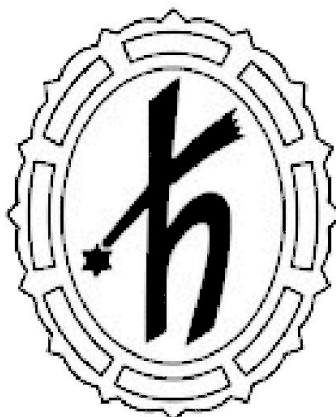


Uniwersytet Mikołaja Kopernika
Wydział Fizyki, Astronomii i Informatyki Stosowanej



Drzewa decyzyjne

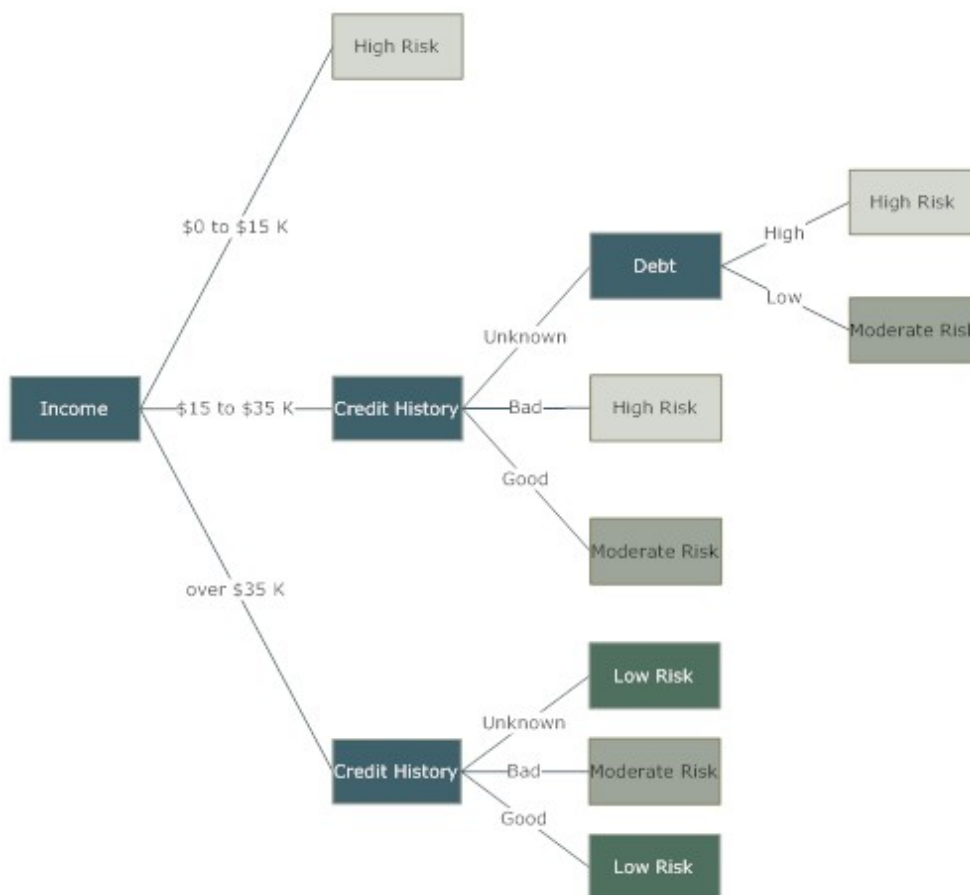
Łukasz Bujak

Toruń 2008

Wstęp

W działalności badawczej, a w szczególności w teorii decyzji, drzewa decyzyjne są narzędziami wspomagającymi podejmowanie decyzji. Kozystają one z graficznego przedstawienia decyzji i ich możliwych konsekwencji, w tym zdarzenia losowego wyników, kosztów zasobów, oraz użyteczności. Drzewa decyzyjne wykorzystuje się do określenia najbardziej prawdopodobnej strategii, tworzenia planu, do osiągnięcia celu. Drzewa decyzyjne są tak skonstruowane aby pomóc w podejmowaniu decyzji. Drzewa decyzyjne mają ustalony porządek:

- *korzeń* odpowiada wszystkim możliwym decyzjom
- każdy wewnętrzny *węzeł* odpowiada pewnej decyzji, którą możemy podjąć
- *liściom* odpowiadają cele



Rysunek 1: Przykładowe drzewo decyzji zastosowane do problemu finansowego [7]

W eksploracji danych i uczeniu maszynowym, drzewa decyzyjne są modelami prognozującymi. Drzewa decyzyjne są często stosowana metodą w data mining. Używane są w kilku przypadkach:

