

Predyktywna Sztuczna Inteligencja

Uczenie maszynowe I



Włodzisław Duch

Katedra Informatyki Stosowanej UMK

Google: Włodzisław Duch

Co było



- Teorie poznania
- Systemy oparte na wiedzy
- Modele kognitywne
- SOAR
- ACT
- Cog
- Watson
- Cyc

Co będzie



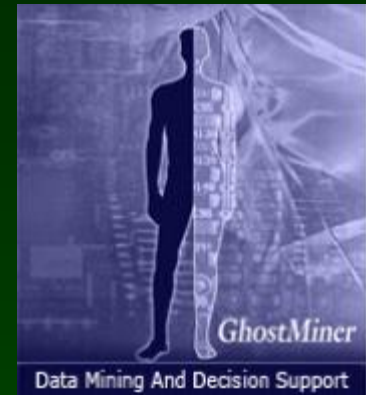
- Uczenie maszyn
- Metody indukcyjne
- Metody oparte na podobieństwie
- Drzewa decyzyjne
- Algorytmy genetyczne
- Sieci neuronowe
- Deep learning

Wykłady z inteligencji obliczeniowej omawiają uczenie maszynowe dokładniej.

Systemy ekspertowe

- Systemy ekspertowe (ES): inżynieria wiedzy, systemy oparte na wiedzy (KBS).
- Program = algorytm + struktury danych.
- ES = interfejs użytkownika + wiedza + system wnioskujący, sterowany zdarzeniami, a nie kolejnością instrukcji. To większa elastyczność rozwiązań.
- ES wykorzystują wiedzę i procedury wnioskowania do rozwiązywania problemów, które wymagają znaczącej ekspertyzy specjalistów.
- Wymagają wiedzy, reprezentacji w postaci reguł, sieci semantycznych, ram.
- Wiedza może być w sieci neuronowej, ale preferowane jest zrozumiałe AI. Uczenie maszynowe może służyć akwizycji wiedzy dla SE.
- Wiedza jest niezbędna, by zapewnić odpowiedni poziom ekspertyzy, wraz z procedurami wnioskowania stanowi model ekspertyzy.
- [SAP LeanIX](#), lider Enterprise Architecture Tools (Gartner Magic Quadrant).
- [AI and Expert Systems](#): Powering the Future of Business.

Business intelligence



- Google: ghostminer, projekt zrealizowany w naszej grupie w latach 1998-2004, służył do data mining, wspomagania decyzji poprzez analitykę danych, uczenie maszynowe, KDD, business intelligence.
- GhostMiner sprzedawany był przez Fujitsu jako narzędzie do analitycznej eksploracji danych, włączony przez FQS PL do różnych narzędzi FinTech.
- Klienci: uniwersytety, politechniki, instytuty badawcze, banki i firmy w Polsce, Austrii, Australii, Chinach, Czechach, Holandii, Indiach, Japonii, Kanadzie, Niemczech, Norwegii, Singapurze, Wielkiej Brytanii i USA.
- Np. w Abbott Laboratories używano GhostMiner do odkrywania właściwości wielowymiarowych danych naukowych.
- Intemi (Intelligent Miner), nasz program nowej generacji, oparty na ideach meta-uczenia, powinien zostawić konkurencję daleko w tyle ...

Systemy kognitywne



- IBM Watson, projekt Q/A rozpoczęty w 2007 roku, określany jako “masowo równoległa architektura probabilistyczna oparta na dowodach”.
- Test: w 2011 roku wygrywa z dwoma mistrzami teleturnieju Jeopardy (Va Banque). Odpowiedzi wymagały głębokiego wnioskowania. Użyto około 100 różnych technik do akwizycji wiedzy, NLP, wnioskowania, analizy informacji z Wikipedii, innych encyklopedii, książek, gazet.
- W 2014 IBM utworzył [Watson Group](#) i ogłosił erę “kognitywnych obliczeń”, od zabawy do zastosowań medycznych i AI dla biznesu, ale postępy nie były imponujące. Duże wymagania obliczeniowe. Brak analizy mowy czy obrazu.
- 2023 [WatsonX](#), połączenie ES z generatywną AI, asystenci do analityki BI, obsługi klienta, rozwoju oprogramowania i wspomaganie zarządzania.
- [CyC Corp](#) (od 1984!), autonomiczne rozumowanie maszynowe podobne do ludzkiego, ogromna baza reguł, zastosowania w szpitalach i cyberbezpieczeństwa.



Przyszłość AI



Frankly, I don't see any reason to set limits on what AI can do.

We have in our heads a wonderful computer. It is made of meat, but it's a computer. It's extremely noisy, but it does parallel processing. It is extraordinarily efficient, but there is no magic there.

So, it is difficult to imagine that, with sufficient data in the future, there will remain things that only humans can do.

You should replace humans by algorithms whenever possible.

Daniel Khaneman (Nobel 2002).

Motywacja: dane => wiedza



- Systemy AI były „karmione” wiedzą zamiast autonomicznego uczenia się. Ale pamięć komputerów wzrosła, technologia baz danych i czujników stworzyła problem eksplozji danych.
- Repozytoria informacji, hurtownie danych przechowują wszelkiego rodzaju dane, Internet stał się źródłem danych o zachowaniu ludzi.
- Pojawiło się hasło: Tonjemy w danych, ale pragniemy wiedzy!
- W latach 1990-2000: eksploracja danych i hurtownie danych, multimedialne bazy danych i internetowe bazy danych zapoczątkowały ruch „big data”.
- Wykorzystano przetwarzanie analityczne on-line (OLAP) i algorytmy uczenia maszynowego do eksploracji danych (data mining), szukając reguł, prawidłowości, wzorców, ograniczeń, korelacji, podsumowań w dużych bazach danych.
- Data scientists w 2018 roku byli najbardziej poszukiwanymi pracownikami!

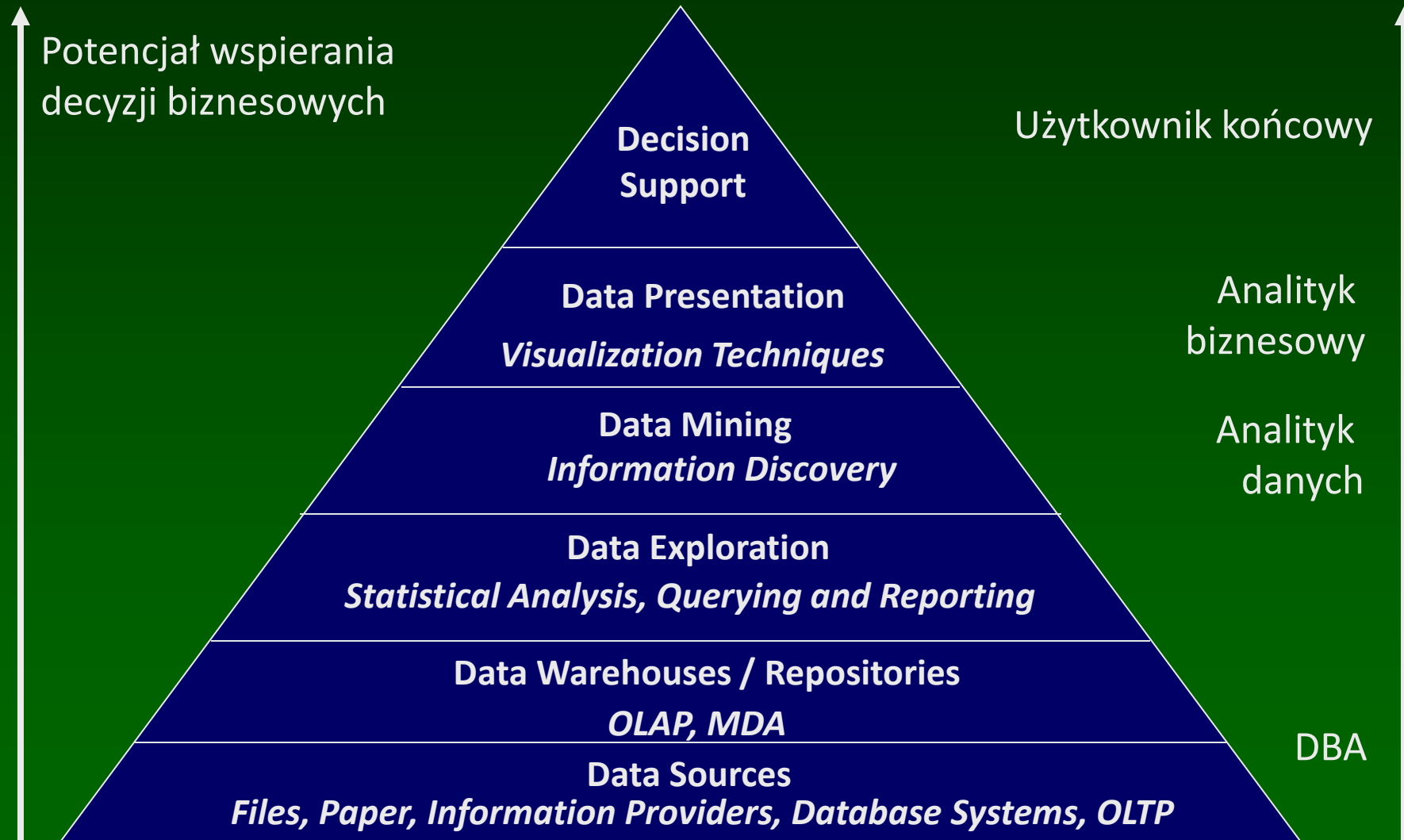
Uczenie maszynowe

- Jeśli problem jest dostatecznie precyzyjnie zdefiniowany może napisać program, który jest implementacją algorytmu. Dla większości oprogramowania nie było potrzeby uczenia się, ale wiele problemów nie da się precyzyjnie opisać.
- Wielkie zainteresowanie ML nastąpiło z powodu dostępności dużych danych, w których starano się zauważyć jakieś regularności, stąd KDD, Knowledge Discovery in Data, data mining, a następnie dzięki dobrym wynikom analizy obrazów w dekadzie 2010-20.
- Uczenie maszynowe umożliwia mechanizację akwizycji wiedzy. To część sztucznej inteligencji, symbolicznej AI lub obliczeniowej.
- To alternatywa dla tworzenia reguł na podstawie analizy zachowań ekspertów lub samodzielnego odkrywania wiedzy.
- ML czerpie inspiracje z ogólnych metod informatyki, takich dziedzin jak data mining, statystyki, rozpoznawania struktur (pattern recognition), sieci neuronowych i kognitywistyki.
- ML tworzy modele zmieniające swoje wewnętrzne parametry tak, by rozpoznać ogólne cechy danych umożliwiając odkrycie ich znaczenia – kategorii, numerycznej oceny wartości.

