→ "Następnik", gdzie "Poprzednik"określa warunki, a "Następnik"to decyzja. Kolumny "Pokr.", "Dł." i "Wsp."oznaczają odpowiednio pokrycie, długość i współczynnik zaufania reguły.

Tabela 9: Zbiór danych do analizy decyzji o spędzaniu wolnego czasu
Opracowano na podstawie [15]

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

4.3 Indukcja reguł z drzew decyzyjnych

Indukcja reguł opartych na drzewach decyzyjnych polega na przekształceniu hierarchicznej struktury drzewa decyzyjnego w zestaw reguł czytelnych dla człowieka, które określają warunki podejmowania decyzji. Aby stworzyć te reguły, przechodzimy przez drzewo decyzyjne od korzenia do każdego liścia, zapisując napotkane warunki atrybutów. W każdym węźle wewnętrznym warunek związany z wybraną gałęzią jest dodawany do reguły. Proces ten kontynuuje się, aż zostanie osiągnięty liść, w którym przewidywana klasa (w przypadku klasyfikacji) lub przewidywana wartość (w przypadku regresji) jest przypisywana do reguły [31].

4.4 Optymalizacja reguł decyzyjnych

Optymalizacja reguł decyzyjnych polega na szukaniu ich najlepszych wariantów pod względem różnych kryteriów, takich jak długość reguł, precyzja klasyfikacji, czy złożoność obliczeniowa. Celem optymalizacji jest znalezienie takich reguł decyzyjnych możliwie krótkich, które będą skuteczne i jednocześnie łatwe do zrozumienia i interpretacji.

Długość reguł decyzyjnych jest ważnym czynnikiem w określaniu ich jakości, zwłaszcza z punktu widzenia reprezentacji wiedzy. Wybór krótszych reguł może również zmniejszyć prawdopodobieństwo nadmiernego dopasowania modelu oraz poprawić jego zdolność do generalizacji [31].

Jednym z głównych kierunków optymalizacji reguł decyzyjnych jest także redukcja ich długości. Krótsze reguły są zazwyczaj prostsze i łatwiejsze do zrozumienia, co ułatwia ich wykorzystanie w praktyce. Istnieje wiele metod optymalizacji reguł decyzyjnych, które skupiają się na minimalizacji długości reguł przy jednoczesnym zachowaniu ich skuteczności klasyfikacyjnej.

W niniejszej pracy zwrócono szczególną uwagę na optymalizację reguł decyzyjnych względem ich długości. W celu przeprowadzenia globalnej optymalizacji wykorzystano koncepcję systemu rozproszonego, definiowanego w następujący sposób:

System rozproszony można zdefiniować jako zbiór powiązanych ze sobą podsystemów informacyjnych, gdzie każdy podsystem reprezentuje część oryginalnego zbioru danych, ale z uwzględnieniem tylko wybranych atrybutów. Bazując na teorii Pawlaka [14], możemy formalnie opisać taki system rozproszony jako:

$$SD = \{S_1, S_2, ..., S_n\} \tag{27}$$

gdzie:

- SD - system rozproszony
- $S_i = (U_i, A_i)$ - i-ty podsystem informacyjny (podtablica)
- $U_i \subseteq U$ - podzbiór obiektów z oryginalnego uniwersum U
- $A_i \subseteq A$ - podzbiór atrybutów (redukt) z oryginalnego zbioru atrybutów A

Każdy podsystem S_i w naszym przypadku odpowiada jednej z podtablic wygenerowanych na podstawie reduktów. Podejście to pozwala na zachowanie integralności informacji zawartych w oryginalnym zbiorze danych, jednocześnie umożliwiając bardziej efektywną analizę poprzez rozproszenie obliczeń na mniejsze, ale powiązane ze sobą podsystemy.

W naszym badaniu, dla każdego z głównych zbiorów danych (Lymphography i Tic-Tac-Toe), system rozproszony składa się z pięciu takich podsystemów (podtablic), co można zapisać jako:

$$SD = \{S_1, S_2, S_3, S_4, S_5\} \tag{28}$$

W tej pracy, każdy podsystem S_i odpowiada jednej z podtablic wygenerowanych na podstawie reduktów. Podtablice te powstały w oparciu o redukty generowane na podstawie algorytmu genetycznego w programie RSES, tworząc lokalne źródła danych w tablicy decyzyjnej.

