Eksploracja danych

klasyfikatory drzewiaste

Piotr Lipiński

Klasyfikacja danych

- Klasyfikacja danych
  - Zagadnienie dotyczy pewnych obiektów, o których możemy rejestrować informacje, i ich podziału na pewne klasy.
  - Każdy obiekt (rekord) opisywany jest przez d cech (atrybutów):
    $$X_1, X_2, \ldots, X_d.$$
  - Niektóre atrybuty przyjmują wartości numeryczne, niektóre przyjmują wartości nominalne. Niech $D_1, D_2, \ldots, D_d$ oznaczają zbiory wartości poszczególnych atrybutów.
  - Każdy obiekt jest więc reprezentowany przez krotkę
    $$\mathbf{x} = (x_1, x_2, \ldots, x_d) \in D_1 \times D_2 \times \ldots \times D_d .$$
  - Każdy obiekt należy do dokładnie jednej ze zbioru k klas:
    $$K = \{K_1, K_2, \ldots, K_k\}.$$
  - Problem klasyfikacji danych polega na skonstruowaniu funkcji
    $$F : D_1 \times D_2 \times \ldots \times D_d \rightarrow \{K_1, K_2, \ldots, K_k\},$$
    która na podstawie opisu obiektu określa jego klasę. Funkcja F zwana jest klasyfikatorem.

Klasyfikacja danych

- Najczęściej klasyfikator konstruuje się na podstawie pewnego zestawu danych uczących.
  - Dysponujemy zestawem danych uczących zawierającym opis pewnej liczby obiektów wraz ze wskazaniem klasy, do której każdy z tych obiektów należy.
  - Niech N oznacza liczność zestawu danych uczących. Niech $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ oznaczają opisy obiektów, zaś $c_1, c_2, \ldots, c_N$ etykiety odpowiadających im klas.
  - Dla klasyfikatora F można określić błąd klasyfikacji err(F) jako liczbę obiektów z zestawu danych uczących, które klasyfikator źle klasyfikuje, czyli takich rekordów $\mathbf{x}_i$, dla których $F(\mathbf{x}_i) \neq c_i$. (i = 1, 2, …, N).
  - Wówczas problem klasyfikacji można sformułować jako problem minimalizacji funkcji błędu klasyfikacji err(F), czyli problem wyznaczenia klasyfikatora F o minimalnym błędzie klasyfikacji spośród wszystkich klasyfikatorów.
  - W praktyce zazwyczaj rozważa się tylko pewną określoną rodzinę klasyfikatorów i spośród nich wyznacza się klasyfikator o minimalnym błędzie klasyfikacji.

Klasyfikacja danych

id	car	age	children	subscription
1	sedan	23	0	+
2	sport	31	1	-
3	sedan	36	1	-
4	truck	25	2	-
5	sport	30	0	-
6	sedan	36	0	-
7	sedan	25	0	+
8	truck	36	1	-
9	sedan	30	2	+
10	sedan	31	1	+
11	sport	25	0	-
12	sedan	45	1	+
13	sport	23	2	-
14	truck	45	0	+

Klasyfikacja danych

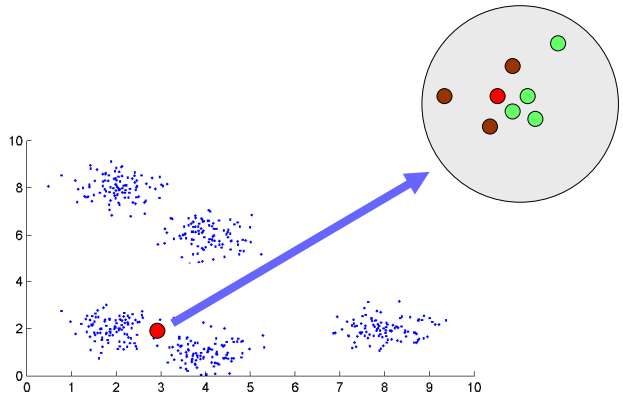
- W praktyce klasyfikator wykorzystuje się do klasyfikacji danych nieznanych, nieużywanych podczas konstruowania klasyfikatora, więc wygodnie jest sformułować problem klasyfikacji danych w języku probabilistyczno-statystycznym:
  - Niech P będzie rozkładem prawdopodobieństwa na przestrzeni $D_1 \times D_2 \times \ldots \times D_d \times K$, zaś
    $$\mathbf{x} = (x_1, x_2, \ldots, x_d, c)$$
    będzie rekordem wygenerowanym losowo z rozkładem prawdopodobieństwa P.
  - Dla klasyfikatora F można określić błąd klasyfikacji err(F) jako wartość oczekiwaną rozkładu prawdopodobieństwa
    $$P(F(x_1, x_2, \ldots, x_d) \neq c)$$
  - Celem jest oczywiście znalezienie klasyfikatora o minimalnym błędzie klasyfikacji.
  - UWAGA: Jeśli zestaw danych uczących jest dostatecznie dużą próbką losową z rozkładu prawdopodobieństwa P, to tak określony błąd klasyfikacji err(F) można estymować liczbą błędnie zaklasyfikowanych obiektów z zestawu danych uczących podzieloną przez liczność zestawu.

