ID3 Algorithm

Opis i implementacja

Wstęp

Algorytmy klasyfikacji stanowią istotny element współczesnej analizy danych i uczenia maszynowego. Jednym z najprostszych, a jednocześnie intuicyjnych podejść do tego problemu są drzewa decyzyjne, które odwzorowują proces podejmowania decyzji w postaci struktury hierarchicznej.

Dzięki swojej przejrzystości oraz łatwej interpretowalności znajdują one zastosowanie zarówno w zadaniach edukacyjnych, jak i w praktycznych systemach wspomagania decyzji. Jednym z klasycznych algorytmów służących do ich budowy jest ID3 (Iterative Dichotomiser 3), wykorzystujący dane treningowe do budowy drzewa decyzyjnego.

1. Opis algorytmu

Celem algorytmu ID3 jest stworzenie drzewa decyzyjnego, które będzie w stanie klasyfikować dane na podstawie atrybutów. Drzewo budowane jest w taki sposób, aby na wyższych poziomach znajdowały się atrybuty które maksymalizują zysk informacyjny, natomiast na niższych poziomach atrybuty o mniejszej wartości informacyjnej. Dzięki temu model może osiągać możliwie wysoką skuteczność klasyfikacji.

1.1 Entropia i zysk informacyjny

Kluczowym elementem algorytmu ID3 jest wybór atrybutu, który będzie użyty do podziału danych na każdym poziomie drzewa. Do tego celu wykorzystuje się pojęcie entropii, które mierzy niepewność lub niejednorodność zbioru danych, czyli stopień nieuporządkowania danych. Na podstawie tego wyliczany jest zysk informacyjny - informacja o tym, jak bardzo dany podział redukuje tę niepewność.

Entropia (*ang. entropy*) jest definiowana jako:

$$H(S) = -\sum_{i=1}^{c} f_i \log_2(f_i), \quad f_i > 0$$

gdzie S to zbiór danych, c to liczba klas, a f_i to proporcja elementów należących do klasy i w zbiorze S, czyli $f_i = \frac{|S_i|}{|S|}$, gdzie S_i to zbiór elementów należących do klasy i.

Zysk informacyjny (*ang. information gain*) jest definiowany jako:

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

gdzie S to zbiór danych, A to rozważany atrybut, $Values(A)$ to zbiór jego możliwych wartości, a S_v to podzbiór danych, dla którego atrybut A przyjmuje wartość v.

1.2 Przebieg algorytmu

Algorytm ID3 działa rekurencyjnie, wykonując następujące kroki:

1. Jeśli wszystkie elementy w zbiorze S należą do tej samej klasy, zwróć liść oznaczony tą klasą.
2. Jeśli zbiór atrybutów jest pusty lub osiągnięto maksymalną głębokość drzewa, zwróć liść z klasą najczęściej występującą w S.
3. Wybierz atrybut A, dla którego zysk informacyjny $IG(S, A)$ jest największy.
4. Dla każdej wartości v atrybutu A:
   (a) Utwórz podzbiór S_v zawierający elementy, dla których $A = v$.
   (b) Jeśli S_v jest pusty, dodaj liść z klasą najczęściej występującą w S.
   (c) W przeciwnym razie, utwórz gałąź i wywołaj algorytm rekurencyjnie dla S_v oraz zbioru atrybutów bez A.

2. Metody oceny jakości klasyfikatora

Gdy już zbudujemy drzewo decyzyjne, ważne jest, aby ocenić jego skuteczność. Dzięki temu możemy określić, jak dobrze model radzi sobie z klasyfikacją nowych danych, które nie były używane podczas treningu. Istnieje wiele metryk pozwalających ocenić jakość klasyfikatora, a wśród najpopularniejszych znajdują się macierz pomyłek (*ang. confusion matrix*) oraz dokładność (*ang. accuracy*).