Globalna optymalizacja reguł decyzyjnych względem długości, oznaczona jako $Rul_{len(T_i,r)}$, reprezentuje zbiór reguł przypisanych do konkretnego wiersza, po przeprowadzeniu optymalizacji względem ich długości. Podczas tego procesu spośród reguł indukowanych z drzew decyzyjnych i przypisanych do danego wiersza r, wybierane są tylko te, które posiadają minimalną wartość długości $Opt_{len(T_i,r)}$.

$$Opt_{len(T_i,r)} = \min\{rule_{len}^{T_i,r}\} \tag{29}$$

Spośród wszystkich reguł odpowiadających danemu wierszowi wybierane są tylko te, dla których:

$$rule_{len}^{T_i,r} = Opt_{len(T_i,r)} \tag{30}$$

W wyniku globalnej optymalizacji względem długości, każdy wiersz r z T_i ma przypisany zbiór reguł decyzyjnych $Rul_{len(T_i,r)}$.

Takie podejście, wykorzystujące system rozproszony, pozwala na efektywną optymalizację reguł decyzyjnych, jednocześnie zachowując integralność informacji zawartych w oryginalnym zbiorze danych.

5 Klasyfikacja

6 Eksperymenty

Przedmiotem niniejszej pracy były badania na dwóch zbiorach danych: Lymphography oraz Tic-Tac-Toe. Proces badawczy obejmował generowanie reduktów, tworzenie podtablic, usuwanie niespójności, tworzenie reguł decyzyjnych, dopasowywanie ich do konkretnych wierszy danych oraz optymalizację pod względem długości. W podrozdziale "Analiza wyników"znalazło się zestawienie wyników dla zbioru Lymphography i zbioru Tic-Tac-Toe oraz pięciu podtablic przed i po optymalizacji.

6.1 Opis zbiorów danych wykorzystanych do eksperymentów

W pracy wykorzystano dwa zbiory danych:

Tic-Tac-Toe zawiera 958 instancji reprezentujących możliwe konfiguracje planszy gry w kółko i krzyżyk pod koniec rozgrywki. Każda instancja składa się z dziewięciu cech, które opisują zawartość poszczególnych pól na planszy (górne lewe, górne środkowe, górne prawe, środkowe lewe, środkowe-środkowe, środkowe prawe, dolne lewe, dolne środkowe, dolne prawe) jako "x"(krzyżyk), "o"(kółko) lub "b"(puste pole). Głównym zadaniem klasyfikacyjnym jest przewidywanie, czy dany układ planszy kończy się wygraną dla gracza "x"[32].

Lymphography zawiera informacje medyczne dotyczące limfografii, czyli techniki obrazowania układu limfatycznego przy użyciu promieniowania rentgenowskiego [33]. Pozyskany z Uniwersyteckiego Centrum Medycznego, Instytutu Onkologii w Lublanie, zbiór ten składa się z 148 instancji i 19 cech, z których większość jest kategoryczna. Atrybut "class"stanowi klasę decyzyjną, reprezentując różne diagnozy związane z wynikami limfografii, takie jak "normal find", "metastases", "malign lymph"oraz "fibrosis"[34].

6.2 Metodologia badań

Proces przekształcania danych i indukcji reguł decyzyjnych, który doprowadził do uzyskania wyników przedstawionych w niniejszej pracy, obejmował zastosowanie narzędzia Rough Set Exploration System [35] oraz własnych skryptów w języku Python. W skryptach Pythona użyto następujących bibliotek: pandas [36], numpy [37], sklearn [38]. Kod źródłowy użyty do przeprowadzenia eksperymentów jest dostępny w repozytorium GitHub: decision-rules-global-optimalization.