Klasyfikacja danych

- Przykładowe rodziny klasyfikatorów:
  - proste klasyfikatory, na przykład KNN
  - klasyfikatory drzewiaste (drzewa klasyfikacyjne)
  - klasyfikatory neuronowe (sieci neuronowe)
  - klasyfikatory ewolucyjne (konstruowane algorytmami ewolucyjnymi)
  - i wiele innych

KNN

- dla rozpatrywanego obiektu **x** wyznacza jego K najbliższych sąsiadów $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_K$ w przestrzeni cech i zwraca klasę reprezentowaną przez większość wyznaczonych sąsiadów



Drzewa klasyfikacyjne

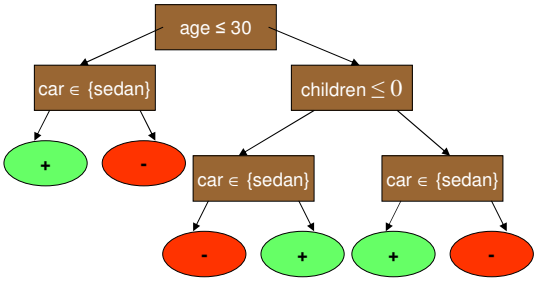
- Klasyfikator może być określony w formie drzewa klasyfikacyjnego:
  - drzewo binarne,
  - w korzeniu i węzłach wewnętrznych są umieszczone warunki postaci $X_i \in A_i$, gdzie X_i jest jednym z atrybutów opisujących obiekty, a $A_i \subset D_i$ jest pewnym podzbiorem zbioru wartości tego atrybutu,
  - w przypadku atrybutów o wartościach numerycznych, warunek $X_i \in A_i$ można zastąpić warunkiem $X_i < a_i$, gdzie a_i jest pewną ustaloną wartością numeryczną,
  - krawędzie drzewa są opisane etykietami FALSE i TRUE,
  - w liściach są umieszczone etykiety klas K_1, K_2, …, K_k,
  - dla danego rekordu, przechodząc drzewo klasyfikacyjne od korzenia do liścia uzyskuje się etykietę klasy stanowiącą wynik klasyfikacji danego rekordu.
  - UWAGA: Czasami rozpatruje się drzewa niebinarne, w których węzły z warunkami dotyczącymi atrybutów dyskretnych mają tyle potomków, ile możliwych wartości danego atrybutu dyskretnego.

Drzewa klasyfikacyjne

- Przykład:

  Dane o czytelnikach pewnego magazynu motoryzacyjnego. Konstruowane drzewo klasyfikacyjne ma określić czy czytelnik jest czy nie jest prenumeratorem magazynu.

id	car	age	children	subscription
1	sedan	23	0	+
2	sport	31	1	-
3	sedan	36	1	-
4	truck	25	2	-
5	sport	30	0	-
6	sedan	36	0	-
7	sedan	25	0	+
8	truck	36	1	-
9	sedan	30	2	+
10	sedan	31	1	+
11	sport	25	0	-
12	sedan	45	1	+
13	sport	23	2	-
14	truck	45	0	+



Konstrukcja drzewa klasyfikacyjnego

- Drzewa klasyfikacyjne konstruuje się na podstawie zestawu danych uczących różnymi metodami.

- Popularne algorytmy konstrukcji drzew klasyfikacyjnych to:
  - algorytm CART
  - algorytm ID3
  - algorytm C4.5
  - algorytm RandomForest
  - i wiele innych

- Popularne narzędzia z gotową implementacją drzew klasyfikacyjnych:
  - WEKA (http://www.cs.waikato.ac.nz/ml/weka/)
  - Microsoft SQL Server Business Intelligence

    Microsoft SQL Server Data Tools - Business Intelligence
  - Oracle Advanced Analytics
  - SAS Enterprise Miner
  - i wiele innych (w tym biblioteki do Matlab, R, Python, Java, C/C++)

