Politechnika Wrocławska

Uczenie się maszyn

Dariusz Banasiak
Katedra Informatyki Technicznej
Wydział Elektroniki

Machine Learning (uczenie maszynowe, uczenie się maszyn, systemy uczące się) – interdyscyplinarna nauka, której celem jest stworzenie systemów (programów komputerowych) automatycznie polepszających swoje działanie w miarę gromadzenia wiedzy (doświadczenia).

Początki uczenia maszynowego:

- 1952-1962 (Arthur Samuel, IBM) – program do szkolenia zawodników szachowych
- 1965 (Edward Feigenbaum, Uniwersytet Stanforda) – system ekspertowy Dendral do analizy i identyfikacji molekuł związków organicznych

Definicje uczenia się

Uczeniem się systemu jest każda autonomiczna zmiana w systemie zachodząca na podstawie doświadczeń, która prowadzi do poprawy jakości jego działania (P. Cichosz).

Uczenie się oznacza zmiany w systemie, które mają charakter adaptacyjny w tym sensie, że pozwalają systemowi wykonać za następnym razem takie same zadanie lub zadania podobne bardziej efektywnie (H. Simon).

System uczący się wykorzystuje zewnętrzne dane empiryczne w celu tworzenia i aktualizacji podstaw do udoskonalonego działania na podobnych danych w przyszłości oraz wyrażania tych podstaw w zrozumiałej i symbolicznej postaci (D. Michie).

Warunki konieczne dla procesu uczenia

- korzystna zmiana (poprawa) – nie każda zmiana w systemie oznacza uczenie się (np. zapominanie)
- autonomiczność – źródłem zmian jest sam system, a nie działanie zewnętrzne (np. ingerencja programisty)
- wpływ czynników zewnętrznych (doświadczenie, obserwacja) – zmiana w systemie dokonuje się w wyniku pewnych czynników lub okoliczności zewnętrznych (informacji związanych z funkcjami systemu)

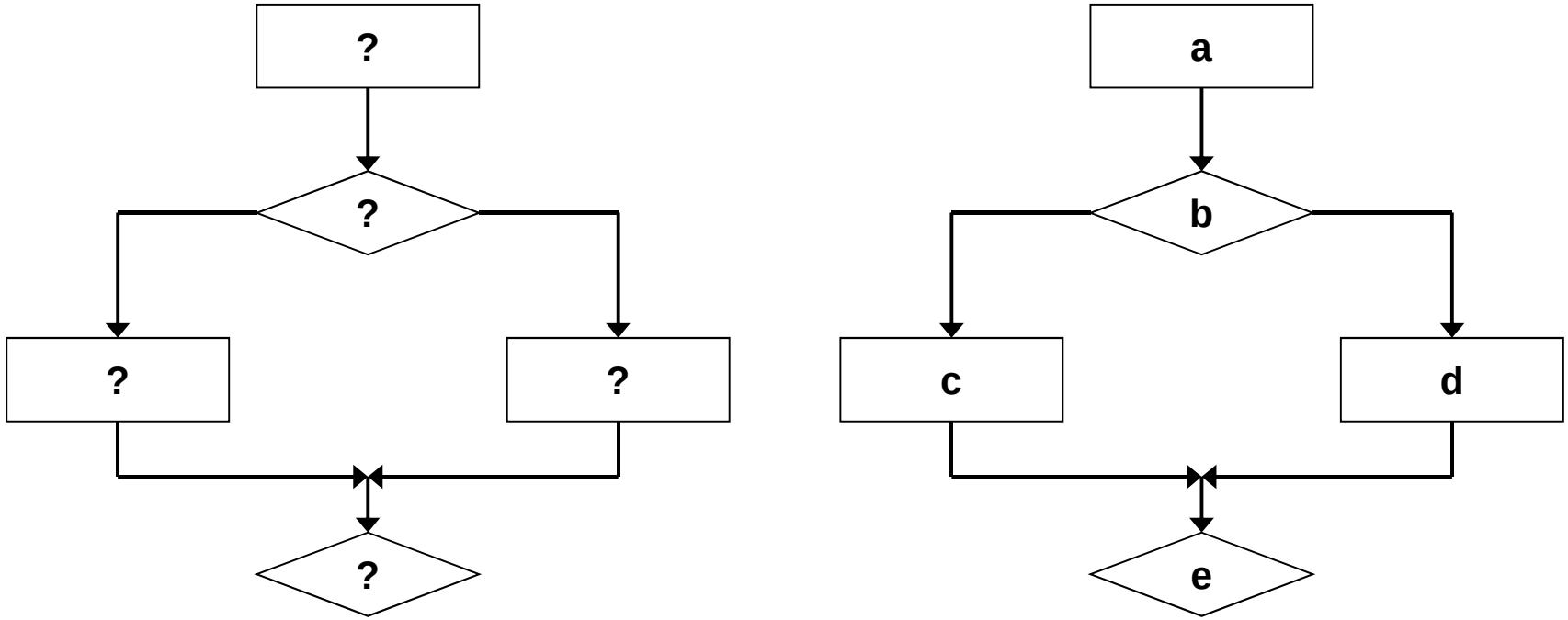
Motywacje tworzenia systemów uczących się