W ramach eksperymentów wybrano 2 zbiory danych z repozytorium uczenia maszynowego UCI (https://archive.ics.uci.edu). Zbiory, na których przeprowadzono badania to:

- Lymphography – zawierający informacje medyczne dotyczące limfografii,
- Tic-Tac-Toe – będący pełnym zbiorem możliwych konfiguracji planszy w grze "kółko i krzyżyk".

Charakterystykę wybranych zbiorów przedstawiono w Tabeli 10.

Tabela 10: Charakterystyka zbiorów danych wybranych do eksperymentów.

Nazwa zbioru	**Liczba obiektów**	**Liczba atrybutów**	**Brakujące wartości**
Lymphography	148	18	Nie
Tic-Tac-Toe	958	9	Nie

Opracowanie własne

W ramach niniejszej pracy stworzony został skrypt w języku Python, za pomocą którego zostały przeprowadzone badania prezentowane w dalszej części pracy.

Żaden z wybranych zbiorów danych nie posiadał atrybutów ciągłych, dlatego też nie było potrzeby przeprowadzania dyskretyzacji danych w ramach ich wstępnej obróbki.

Proces badawczy obejmował następujące etapy:

1. **Generowanie reduktów:**
   Redukty zostały wygenerowane przy użyciu programu RSES, z wykorzystaniem algorytmu genetycznego. Dla każdego zbioru danych wygenerowano 5 reduktów.
2. **Tworzenie podtablic:**
   Na podstawie wygenerowanych reduktów utworzono podtablice dla każdego zbioru

danych. Każda podtablica zawierała atrybuty odpowiadające danemu reduktowi oraz kolumnę z klasą decyzyjną.

3. **Usuwanie niespójności:**
   Z utworzonych podtablic usunięto niespójności, przyjmując najczęściej występującą wartość decyzyjną dla zestawów o tych samych wartościach atrybutów warunkowych.

4. **Indukcja reguł decyzyjnych:**
   Dla każdej podtablicy wygenerowano reguły decyzyjne przy użyciu algorytmu CART z biblioteki scikit-learn. Parametry użyte do generowania drzew decyzyjnych to:

   - criterion = 'gini',
   - random_state = 1234.

5. **Dopasowanie reguł do danych:**
   Wygenerowane reguły zostały dopasowane do konkretnych wierszy w podtablicach. Dla każdego wiersza zidentyfikowano wszystkie pasujące reguły.

6. **Optymalizacja reguł decyzyjnych:**
   Przeprowadzono globalną optymalizację reguł decyzyjnych względem ich długości. Dla każdego wiersza wybrano najkrótsze pasujące reguły.

Analiza wyników obejmowała porównanie liczby unikalnych reguł oraz ich długości przed i po optymalizacji dla każdej podtablicy oraz dla całego zbioru danych.

6.3 Analiza wyników

W poniższym podrozdziale podsumowano wyniki szczegółowej analizy dwóch zbiorów danych: Lymphography oraz Tic-Tac-Toe. Przedstawiono zestawienie danych dla obu powyższych zbiorów oraz pięć podtablic wygenerowanych z reduktów dla każdego z nich. Co pozwoliło na określenie liczby wierszy oraz atrybutów dla każdego zbioru danych. Dodatkowo, przeprowadzono analizę wyników przed i po optymalizacji, prezentując zmiany w liczbie unikalnych reguł oraz ich długości dla każdej podtablicy.

Lymphography

W tabeli 10 przedstawiono zestawienie danych dla zbioru Lymphography oraz pięciu podtablic utworzonych z reduktów tj.: Lymphography-1, Lymphography-2, Lymphography-3, Lymphography-4 i Lymphography-5.