Konstrukcja drzewa klasyfikacyjnego

- PROBLEM 1: Dla zestawu danych uczących – tabeli zawierającej rekordy $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$ wraz z przypisanymi etykietami klas c_1, c_2, …, c_N – skonstruować drzewo decyzyjne F o minimalnym błędzie klasyfikacji err(F).
  - Jeśli dane uczące są niesprzeczne – tabela nie zawiera dwóch identycznych rekordów $\mathbf{x}_i$ i $\mathbf{x}_j$, $\mathbf{x}_i = \mathbf{x}_j$, $i \neq j$, z różnymi etykietami klas c_i i c_j, $c_i \neq c_j$ – to zawsze można skonstruować drzewo klasyfikacyjne poprawnie klasyfikujące taki zestaw danych uczących.
  - Jak to zrobić ?
  - Jakie mogą pojawić się problemy ?
  - Często możliwe jest skonstruowanie wielu drzew poprawnie klasyfikujących zestaw danych uczących. Które z nich uznać za lepsze ? Jak oceniać drzewa decyzyjne ?

Konstrukcja drzewa klasyfikacyjnego

- PROBLEM 2: Przy konstrukcji drzewa należy pamiętać, że zestaw danych uczących to tylko próbka losowa z rozkładu prawdopodobieństwa opisującego dane do klasyfikacji. Innymi słowy – drzewo klasyfikacyjne powinno też poprawnie klasyfikować dane nieznane, nieużywane podczas konstruowania klasyfikatora, a możliwe do otrzymania w przyszłości.
  - W praktyce wyniki wszelkich pomiarów są obarczone pewnym błędem pomiarowym, wszystkie atrybuty należy więc traktować jako zmienne losowe i nie należy specjalnie przywiązywać się do ich dokładnych wartości.
  - Jak to zrobić ?
  - Jakie mogą pojawić się problemy ?
  - Poprawna klasyfikacja zestawu danych uczących przestaje być najważniejsza. Jak więc oceniać drzewa decyzyjne ?

Konstrukcja drzewa klasyfikacyjnego

- CEL1: znalezienie drzewa poprawnie klasyfikującego jak największą liczbę rekordów z zestawu danych uczących
  - nie jest to ani jedyne, ani najważniejsze kryterium,
  - dla niesprzecznych danych uczących, można znaleźć idealne drzewo poprawnie klasyfikujące wszystkie dane uczące.
- CEL2: znalezienie drzewa poprawnie klasyfikującego dane „podobne" do zestawu danych uczących
  - kryterium należy doprecyzować (co znaczy „podobne" ?).
- CEL pomocniczy: znalezienie drzewa z wysoką różnorodnością klas w swoich węzłach (wyraźnie odseparowującego klasy w swoich węzłach)
  - kryterium należy doprecyzować definiując miarę różnorodności klas.

Miary różnorodności klas

- Zestaw danych uczących to zbiór:

$$\{(\mathbf{x}_i, c_i) : i = 1, 2, \ldots, N\}$$
$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \in D_1 \times D_2 \times \ldots \times D_d$$
$$c_i \in K = \{K_1, K_2, \ldots, K_k\}.$$

- Dla każdego węzła *m* drzewa klasyfikacyjnego:
  - Niech R_m będzie podzbiorem zestawu danych uczących związanym z tym węzłem (tzn. spełniającym – lub nie – wszystkie warunki logiczne w węzłach nadrzędnych, zgodnie z etykietami krawędzi).
  - Niech N_m będzie liczbą elementów zbioru R_m.
  - Niech N_{ms} będzie liczbą elementów zbioru R_m klasy K_s.
  - Niech $p_{ms} = N_{ms} / N_m$.

Miary różnorodności klas

- Wówczas wszystkie obserwacje w węźle m przypisujemy do klasy najmocniej reprezentowanej przez zestaw danych uczących w węźle m:

$$c(m) = \arg\max_s p_{ms}.$$

- Jeśli węzeł m jest liściem, to jest to ostateczny wynik klasyfikacji. W przeciwnym przypadku, $c(m)$ ma charakter wyłącznie informacyjny i jest wykorzystywane w dalszym działaniu algorytmu (ostateczny wynik klasyfikacji obliczany jest wówczas po przejściu poddrzewa zakorzenionego w m).

- Miara różnorodności klas w węźle m:
  - powinna przyjmować wartość 0, jeśli wszystkie obserwacje w węźle m należą do tej samej klasy,
  - powinna przyjmować wartość maksymalną, jeśli wszystkie klasy w węźle są reprezentowane przez taką samą liczbę obserwacji, tzn. $p_{m1} = p_{m2} = \ldots = p_{mk} = 1 / k$).