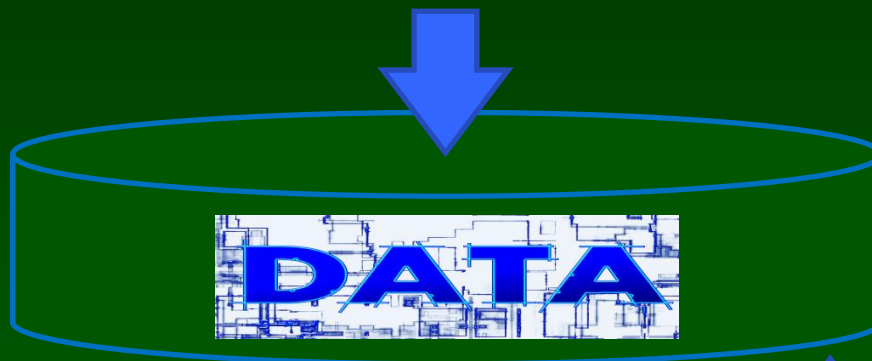
Uczenie maszynowe

- System uczący się używa danych by utworzyć ich model, pozwalający na przewidywanie zachowania dla przyszłych danych. Celem jest generalizacja zdobytej wiedzy.
- Uczenie: zmiany w systemie adaptującym się pozwalające mu w przyszłości działać bardziej efektywnie na zadaniach o podobnym lub analogicznym charakterze. Potrzebne jest wszędzie tam, gdzie nie potrafimy zaprogramować rozwiązania zadania, z braku możliwości symbolicznej reprezentacji, lub nieprzewidywalności opisu wszystkich możliwości.
- Eksperci opisują sytuacje podając przykłady. W GOFAI był to głównie opis symboliczny ale to bardzo ograniczona metoda.
Czasami wiedzę można wyrazić w postaci symbolicznej, ale często zdobywamy ją bezpośrednio z obserwacji za pomocą zmysłów i trudna to opisać symbolicznie.
- Reguły potrzebne do wnioskowania otrzymane metodami ML mogą być lepsze niż reguły wydedukowane przez ludzi. Jednak wiele zagadnień nie da się przedstawić w postaci reguł, ale nawet mało precyzyjne skojarzenia mogą być przydatne do rozumowania.
- Brak wyjaśnialności jest jednym z ważnych problemów ML, który próbuje rozwiązać XAI.

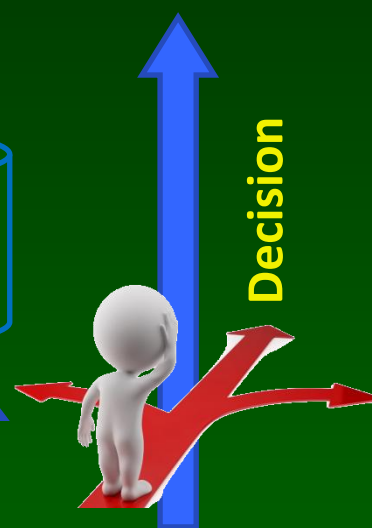
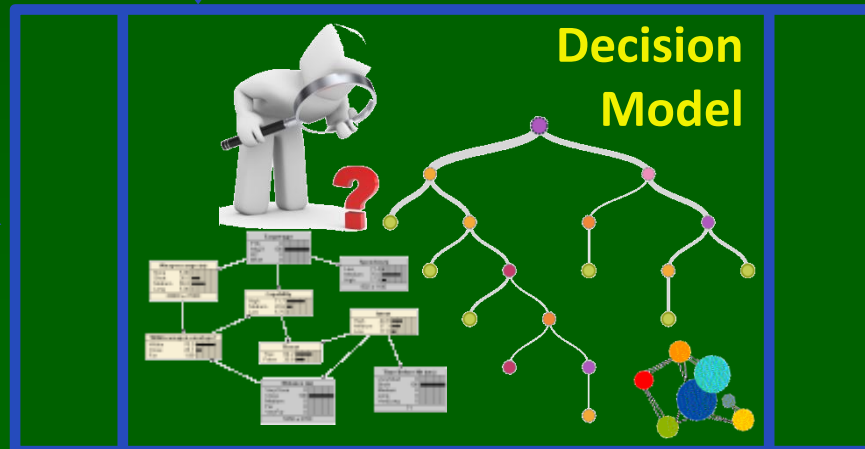
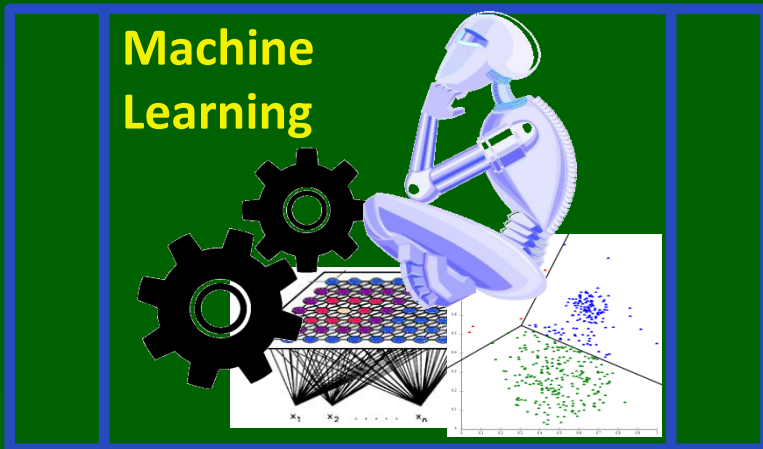
Data Mining i Business Intelligence