Tabela 11: Zestawienie danych dla zbioru danych Lymphography

Zbiór	**Liczba wierszy**	**Liczba atrybutów**	**Atrybuty**
Lymphography	148	17	A1 = lymphatics, A2 = block_of_affere, A3 = bl_of_lymph_c, A4 = bl_of_lymph_s, A5 = by_pass, A6 = extravasates, A7 = regeneration_of, A8 = early_uptake_in, A9 = lym_nodes_dimin, A10 = lym_nodes_enlar, A11 = changes_in_lym, A12 = defect_in_node, A13 = changes_in_node, A14 = changes_in_stru, A15 = special_forms, A16 = dislocation_of, A17 = exclusion_of_no, A18 = no_of_nodes_in
Lymphography-1	129	6	A2, A13, A14, A15, A16 A18,

Lymphography-2	136	7	A2, A12, A13, A14, A15, A17, A18
Lymphography-3	138	7	A2, A11, A12, A13, A14, A15, A18
Lymphography-4	134	7	A2, A8, A11, A13, A14, A15, A18
Lymphography-5	133	7	A2, A8, A13, A14, A15, A17, A18,

Liczba wierszy dla zbiorów wynosiła 148, a atrybuty obejmowały następujące zmienne: lymphatics, block_of_affere, bl_of_lymph_c, bl_of_lymph_s, by_pass, extravasates, re-

generation_of, early_uptake_in, lym_nodes_dimin, lym_nodes_enlar, changes_in_lym, defect_in_node, changes_in_node, changes_in_stru, special_forms, dislocation_of, exclusion_of_no oraz no_of_nodes_in.

Wyniki dla zbioru danych Lympography i podtablic wygenerowanych z reduktów przed i po optymalizacji obrazują tabele 11 i 12.

Tabela 12: Analiza wyników dla zbioru Lymphography

Zbiór	**Głębokość**	**Unikalne reguły**	**min**	**avg**	**max**
Lymphography	8	30	3	5,291	8
Lymphography-1	10	39	2	5,364	10
Lymphography-2	11	43	3	6,022	11
Lymphography-3	11	43	3	5,957	11
Lymphography-4	10	39	3	5,709	10
Lymphography-5	10	42	3	5,917	10

W tabeli 11 zaprezentowano wyniki analizy dla zbioru danych Lymphography przed optymalizacją. Najmniejsza liczba unikalnych reguł wystąpiła dla pełnego zbioru Lymphography i wynosiła 30, a największa dla zbiorów Lymphography-2 i Lymphography-3 w liczbie 43. Najmniejszą głębokość miał zbiór Lymphography, a największą Lymphography-2 i Lymphography-3 odpowiednio 8 i 11. Najniższa średnia długość reguł wystąpiła w zbiorze Lymphography, a najwyższa dla zbioru Lymphography-3. Najkrótsza reguła o długości 2 wystąpiła w zbiorze Lymphography-1, a najdłuższa o długości 11 wystąpiła w podtablicach Lypmhography-2 i Lymphography-3.

Tabela 13: Analiza wyników dla zbioru Lymphography po optymalizacji

Zbiór	**Liczba unikalnych reguł**	**min**	**avg**	**max**
Lymphography	171	2	4,849	8

Po optymalizacji zaobserwowano znacząca redukcja średniej długości reguł decyzyjnych. Najdłuższe i najkrótsze reguły to te, które w podtablicach miały najmniejsze wartości.

Tic-Tac-Toe

W tabeli 18 przedstawiono zestawienie danych dla zbioru Tic-Tac-Toe oraz pięciu podtablic utworzonych z reduktów tj.: Tic-Tac-Toe-1, Tic-Tac-Toe-2, Tic-Tac-Toe- 3, Tic-Tac-Toe-4 i Tic-Tac-Toe-5.

Tabela 14: Zestawienie danych dla zbioru danych Tic-Tac-Toe

Zbiór	**Liczba wierszy**	**Liczba atrybutów**	**Atrybuty**
Tic-Tac-Toe	958	9	B1 = top-left-square, B2 = top-middle-square, B3 = top-right-square, B4 = middle-left-square, B5 = middle-middle-square, B6 = middle-right-square, B7 = bottom-left-square, B8 = bottom-middle-square, B9 = bottom-right-square
Tic-Tac-Toe-1	958	8	B1, B2, B3, B4, B6, B7, B8, B9

Tic-Tac-Toe-2	958	8	top-left-square, B2, B3, B4, B5, B6, B8, B9
Tic-Tac-Toe-3	958	8	B1, B2, B3, B4, B5, B6, B7, B8
Tic-Tac-Toe-4	958	8	B1, B3, B4, B5, B6, B7, B8, B9
Tic-Tac-Toe-5	958	8	B1, B2, B3, B4, B5, B7, B8, B9

Liczba wierszy dla zbiorów wynosiła 948, a atrybuty obejmowały następujące zmienne: top-left-square, top_middle_square, top_right_square, middle_left_square, middle_middle_square, middle_right_square, bottom_left_square, bottom_middle_square, bottom_right_square.