Miary różnorodności klas

- Popularne miary różnorodności klas w węźle m drzewa F:
  - proporcja błędnych klasyfikacji

$$Q_m(F) = 1 - p_{mc(m)}$$

  - wskaźnik Giniego (używany w algorytmie CART)

$$Q_m(F) = \Sigma_{s \neq s'} p_{ms} p_{ms'} = \Sigma_{t = 1, 2, \ldots, k} p_{mt}(1 - p_{mt})$$

  - entropia (używana w algorytmie ID3)

$$Q_m(F) = - \Sigma_{t = 1, 2, \ldots, k} p_{mt} \log p_{mt}$$

Miary różnorodności klas

- W przypadku dwóch klas otrzymujemy:
  - proporcja błędnych klasyfikacji
    $$Q_m(F) = 1 - \max(p, 1 - p)$$
  - wskaźnik Giniego (używany w algorytmie CART)
    $$Q_m(F) = 2\, p\, (1 - p)$$
  - entropia (używana w algorytmie ID3)
    $$Q_m(F) = -\, p \log p - (1 - p) \log (1 - p)$$

  gdzie $p = p_{m1}$.

Miary różnorodności klas

- Niech m będzie węzłem wewnętrznym drzewa F (lub korzeniem), zaś m_L i m_R jego potomkami.
- Niech
  $$p_L = N_{mL} / N_m$$
  $$p_R = N_{mR} / N_m$$
  będzie proporcją liczby rekordów z podzbioru zestawu danych uczących związanego z węzłem m, które przeszły do węzła m_L i m_R odpowiednio.
- Łączna miara różnorodności klas w potomkach węzła m to:
  $$Q_{mL,\, mR}(F) = p_L Q_{mL}(F) + p_R Q_{mR}(F).$$
- Różnica różnorodności klas w węźle m i jego potomkach to:
  $$\Delta Q_{m,\, mL,\, mR}(F) = Q_m(F) - Q_{mL,\, mR}(F).$$

Konstrukcja drzewa decyzyjnego

- IDEA ALGORYTMU:
  - drzewo klasyfikacyjne można konstruować zachłannie:
  - w każdym węźle m, poczynając od korzenia, staramy się znaleźć odpowiedni podział podzbioru zestawu danych uczących przez wyznaczenia warunku logicznego prowadzącego do odpowiednich węzłów potomnych m_L i m_R
  - sprowadza się to do maksymalizacji funkcji $\Delta Q_{m, mL, mR}(F)$ w każdym węźle m (dla danego m i nieznanych m_L i m_R).

Konstrukcja drzewa decyzyjnego

- ALGORYTM:
  - Stwórz korzeń drzewa F.
    - Oznaczmy go przez m.
    - R_m jest oczywiście całym zestawem danych uczących.
  - Znajdź taki warunek logiczny dla węzła m, który prowadzi do węzłów potomnych m_L i m_R oraz podziału zbioru R_m na zbiory R_{mL} i R_{mR} maksymalizującego funkcję $\Delta Q_{m, mL, mR}(F)$.
  - Jeśli znalezione rozwiązanie nie poprawia różnorodności klas (tzn. $\Delta Q_{m, mL, mR}(F) <= 0$), to przerwij tę procedurę (węzeł m zostanie liściem drzewa F).
  - Jeśli znalezione rozwiązanie poprawia różnorodność klas, to dołącz węzły m_L i m_R do drzewa F oraz
    - uruchom taką samą procedurę dla m_L,
    - uruchom taką samą procedurę dla m_R.

Konstrukcja drzewa decyzyjnego

- Pozostaje ustalić jak znaleźć odpowiedni warunek logiczny podziału węzła m na węzły potomne m_L i m_R.
  - Ograniczamy się do warunków dotyczących tylko jednego atrybutu (tzn. warunków postaci $X_i \in A_i$, gdzie X_i jest jednym z atrybutów, a $A_i \subset D_i$ jest pewnym podzbiorem zbioru wartości tego atrybutu).
  - Dla każdego atrybutu X_i ($i = 1, 2, \ldots, d$):
    - niech $V(X_i)$ oznacza zbiór zarejestrowanych wartości atrybutu X_i w całym zestawie danych uczących (nawet dla atrybutów ciągłych ten zbiór jest skończony i $|V(X_i)| \leq N$),
    - sprawdźmy wszystkie podzbiory A zbioru $V(X_i)$ i warunki postaci $X_i \in A$ (do sprawdzenia jest więc $2^{|V(Xi)|}$ warunków),
    - razem do sprawdzenia jest więc $2^{|V(X1)|} + 2^{|V(X2)|} + \ldots + 2^{|V(Xd)|}$ warunków
    - można zauważyć, że dla ciągłych atrybutów X_i wystarczy sprawdzać warunki postaci $X_i \leq a$ zamiast $X_i \in A$, co redukuje liczbę warunków z $2^{|V(Xi)|}$ do $|V(X_i)|$.