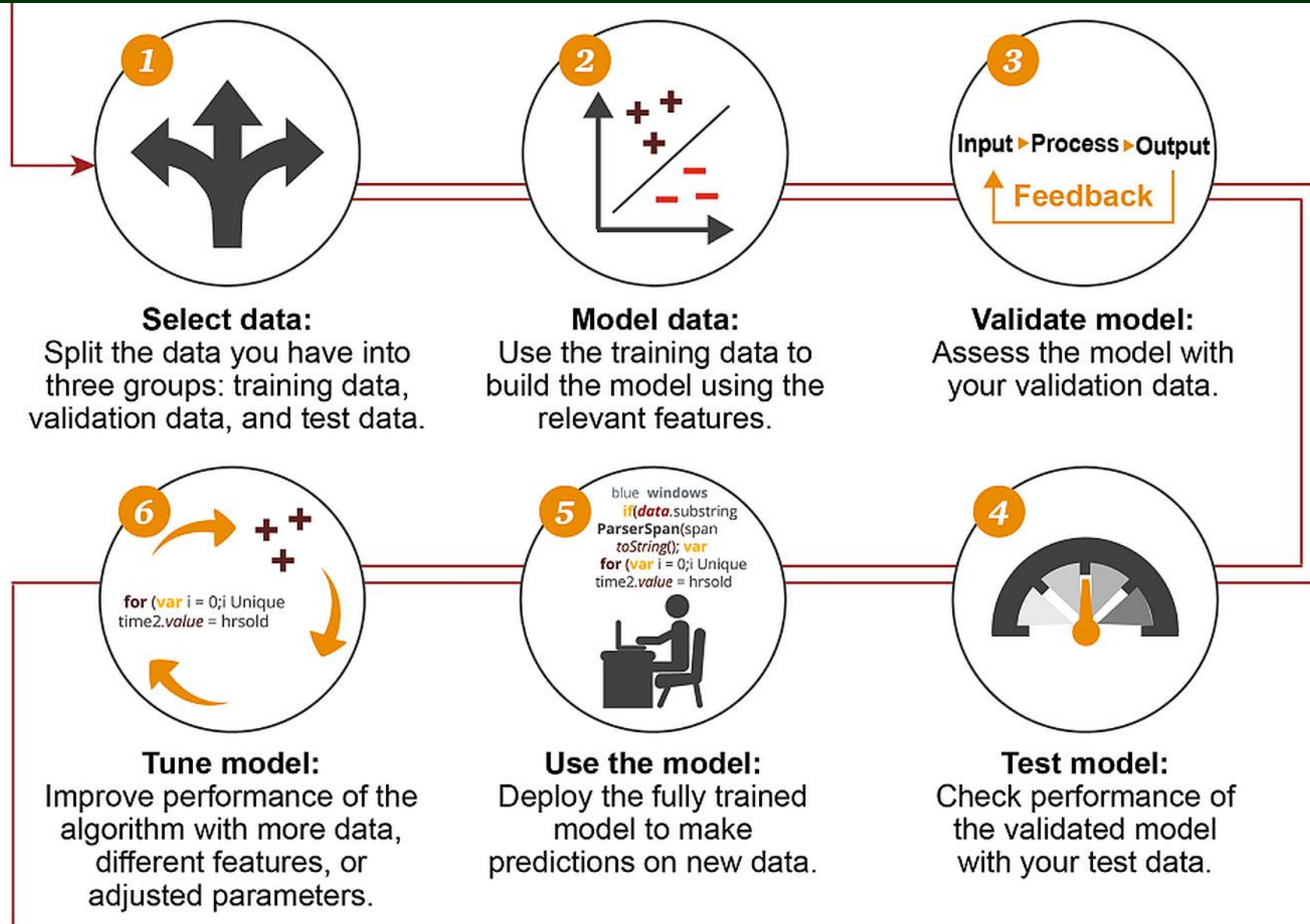
Od obserwacji do decyzji



Data Mining & Knowledge Discovery



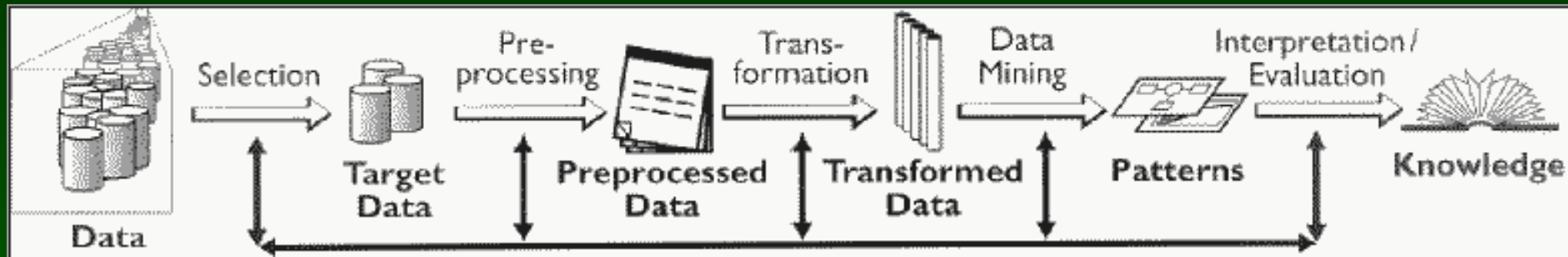
Proces analizy danych: Data Mining



Data Mining

DM część KDD - Knowledge Discovery in Databases.

Szukanie wiedzy w bazach danych, oparte na metodach ML/CI.



Dane - duże bazy, komercyjne, techniczne, tekstowe (WWW).

Selekcja - wybieranie podzbioru danych i atrybutów do analizy.

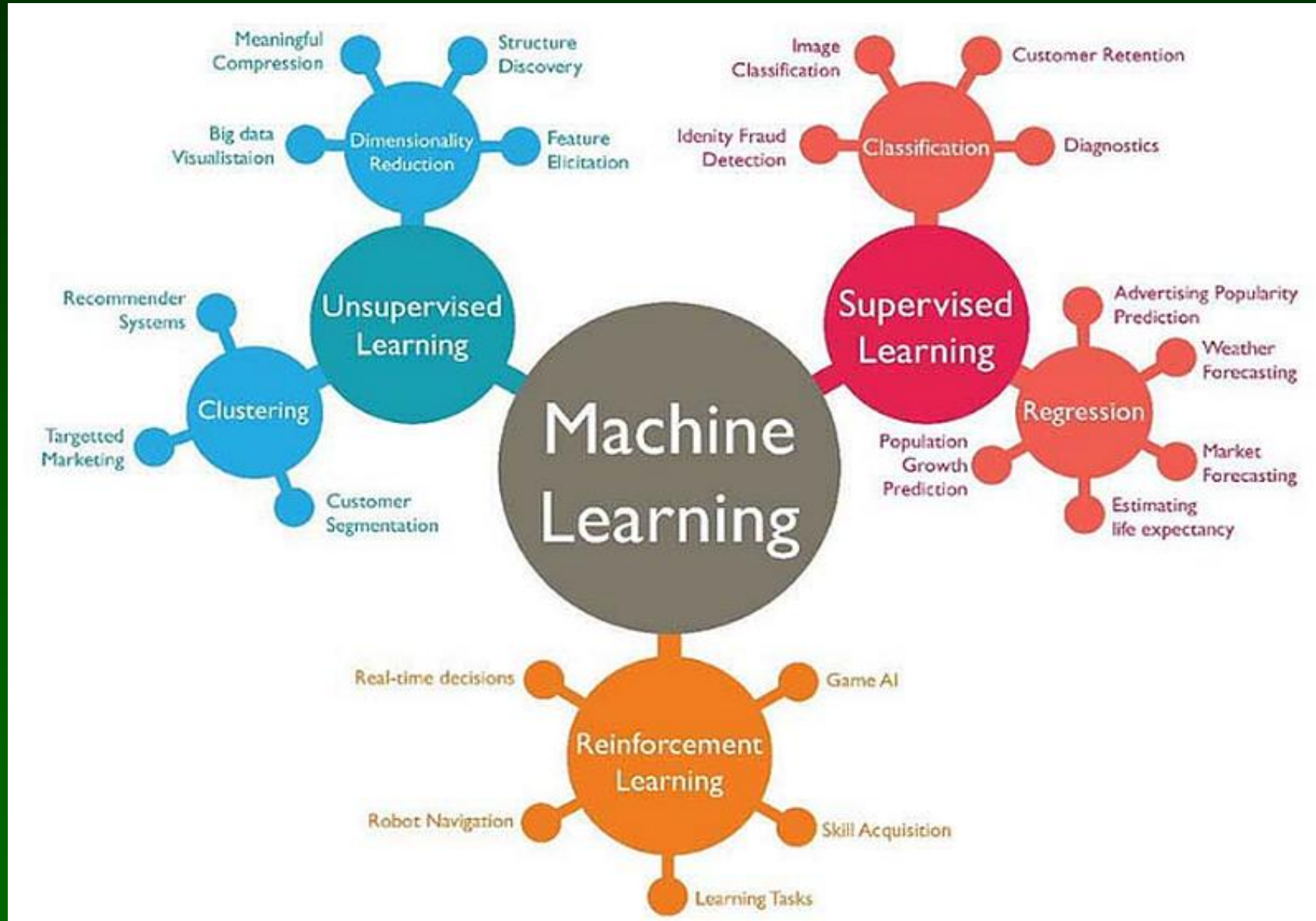
Pre-processing - wstępne przetwarzanie, czyszczenie danych (szum i wyrzutki), uzupełnianie braków, standaryzacja ...

Transformacja - do postaci akceptowalnej przez program.

DM - klasyfikacja, regresja, klasteryzacja, wizualizacja ...

Tutorial na [temat data mining](#).

3 typy ML



Uczenie z nadzorem

Uczenie z nadzorem: podział na znane klasy, przekazywanie znanej wiedzy, heteroasocjacja - kojarzenie obiektów/własności, jak w szkole.

Wymagany jest:

Opis obiektów/danych X , zwykle w postaci wektorów cech, pomiarów własności.

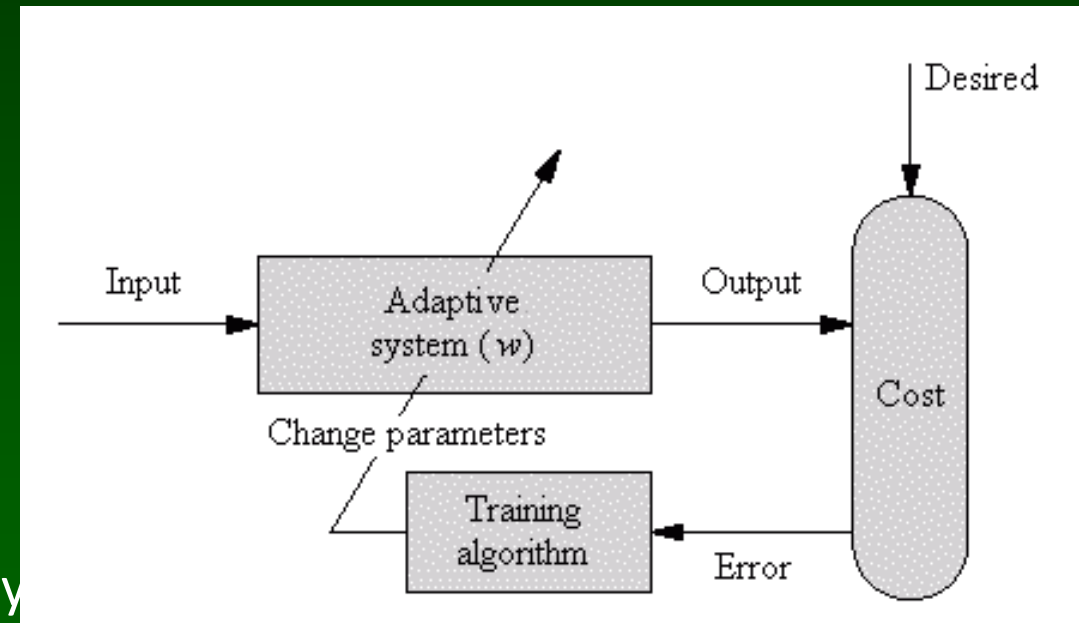
System A adaptujący swoje parametry W do danych.