W tabeli 14 zaprezentowano wyniki analizy dla zbioru danych Tic-Tac-Toe przed optymalizacją.

Tabela 15: Analiza wyników dla zbioru Tic-Tac-Toe

Zbiór	**Głębokość**	**Liczba unikalnych reguł**	**min**	**avg**	**max**
Tic-Tac-Toe	12	73	3	4,908	12
Tic-Tac-Toe-1	16	347	3	8,054	16
Tic-Tac-Toe-2	15	213	4	6,868	15
Tic-Tac-Toe-3	15	213	3	6,684	15
Tic-Tac-Toe-4	14	119	3	5,200	14
Tic-Tac-Toe-5	13	118	3	5,167	13

Najmniejsza liczba unikalnych reguł wystąpiła dla pełnego zbioru Tic-Tac-Toe i wynosiła 73, a największa dla zbiorów Tic-Tac-Toe-1 w liczbie 347. Najmniejszą głębokość miał zbiór Tic-Tac-Toe, a największą Lymphography-1 odpowiednio 12 i 16. Najwyższa średnia długość reguł wystąpiła w zbiorze Tic-Tac-Toe-1, a najniższa była dla zbioru Tic-Tac-Toe-5.

W tabeli 15 przedstawiono wyniki analizy po optymalizacji.

Tabela 16: Analiza wyników dla zbioru Tic-Tac-Toe po optymalizacji

Zbiór	**Liczba unikalnych reguł**	**min**	**avg**	**max**
Tic-Tac-Toe	x	3	3,951	10

7 Podsumowanie

8 Bibliografia

[1] *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025*. 2023. URL: https://www.statista.com/statistics/871513/worldwide-data-created/.

[2] P. Cichosz. *Systemy uczące się*. Warszawa: WNT, 2000, s. 211–218.

[3] T. Morzy. *Eksploracja danych: Metody i algorytmy*. Warszawa: Wydawnictwo Naukowe PWN, 2013, s. 193–203.

[4] K. Racka. *Metody eksploracji danych i ich zastosowanie*. 2015, s. 1–5.

[5] GeeksforGeeks. *KDD Process in Data Mining*. GeeksforGeeks. 2023. URL: https://www.geeksforgeeks.org/kdd-process-in-data-mining/ (term. wiz. 02. 07. 2024).

[6] F. Duarte. 2023. URL: https://explodingtopics.com/blog/data-generated-per-day.

[7] D. T. Larose. *Odkrywanie wiedzy z danych Wprowadzenie do eksploracji danych*. 2006, s. 342–351. ISBN: 978-83-01-14836-2.

[8] I. H. Witten, E. Frank i M. A. Hall, red. Boston: Morgan Kaufmann, 2011, s. 587–605. ISBN: 978-0-12-374856-0. DOI: https://doi.org/10.1016/B978-0-12-374856-0.00023-7. URL: https://www.sciencedirect.com/science/article/pii/B9780123748560000237.

[9] P. Cichosz. *Systemy uczące się*. Warszawa: WNT, 2000, s. 384.

[10] H. Mannila D. Hand i P. Smyth. *Eksploracja danych*. Warszawa: WNT, 2005.

[11] J. Pei J. Han M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[12] Leo Breiman i in. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[13] K. Krawiec i J. Stefanowski. *Sieci neuronowe i uczenie maszynowe*. Poznan: Wydawnictwo Politechniki Poznańskiej, 2003. ISBN: 83-7143-455-3.

[14] Zdzisław Pawlak. *Systemy informacyjne: podstawy teoretyczne*. Warszawa: Wydawnictwo Naukowe PWN, 1983.