- redukcja ilości danych, poprzez przekształcenie ich do formy zajmującej mniej “miejsca”, przy jednoczesnym zachowaniu cech oraz zapewnieniu dokładności rozwiązania problemu
- sprawdzanie czy dane zawierają oddzielne grupy obiektów, takie że te grupy można sensownie interpretować w kontekście teorii
- wykrycie mapowania z niezależnych do zależnych zmiennych, które jest użyteczne przy przewidywaniu wartości zależnej zmiennej w przyszłości

W data mining na drzewa decyzji składają się, podobnie jak w teorii decyzji, z:

- *korzeń* zawiera wszystkie próbki treningowe
- *węzły* posiadają pojedynczą cechę lub kombinację cech z próbek treningowych
- *liście* ostatecznie uszeregowane dane z próbek treningowych wg. pewnych cech

Przykłady tworzenia drzew decyzyjnych

Przykład 1 [8]

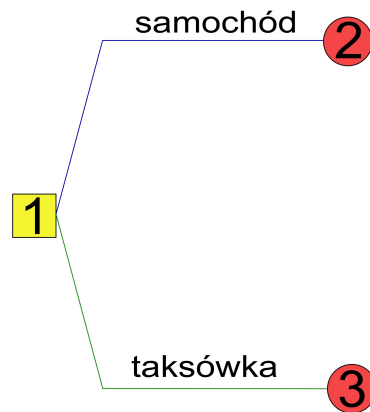
Student Leszek obudził się 25 minut przed egzaminem. Profesor był formalistą i nawet minuta spóźnienia wykluczała możliwość pisania egzaminu, Leszkowi groziła więc sesja poprawkowa. Sprawa była jednak znacznie bardziej skomplikowana, gdyż Leszek miał zamiar wyjechać do pracy do Stanów Zjednoczonych, zaś wcześniejszy powrót we wrześniu oznaczałby dla niego utratę 4 tys. zł zarobków. Jeżeli jednak zdążyłby dotrzeć na czas, to rodzice w nagrodę za dobre oceny kupiliby mu prezent (zazwyczaj wartości 500 zł).

Leszek musiał więc starannie przemyśleć, w jaki sposób dotrzeć do szkoły. Autobus nie wchodził w grę, jazda nim zajmowała co najmniej 40 minut. Pozostawała jeszcze inna możliwość - pożyczyć samochód ojca. Samochód ten jednak był w kiepskim stanie i szanse na dojechanie do szkoły wynosiły 90%, zaś w przypadku awarii Leszek musiałby pokryć część kosztów naprawy, gdyż ojciec od dawna winił go za zły stan samochodu (3 tys. zł). Oprócz tego musiał zdecydować, czy opłaca się jechać przez miasto szybko, czy wolno. Przy wolnej jeździe dotarłby na czas z prawdopodobieństwem 60 %, natomiast przy szybkiej zdążyłby na pewno (Leszek potrafił naprawdę szybko jeździć). Mógł jednak zostać zatrzymany przez patrol policji, co groziło mandatem w wysokości 500 zł (prawdopodobieństwa trafienia na patrol - 20%).

Druga możliwość to wynajęcie taksówki, za którą trzeba będzie zapłacić 30 zł. Jednak taksówkę trzeba będzie znaleźć w ciągu kilku minut, co można wykonać z prawdopodobieństwem sukcesu 80 % (jeżeli nie będzie taksówki w pobliżu, Leszek już nie zdąży). Aby ponaglić kierowcę Leszek może wręczyć napiwek w wysokości 20 zł, co zwiększy szanse na dotarcie na czas do 85 % (w przeciwnym wypadku tylko 70 %).

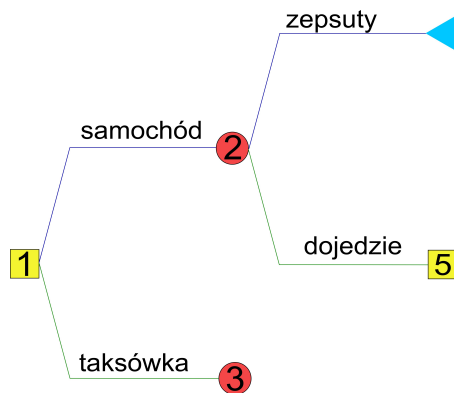
Jaką decyzję powinien podjąć Leszek?