Pożądaną odpowiedź $Y(X)$

Wyniki działania: $Y' = A(X; W)$

Funkcja błędu/kosztu $E(Y, Y')$

Algorytm minimalizujący błędy na danych treningowych



Cel uczenia: dla nowych danych X otrzymać wyniki $A(X; W) \approx Y(X)$, czyli generalizacja zdobytej wiedzy. Dane treningowe mają tu etykiety.

Uczenie bez nadzoru

Chcemy uprościć opis złożonych danych, bez nadzoru, spontanicznie.

Dominuje w okresie niemowlęcym, nadawanie sensu danym zmysłowym, nauka chodzenia wymaga odkrycia struktur w sygnałach, np. rozumienie mowy wymaga kategoryzacji fonemów w sygnale dźwiękowym.

odkrywanie ciekawych struktur w przestrzeni danych,
korelacja zachowań systemu ze zmianą tych struktur.

Opis danych/sygnałów $p_i \in P$

System adaptujący się

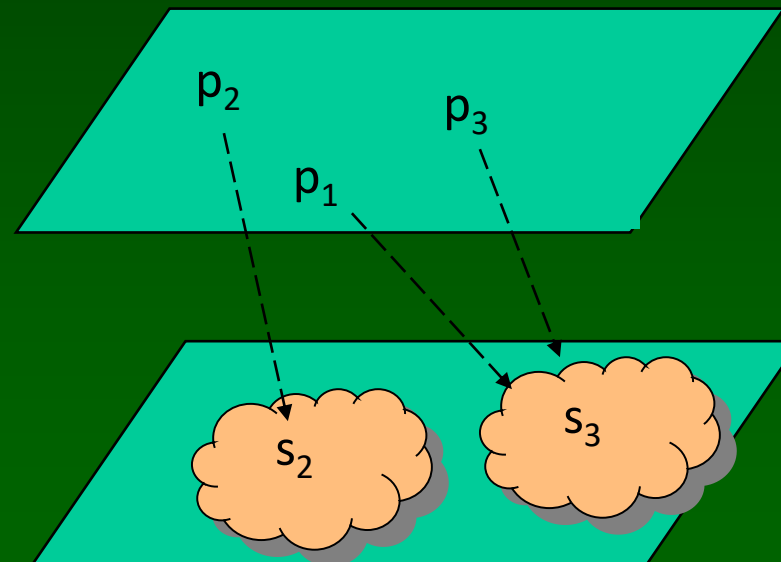
Miary podobieństwa $S(p_i, p_j)$

Algorytm uczący, analizujący rozkłady
miar podobieństwa.

Tworzenie odróżnialnych skupień S_i

Funkcja jakości grupowania $E(S_i, S_j)$

Bioinformatyka – motywy; segmentacja
klientów, analiza sygnałów ...



Uczenie zachowań (reinforcement learning)

Uczymy się też z sukcesów i porażek po wykonaniu jakiegoś zadania.

Nie ma wtedy informacji o błędach po każdym kroku, potrzebna jest strategia, optymalizacja zysków na dłuższą metę. To uczenie behawioralne, nabieranie „mądrości”. Uczenie z krytykiem lub „z wzmocnieniem” ([reinforcement learning](#)) pozwala nauczyć się przydatnych strategii zachowań, np. agenty, roboty, gry z przeciwnikiem: przegrana lub wygrana dopiero na końcu partii.

Potrzebny jest:

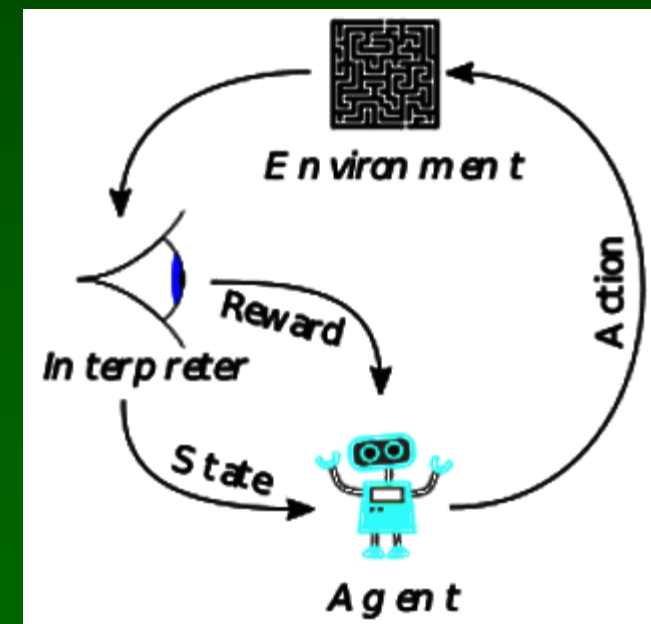
Opis stanów X w obserwowanym środowisku.

Opis akcji a , działania agenta A w stanie s .

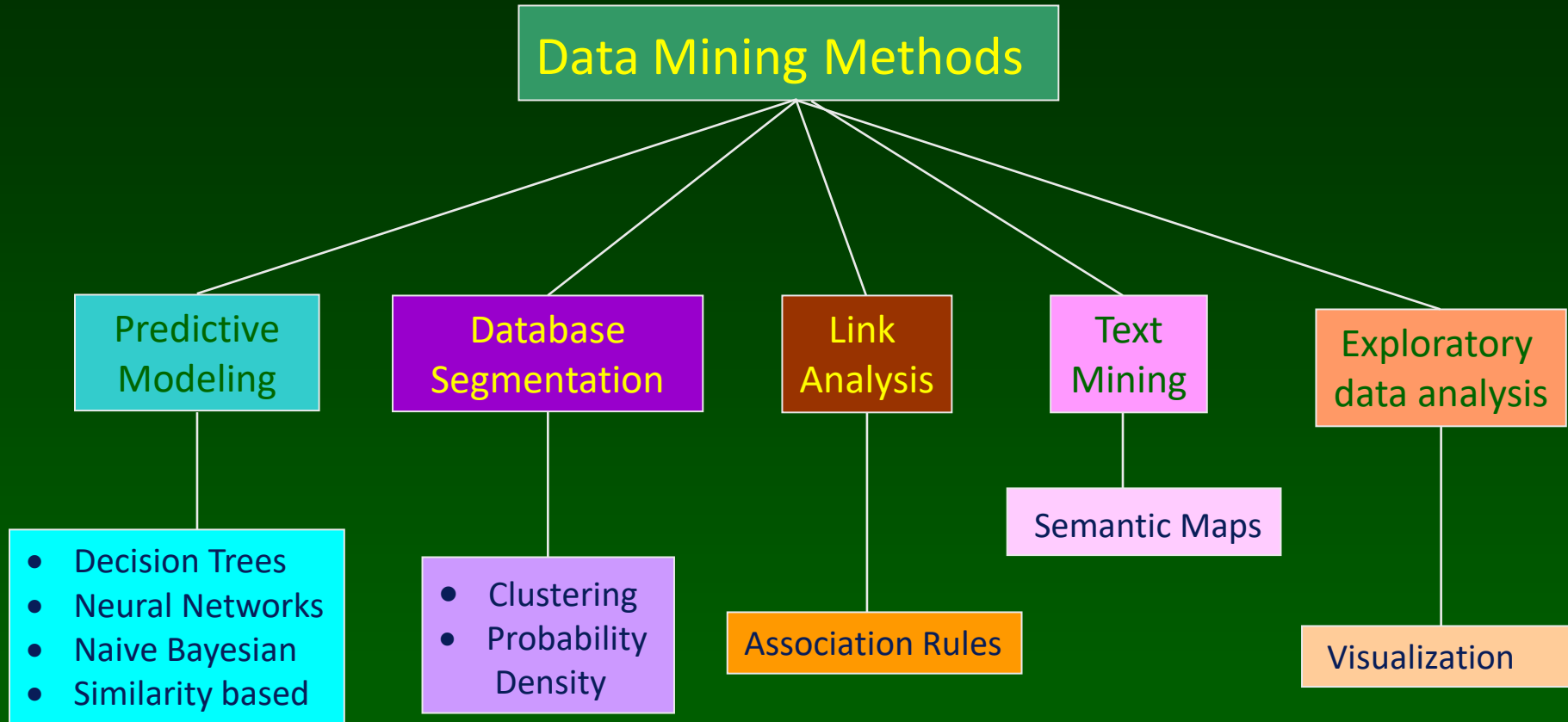
System adaptujący prawdopodobieństwa zmiany stanu $s \Rightarrow s'$ pod wpływem akcji a .

Zmiana stanu środowiska i miara nagrody.

Algorytm tworzący plany zwiększające końcową nagrodę.