- złożoność problemu (uzyskanie modelu teoretycznego dla danego problemu jest bardzo kosztowne lub mało wiarygodne)
- rozmiary problemu (system analizuje dużą ilość danych, które mogą być przetwarzane tylko automatycznie np. analiza ekonomicznych lub medycznych baz danych)
- dynamiczne środowisko (system musi dostosować swoje działanie do dynamicznie zmieniających się warunków np. systemy sterowania)

Zastosowania systemów uczących się

- rozpoznawanie mowy, pisma itp. (np. SPINX)
- określanie strategii w grach (np. szachy)
- kierowanie pojazdem (np. ALVINN)
- nawigacja w nieznanym środowisku
- rozpoznawanie chorób na podstawie symptomów
- klasyfikacja obiektów astronomicznych (np. Sky Survey)
- klasyfikacja tekstów do grup tematycznych
- odkrywanie wiedzy w bazach danych (data mining)
- aproksymacja nieznanej funkcji na podstawie próbek
- wykrywanie nadużyć (bankomaty, rozmowy telefoniczne)

Uczenie się jako konkretyzacja algorytmu



Podstawowe kryteria podziału systemów uczących się

- metoda reprezentacji nabywanej wiedzy
- sposób wykorzystania wiedzy
- źródło i postać informacji trenującej
- metoda nabywania i doskonalenia wiedzy

Metody reprezentacji wiedzy w systemach uczących się

- reguły i drzewa decyzyjne
- wyrażenia logiczne (np. formuły logiki predykatów)
- parametry w równaniach algebraicznych
- automaty skończone
- grafy i sieci semantyczne

Typy zadań realizowane przez systemy uczące się

- klasyfikacja – ustalenie przynależności obiektów do określonej kategorii
- aproksymacja
- podejmowanie decyzji
- modelowanie środowiska

Rodzaje uczenia się ze względu na źródło i postać informacji trenującej

- uczenie się z nadzorem
- uczenie się bez nadzoru
- uczenie się ze wzmocnieniem

W przypadku uczenia z nadzorem system otrzymuje informację określającą w pewien sposób jego pożądane odpowiedzi dla danego zbioru wektorów wejściowych jako przykłady zachowania, jakiego się od niego oczekuje.

Przy uczeniu się bez nadzoru system nie otrzymuje informacji instruktażowej. System musi nauczyć się właściwego zachowania wyłącznie na podstawie obser-wacji wektorów wejściowych.

Uczenie się ze wzmocnieniem wykorzystuje informację oceniającą jakość działania systemu. Informacja trenująca ma charakter wartościujący (informuje system, czy działa dobrze, czy źle).

Metody (strategie) nabywania wiedzy

System uczący się transformuje otrzymane informacje (np. dostarczone przez nauczyciela) na nową postać, którą zapamiętuje. Sposób tej transformacji jest zależny od przyjętej strategii uczenia się. Główne strategie to:

- uczenie się na pamięć
- uczenie się przez indukcję
- uczenie się przez dedukcję
- uczenie się przez analogię
- uczenie się przez instrukcję

Uczenie się przez indukcję

Uczenie indukcyjne jest uczeniem się poprzez przykłady. Polega na znajdowaniu hipotez, które najlepiej opisują obserwowane fakty oraz znajdują wśród nich zależności (znajdowanie ogólnych zasad na podstawie pojedynczych przykładów). Znalezione hipotezy nie muszą sprawdzać się dla każdego przykładu, powinny jednak umożliwić analizę kolejnych przykładów (np. ich klasyfikację).

