



Zaawansowana analityka z SAS Enterprise Miner

2018/2019 – edycja 5

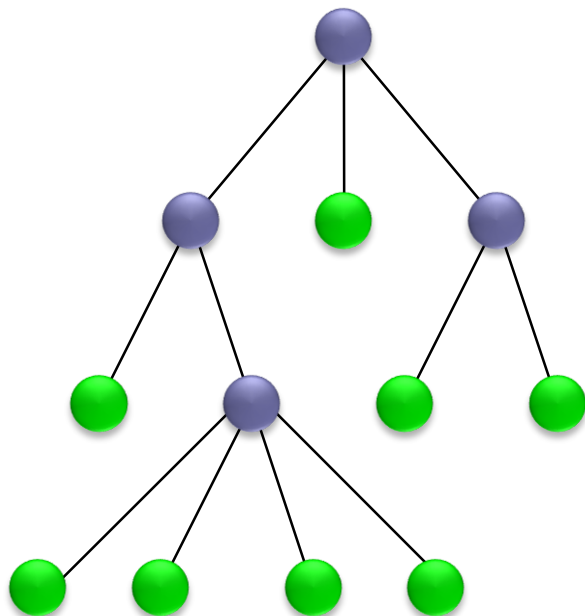
Wykład - materiały dodatkowe



MODELE: DRZEWA DECYZYJNE

Drzewa decyzyjne - Idea

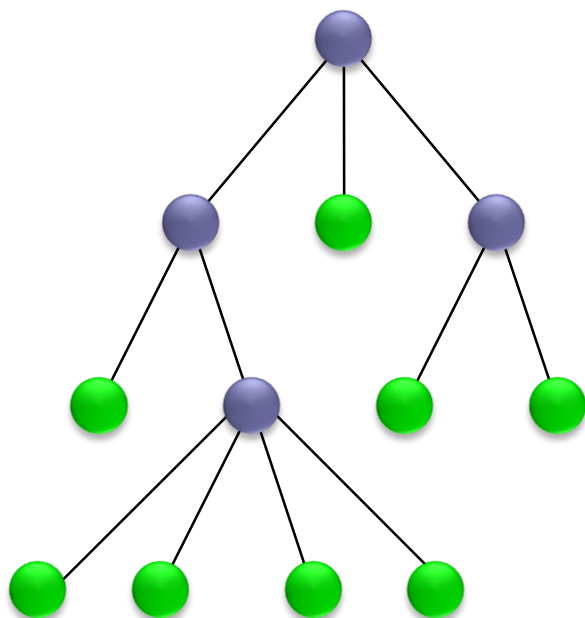
- Drzewo decyzyjne jest zbiorem reguł przedstawionych za pomocą struktury drzewiastej



- Łatwo zrozumieć i przedstawić sposób, w jaki dokonywana jest predykcja (wystarczy spojrzeć na drzewo)
- Łatwe do budowy i wizualizacji
- Najczęściej od drzew zaczyna się proces modelowania

Drzewa decyzyjne - Idea

- Drzewo decyzyjne służy do klasyfikacji obiektów / zdarzeń o znanych atrybutach do jednej ze skończonej liczby klas



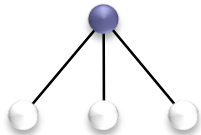
- Budowa drzewa polega na hierarchicznym podziale wielowymiarowej przestrzeni cech na rozłączne podzbiory
- Kryterium podziału są warunki nakładane na zmienne (warunki w węzłach drzewa)
- Na najniższym poziomie znajdują się liście
- Element od którego rozpoczyna się budowa drzewa to korzeń
- Drzewo na ilustracji obok posiada 3 poziomy (korzeń to poziom 0)

- Konstrukcja

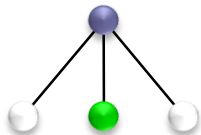
Budowa drzewa rozpoczyna się od elementu głównego, który reprezentuje wszystkie analizowane przypadki. Następnie proces jest powtarzany dla każdego otrzymanego liścia, który może stać się węzłem



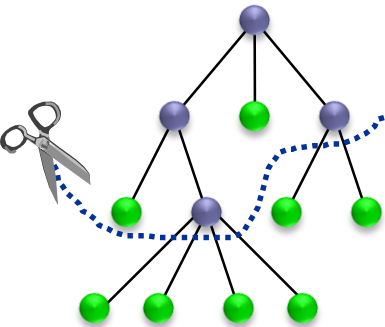
Spośród wszystkich możliwych podziałów biorąc pod uwagę wszystkie zmienne wybieramy te najlepsze.