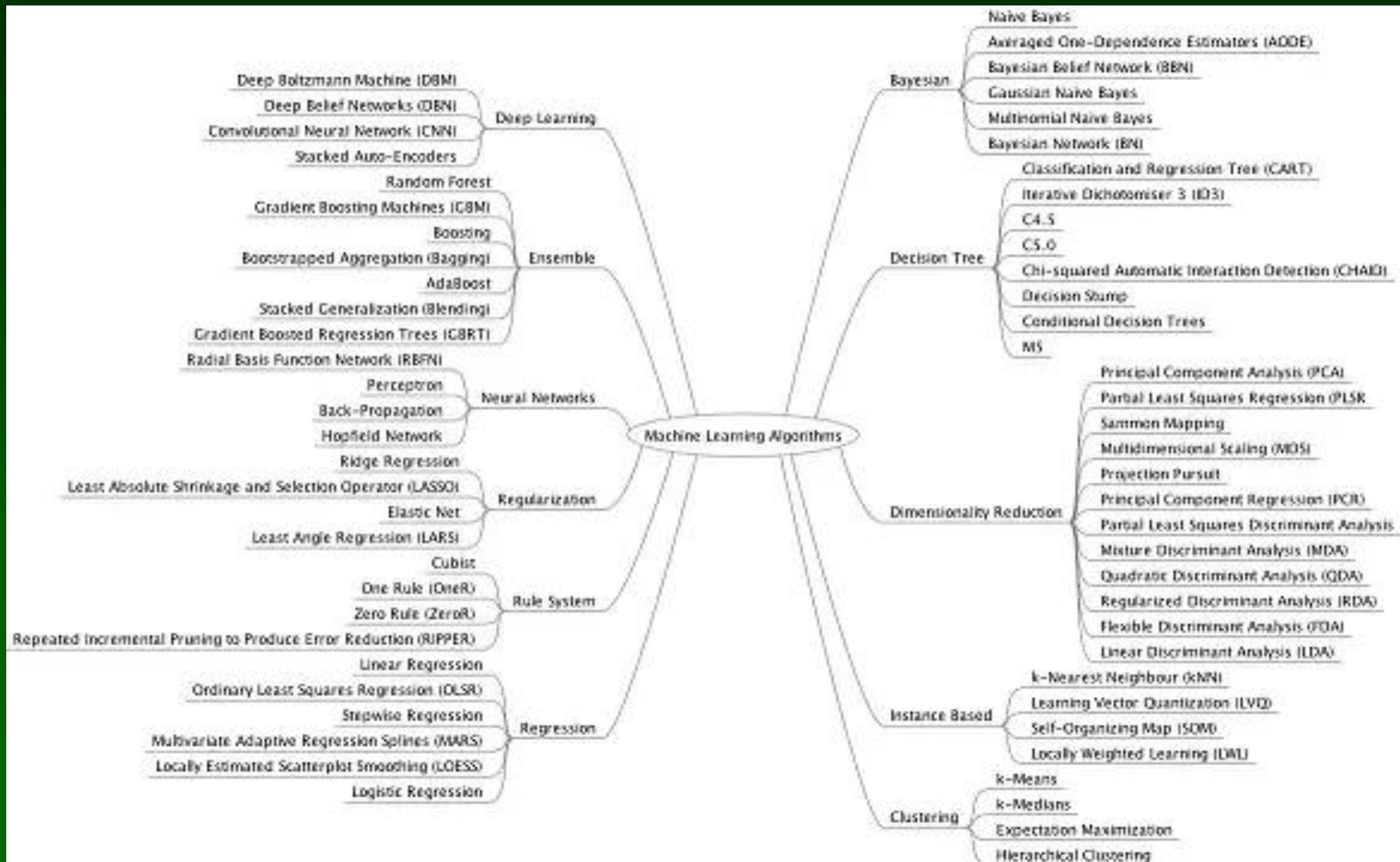
Taksonomia metod Data Mining



Machine Learning

Every Machine Learning Model Explained in 15 minutes.

Rodzaje algorytmów ML



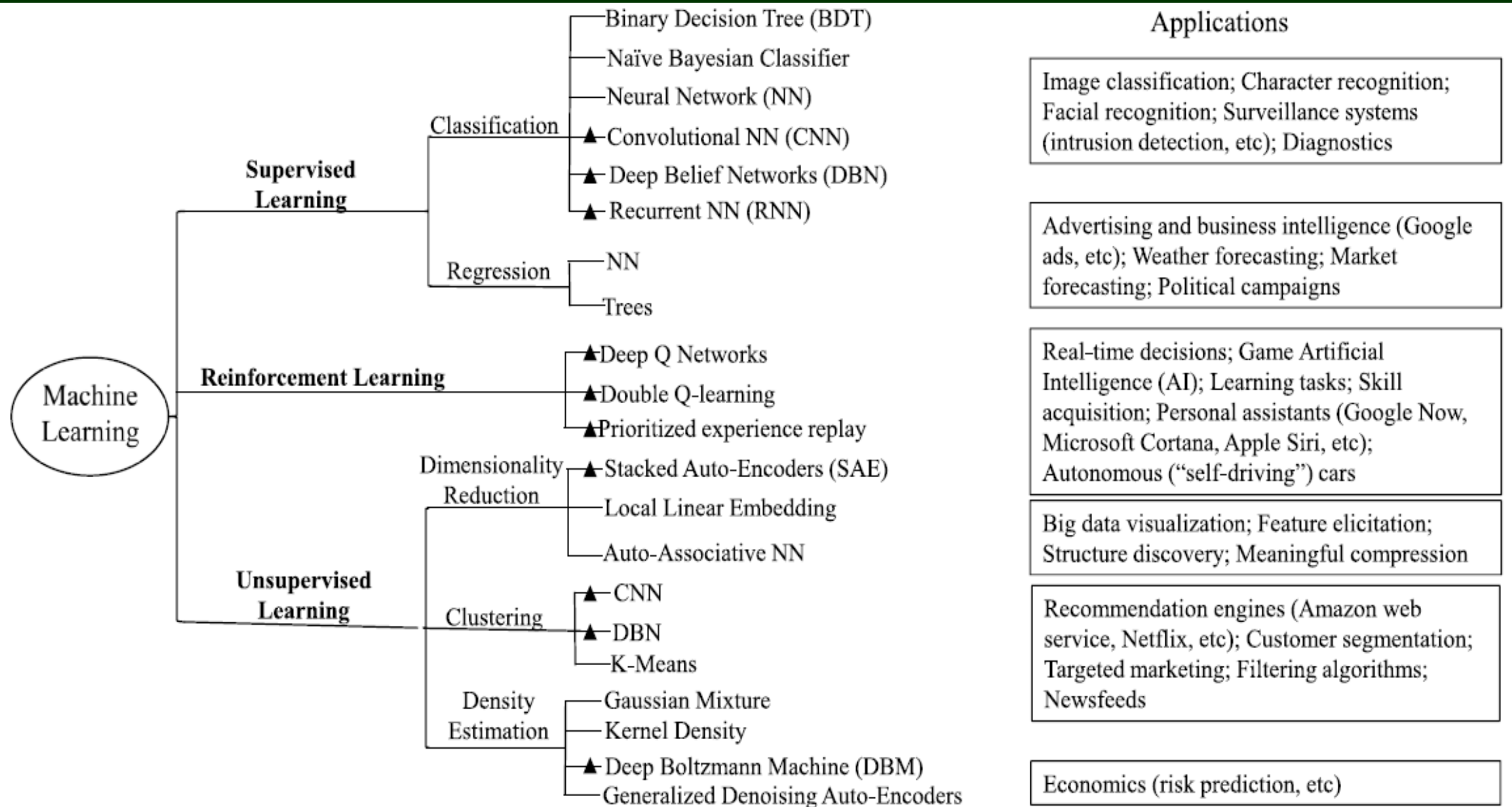
Klasy metod ML

- Powszechnie stosowane metody, ML w Python, tutoriale [scikit-learn](#)
- Drzewa decyzji/indukcja reguł.
- Logika rozmyta, systemy na regułach rozmytych.
- Rozumowanie oparte na analogiach, precedensach (case-based), pamięci (memory-based).
- Dyskryminacja liniowa LDA, SVM - Maszyny Wektorów Wsparcia (klasyfikacja, regresja).
- Sieci neuronowe w wielu odmianach, głębokie sieci neuronowe, sieci konwolucyjne.
- Sieci probabilistyczne (Baysowskie), Naiwny Bayes.
- Algorytmy genetyczne.
- Redukcja wymiarowości, rozmaitości (manifolds).
- Indukcyjne programowanie logiczne (ILP)
- Uczenie ze wzmocnieniem (reinforcement), sterowanie, kontrola, strategie.
- HMM, ukryte modele Markova, modele przestrzeni stanów.
- Modele topologiczne (TDA, topological data analysis).

Więcej: [scikit projects](#). Kombinacja wszystkich modułów => miliardy możliwości ...

- Metalearning - automatyczne tworzenie modeli, kombinacja zwiększająca złożoność/dokładność.

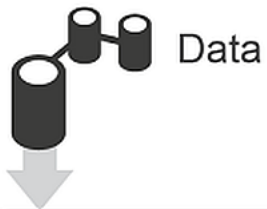
Subtypes of ML



DM i statystyka

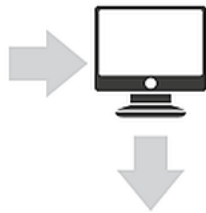
1 Traditional programming

The software engineer writes a program that solves a problem.



Software engineer writes a procedure that tells the machine what to do to solve the problem.