Przykłady indukcyjnego uczenia się:

- uczenie się pojęć
- tworzenie pojęć
- uczenie się aproksymacji funkcji

Drzewa decyzyjne

Drzewo decyzyjne jest strukturą złożoną z węzłów, z których wychodzą gałęzie prowadzące do innych węzłów lub liści, oraz z liści, z których nie wychodzą żadne gałęzie.

Najczęściej przyjmuje się, że każdy wierzchołek ma co najwyżej jedną krawędź wchodzącą, a liczba krawędzi wychodzących wynosi 0 (liście) lub jest większa niż 1.

W drzewie decyzyjnym węzły, gałęzie i liście posiadają dodatkowo pewną specjalną interpretację.

Interpretacja elementów drzewa decyzyjnego

- węzły – odpowiadają testom przeprowadzanym na wartościach atrybutów
- gałęzie – odpowiadają możliwym wynikom przeprowadzonych testów
- liście – odpowiadają etykietom kategorii.

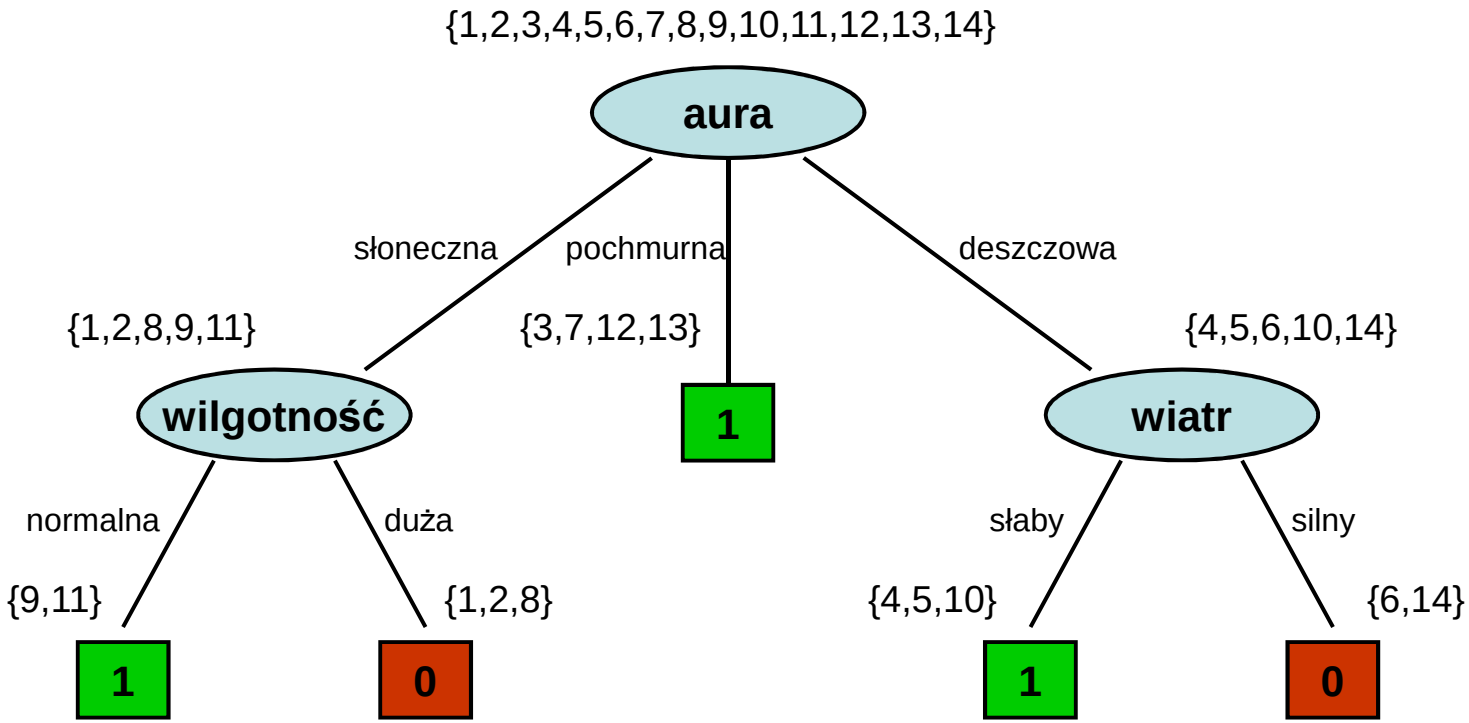
Wyznaczenie kategorii przykładu za pomocą drzewa decyzyjnego polega na przejściu od korzenia drzewa do jednego z liści, przez wykonywanie w odwiedzanych kolejno węzłach umieszczonych w nich testów i przemieszczanie się wzdłuż gałęzi odpowiadających uzyskanym wynikom.