Każdy podział generuje kolejne liście (węzły)

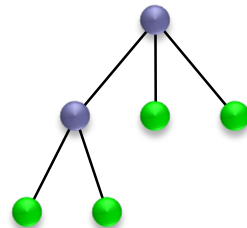


Jeżeli liść składa się tylko z jednorodnych elementów albo wystąpiła inna przyczyna nie pozwalająca na dalsze rozrastanie się drzewa proces dla wybranego liścia zostaje zatrzymany.



Drzewo jest gotowe, gdy nie ma już więcej możliwości podziału.

Zwykle przy tego typu algorytmie drzewo jest przetrenowane. Dlatego na koniec procesu szkolenia drzewo jest przycinane w oparciu o dane ze zbioru walidacyjnego.



- Prawdopodobieństwo w liściach



Po zbudowaniu drzewa każdy liść zawiera proporcję wystąpień modelowanego zjawiska.

Prawdopodobieństwo, że zmienna celu jest równa poszukiwanej wartości określamy poprzez gęstość danego liścia

W podanym przykładzie jest ono równe 0.965 dla wartości zmiennej celu = *Nie*



Podział węzła jest niejako dwuetapowy

- Wybór najlepszego podziału dla każdej zmiennej objaśniającej
- Wybór najlepszej zmiennej wraz z podziałem

Porównanie jakości podziałów dla nominalnych zmiennych celu

- Porównanie podziałów dla tej samej zmiennej
 - χ^2 , ΔGini , $\Delta\text{Entropia}$ – częściej wybierają podział z większą liczbą gałęzi
 - P-value, LOGWORTH (zmienne celu nominalne: $\text{LOGWORTH} = -\log(\chi^2 \text{ p-value})$; ciągłe zmienne celu: $\text{LOGWORTH} = -\log(\text{F test p-value})$) – nakładają karę za zbyt dużą liczbę gałęzi – częściej wybierają podziały binarne
- Porównanie podziałów między zmiennymi jest obciążone:
 - Im większa liczba podziałów, tym większe wartości LOGWORTH
 - Dla zmiennych binarnych, gdzie możliwy jest tylko 1 podział, LOGWORTH będzie niższy
 - Liczba potencjalnych podziałów dla zmiennych nominalnych jest wyższa niż dla zmiennych porządkowych, dlatego uzyskują one wyższe wartości LOGWORTH
- Korekta obciążenia:
 - Information Gain Ratio (ΔGini /liczba poziomów zmiennej w węźle rodzicu)
 - Bonferroni adjustment (korekta dla p-value)



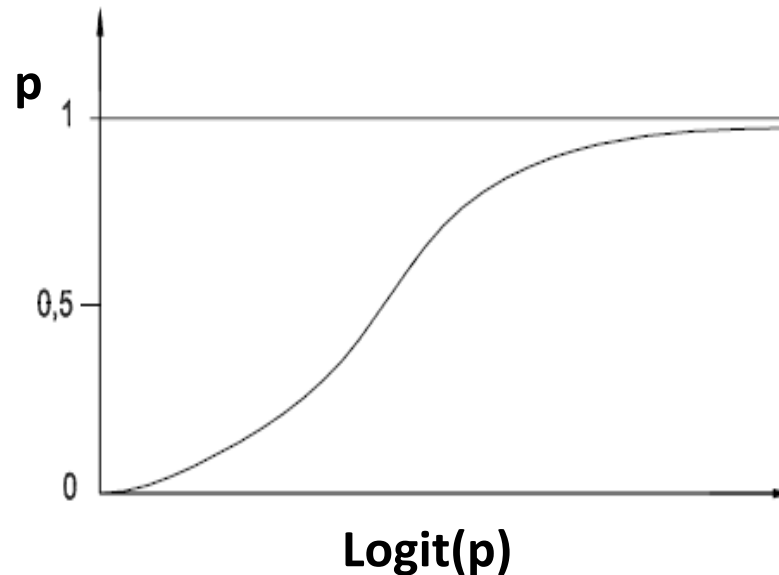
Porównanie jakości podziałów dla ciągłych zmiennych celu