Konstrukcja drzewa decyzyjnego

- Przykład: Rozważmy korzeń drzewa m i warunek AGE $\leq$ a:
  - w zestawie danych uczących atrybut AGE przyjmuje wartości: 23, 25, 30, 31, 36 i 45, nie ma sensu rozpatrywać innych wartości a, bo nie występują one w danych uczących,
  - $N_m = 14$, $p_{m1} = 6/14$, $p_{m2} = 8/14$
  - $Q_m(F) = 2 * 6/14 * 8/14 = 0.4898$
  - dla a = 23:
    $N_{mL} = 2$, $N_{mR} = 12$
    $Q_{mL}(F) = 2 * 1/2 * 1/2 = 0.5000$
    $Q_{mR}(F) = 2 * 5/12 * 7/12 = 0.4861$
    $Q_{mL, mR}(F) = 2/14 * 0.5000 + 12/14 * 0.4861 = 0.4881$
  - dla a = 25:
  - …

(w przykładzie używamy wskaźnika Giniego)

id	car	age	children	subscription
1	sedan	23	0	+
2	sport	31	1	-
3	sedan	36	1	-
4	truck	25	2	-
5	sport	30	0	-
6	sedan	36	0	-
7	sedan	25	0	+
8	truck	36	1	-
9	sedan	30	2	+
10	sedan	31	1	+
11	sport	25	0	-
12	sedan	45	1	+
13	sport	23	2	-
14	truck	45	0	+

Algorytm C4.5

- Przedstawione podejście jest podstawą kilku algorytmów tworzenia drzew klasyfikacyjnych:
  - CART (indeks Giniego)
  - ID3 (entropia)
- Algorytm C4.5 jest rozszerzeniem algorytmu ID3:
  - dopuszcza nieokreślone wartości niektórych atrybutów w niektórych rekordach danych (wyliczenia oparte są jedynie na tych rekordach, które mają określone wartości wymaganych atrybutów)
  - dopuszcza ciągłe wartości niektórych atrybutów (oryginalny algorytm ID3 zakłada, że wszystkie atrybuty mają wartości dyskretne)
  - przycina utworzone drzewo (przechodząc od liści do korzenia, tzn. z dołu do góry drzewa, próbuje zastąpić poddrzewo zaczepione w danym węźle liściem, jeśli nie zwiększa to błędu klasyfikacji o więcej niż ustalony próg)

Przycinanie drzewa decyzyjnego

- Proste metody
  - uznanie za liść węzła *m* jeśli $|R_m| < 5$
  - ograniczenie głębokości drzewa
  - ograniczenie liczby węzłów drzewa
- Metoda poddrzew zakorzenionych
  - Rozpoczynamy od skonstruowanego drzewa T_0 o $|T_0|$ liściach.
  - Wyznaczamy drzewo T_1 o $|T_0|$-1 liściach, zakorzenione w T_0, o minimalnej liczbie błędów klasyfikacji.
  - Wyznaczamy drzewo T_2 o $|T_0|$-2 liściach, zakorzenione w T_0, o minimalnej liczbie błędów klasyfikacji, itd.
  - Z drzew T_0, T_1, T_2, … wybieramy drzewo o minimalnej liczbie błędów klasyfikacji (można tutaj użyć innego zestawu danych niż przy konstrukcji drzewa T_0).
- Metoda kosztu-złożoności:
  - Niech R(F) oznacza niedoskonałość drzewa F, na przykład liczbę błędów klasyfikacji.
  - Szukamy drzewa F zakorzenionego w drzewie F_0, dla którego wartość minimalną osiąga kryterium kosztu-złożoności:

$$R_a(F) = R(F) + a\,|F|$$

gdzie a jest ustaloną stałą, zwaną współczynnikiem złożoności.

