Eryk Wawrzyn 291018

# UM Projekt

Temat: Drzewo decyzyjne, które potrafi zmieniać się wskutek nadchodzenia nowych danych trenujących (uczenie przyrostowe).

**Budowanie podstawowego drzewa**

Na początku należy znaleźć najlepsze kryterium, według którego dane zostaną rozdzielone na dwa zbiory (pierwszy, „prawy” zbiór z danymi spełniającymi założenia i drugi „lewy” zbiór z danymi niespełniającymi założenia).

W tym celu należy przejść po wszystkich znajdujących się w zbiorze danych chrakteryzujących przedmioty (dla których ma nastąpić podział). Dla każdej danej „opisującej” należy sprawdzić jaki nastąpiłby podział i jaki zysk informacyjny (niżej wytłumaczenie i sposób liczenia) zostałby uzyskany, jeżeli jako kryterium podziału użyto by wartość tego elementu.

Zysk informacyjny to miara pozwalająca wybrać atrybut, który najbardziej zminimalizuje ilość informacji niezbędnej do klasyfikacji przykładów w partycjach uzyskanych w wyniku podziału.

$$I = GINY(c) - pGINY(f) - (1-p)GINY(t)$$

I – zysk informacyjny

GINY – współczynnik Giniego

f – zbiór danych niespełniających warunków według którego zbiór miałby zostać podzielony

t – zbiór danych spełniający warunki według którego zbiór miałby zostać podzielony

c – cały zbór danych

$$p = \frac{ilość\ elementów\ f}{ilość\ elementów\ f + ilość\ elementów\ t}$$

Współczynnik Giniego określa jak rozproszony jest zbór danych, im wychodzi mniejszy tym lepiej, ponieważ wtedy nasz zbór jest bardziej jednolity.

$$GINY = 1 - \Sigma \left(\frac{Ilość\ wystąpienia\ poszczególnych\ "produktów"}{Ilość\ wszyskich\ danych}\right)^2$$

Po przejściu przez wszystkie elementy należy dokonać podziału według elementów, dzięki któremu uzyskano największy współczynnik informacji i zapisać poniższe elementy

- kryterium, według którego został dokonany podział

- uzyskaną wartość współczynnika zysku informacji

- dane spełniające danego kryterium

- dane niespełniające danego kryterium

Następnie powyższe operacje należy wykonywać dla uzyskanych danych spełniających i niespełniających wybrane kryterium (rekurencyjnie) aż zostanie uzyskany zysk informacyjny równy zero. Na końcu dla każdego liścia należy policzyć jak dany zbiór jest rozproszony (współczynnik Giny).

Aby zadbać żeby drzewo nie było nadmiernie dopasowane, gdy będzie chciał utworzyć się liść z liczebnością danych równą np. jednej tysięcznej liczebności całego zbioru należy to uniemożliwić i zakończyć na poprzedzającym węźle i go „ustawić" jako liść.

**Prosty przykład**

Dane trenujące:

| Kolor | Długoś ogona | |
|---|---|---|
| Czarny | Długi | Kot |
| Biały | Krótki | Kot |
| Czarny | Krótki | Pies |

1. Korzystając z wzorów z poprzedniej strony obliczany zysk informacyjny dla każdej danej opisującej. Poniżej przykład dla atrybutu „czarny".

$GINY(c)$ = 1 – $\Sigma \left(\frac{Ilość\ wystąpienia\ poszczególnych\ "produktów"}{Ilość\ wszyskich\ danych}\right)^2$ = 1 – (2/3)^2 - (1/3)^2 = 4/9

$GINY(f)$ = 1 – 1 = 0

$$GINY(t) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = 0{,}5$$

$$p = \frac{ilość\ elementów\ f}{ilość\ elementów + ilość\ elementów\ t} = \frac{1}{1+2} = \frac{1}{3}$$

$I = GINY(c) - pGINY(f) - (1-p)GINY(t)$ = 0,11

Powyższą operację należy wykonać jeszcze dla atrybutów „Biały” (I l=0,11), „Długi” (I=0,11), „Krótki” (I=0,11), w tym wypadku wszystkie zyski informacyjne są równe więc bierzemy pierwszy element, czyli czarny i zapisujemy:

- kryterium, według którego został dokonany podział

- uzyskaną wartość współczynnika zysku informacji

- dane spełniające danego kryterium

- dane niespełniające danego kryterium

Niespełnione

| Kolor | Długość ogona | |
|---|---|---|
| Czarny | Długi | Kot |
| Czarny | Krótki | Pies |

Spełnione

| Kolor | Długość ogona | |
|---|---|---|
| Biały | Krótki | Kot |

Powtarzamy procedurę dla danych po prawej stronie, uzyskujemy I = 0, więc zostawiamy już te dane.