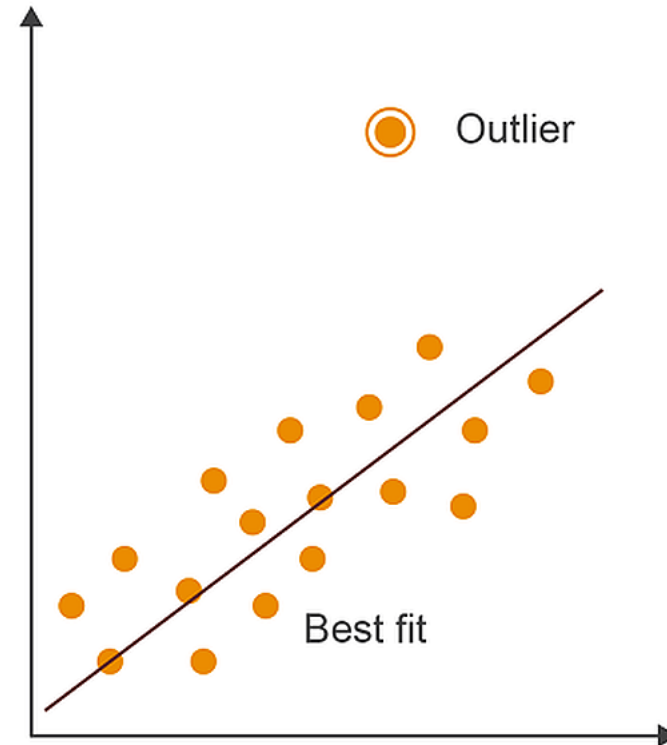
```
class FREE!!! implements SpamDetector
{
    public function detect($string)
    {
        if (str_word_count($string) = FREE!!!) {
            return true;
        }
        return false;
    }
}
```



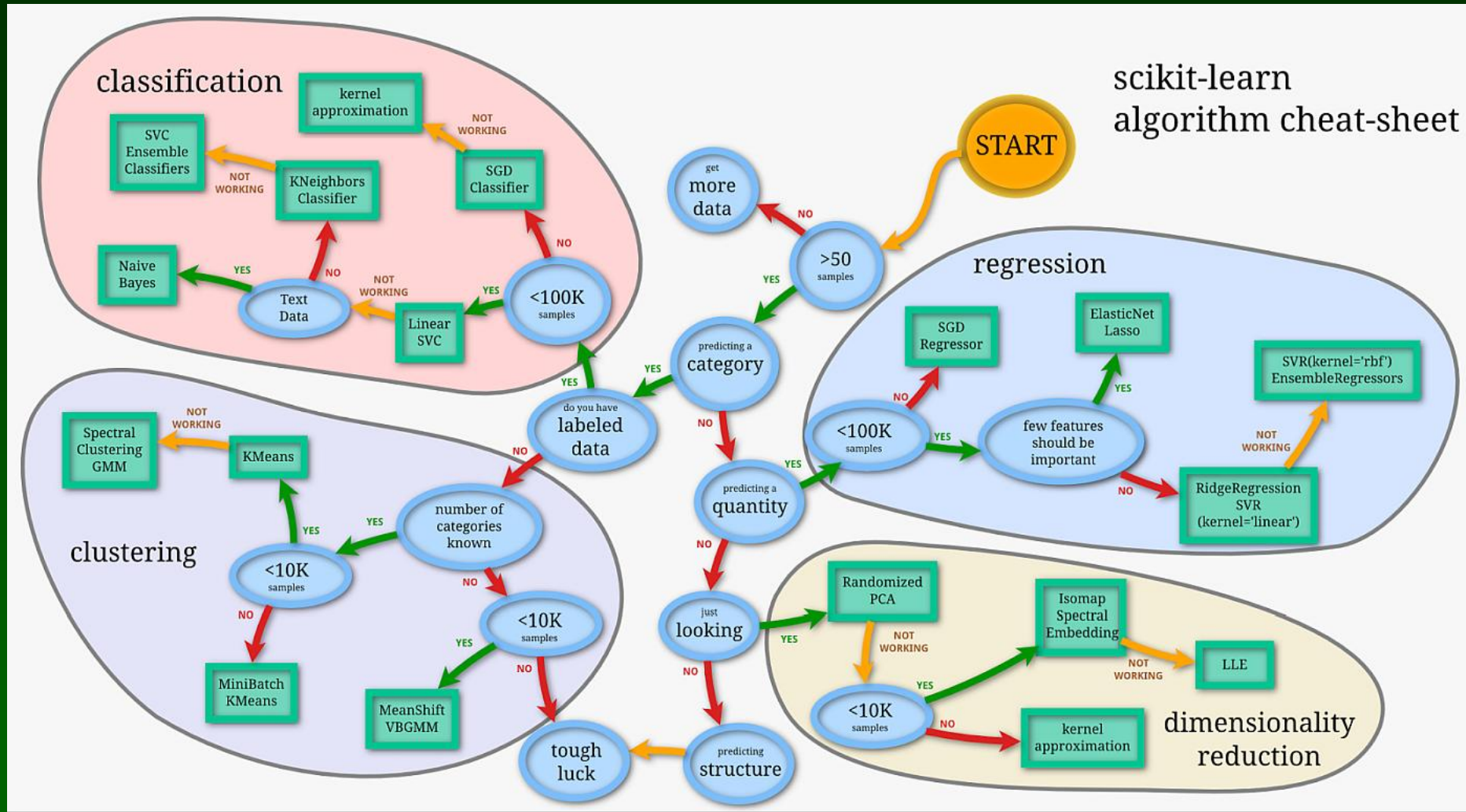
Computer follows the procedure and generates a result.

2 Statistics

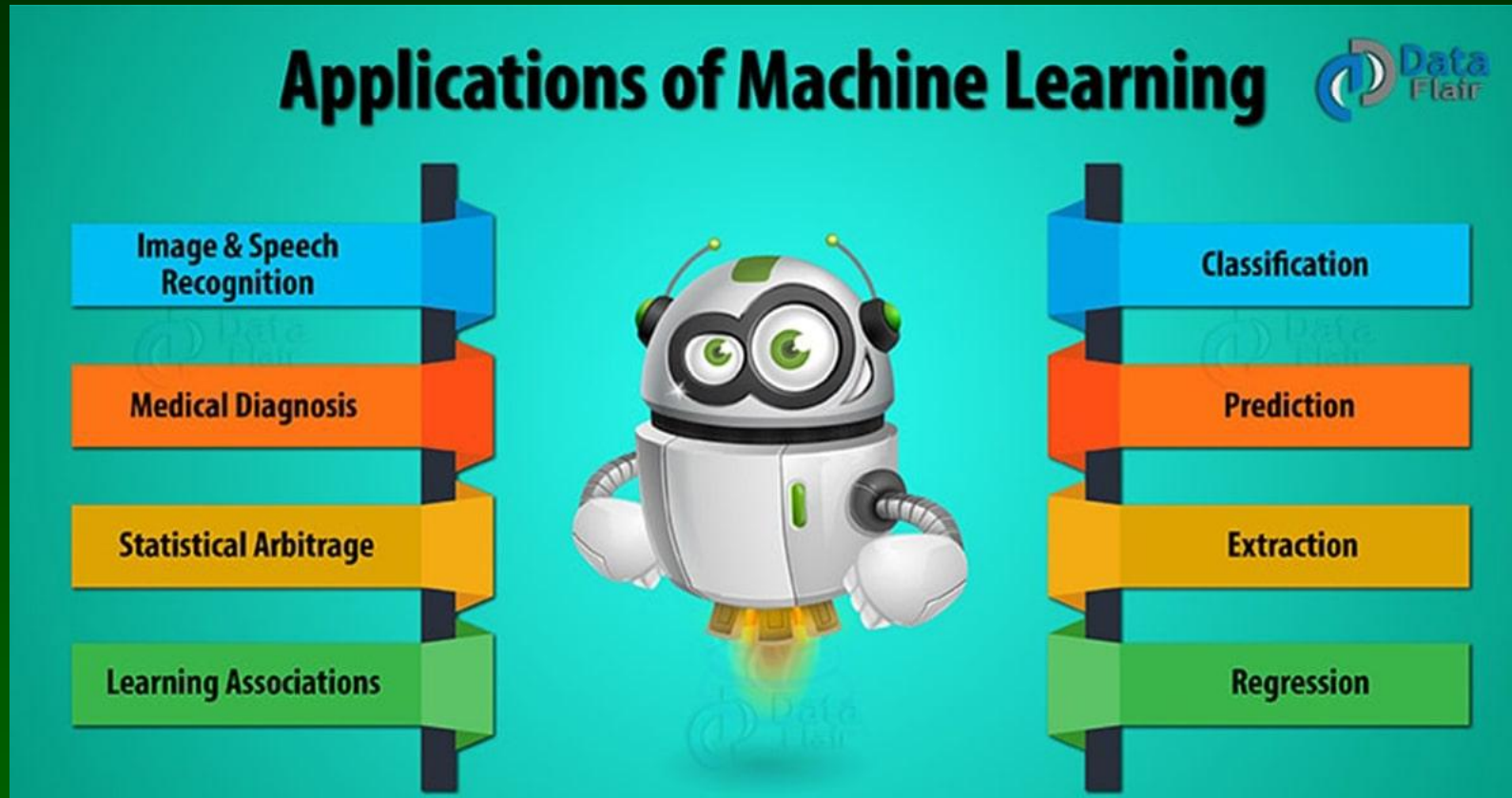
An analyst compares the relationships of variables.



Wybierz co chcesz zrobić



Zastosowania ML



Obecnie ML do wszystkiego ...