- Różnica w stosunku do nominalnych zmiennych cel - kryteria:
 - Redukcja wariancji
 - P-value dla testu F
- Porównanie podziałów:
 - Obliczenie wariancji zmiennej celu w każdym z węzłów
 - Cel: różnorodność między węzłami oraz między węzłami a rodzicem



MODELE: REGRESJA

- Funkcja logitowa przyjmuje wartości ciągłe z zakresu $(-\infty, \infty)$, dlatego może być ona przybliżona przez regresję liniową. Wartość $\text{Logit}(p)$ staje się więc zmienną celu dla modelu liniowego.



Powstały model jest więc postaci: $\text{Logit}(p) = a \cdot x_1 + b \cdot x_2 + \dots + f \cdot x_n + C$

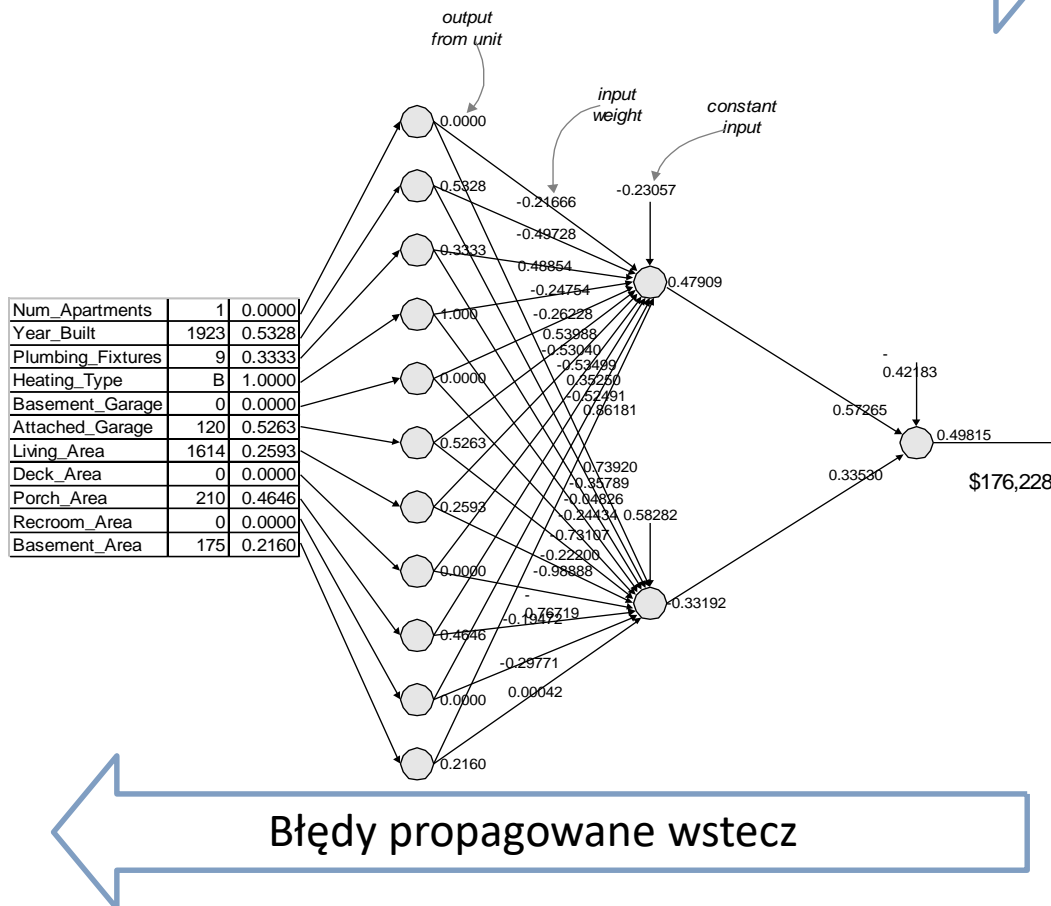


MODELE: PODSTAWY SIECI NEURONOWYCH



UCZENIE SIECI NEURONOWEJ

Wartości propagowane do przodu



Błędy propagowane wstecz

- Sieci podawane są przykłady poprawnego działania, które powinna ona potem naśladować w swoim bieżącym działaniu
- Celem szkolenia sieci jest minimalizacja błędu liczonego jako suma kwadratów różnicy pomiędzy wartościami rzeczywistymi a oczekiwanymi
- Odpowiednie algorytmy (Quasi-Newton, metoda najszybszego spadku gradientu) w każdym kolejnym kroku dopasowują wagi w kierunku minimalizacji błędu