Rozpatrzmy węzeł samochód. Mamy do czynienia z dwoma stanami natury: samochód zepsuje się lub dojedzie. Sytuacja, w której samochód zepsuje się, jest jednocześnie węzłem końcowym, gdyż Leszek nie ma już żadnego wyboru:



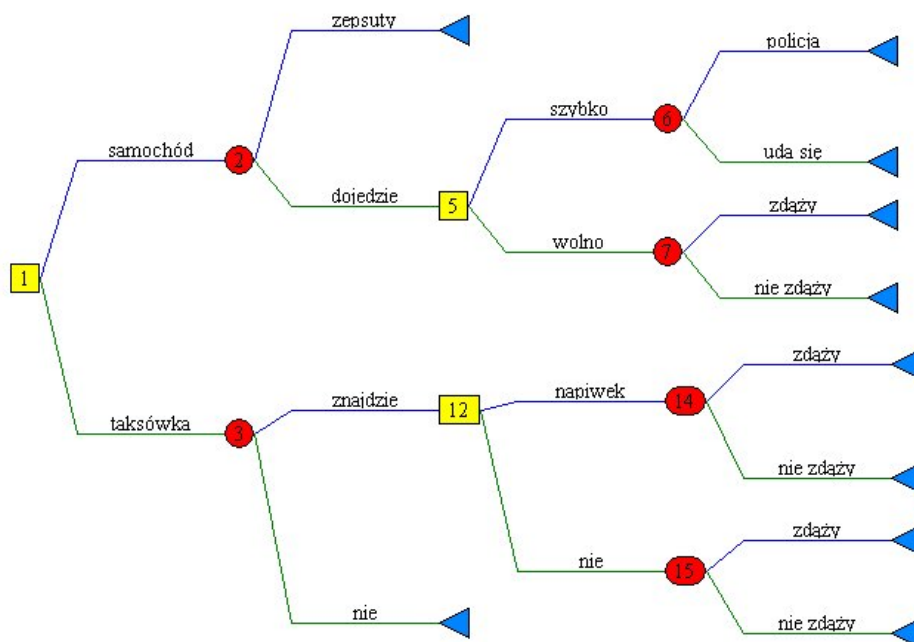
Rysunek 2: Pierwszy etap drzewa decyzji do rozwiązania problemu Leszka

Rozpatrzmy węzeł samochód. Mamy do czynienia z dwoma stanami natury: samochód zepsuje się lub dojedzie. Sytuacja, w której samochód zepsuje się, jest jednocześnie węzłem końcowym, gdyż Leszek nie ma już żadnego wyboru:



Rysunek 3: Drugi etap drzewa decyzji do rozwiązania problemu Leszka

W ten sposób kontynuujemy konstrukcję całego drzewa:



Rysunek 4: Drzewo decyzji do rozwiązania problemu Leszka

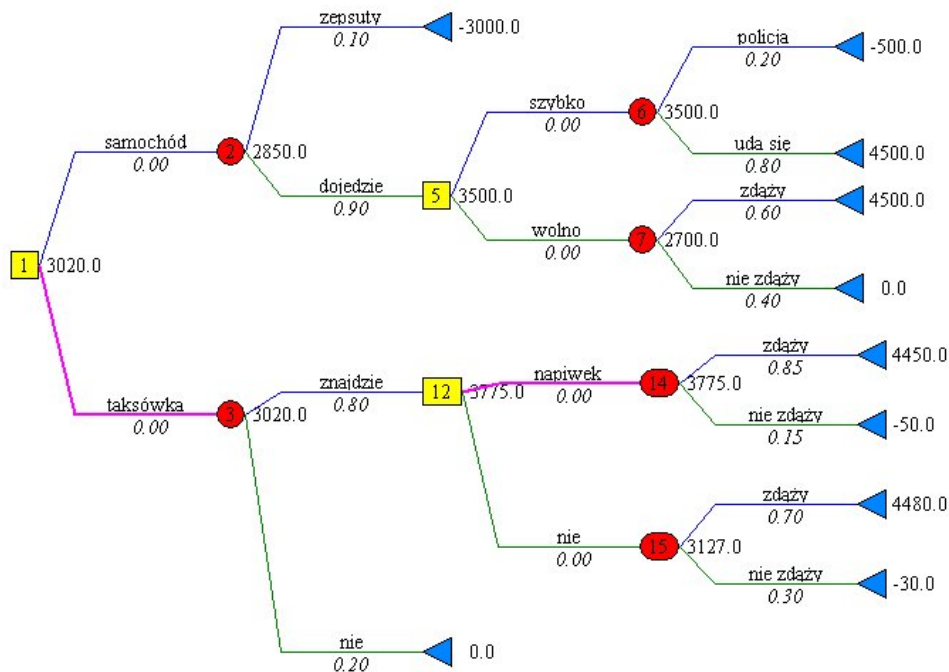
W prawidłowo skonstruowanym drzewie węzły decyzyjne i stanów natury powinny występować na przemian, zaś każda ścieżka powinna być zakończona węzłem końcowym.

Rozwiązywanie problemu przy pomocy drzewa decyzyjnego rozpoczynamy od węzłów końcowych tego drzewa, przypisując im końcowe wypłaty. Przykładowo, dla sekwencji: taksówka - znajdzie - napiwek - zdaży, końcowa wypłata wynosi 4000 zł (zarobek w Stanach) + 500 zł (prezent) - 30 zł (koszt taksówki) - 20 zł (napiwek) = 4450 zł.