Algorytm Tree Bagging

- Tree Bagging to algorytm typu bagging agregujący wiele drzew klasyfikacyjnych w jeden klasyfikator.
- IDEA ALGORYTMU:
  - z zestawu danych uczących losujemy ze zwracaniem N rekordów tworząc nowy zestaw danych uczących
  - używając nowego zestawu danych uczących tworzymy drzewo za pomocą ustalonego algorytmu konstrukcji drzewa, na przykład CART,
  - powyższą procedurę powtarzamy *n* razy, gdzie *n* jest wielkością tworzonego lasu
- Klasyfikacja odbywa się następująco:
  - każdy rekord jest klasyfikowany przez każde z *n* drzew klasyfikacyjnych
  - wynikiem klasyfikatora jest etykieta klasy zwrócona przez największą liczbę drzew klasyfikacyjnych

Algorytm RandomForest

- RandomForest to algorytm typu bagging, podobny do algorytmów bootstrapowych, agregujący wiele drzew klasyfikacyjnych w jeden klasyfikator.
- IDEA ALGORYTMU:
  - z zestawu danych uczących losujemy ze zwracaniem N rekordów tworząc nowy zestaw danych uczących
  - używając nowego zestawu danych uczących tworzymy drzewo losowe:
    - w każdym węźle wybieramy losowo atrybut, atrybuty wybrane w węzłach nadrzędnych nie mogą być losowane
    - wykorzystywany jest indeks Giniego
  - powyższą procedurę powtarzamy *n* razy, gdzie *n* jest wielkością tworzonego lasu
- Klasyfikacja odbywa się następująco:
  - każdy rekord jest klasyfikowany przez każde z *n* drzew klasyfikacyjnych
  - wynikiem RandomForest jest etykieta klasy zwrócona przez największą liczbę drzew klasyfikacyjnych

Metody oceny klasyfikatora

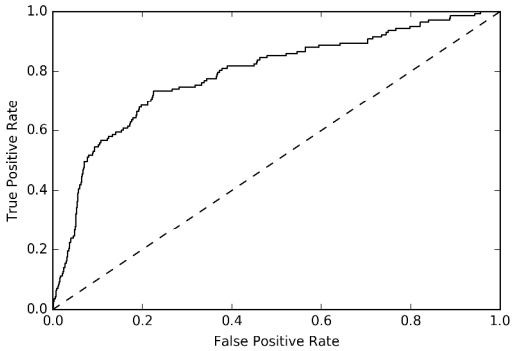
- Confusion Matrix

		zaklasyfikowany jako	
		pozytywny	negatywny
w rzeczywistości	pozytywny	**TP**	**FN**
	negatywny	**FP**	**TN**

- Accuracy = (TP + TN) / (TP + FN + FP + TN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 = 2 Precision Recall / (Precision + Recall)
- Fb = $(1+b^2)$ / b^2 Precision Recall / (Precision + Recall)

Metody oceny klasyfikatora

- Receiver Operating Characteristic (ROC)
  - Sensitivity = TPR = TP / (TP + FN) = Recall
  - Specificity = TNR = TN / (TN + FP)
  - Fall Out = FPR = 1 – TNR
  - ROC to wykres TPR vs. FPR (dla różnych parametrów algorytmu/klasyfikatora)
- Interpretacja:
  - szara przerywana linia oznacza klasyfikator losowy (przekątna)
  - klasyfikator idealny to lewy górny róg (punkt (0, 1))



Drzewa regresyjne

- Atrybut zależny jest numeryczny. Najczęściej oznaczamy go symbolem Y.
- Większość definicji jest analogiczna do przypadku drzew decyzyjnych. Zamiast określenia klasyfikator używamy określenia reguła predykcji.
- Błędem predykcji dla reguły predykcji d nazywamy błąd średniokwadratowy

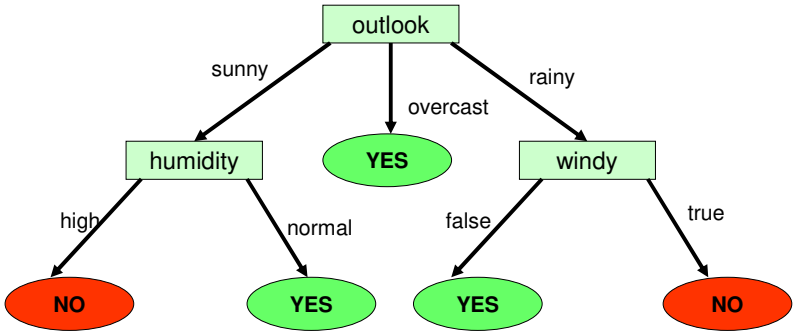
$$R_d = E(d(t.X_1, t.X_2, \ldots, t.X_m) - t.Y)^2$$

- Celem jest znalezienie reguły predykcji o minimalnym błędzie predykcji.
- Drzewo regresyjne definiuje się analogicznie jak drzewo decyzyjne.