Przykład drzewa decyzyjnego

Dany jest zbiór trenujący dla pojęcia „odpowiednia pogoda do gry w golfa”

x	aura	temperatura	wilgotność	wiatr	c(x)
1	słoneczna	ciepła	duża	słaby	0
2	słoneczna	ciepła	duża	silny	0
3	pochmurna	ciepła	duża	słaby	1
4	deszczowa	umiarkowana	duża	słaby	1
5	deszczowa	zimna	normalna	słaby	1
6	deszczowa	zimna	normalna	silny	0
7	pochmurna	zimna	normalna	silny	1
8	słoneczna	umiarkowana	duża	słaby	0
9	słoneczna	zimna	normalna	słaby	1
10	deszczowa	umiarkowana	normalna	słaby	1
11	słoneczna	umiarkowana	normalna	silny	1
12	pochmurna	umiarkowana	duża	silny	1
13	pochmurna	ciepła	normalna	słaby	1
14	deszczowa	umiarkowana	duża	silny	0

Drzewo decyzyjne dla stanów pogody



Drzewo decyzyjne jako zbiór reguł

Drzewo decyzyjne można przedstawić jako zbiór reguł postaci:

JEŻELI *warunki* TO *kategoria* (*warunki* → *kategoria*)

Każdej ścieżce w drzewie prowadzącej od korzenia do liści odpowiada równoważna reguła (*warunki* reguły to koniunkcja warunków elementarnych – testów i wyników testów, *kategoria* to etykieta związana z liściem w danej ścieżce):

aura(x)=słoneczna Λ wilgotność(x)=duża → 0
aura(x)=słoneczna Λ wilgotność(x)=normalna → 1
aura(x)=pochmurna → 1
aura(x)=deszczowa Λ wiatr(x)=silny → 0
aura(x)=deszczowa Λ wiatr(x)=słaby → 1

Konstrukcja drzewa decyzyjnego

Konstrukcję drzewa decyzyjnego można przedstawić w postaci algorytmu rekurencyjnego uruchamianego dla każdego węzła w drzewie. Dany węzeł może być:

- liściem (spełnione kryterium stopu) - koniec wywołania rekurencyjnego
- węzłem rozgałęziającym się - dokonujemy wyboru atrybutu, tworzymy rozgałęzienia węzła na podstawie wartości przyjmowanych przez dany atrybut i dla każdego węzła potomnego tworzymy rekurencyjne wywołanie algorytmu, z listą atrybutów zmniejszoną o właśnie wybrany atrybut.

Ważnym elementem każdego algorytmu jest kryterium stopu oraz metoda wyboru atrybutu.

Wprowadźmy oznaczenia:

P – zbiór przykładów dla danego pojęcia (kategorii)

S – zbiór możliwych testów (atrybutów)