Przechodzimy do lewych danych i poważamy operacje, przechodzimy przez atrybuty „Długi” (I=0,5), „Krótki” (I=0,5), „Długi” (I=0), wybieramy kryterium Długi.

Niespełnione

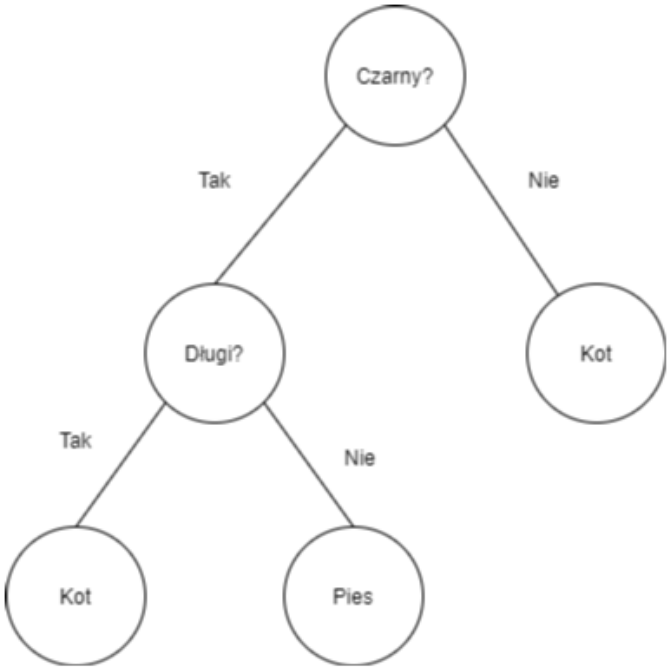
| Kolor | Długość ogona | |
|---|---|---|
| Czarny | Krótki | Pies |

Spełnione

| Kolor | Długość ogona | |
|---|---|---|
| Czarny | Długi | Kot |

Powtarzamy procedurę dla danych po prawej stronie, uzyskujemy I = 0, więc zostawiamy już te dane.

Powtarzamy procedurę dla danych po lewej stronie, uzyskujemy I = 0, więc zostawiamy już te dane, a budowę drzewa ukończono.




*Rys. 1 Uzyskane drzewo decyzyjne*

**Uczenie przyrostowe drzewa**

W przypadku gdy zostały dostarczone nowe dane trenujące, a drzewo utworzone z poprzednich danych jest duże, nie opłaca się budować drzewa od nowa. Dlatego gdy nowych danych trenujących będzie znacznie więcej (np. 10 razy) od danych, które utworzyły dotychczasową strukturę, drzewo zostanie zbudowane od nowa, w przeciwnym wypadku zostaną zastosowane przedstawione poniżej operacje, które są bardziej skomplikowane.

Należy przejść po prawie wszystkich węzłach po kolei i sprawdzić nowy współczynnik zysku informacji w wypadku, gdy jest on większy od poprzedniego (np. o więcej niż 0,1), należy przebudować poddrzewo, za które odpowiada dany węzeł (czyli korzeń danego poddrzewa), lecz z zachowaniem w innej zmiennej poprzedniego poddrzewa może się przydać w tworzeniu nowego poddrzewa.

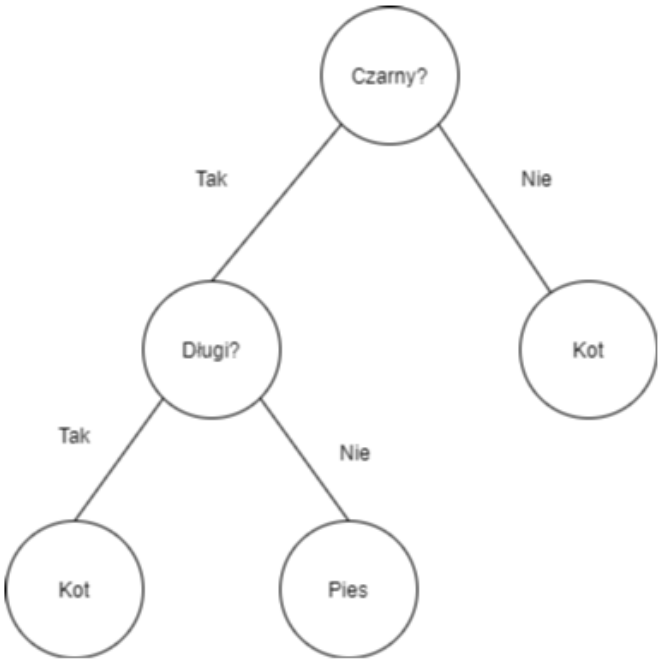
Jeżeli dla węzła zostało wybrane nowe pytanie, należy podzielić według niego dane, a dla nowo powstałych węzłów należy sprawdzić czy można wykorzystać stare poddrzewo. Może się zdarzyć, że wszystkie nowe dane trenujące „pójdą" na jedną stronę, wycięte poddrzewo (ze starego drzewa) będzie można wstawić w miejsce, dla którego nowy węzeł ma takie samo zapytanie i na takie same zbiory by podzielił dane. Jeżeli takie coś nie było by możliwe, należy zbudować od nowa całe poddrzewo pod węzłem, który uległ zmianie. Może też zajść potrzeba zamiany liścia na węzeł.