Następnym krokiem jest zaznaczenie przy gałęziach wychodzących ze stanów natury odpowiadających im prawdopodobieństw. Na przykład dla węzła nr 6 prawdopodobieństwa wynoszą: policja - 0.20, uda się - 0.80.

Kolejny krok to wyznaczenie dla każdego węzła - stanu natury wartości oczekiwanej. Np. dla węzła nr 6 wartość oczekiwana wyniesie: $0.20 * (-500) \text{ zł} + 0.80 * 4500 \text{ zł} = 3500 \text{ zł}$. Przy każdym węźle decyzyjnym zapisujemy największą wartość z wyznaczonych wartości oczekiwanych (odpowiada to najkorzystniejszej decyzji).

Teraz cofając się po drzewie do korzenia, wypełniamy kolejno wszystkie węzły:



Rysunek 5: Całe drzewo decyzji do rozwiązania problemu Leszka

Optymalna ścieżka decyzji jest wyznaczona przez największe wartości oczekiwane.

W podanym przykładzie z obliczeń wynika, że Leszek powinien poszukać taksówki i dać taksówkarzowi napiwek.

Przykład 2 [9]

Dawid jest kierownikiem słynnego klubu golfowego. Niestety ma on pewne problemy. Istnieją dni kiedy każdy chce grać w golfa i wtedy jego pracownicy są przepracowani. Natomiast w inne dni, bez wyraźnej przyczyny, nikt nie chce grać w golfa i jego pracownicy mają zbyt wiele czasu wolnego. Dawid chce tak zoptymalizować dostępność personelu dla graczy. Jednak aby tego dokonać musi zrozumieć jakimi decyzjami kierują się gracze. Założył że pogoda jest ważnym czynnikiem. Także przez kolejne tygodnie zapisywał w tabeli:

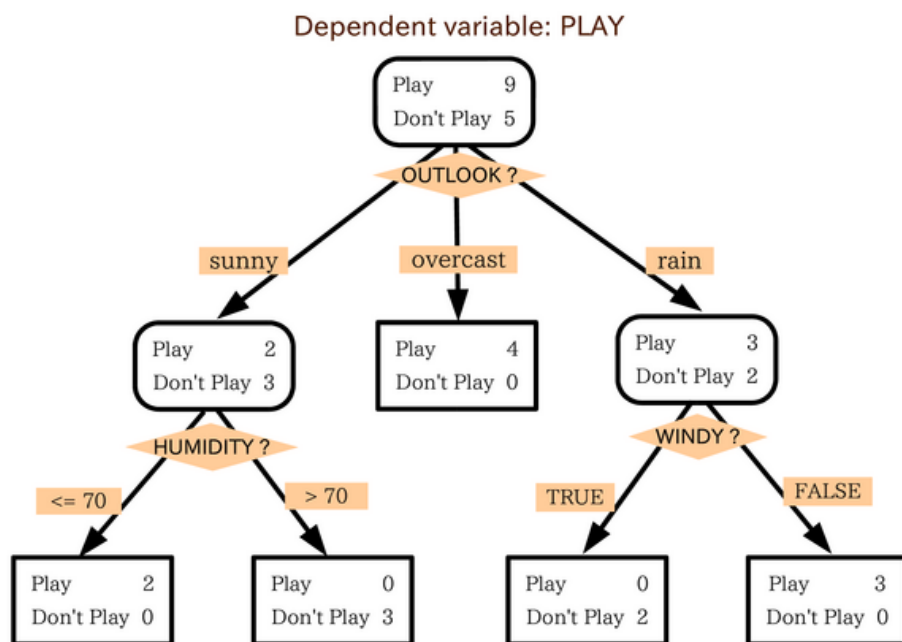
- zachmurzenie(outlook), czy było słonecznie(sunny), pochmurno(overcast), czy deszczowo(raining)
- temperaturę w stopniach Fahrenheita

- wilgotność względna(humidity) w procentach
- wietrzność(windy) tak(true) czy nie(false)
- obecność graczy w klubie (play)