PROCES SCORINGU


- Węzły SAS Enterprise Miner generują kody scoringowe
- Kody scoringowe pochodzące z jednej ścieżki modelowania mogą zostać połączone dzięki zastosowaniu węzła SCORE
- Istnieją 2 typy kodów scoringowych
 - Data step
 - Procedure
- Wygenerowany w ten sposób kod może zostać zastosowany dla źródła danych znajdującego się poza SAS Enterprise Miner



Algorytm k-średnich

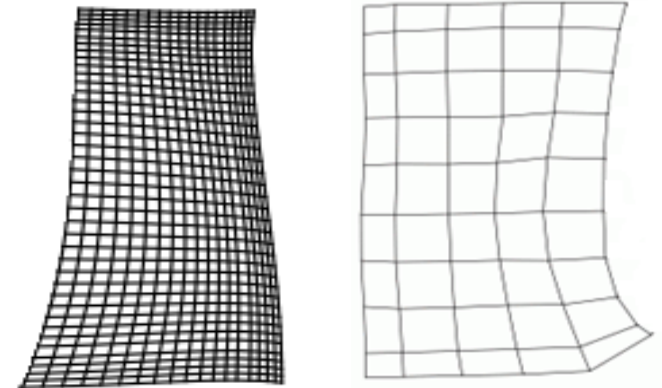
METODY NIEHIERARCHICZNE

Sieci samoorganizujące - SOM (ang. *Self-Organising Maps*)

- Sieci samoorganizujące się powstały jako inspiracja sposobu neurologicznego odwzorowania w mózgu różnorodnych bodźców odbieranych przez ludzi, gdzie odległość bądź inna relacja między bodźcami odnosi się do odległości między neuronami.
 - Neurony zorganizowane są w siatkę, która jest odizolowana od przestrzeni wejściowej.
 - SOM próbują znaleźć takie klastry, które będąc blisko siebie w przestrzeni neuronowej (siatce), znajdują się blisko w przestrzeni wejściowej.
 - Zależność w drugą stronę nie musi być spełniona
- 
- The diagram shows a 3D model of a brain with a section of the somatosensory cortex highlighted in orange. Labels with leader lines point to 'Leg' (pointing to the upper limb area), 'Toes' (pointing to the lower limb area), and 'Somatosensory cortex' (pointing to the highlighted orange area).



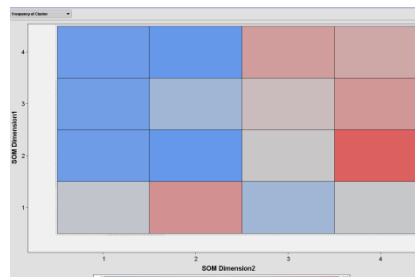
- Inny sposób wyobrażenia idei sieci samoorganizujących
 - Sieci samoorganizujące próbują zanurzyć siatkę w przestrzeń wejściową w taki sposób, aby każdy przykład był możliwie blisko ziarna przy minimalnym odkształceniu siatki (ściskaniu, rozciąganiu)
 - Sieci samoorganizujące są wygładzonym odwzorowaniem regionów z przestrzeni wejściowej w punkty z siatki