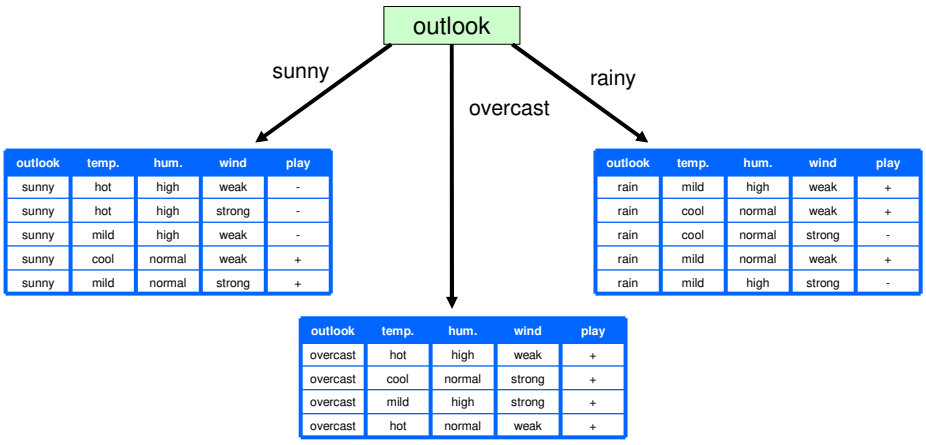
Konstrukcja drzewa klasyfikacyjnego z ID3

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	-
2	sunny	hot	high	strong	-
3	overcast	hot	high	weak	+
4	rain	mild	high	weak	+
5	rain	cool	normal	weak	+
6	rain	cool	normal	strong	-
7	overcast	cool	normal	strong	+
8	sunny	mild	high	weak	-
9	sunny	cool	normal	weak	+
10	rain	mild	normal	weak	+
11	sunny	mild	normal	strong	+
12	overcast	mild	high	strong	+
13	overcast	hot	normal	weak	+
14	rain	mild	high	strong	-

Konstrukcja drzewa klasyfikacyjnego z ID3



Konstrukcja drzewa klasyfikacyjnego z ID3

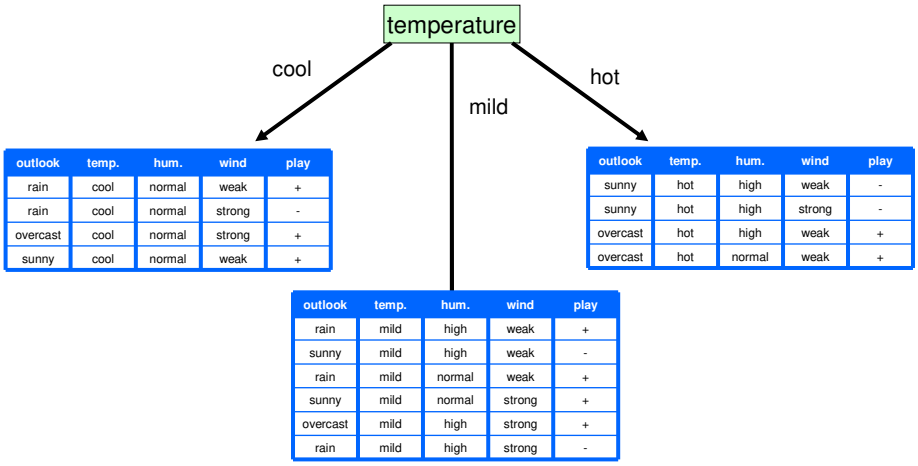


outlook	temp.	hum.	wind	play
sunny	hot	high	weak	-
sunny	hot	high	strong	-
sunny	mild	high	weak	-
sunny	cool	normal	weak	+
sunny	mild	normal	strong	+

outlook	temp.	hum.	wind	play
rain	mild	high	weak	+
rain	cool	normal	weak	+
rain	cool	normal	strong	-
rain	mild	normal	weak	+
rain	mild	high	strong	-

outlook	temp.	hum.	wind	play
overcast	hot	high	weak	+
overcast	cool	normal	strong	+
overcast	mild	high	strong	+
overcast	hot	normal	weak	+

Konstrukcja drzewa klasyfikacyjnego z ID3



outlook	temp.	hum.	wind	play
rain	cool	normal	weak	+
rain	cool	normal	strong	-
overcast	cool	normal	strong	+
sunny	cool	normal	weak	+

outlook	temp.	hum.	wind	play
sunny	hot	high	weak	-
sunny	hot	high	strong	-
overcast	hot	high	weak	+
overcast	hot	normal	weak	+

outlook	temp.	hum.	wind	play
rain	mild	high	weak	+
sunny	mild	high	weak	-
rain	mild	normal	weak	+
sunny	mild	normal	strong	+
overcast	mild	high	strong	+
rain	mild	high	strong	-