Tabela 1: Wyniki obserwacji Dawida

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

Na podstawie zebranych danych Dawid sporządził drzewo decyzyjne, które pomoże mu w rozwiązaniu problemu. *Korzeń* zawiera wszystkie możliwe sytuacje, a jest ich 14(9 dni gracze grali, a 5 niegrali). Więc podzielimy je na zależne od zachmurzenia na słoneczne (5 dni spośród których 2 grali, 3 niegrali), zachmurzone (4 dni i wszystkie grali), deszczowe (5 dni, spośród których 3 grali, a 2 niegrali). Pierwszy wniosek jaki się nasuwa jest taki, że gdy niebo jest zachmurzone ludzie zawsze grają w golfa. W pozostałych przypadkach jest to już bardziej skomplikowane. Jeżeli jest słonecznie to gracze wybierają się na grę gdy wilgotność nie przekracza 70%. Natomiast gdy jest deszczowo, anacyzni gracze przychodzą grać, jednak tylko w wypadku gdy jest bezwietrznie. Teraz Dawid może podjąć decyzję w celu zoptymalizowania wykorzystania swoich pracowników. Tak więc w dni słoneczne i wilgotne, oraz deszczowe i wietrzne, Dawid może dać wolne większości swoich pracowników, ponieważ i tak prawie nikt nie będzie grać w tych dniach w golfa. Natomiast w pozostałe dni gdy wiele osób będzie grało, musi on wynająć dodatkowych pracowników.



Rysunek 6: Drzewo decyzji dla problemu Dawida

Wnioski z przykładów

Jak widać w załączonych przykładach drzewa decyzji mają wiele zalet, m.in.:

- Są proste do zrozumienia i interpretacji. Większość ludzi jest w stanie zrozumieć na czym polega model podejmowania decyzji na podstawie drzew decyzji po krótkim wyjaśnieniu.
- Nie wymagają przygotowywania danych. Inne techniki często wymagają normalizacji danych, tworzenia ślepych zmiennych czy usuwania pustych wartości.
- Drzewa są w stanie obsługiwać zarówno dane numeryczne jak i katagoryczne. Inne techniki są wyspecjalizowane w analizie zestawów danych, które mają tylko jeden typ zmiennych. Przykładowo sieci neuronowe mogą obsługiwać tylko dane numeryczne.
- Możliwość poprawienia modelu za pomocą metod statystycznych. Co powoduje niezawodność modelu.
- Są solidne, dobrze i szybko wykonują operacje na dużych ilościach danych. Także duże ilości danych mogą być szybko przetworzone na komputerach osobistych, umożliwiając zainteresowanym podjęcie decyzji.

- Używają modelu “białego pudła”, jeżeli wyjaśnimy stan w danym modelu, to jest łatwo go wytłumaczyć, gdyż możemy przejść przez całą ścieżkę decyzji. Jest to zupełnie inne podejście niż przy sztucznej sieci neuronowej która jest jak “czarne pudło” gdzie wrzucamy dane, otrzymujemy wyniki, ale nie wiemy dokładnie jak podjęte zostały decyzje.

Tworzenie i optymalizacja drzew decyzyjnych

Entropia

Entropia w ramach teorii informacji jest definiowana jako średnia ilość informacji (liczba bitów), przypadająca na znak symbolizujący zajście zdarzenia z pewnego zbioru. Zdarzenia w tym zbiorze mają przypisane prawdopodobieństwa wystąpienia.

Wzór na entropię:

$$H(x) = \sum_{i=1}^n p(i) \log \frac{1}{p(i)} = - \sum_{i=1}^n p(i) \log p(i)$$

gdzie $p(i)$ - prawdopodobieństwo zajścia zdarzenia i .

Własności entropii:

- jest nieujemna
- jest maksymalna, gdy prawdopodobieństwa zajść zdarzeń są takie same
- jest równa 0, gdy stany systemu przyjmują wartości 0 albo 1
- własność superpozycji - gdy dwa systemy są niezależne to entropia sumy systemów równa się sumie entropii.

W celu utworzenia drzewa decyzyjnego liczymy entropię dla różnych przypadków. Gdy entropia jest równa 0 to jest to koniec “konaru” drzewa decyzyjnego. I tak np. dla przykładu drugiego mamy, że średnia informacja I w danym węźle wynosi:

Outlook = sunny

$$I([2, 3]) = H\left(\frac{2}{5}, \frac{3}{5}\right) = -2/5 \log\left(\frac{2}{5}\right) - \frac{3}{5} \log\left(\frac{3}{5}\right) = 0.971 \text{ bits}$$

Outlook = overcast

$$I([4, 0]) = H(1, 0) = -1 \log(1) - 0 \log(0) = 0 \text{ bits}$$

Outlook = rainy

$$I([3, 2]) = H\left(\frac{3}{5}, \frac{2}{5}\right) = -\frac{3}{5} \log\left(\frac{3}{5}\right) - \frac{2}{5} \log\left(\frac{2}{5}\right) = 0.971 \text{ bits}$$

Wartość dla atrybutu:

$$I([3, 2], [4, 0], [3, 2]) = \left(\frac{5}{14}\right) * 0.971 + \left(\frac{4}{14}\right) * 0 + \left(\frac{5}{14}\right) * 0.971 = 0.693 \text{ bits}$$

Przyrost informacji(information gain)