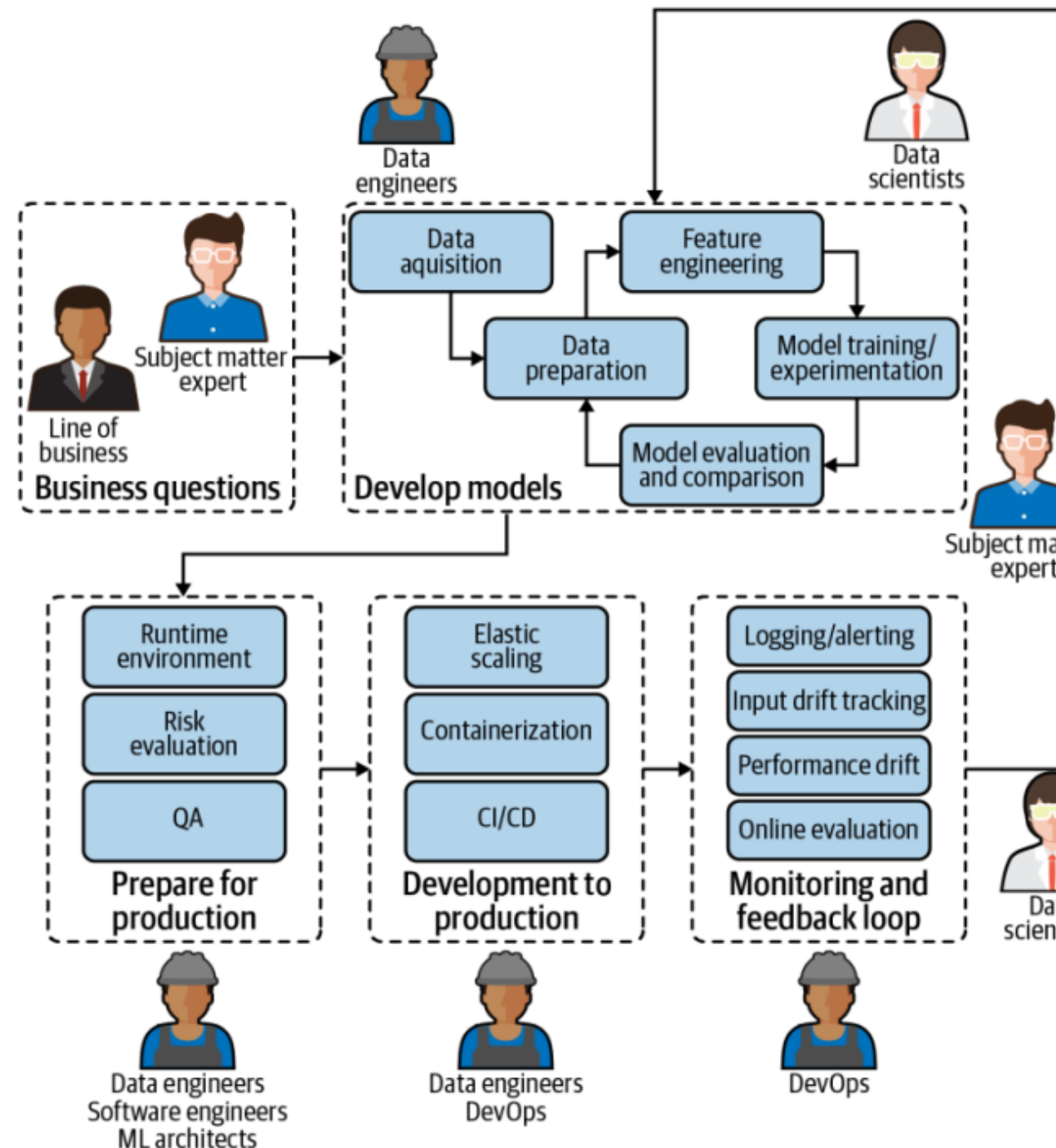
- ✓ Zintegrowane pakiety programów z elementami ML do data mining.
- ✓ Analiza obrazów i sygnałów.
- ✓ Sterowanie: automatyczny kierowca/pilot/czołg ...
- ✓ Analiza tekstów.

MLOps

MLOps, Machine Learning Operations, ModelOPs, AIOps, DataOps, DevOps ... to metodologie usprawniające współpracę pomiędzy Data Scientist i ekspertów AI a pracownikami różnych firm.

Pomagają skutecznie wdrażać AI do systemów produkcyjnych w wielkiej skali.

To problemy inżynieryjno-naukowo-biznesowe, a nie samo rozwijanie algorytmów. Analiza ryzyka, responsible AI



Exploracyjna analiza danych

Wiki

Projekcje 2D: skaterogramy

Najprostsze projekcje: użyj projekcji, wybierz tylko 2 cechy.

Przykład: cukier - próchnica zębów.

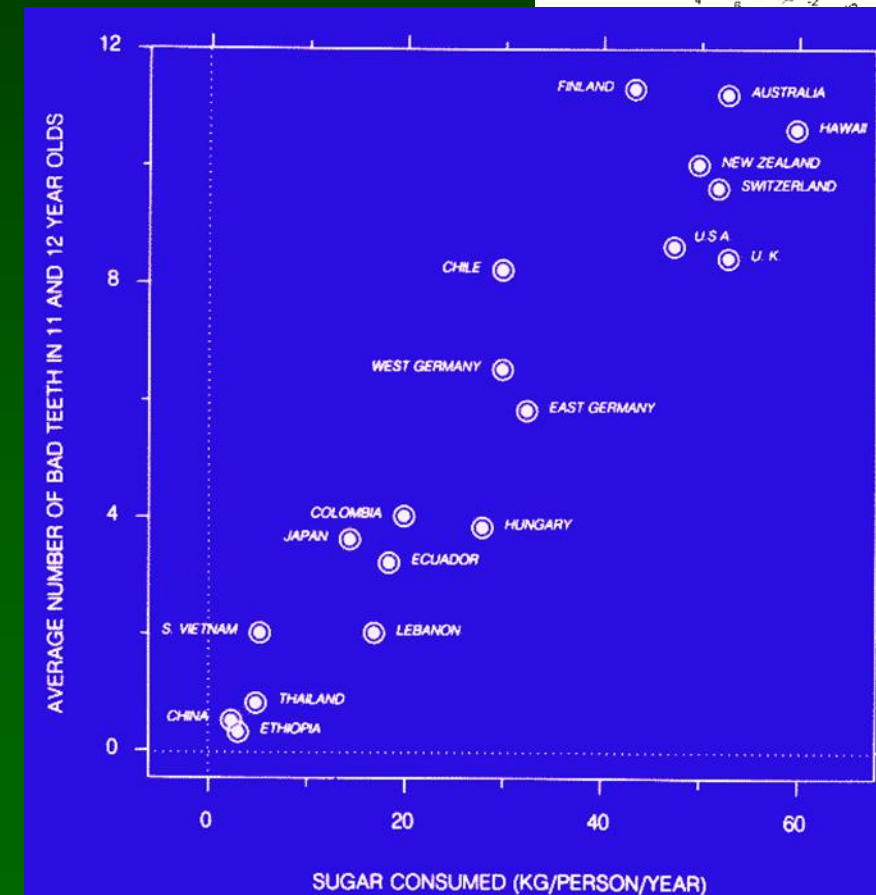
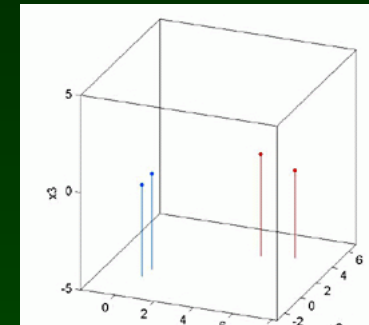
Dla d zmiennych mamy $d(d-1)/2$ podzbiorów 2D i czasami wyświetlane są na jednym rysunku.

Każdy punkt 2D jest rzutem ortogonalnym ze wszystkich pozostałych $d-2$ wymiarów.

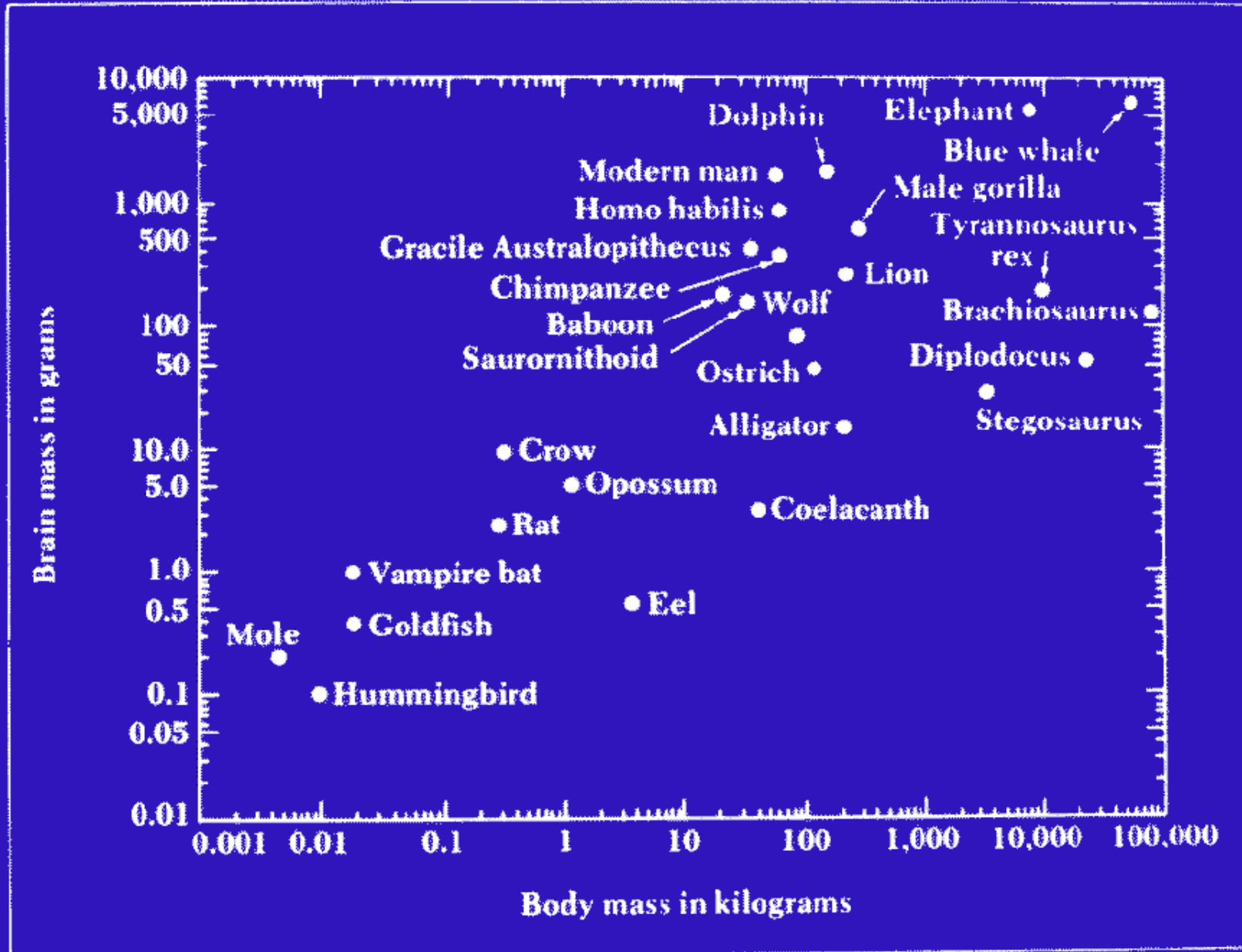
Czego szukać:

korelacje między zmiennymi,
grupowanie różnych obiektów.