[15] S. Colton. *Decision Tree Learning*. 2006. URL: https://www.sfu.ca/iat813/lectures/lecture6.html.

[16] Ł. Bujak. „Drzewa decyzyjne". W: *Uniwersytet Mikołaja Kopernika* (2008), s. 4–9.

[17] J. Mulawka. *Systemy ekspertowe*. WNT, 1996.

[18] H. S. Nguyen i H. S. Nguyen. „Pattern extraction from data". W: *Fundamenta Informaticae* 1-2 (1998), s. 129–144.

[19] Y. Song i Y. Lu. „Decision tree methods: applications for classification and prediction". W: *Shanghai Arch Psychiatry* (2015), s. 130–135. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/.

[20] V. Kumar P. Tan M. Steinbach i A. Karpatne. *Introduction to Data Mining*. Pearson Education, 2019. ISBN: 9780273775324. URL: https://books.google.pl/books?id=274oEAAAQBAJ.

[21] B. Zielosko i U. Stańczyk. „Reduct-based ranking of attributes". W: *Procedia Computer Science* (2020), s. 2576–2585.

[22] F. Farris. *The Gini Index and Measures of Inequality*. 2010, s. 851–864.

[23] Teresa Zawadzka i Wojciech Waloszek. *Proces eksploracji danych*. https://enauczanie.pg.edu.pl/moodle/pluginfile.php/213583/mod_resource/content/1/ED_ProcesEksploracjiDanych.pdf.

[24] J. Ross Quinlan. „C4.5: Programs for Machine Learning". W: *Machine Learning* 16.3 (1993), s. 235–240. DOI: 10.1007/BF00993309.

[25] J. Ross Quinlan. „Induction of Decision Trees". W: *Machine Learning* 1.1 (1986), s. 81–106. DOI: 10.1007/BF00116251.

[26] R. Quinlan. „Data Mining from an Al Perspective". W: *Proceedings 15th International Conference on Data Engineering*. 1999, s. 186. DOI: 10.1109/ICDE.1999.754923.

[27] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann Publishers, 1993.

[28] Fadwa Aaboub, Hasna Chamlal i Tayeb Ouaderhman. „Statistical analysis of various splitting criteria for decision trees". W: *Journal of Algorithms & Computational Technology* (paź. 2023). DOI: 10.1177/17483026231198181. URL: https://www.researchgate.net/publication/374795587_Statistical_analysis_of_various_splitting_criteria_for_decision_trees.

[29] G. D. Greenwade. *Proces eksploracji danych*. 1993, s. 342–351. URL: https://enauczanie.pg.edu.pl/moodle/pluginfile.php/213583/mod_resource/content/1/ED_ProcesEksploracjiDanych.pdf.

[30] F. Chollet. *Deep Learning Praca z językiem Python i biblioteką Keras*. 2019, s. 342–351.

[31] B. Zielosko, E. T. Tetteh i D. Hunchak. „Multi-heuristic Induction of Decision Rules". W: *Rough Sets, International Joint Conference*. Springer Nature Switzerland, 2023, s. 18–30. ISBN: 978-3-031-50959-9.

[32] D. Aha. *Tic-Tac-Toe Endgame*. 1991. URL: https://archive.ics.uci.edu/dataset/101/tic+tac+toe+endgame.

[33] *Stanford Medicine Children's Health*. 2024.

[34] M. Zwitter i M. Soklic. *Lymphography*. 1988. URL: https://archive.ics.uci.edu/dataset/63/lymphography.

[35] D. Szczuka. *Rachunek różniczkowy i całkowy*. URL: https://www.mimuw.edu.pl/~szczuka/rses/start.html (term. wiz. 20. 05. 2024).

[36] *pandas*. URL: https://pandas.pydata.org/ (term. wiz. 20. 05. 2024).

[37] *NumPy*. URL: https://numpy.org/ (term. wiz. 20. 05. 2024).

[38] *scikit-learn*. URL: https://scikit-learn.org/ (term. wiz. 20. 05. 2024).

9 Ilustracje i wykresy

Spis rysunków

Spis tabel i tablic decyzyjnych

A Spis algorytmów