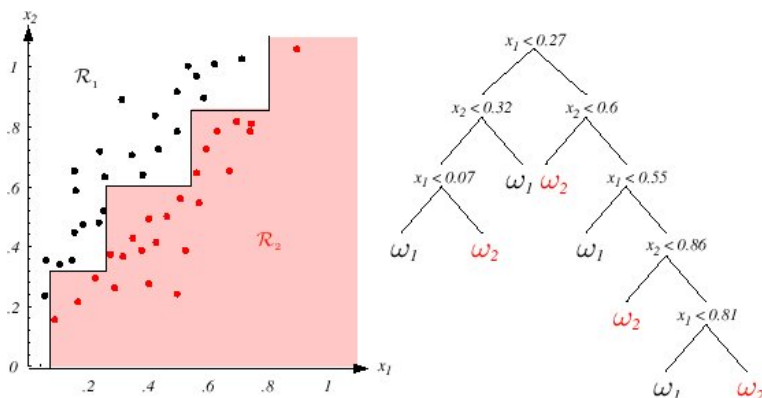
Przyrost informacji IG jest to różnica ilości informacji przed rozdzieleniem oraz po rozdzieleniu. Dla danych z przykładu 2 wygląda to następująco

$$IG(Outlook) = I([9, 5]) - I([2, 3], [4, 0], [3, 2]) = 0.940 - 0.693 = 0.247 \text{ bits}$$

Możemy budować drzewa decyzyjne również w oparciu o zysk informacyjny. Proces budowy drzewa zatrzymujemy gdy dalszy podział nie jest możliwy.

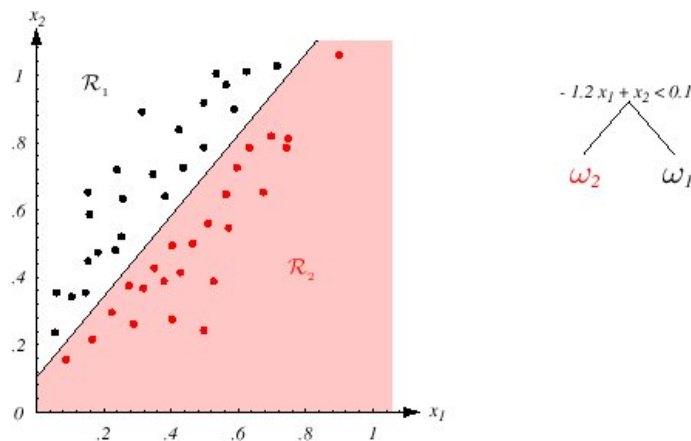
Drzewa proste i skośne

W celu podjęcia decyzji możemy korzystać z drzew decyzyjnych skośnych lub prostych.



Rysunek 7: Drzewo decyzyjne skośne [6]

Skośne drzewa decyzyjne korzystają zazwyczaj ze skomplikowanych warunków co powoduje, że są dokładne jednak nie są optymalne, gdyż podobne wyniki można otrzymać korzystając z drzew prostych. Proste drzewa są częściej wykorzystywane, gdyż z reguły są lepsze. Z powodu małej ilości prostych hipotez jest mniejsza szansa, że pasują do danych przypadkowo. Proste drzewa nie dopasowują się zbyt do danych. Dzięki prostym drzewom unikamy złej generalizacji, zbytniego przetrenowania złożonych drzew które mogą



Rysunek 8: Drzewo decyzji proste[6]

dostosować się do danych na których uczymy drzewo, natomiast nie będzie ono odpowiadać danym testowym. Stosujemy tutaj tak zwaną “brzytwę Ochmana”, która to zasada mówi że “nie należy mnożyć bytów bez potrzeby”.

Przycinanie drzew decyzyjnych

W trakcie powstawania drzewa decyzyjnego istnieje ryzyko nadmiernego dopasowania do przykładów trenujących. Drzewo decyzyjne może okazać się zbyt złożone i odzwierciedlać przypadkowe zależności występujące w zbiorze przykładów, zwłaszcza, jeśli jest ono “zaszumiony” (zawiera małą ilość błędnych danych), dlatego oprócz samej konstrukcji drzewa używa się też sposobów służących do uproszczenia drzewa w taki sposób, aby zachować jak największą spójność z zbiorem przykładów i zminimalizować błąd rzeczywisty (powstający podczas testowania obiektami nienależącymi do zbioru przykładów). Takie algorytmy noszą potocznie nazwę algorytmów przycinania.