Aby nie sprawdzać każdego węzła czy można lepiej podzielić dane, sprawdzanie dla danego poddrzewa należy zakończyć, jeżeli dla jego korzenia nie zmieniło się zapytanie ani zbiór na jaki dzieli dane, w takim wypadku wszystkie nowe dane trenujące przy podziale na poprzednim węźle „poszły" do drugiego węzła i tam prawdopodobnie trzeba dokonać zmian.

**Prosty przykład**

Dane trenujące z poprzedniego przykładu

| Kolor | Długość ogona | |
|---|---|---|
| Czarny | Długi | Kot |
| Biały | Krótki | Kot |
| Czarny | Krótki | Pies |




*Rys. 2 Drzewo decyzyjne z poprzedniego punktu*

Dane doszkalające

| Kolor | Długość ogona | |
|---|---|---|
| Szary | Długi | Kot |
| Szary | Krótki | Kot |
| Szary | Krótki | Kot |
| Szary | Długi | Kot |

Zaczynamy od korzenia, liczymy nowy zysk informacji jaki możemy uzyskać po dodaniu nowych danych doszkalających dla tego węzła. Uzyskujemy znacznie większy niż poprzednio zysk informacji, więc ustalamy nowe zapytanie, którym jest czy jest szary, a stare drzewo lub poddrzewo zapisujemy do innej zmiennej.

Dane spełniające zapytanie

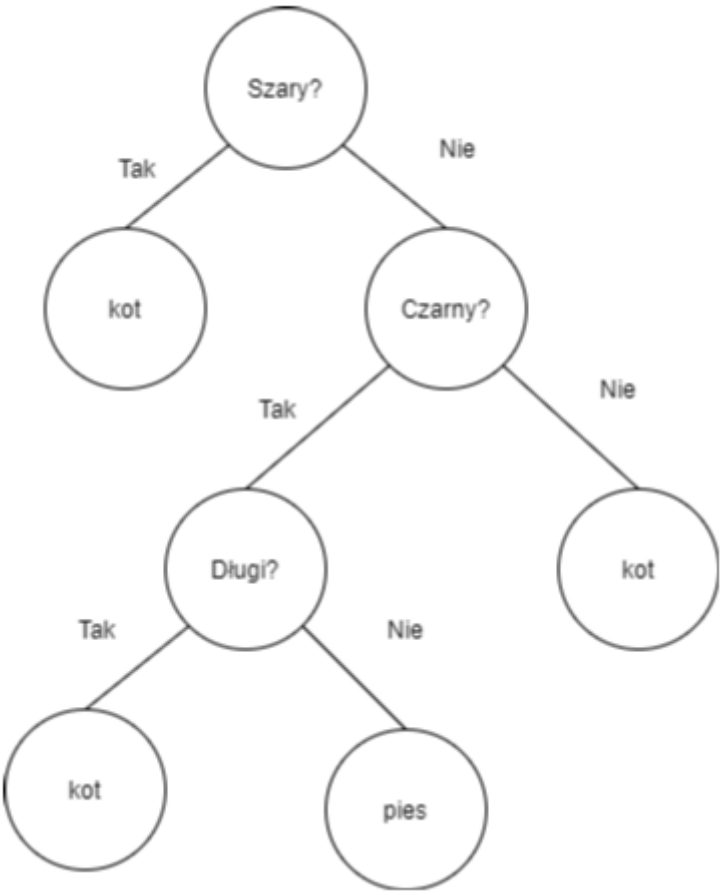
| Kolor | Długość ogona | |
|---|---|---|
| Szary | Długi | Kot |
| Szary | Krótki | Kot |
| Szary | Krótki | Kot |
| Szary | Długi | Kot |

Dane niespełniające zapytania

| Kolor | Długość ogona | |
|---|---|---|
| Czarny | Długi | Kot |
| Biały | Krótki | Kot |
| Czarny | Krótki | Pies |

Teraz jak przy budowie drzewa przechodzimy do zbioru spełniającego warunki zapytania, tutaj nasz zysk informacji jest równy zero więc kończymy operacje dla tego zbioru.