2.1 Macierz pomyłek

Macierz pomyłek to narzędzie pozwalające zobrazować, jak klasyfikator radzi sobie z różnymi klasami danych. Jest ona wyznaczana poprzez porównanie przewidywanych klas z rzeczywistymi klasami w zbiorze testowym. Dla N klas macierz pomyłek jest macierzą kwadratową o wymiarach $N \times N$:

$$\begin{array}{c} \\ \text{True 1} \\ \text{True 2} \\ \vdots \\ \text{True N} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Pred 1} & \text{Pred 2} & \cdots & \text{Pred N} \end{array} \\ \begin{bmatrix} n_{1,1} & n_{1,2} & \cdots & n_{1,N} \\ n_{2,1} & n_{2,2} & \cdots & n_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ n_{N,1} & n_{N,2} & \cdots & n_{N,N} \end{bmatrix} \end{array}$$

gdzie wiersze odpowiadają klasom rzeczywistym, a kolumny klasom przewidywanym przez model. Element $n_{i,j}$ oznacza liczbę przypadków, dla których rzeczywista klasa to i, a model przewidział klasę j. Wtedy, elementy na przekątnej $n_{i,i}$ reprezentują poprawne klasyfikacje, natomiast pozostałe elementy to błędne klasyfikacje, co pozwala na analizę, które klasy są mylone przez model oraz wyliczenie dokładności.

2.2 Dokładność

Dokładność to jedna z najprostszych metryk oceny klasyfikatora, która mierzy stosunek liczby poprawnych klasyfikacji do całkowitej liczby przypadków. Można ją obliczyć jako:

$$Accuracy = \frac{\sum_{i=1}^{N} n_{i,i}}{\sum_{i=1}^{N} \sum_{j=1}^{N} n_{i,j}}$$

3. Algorytm w działaniu

Aby zobrazować działanie algorytmu ID3, rozważmy przykład klasyfikacji końcówek w grze kółko i krzyżyk [1]. Załóżmy, że mamy zbiór danych zawierający różne konfiguracje planszy oraz etykiety wskazujące, czy dana konfiguracja jest zwycięska dla gracza X.

Każda konfiguracja planszy reprezentowana jest jako wektor atrybutów, gdzie każdy atrybut odpowiada jednemu polu planszy i przyjmuje wartość: x, o lub b (puste pole).

Etykieta **Positive** oznacza sytuację wygrywającą dla gracza X, natomiast **Negative** w przeciwnym przypadku.

A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**Label**
x	o	x	o	x	b	b	b	x	Positive
o	x	o	x	o	b	b	b	o	Negative
...	...	...	...	...	...	...	...	...	...

Zbiór danych zostaje losowo podzielony na zbiory: treningowy, walidacyjny oraz testowy w proporcjach odpowiednio 70%, 15% oraz 15%.

3.1 Trening i walidacja

Podczas treningu algorytm ID3 analizuje atrybuty planszy, obliczając entropię oraz zysk informacyjny dla każdego z nich.

- Dla każdej wartości parametru *max depth* od 1 do 9 budowane jest drzewo decyzyjne na zbiorze treningowym.
- Następnie dla każdego drzewa obliczana jest dokładność na zbiorze walidacyjnym.
- Na podstawie wyników walidacji wybierana jest optymalna wartość parametru *max depth*, która zapewnia najwyższą dokładność.
- Wybrane drzewo jest następnie wykorzystywane do klasyfikacji danych testowych.