Przycinanie jest to czynność polegająca na zastępowaniu wybranych poddrzew liśćmi. Samą implementację algorytmów przycinania drzew decyzyjnych można przeprowadzić na kilka sposobów:

- Przycinanie w trakcie wzrostu – jest to przycinanie podczas konstrukcji drzewa – taki mechanizm jest jednak mało precyzyjny gdyż podczas konstrukcji nie znamy jeszcze wyglądu całego drzewa i powoduje to stosunkowo duży błąd rzeczywisty.
- Przycinanie od środka – jest to zastępowanie węzła jednym z jego węzłów potomnych taki algorytm jest bardzo złożony i wymaga reorganizacji struktury poddrzewa jednak daje najbardziej optymalne rezultaty.

- Przycinanie proste - polega na zastąpieniu wybranego węzła węzłem zawierającym etykietę klasy-decyzji. Taki mechanizm jest stosunkowo prosty w implementacji i bardzo często stosowany. Algorytm polega na przejrzaniu całego drzewa i sprawdzeniu czy szacowany błąd rzeczywisty dla każdego z węzłów jest większy niż przedział ufności, jeśli tak to powinno nastąpić zamiana węzła rozgałęzionego na węzeł zawierający etykietę klasy-decyzji, która występuje najczęściej w przykładach dla tego węzła.

Sam sposób wyboru na odpowiedniego algorytmu modyfikacji drzewa nie wystarcza, należy jeszcze zdecydować, w jaki sposób szacowany będzie współczynnik błędu rzeczywistego:

- Przycinanie z oddzielnym zbiorem przycinania – do obliczenia współczynnika błędu używany jest oddzielny zestaw przykładów niebiorący udziału w tworzeniu drzewa. Zwykle przy dużej ilości danych przykładowych używa się 2/3 przykładów do algorytmu rekurencyjnego tworzenia drzewa, natomiast pozostała 1/3 służy do wyznaczenia błędów rzeczywistych konkretnych węzłach gotowego już drzewa decyzyjnego.
- Przycinanie na podstawie przykładów trenujących – do obliczania współczynnika błędu wykorzystywany jest ten sam zestaw przykładów. Daje to gorsze efekty jednak przy małym zbiorze przykładów takie rozwiązanie jest konieczne.

Algorytm C4.5

C4.5 jest algorytmem do tworzenia drzew decyzyjnych zaproponowanym przez Rossa Quilana. Drzewa decyzyjne wygenerowane algorytmem C4.5 mogą być wykorzystane do klasyfikacji, z tego powodu algorytm ten jest stosowany jako statystyczny klasyfikator.

Poszczególne kroki algorytmu można zapisać jako:

1. wybierz atrybut, który najbardziej różnicuje wyjściowe wartości atrybutów
2. stwórz oddzielną gałąź dla każdej wartości wybranego atrybutu
3. podziel przypadki na podgrupy, tak aby odzwierciedlały wartości w wybranym węźle
4. dla każdej podgrupy zakończ proces selekcji jeżeli:
 - wszystkie podgrupy mają taką samą wartość dla wyjściowego atrybutu

- podgrupa zawiera pojedynczy węzeł

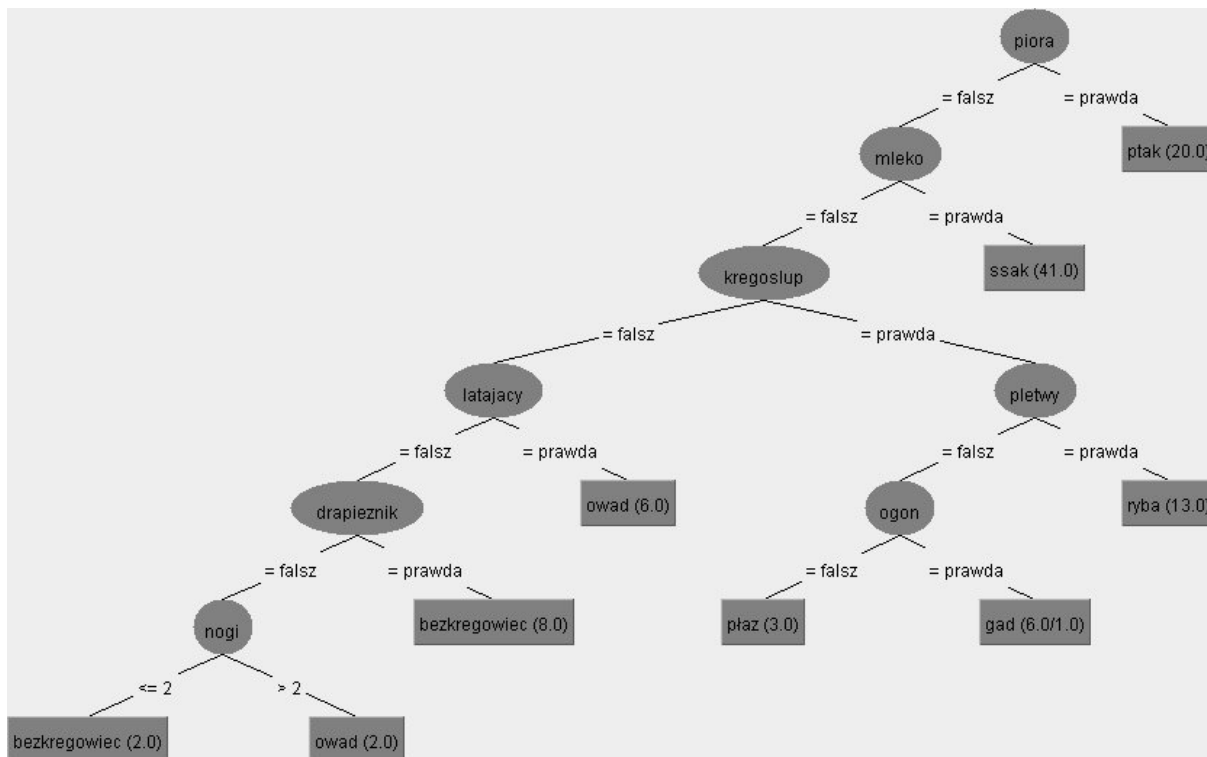
5. dla każdej podgrupy stworzonej w 3 punkcie, która nie została określona jako liść powtórz powyższy proces