d – domyślna etykieta kategorii

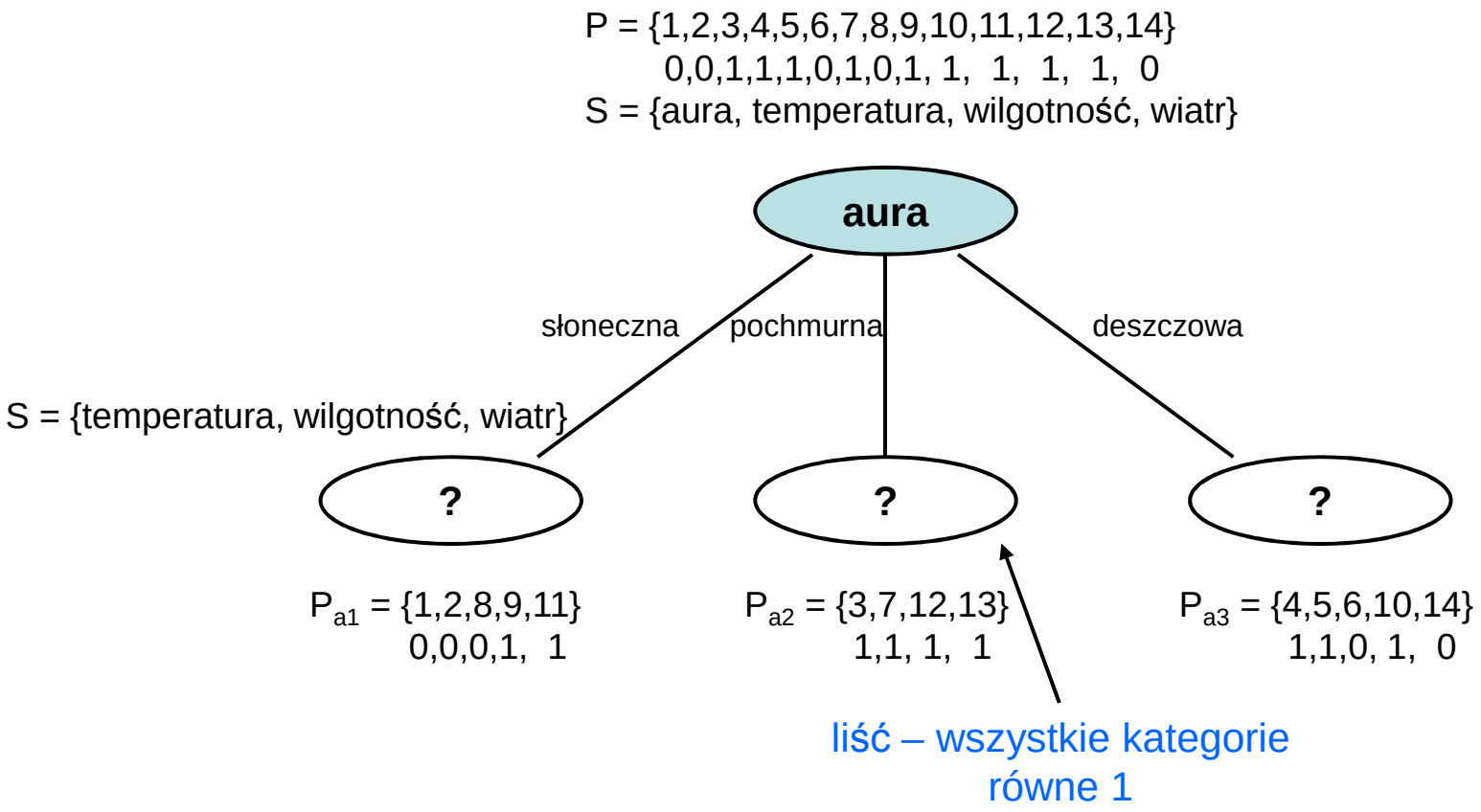
Dla naszego przykładu mamy:

P = {1,2,3,4,5,6,7,8,9,10,11,12,13,14}

S = {aura, temperatura, wilgotność, wiatr}

d = 1 (większość przykładów w zbiorze trenującym ma kategorię 1)

Działanie algorytmu:



Kryterium stopu

Kryterium stopu określa, czy dany węzeł drzewa powinien być traktowany jako końcowy liść drzewa, czy powinien być dalej rozwijany. Kryterium stopu przyjmuje najczęściej następującą postać:

- kiedy w trakcie wywołań rekurencyjnych w zestawie przykładów znajdują się już tylko przykłady opisujące jedną kategorię dany wierzchołek zostaje liściem (otrzymuje etykietę odpowiadającą tej kategorii)
- kiedy zbiór atrybutów jest pusty kryterium stopu powinno zgłosić jedną z możliwości:
  - błąd (na podstawie przykładów nie można jednoznacznie ustalić odpowiedniej kategorii np. niepoprawny zbiór przykładów zawierający przekłamania)
  - wierzchołek potraktować jako liść (przypisać mu etykietę odpowiadającą tej kategorii, która najliczniej występuje w zestawie przykładów)

Kryterium wyboru atrybutów

Najważniejszą częścią algorytmu jest odpowiednia kolejność wyboru atrybutów, która w znaczącym stopniu wpływa na wygląd drzewa (np. jego złożoność).