Przechodzimy do zbioru niespełniającego warunki, jak w przypadku budowania, wybieramy najlepsze zapytanie, teraz sprawdzamy czy w starym drzewie był taki węzeł z takim zapytaniem i z takim zbiorem danych okazuje się, że tak, więc wstawiamy stare drzewo (lub poddrzewo) w miejsce tego węzła i kończymy operacje.




*Rys. 3 Drzewo decyzyjne po dotrenowaniu*

Udało się uzyskać nowe drzewo *w* trzech krokach a nie w siedmiu jak by to miało miejsce w przypadku budowy od nowa drzewa.

**Plan testów**

Poniższe operacje zostaną przeprowadzone trzykrotnie dla różnych liczebności zbiorów (za pierwszym razem do trenowania podstawowego drzewa zostanie użytych 50% danych znajdujących się w bazie, do pierwszego dotrenowania 25% zbioru i do drugiego dotrenowania 25% zbioru, za drugim razem odpowiednia 25% zbioru, 25% zbioru i 50% zbioru, za trzecim razem odpowiednia 10%, 85% i 5%).

Stworzenia podstawowego drzewa:

- stworzenie zbioru danych

- podzielenie zbioru, 80% zbiór trenujący, 20% zbiór testujący

- wytrenowanie drzewa

- testowanie drzewa, sprawdzenie dopasowania zbioru testowego

- wyświetlenie wykresów ukazujących zbiór testowy, treningowy i przedziały drzewa

- skontrolowanie czy drzewo nie jest przetrenowane (np. jeżeli mieliśmy zbiór z 1000 elementów, a występują drzewa z liśćmi, z pojedynczymi elementami).

Dotrenowanie drzewa:

- stworzenie zbioru danych

- podzielenie zbioru, 80% zbiór trenujący, 20% zbiór testujący

- dotrenowanie drzewa

- testowanie drzewa, sprawdzenie dopasowania zbioru testowego

- wyświetlenie wykresów ukazujących zbiór testowy, treningowy i przedziały drzewa

- skontrolowanie czy drzewo nie jest przetrenowane (np. jeżeli mieliśmy zbiór z 1000 elementów, a występują drzewa z liśćmi z pojedynczymi elementami).

Planowane są dwa dotrenowania dla jednego drzewa podstawowego.

**Zbiór danych**

Do badania zostaną użyte dane znajdujące się na stronie:

http://archive.ics.uci.edu/ml/machine-learning-databases/00222/

w zipie bank.zip, plik o nazwie bank.csv który załączam.

Drzewo decyzyjne będzie miało za zadanie rozpoznać wykształcenie osoby na podstawie danych zgromadzonych w danej bazie.

Aby ułatwić prace z danymi zostanie wykorzystana biblioteka scikit-learn, aby podzielić zbiór na trenujący i testowy dane z dużymi wartościami przestawić w postaci unormowanej i dane słowne w postaci zero jedynkowej, niżej przykład.

Część oryginalnej tabeli

| loan |
|---|
| no |
| yes |
| no |

Wersja docelowa

| Loan_yes |
|---|
| 0 |
| 1 |
| 0 |

Część oryginalnej tabeli

| job |
|---|
| unemployed |
| services |
| management |

Wersja docelowa

| Job_ unemployed | Job_ services |
|---|---|
| 1 | 0 |
| 0 | 1 |
| 0 | 0 |

W zbiorze występują poniższe dane

1 - age (numeric)

2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services")

3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)

4 - education (categorical: "unknown","secondary","primary","tertiary")

5 - default: has credit in default? (binary: "yes","no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes","no")

8 - loan: has personal loan? (binary: "yes","no")

# related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: "unknown","telephone","cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")

12 - duration: last contact duration, in seconds (numeric)

# other attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")

Dane zostaną podzielone następująco: za pierwszym razem do trenowania podstawowego drzewa zostanie użytych 50% danych znajdujących się w bazie, do pierwszego dotrenowania 25% zbioru i do drugiego dotrenowania 25% zbioru, za drugim razem odpowiednia 25% zbioru, 25% zbioru i 50% zbioru, za trzecim razem odpowiednia 10%, 85% i 5%.