Zastosowanie algorytmu do danych z ZOO

Dane zawierają informacje na temat różnych zwierząt, które są w ZOO (<http://www.let.rug.nl/tiedeman/ml05/zoo.arff>). Dla każdego zwierzęcia dodano następującą listę atrybutów: futro, pióra, jajka, mleko, latający, wodny, drapieżnik, zęby, kregosłup, breathes, jadowity, pletwy, nogi, ogon, lokalny, catsize, oraz typ (ssak, ptak, gad, ryba, płaz, owad, bezkręgowiec).

Mając tak określoną próbkę, możemy spróbować skonstruować drzewo decyzyjne, które pomoże nam w określeniu typu zwierzęcia na podstawie określonych atrybutów. Do znalezienia odpowiedniego drzewa wykorzystam open source'owy program WEKA. Jest to program do data miningu napisany w języku Java na Uniwersytecie Waikato.

Program utworzył drzewo decyzji z danych treningowych jak na rysunku (9). Drzewo jak



Rysunek 9: Drzewo decyzji dla zwierząt w ZOO

widać zostało dobrze utworzone.

Algorytm znalazł zależność, że najprościej jest rozpoznać ptaki, ponieważ tylko one posiadają pióra, natomiast z pozostałych zwierząt tylko ssaki dają mleko, itd. Tylko jedno zwierze zostało źle sklasyfikowane co daje $99,0099 \pm 2.1552$ % dobrze sklasyfikowanych zwierząt.

Niestety po wprowadzeniu danych testowych program źle rozpoznał 8 z 101 zwierząt co daje $92,0792 \pm 10.2478$ % poprawnie sklasyfikowanych zwierząt.

Confusion Matrix dla danych testowych wygląda następująco:

```

a b c d e f g |— classified as
41 0 0 0 0 0 0 — a = ssak
0 20 0 0 0 0 0 — b = ptak
0 0 3 1 0 1 0 — c = gad
0 0 0 13 0 0 0 — d = ryba
0 0 1 0 3 0 0 — e = płaz
0 0 0 0 0 5 3 — f = owad
0 0 0 0 0 2 8 — g = bezkręgowiec

```

Na podstawie macierzy pomyłek można zobaczyć dokładnie między jakimi typami zwierząt omawiany algorytm ma problemy. W 3 typach zwierząt nie było pomyłek. Mimo tych kilku źle sklasyfikowanych zwierząt można uznać, że dokładność algorytmu jest dobra.

Literatura

- [1] Cholewa W., Pedrycz W., *Systemy doradcze*. Gliwice : Politechnika Śląska, 1987
- [2] Mulawka J., *Systemy ekspertowe*. Warszawa: Wydawnictwo Naukowo-Techniczne, 1996
- [3] Cichosz P. , *Systemy uczące się*. Warszawa: Wydawnictwo Naukowo – Techniczne, 2000
- [4] Bolc L., Zaremba P., *Wprowadzenie do uczenia się maszyn*. Akademicka Oficyna Wydawnicza, 1993
- [5] Witten, I. A., Frank, E. , *Data Mining*. Morgan Kaufmann, 2000
- [6] prof. dr hab. Włodzisław Duch , *Inteligencja obliczeniowa - wykład*. 2008
- [7] <http://www.smartdraw.com>
- [8] http://pl.wikipedia.org/wiki/Drzewo_decyzyjne
- [9] http://en.wikipedia.org/wiki/Decision_tree_learning
- [10] http://en.wikipedia.org/wiki/Decision_tree