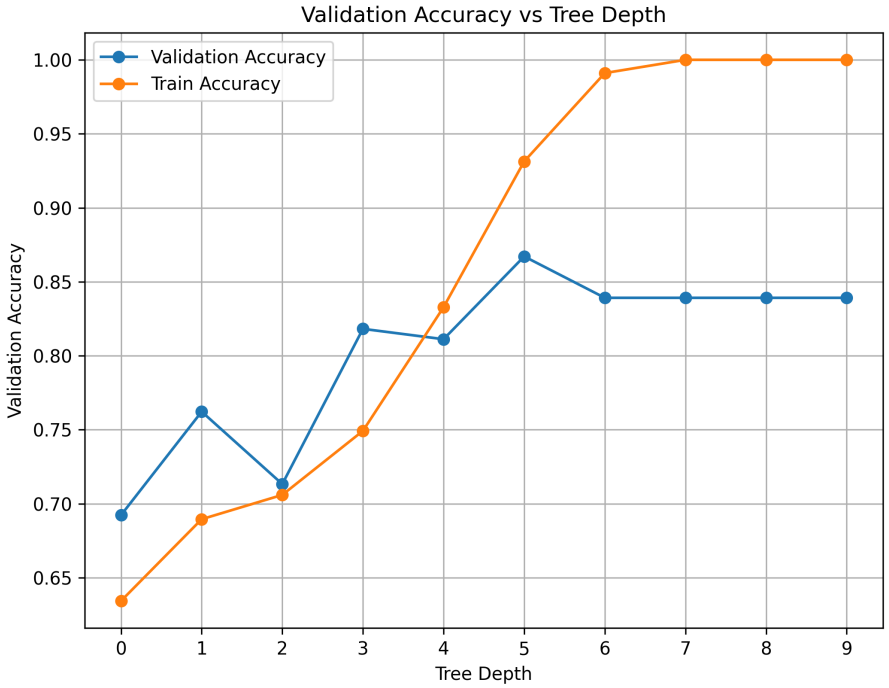
Ostateczna ocena jakości modelu przeprowadzana jest na zbiorze testowym, który nie był wykorzystywany podczas procesu uczenia ani doboru parametrów.

[1] Tic-Tac-Toe Endgame

3.2 Po co właściwie walidacja?

Walidacja hiperparametrów jest kluczowym etapem w procesie budowy drzewa decyzyjnego, ponieważ pozwala na wyznaczenie optymalnej głębokości drzewa, zapewniającej najlepszą zdolność generalizacji modelu.

Wyniki walidacji przedstawiono na wykresie poniżej, gdzie na osi X znajduje się wartość parametru *max depth*, natomiast na osi Y dokładność modelu na zbiorze treningowym oraz walidacyjnym.