Wybór odpowiedniego atrybutu ze zbioru dostępnych atrybutów jest dokonywany dzięki wprowadzeniu systemu ocen (kryteria stosowane w teorii informacji i statystyce).

Wybierając jeden z atrybutów jesteśmy w stanie podzielić zbiór przykładów na mniejsze zbiory. System ocen atrybutów opiera się na założeniu, iż najbardziej korzystnym atrybutem jest taki, dzięki któremu kategorie przykładów należących do uzyskanych podzbiorów są mało zróżnicowane.

Najczęściej stosowane kryterium atrybutu pochodzi z teorii informacji - dla danego węzła należy wybrać ten atrybut, który zapewnia największy przyrost informacji.

Informację zawartą w zbiorze etykietowanych przykładów można wyrazić wzorem:

$$I(P) = \sum_{d \in C} - \frac{|P^d|}{|P|} \log \frac{|P^d|}{|P|} \text{ , gdzie}$$

C – zbiór kategorii występujących w przykładach P

|P| - ilość wszystkich przykładów

$|P^d|$ - ilość przykładów z kategorią d

Tak określona informacja jest duża, gdy liczba przykładów poszczególnych kategorii jest w zbiorze P zbliżona.

Entropię zbioru przykładów P ze względu na wartość r argumentu t określa wzór:

$$E_{tr}(P) = \sum_{d \in C} - \frac{|P_{tr}^{d}|}{|P_{tr}|} \log \frac{|P_{tr}^{d}|}{|P_{tr}|}$$, gdzie

$|P_{tr}|$ - ilość wszystkich przykładów dla których atrybut t przyjmuje wartość r

$|P_{tr}^{d}|$ - ilość przykładów dla których atrybut t przyjmuje wartość r z przypisaną kategorią d

Tak określona entropia przyjmuje dużą wartość, jeżeli wśród przykładów ze zbioru P, dla których atrybut t przyjmuje wartość r, rozkład na kategorie jest równomierny.

Entropia zbioru przykładów P ze względu na dany atrybut t jest definiowana jako średnia ważona entropia dla poszczególnych wartości tego atrybutu:

$$E_t(P) = \sum_{r \in R_t} \frac{|P_{tr}|}{|P|} E_{tr}(P) \text{ , gdzie}$$

R_t – zbiór wartości jakie przyjmuje atrybut t

Ten parametr przyjmuje duże wartości, gdy atrybut t dzieli przykłady ze zbioru P na podzbiory, w których kategorie reprezentowane są równomiernie.

Przyrost informacji wynikający z zastosowania atrybutu t dla zbioru przykładów P określa wzór:

$$g_t(P) = I(P) - E_t(P)$$

Ponieważ informacja I(P) charakteryzuje zbiór przykładów P i jest niezależna od wybranego atrybutu jako kryterium wyboru atrybutu przyjmuje się minimalizację wartości entropii $E_t(P)$ dla $t \in S$.