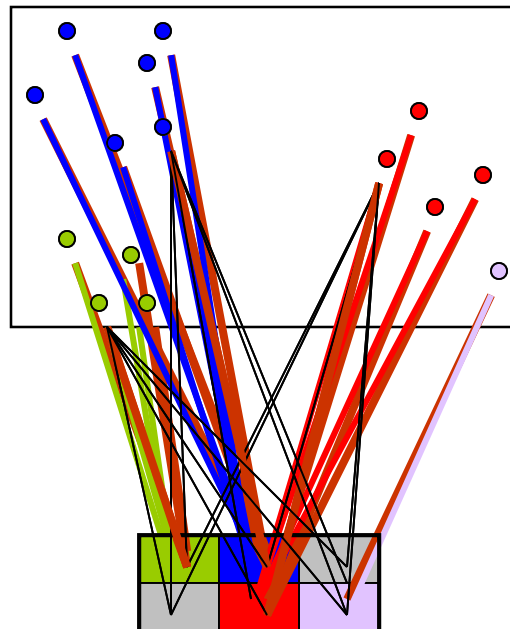
Algorytm sieci SOM

na przykładzie badania skupień wśród klientów banku

- Na początku określa się rozmiary mapy (w wierszach i kolumnach) i losowo inicjalizuje wzorce w neuronach
- Algorytm generuje segmenty w następujący sposób:
 1. Obliczane jest podobieństwo klienta do poszczególnych wzorców w komórkach
 2. Następuje przypisanie klienta do najbliższego segmentu (na podstawie odległości od wzorca)
 3. Przeliczenie wartości wzorca dla zwycięskiego segmentu (i jego sąsiadów)
 4. Procedura powtarzana kolejno dla wszystkich klientów



Ilustracja działania sieci SOM



SOM/Kohonen - dostępne metody

- Kwantowanie wektorowe Kohonena
(ang. *Kohonen Vector Quantization*)
 - Zgodnie z Kohonena prawem uczenia dla każdego przykładu znajdujący się najbliższy klaster (ziarno, wzorzec). Następnie jest on przesuwany w stronę przykładu zgodnie ze współczynnikiem uczenia (ang. *learning rate*) według wzoru:

$$C_n^{s+1} = C_n^s (1 - L^s) + X_i L^s$$

gdzie:

- C_n^s - ziarno dla klastra n w kroku s
- X_i - wektor wejściowy dla i -tego przykładu
- L^s - współczynnik uczenia w kroku s

dla pozostałych ziaren: $C_n^{s+1} = C_n^s$

Współczynnik uczenia jest zmniejszany w trakcie uczenia, aby uzyskać zbieżność

- Sieci samoroganizujące Kohonena
(ang. *Kohonen SOM*)

- Kohonena algorytm dla sieci samoorganizujących wymaga wykorzystania funkcji jądrowej $K^S(j,n)$, gdzie
 - $K^S(j,j)=1$
 - $K^S(j,n)$ jest zazwyczaj nierosnącą funkcją odległości między ziarnami j i n w przestrzeni siatki
 - Zazwyczaj $K^S(j,n)=0$, jeśli ziarna są daleko od siebie w przestrzeni siatki
- Dla każdego przykładu trenującego wszystkie ziarna uaktualniane są według wzoru:

$$C_n^{s+1} = C_n^s (1 - K^s(j,n)L^s) + X_j K^s(j,n)L^s$$

- Wygładzanie odbywa się w przestrzeni siatki
- Sąsiedztwo funkcji jądrowej jest zmniejszane w trakcie uczenia
- Im bardziej zmniejszane jest sąsiedztwo, tym bardziej metoda zbliża się do kwantowania wektorowego Kohonena. Dla wartości sąsiedztwa równej 0 staje się kwantowaniem wektorowym Kohonena
- Współczynnik uczenia jest także zmniejszany w trakcie procesu uczenia

- Wsadowe sieci samoroganizujące
(ang. *Batch SOM*)
 - Algorytm wykorzystuje nieparametryczną regresję z użyciem funkcji jądrowej $K^S(j,n)$
 - Po zainicjowaniu wzorców wykonywane są następujące dwa kroki
 - Przypisanie każdego przykładu trenującego do najbliższego wzorca (wg miary euklidesowej). Rejestrowane są średnie dla klastrów i liczba przykładów do nich należących
 - Wykonanie nieparametrycznej regresji z wykorzystaniem funkcji jądrowej $K^S(j,n)$. Punkty z siatki to dane wejściowe, średnie klastrów to zmienna celu, liczba przykładów stanowi wagę. Następnie każdy wzorzec zastępowany jest wartością z regresji nieparametrycznej wyliczoną w punkcie siatki odpowiadającej wzorcowi
 - Kroki powtarzane są do momentu uzyskania zbieżności bądź przekroczenia maksymalnej liczby iteracji
 - Brak współczynnika uczenia
 - Dostępne są następujące typy wygładzania:
 - wygładzanie Nadaraya-Watsona (ang. *Nadaraya-Watson Smoothing*)
 - lokalnie liniowe wygładzanie (ang. *Local-Linear Smoothing*)