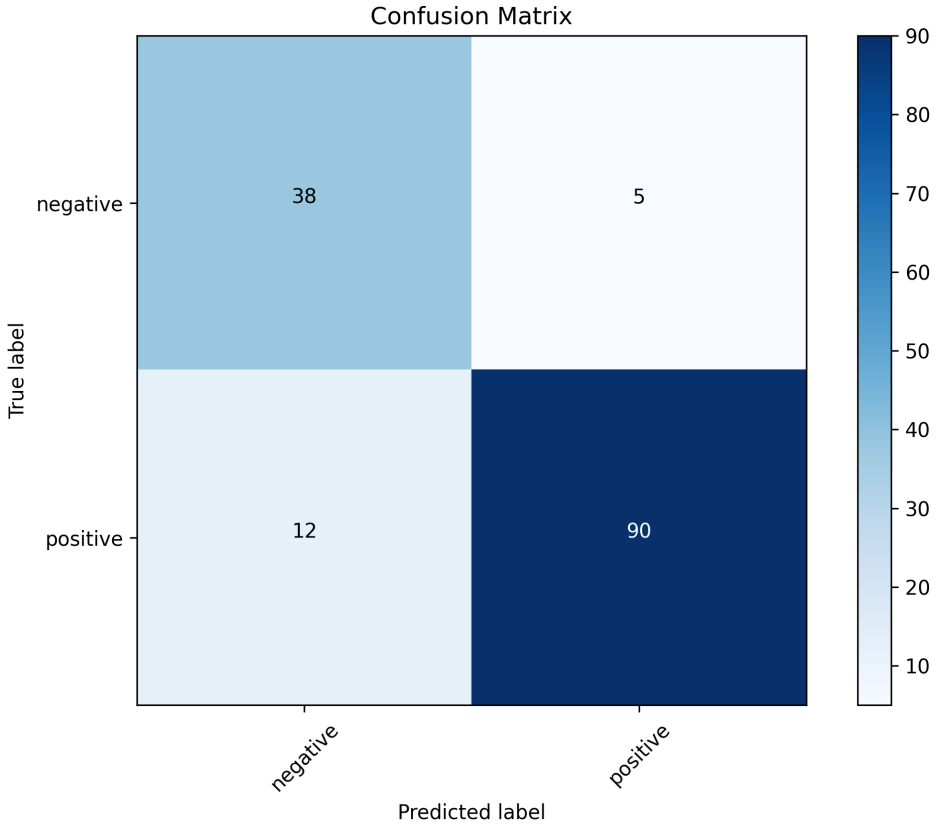


Na podstawie wykresu można sformułować następujące wnioski:

- Dla małych wartości *max depth* (np. 1–3) model jest zbyt prosty, co prowadzi do niskiej dokładności zarówno na zbiorze treningowym, jak i walidacyjnym (underfitting).
- Wraz ze wzrostem głębokości drzewa (np. 4–5) dokładność na zbiorze walidacyjnym rośnie, osiągając maksimum w okolicach *max depth* = 5, co wskazuje na dobrą zdolność generalizacji.
- Dla większych wartości *max depth* (np. 6–9) dokładność na zbiorze treningowym nadal rośnie, natomiast na zbiorze walidacyjnym zaczyna spadać, co świadczy o przeuczeniu modelu (overfitting) – zbytniego przystosowania do danych treningowych.
- Odpowiednia wartość parametru *max depth* wynosi 5, ponieważ zapewnia największą dokładność na zbiorze walidacyjnym, co sugeruje, że model jest wystarczająco złożony, aby uchwycić istotne wzorce w danych, ale nie na tyle złożony, aby dopasować się do szumu w zbiorze treningowym.

4. Testy i wyniki

Po wybraniu optymalnej głębokości drzewa model został przetestowany na zbiorze testowym, który nie był wykorzystywany podczas treningu ani walidacji, dzięki czemu uzyskane wyniki stanowią wiarygodną ocenę zdolności generalizacji modelu.



4.1 Macierz pomyłek

Dla tego problemu macierz pomyłek można opisać czterema wartościami: *TP*, *TN*, *FP* oraz *FN*. Jako klasę pozytywną przyjęto **Positive** (wygrana gracza X).

- $TN = 38$ - poprawnie rozpoznane **Negative**
- $FP = 5$ - **Negative** błędnie jako **Positive**
- $FN = 12$ - **Positive** błędnie jako **Negative**
- $TP = 90$ - poprawnie rozpoznane **Positive**

Łącznie: 145 przykładów, z czego 17 błędnych.

4.2 Metryki jakości

Dokładność jest najprostszą miarą jakości klasyfikatora, jednak nie zawsze jest wystarczająca. W szczególności w przypadku niezrównoważonych danych może prowadzić do mylących wniosków, ponieważ nie uwzględnia rodzaju popełnianych błędów. W analizowanym zbiorze dane są niezrównoważone (więcej przykładów klasy Positive), co uzasadnia konieczność użycia innych metryk.

Dlatego oprócz accuracy analizuje się również inne metryki oparte na macierzy pomyłek, Wszystkie metryki obliczono względem klasy Positive jako klasy pozytywnej:

- **Accuracy**:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{90 + 38}{145} = \frac{128}{145} \approx 0{,}883$$

- **Precision**:

$$Precision = \frac{TP}{TP + FP} = \frac{90}{90 + 5} = \frac{90}{95} \approx 0{,}947$$

- **Recall**:

$$Recall = \frac{TP}{TP + FN} = \frac{90}{90 + 12} = \frac{90}{102} \approx 0{,}882$$

- **F1-score**:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{0{,}947 \cdot 0{,}882}{0{,}947 + 0{,}882} \approx 0{,}914$$

4.3 Wnioski

Model osiągnął dokładność $\approx 88{,}3\%$ na zbiorze testowym, co potwierdza dobrą zdolność generalizacji przy głębokości drzewa równej 5.