Przykład

$|P| = |\{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}| = 14$

$|P^0| = |\{1,2,6,8,14\}| = 5$

$|P^1| = |\{3,4,5,7,9,10,11,12,13\}| = 9$

$|P_{aura,słoneczna}| = |\{1,2,8,9,11\}| = 5$

$|P^0_{aura,słoneczna}| = |\{1,2,8\}| = 3$

$|P^1_{aura,słoneczna}| = |\{9,11\}| = 2$

$|P_{aura,pochmurna}| = |\{3,7,12,13\}| = 4$

$|P_{aura,deszczowa}| = |\{4,5,6,10,14\}| = 5$

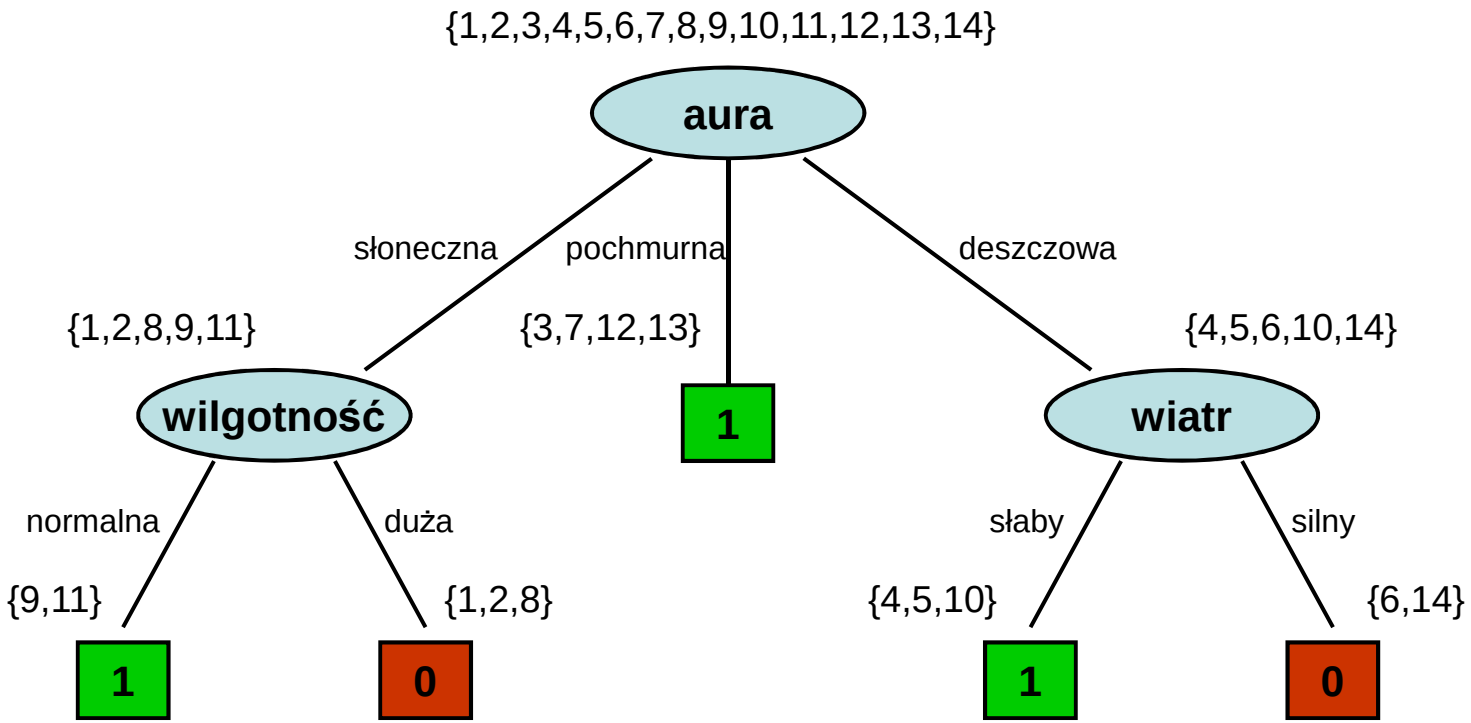
$$I(P) = - 9/14 \log_2 9/14 – 5/14 \log_2 5/14 = 0{,}940$$

$$E_{aura,słoneczna}(P) = -2/5 \log_2 2/5 – 3/5 \log_2 3/5 = 0{,}971$$
$$E_{aura,pochmurna}(P) = -4/4 \log_2 4/4 – 0/4 \log_2 0/4 = 0$$
$$E_{aura,deszczowa}(P) = -3/5 \log_2 3/5 – 2/5 \log_2 2/5 = 0{,}971$$

$$E_{aura}(P) = 5/14*0{,}971 + 4/14*0 + 5/14*0{,}971 = 0{,}694$$
$$E_{temperatura}(P) = 0{,}911$$
$$E_{wilgotność}(P) = 0{,}788$$
$$E_{wiatr}(P) = 0{,}892$$

Dla pierwszego wierzchołka wybieramy atrybut *aura*.

Powtarzając algorytm dla kolejnych wierzchołków otrzymujemy:



Zalety drzew decyzyjnych

- mogą reprezentować dowolnie złożone pojęcia pojedyncze (lub wielokrotne)
- czas decyzyjny ograniczony liniowo przez liczbę atrybutów
- forma reprezentacji czytelna dla człowieka
- łatwość przejścia od reprezentacji drzewiastej do reprezentacji regułowej

Wady drzew decyzyjnych

- na danym etapie testuje się wartość tylko jednego atrybutu (np. niepotrzebny rozrost drzewa)
- kosztowna reprezentacja alternatyw pomiędzy atrybutami
- bardzo trudna modyfikacja drzewa np. aktualizacja na podstawie nowych przykładów