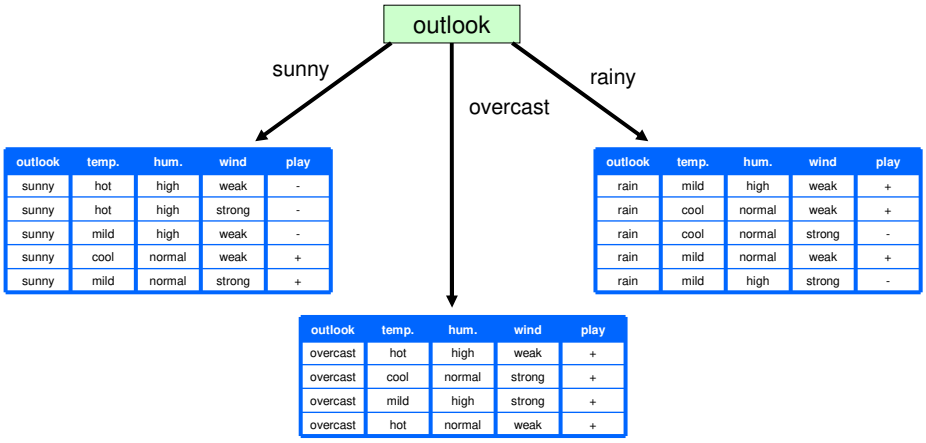
Konstrukcja drzewa klasyfikacyjnego z ID3

- Rozpatrzmy węzeł *m* drzewa klasyfikującego F.
- Niech $p = N_m^+ / N_m$, gdzie N_m^+ to liczba pozytywnych rekordów (klasy K_1) w podzbiorze zestawu danych uczących związanym z węzłem *m*, zaś to N_m liczba wszystkich rekordów związanych z węzłem *m*.
- Entropia dla węzła *m* wynosi

$$Q_m(F) = - p \log p - (1 - p) \log (1 - p).$$

- Entropia mierzy pewnego rodzaju zaburzenie w węźle *m*.

Konstrukcja drzewa klasyfikacyjnego z ID3

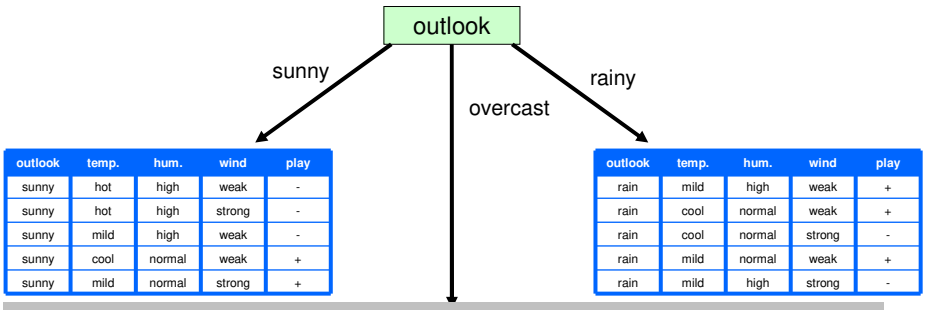


outlook	temp.	hum.	wind	play
sunny	hot	high	weak	-
sunny	hot	high	strong	-
sunny	mild	high	weak	-
sunny	cool	normal	weak	+
sunny	mild	normal	strong	+

outlook	temp.	hum.	wind	play
rain	mild	high	weak	+
rain	cool	normal	weak	+
rain	cool	normal	strong	-
rain	mild	normal	weak	+
rain	mild	high	strong	-

outlook	temp.	hum.	wind	play
overcast	hot	high	weak	+
overcast	cool	normal	strong	+
overcast	mild	high	strong	+
overcast	hot	normal	weak	+

Konstrukcja drzewa klasyfikacyjnego z ID3



outlook	temp.	hum.	wind	play
sunny	hot	high	weak	-
sunny	hot	high	strong	-
sunny	mild	high	weak	-
sunny	cool	normal	weak	+
sunny	mild	normal	strong	+

outlook	temp.	hum.	wind	play
rain	mild	high	weak	+
rain	cool	normal	weak	+
rain	cool	normal	strong	-
rain	mild	normal	weak	+
rain	mild	high	strong	-

Który atrybut sprawdzać tutaj?

Gain (m_{sunny} , *Humidity*) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970

Gain (m_{sunny} , *Temperature*) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570

Gain (m_{sunny} , *Wind*) = .970 - (2/5) 1.0 - (3/5) .918 = .019