- Analiza macierzy pomyłek ujawnia asymetrię błędów: liczba wyników fałszywie negatywnych ($FN = 12$) jest większa niż fałszywie pozytywnych ($FP = 5$), co oznacza, że model częściej nie rozpoznaje wygranych pozycji gracza X.
- Wysoka wartość precision ($\approx 94{,}7\%$) wskazuje, że predykcje klasy **Positive** są zazwyczaj poprawne, natomiast niższy recall ($\approx 88{,}2\%$) oznacza, że część wygranych konfiguracji pozostaje nierozpoznana.
- Miara $F_1 \approx 91{,}4\%$ potwierdza dobry kompromis między precyzją a czułością, co jest istotne w kontekście oceny jakości klasyfikatora.
- W kontekście analizowanego problemu większa liczba błędów typu FN oznacza, że model może nie wykrywać części wygrywających pozycji gracza X. W zależności od zastosowania może to być bardziej niepożądane niż błędy typu FP, dlatego w praktyce można rozważać dostrajanie modelu w kierunku zwiększenia czułości (recall), nawet kosztem niewielkiego spadku precyzji.
- Uzyskane wyniki pokazują, że sama dokładność nie oddaje w pełni jakości modelu, ponieważ nie rozróżnia typów błędów, dlatego konieczna jest analiza dodatkowych metryk.

Podsumowanie

W ramach niniejszego sprawozdania przedstawiono zasadę działania algorytmu ID3 oraz jego praktyczne zastosowanie do budowy drzewa decyzyjnego. Omówiono kluczowe pojęcia wykorzystywane przez metodę, tj. entropię i zysk informacyjny, a także opisano rekurencyjny sposób konstruowania drzewa poprzez wybór atrybutów najlepiej rozdzielających dane na kolejne podzbiory.

Część eksperymentalna dotyczyła klasyfikacji końcowych konfiguracji w grze kółko i krzyżyk, gdzie atrybutami były pola planszy o wartościach x, o lub b, a etykieta określała, czy układ jest zwycięski dla gracza X. Dane podzielono na zbiory treningowy, walidacyjny i testowy, a walidację wykorzystano do doboru hiperparametru *max depth* w taki sposób, aby zwiększać złożoność modelu tylko do momentu, w którym poprawia to generalizację.

Na podstawie krzywych dokładności dla różnych głębokości wybrano drzewo o głębokości równej 5, co dało najlepszy kompromis między niedouczeniem a przeuczeniem. Uzyskana na zbiorze testowym dokładność $\approx 88{,}3\%$ potwierdza, że nawet relatywnie prosta metoda, jaką jest ID3, potrafi uchwycić istotne zależności w danych opisujących stan gry. Analiza macierzy pomyłek wskazała jednocześnie na przewagę błędów typu fałszywie negatywnego, co oznacza, że model częściej pomija pozycje wygrywające niż błędnie je wskazuje.

Ostatecznie otrzymane wyniki pokazują, że ograniczanie głębokości drzewa jest skutecznym i prostym sposobem kontroli nadmiernego dopasowania w ID3. W dalszych pracach jako naturalne kierunki ulepszeń można rozważyć m.in. przycinanie drzewa (pruning), walidację krzyżową zamiast pojedynczego podziału danych oraz zastosowanie algorytmów będących rozwinięciem ID3 (np. C4.5), które lepiej radzą sobie z szumem, brakami danych i doborem podziałów.