Dla wartości dyskretne punkty danych nakładają się na siebie.
Ekstremalny przypadek: dane binarne w wielu wymiarach, cała struktura jest ukryta, każdy scatterogram to 4 punkty.



Masa mózgu vs cięła



4 Gaussy w 4D

Rozkłady mają często postać Gaussów.

Skaterogramy danych 4D w dwóch wymiarach.

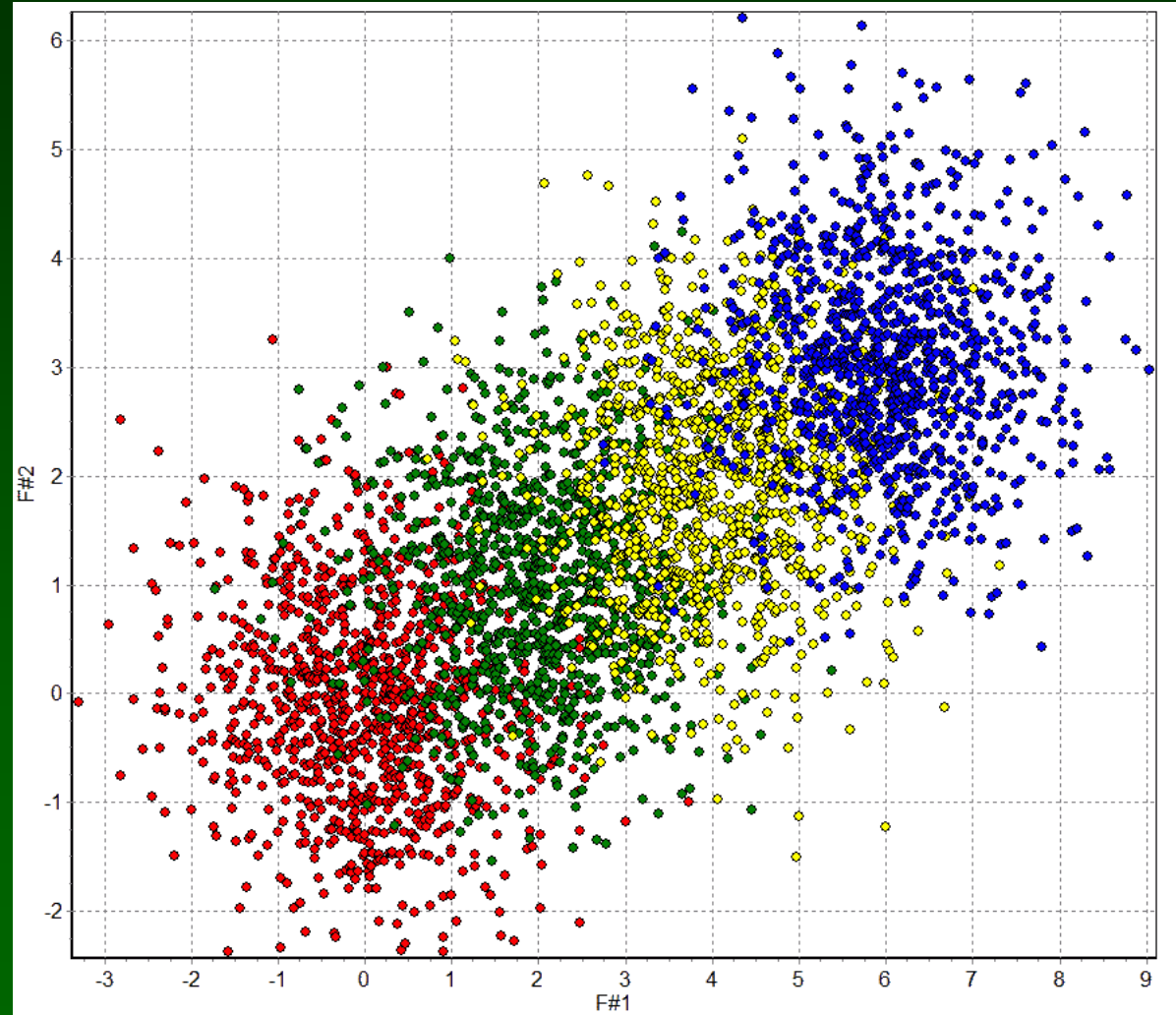
4 rozkłady w 4D, przesunięte względem siebie:

czzerwony $(0,0,0,0)$,

zielony $(1,1/2,1/3,1/4)$,

żółty $2(1,1/2,1/3,1/4)$

niebieski $3(1,1/2,1/3,1/4)$

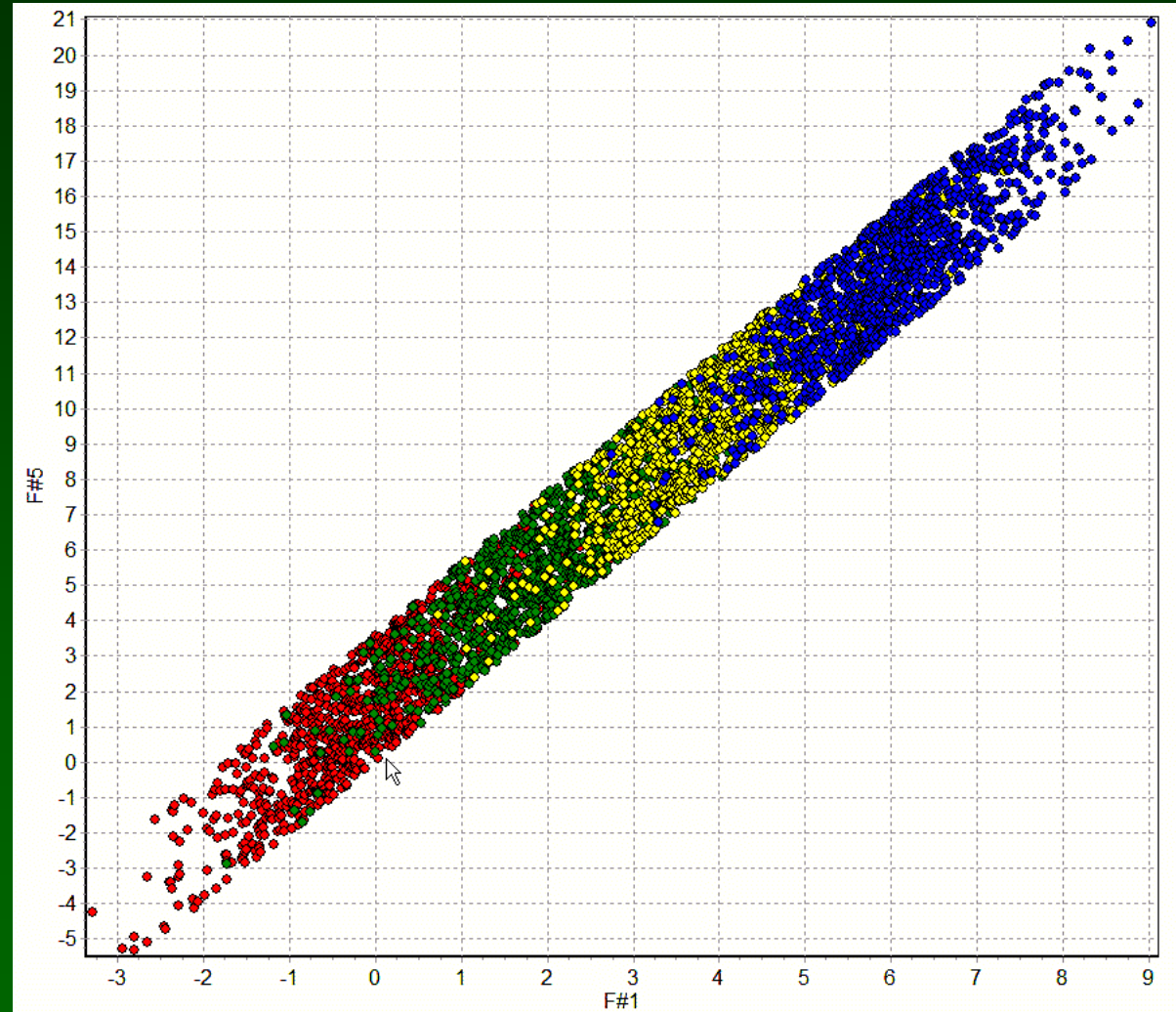


4 Gaussy w 8D, X_1 vs. X_5

O czym świadczy ten rysunek?

Wszystkie X_i vs. X_{i+4} tak wyglądają.

Jak te dodatkowe 4 cechy zostały utworzone?



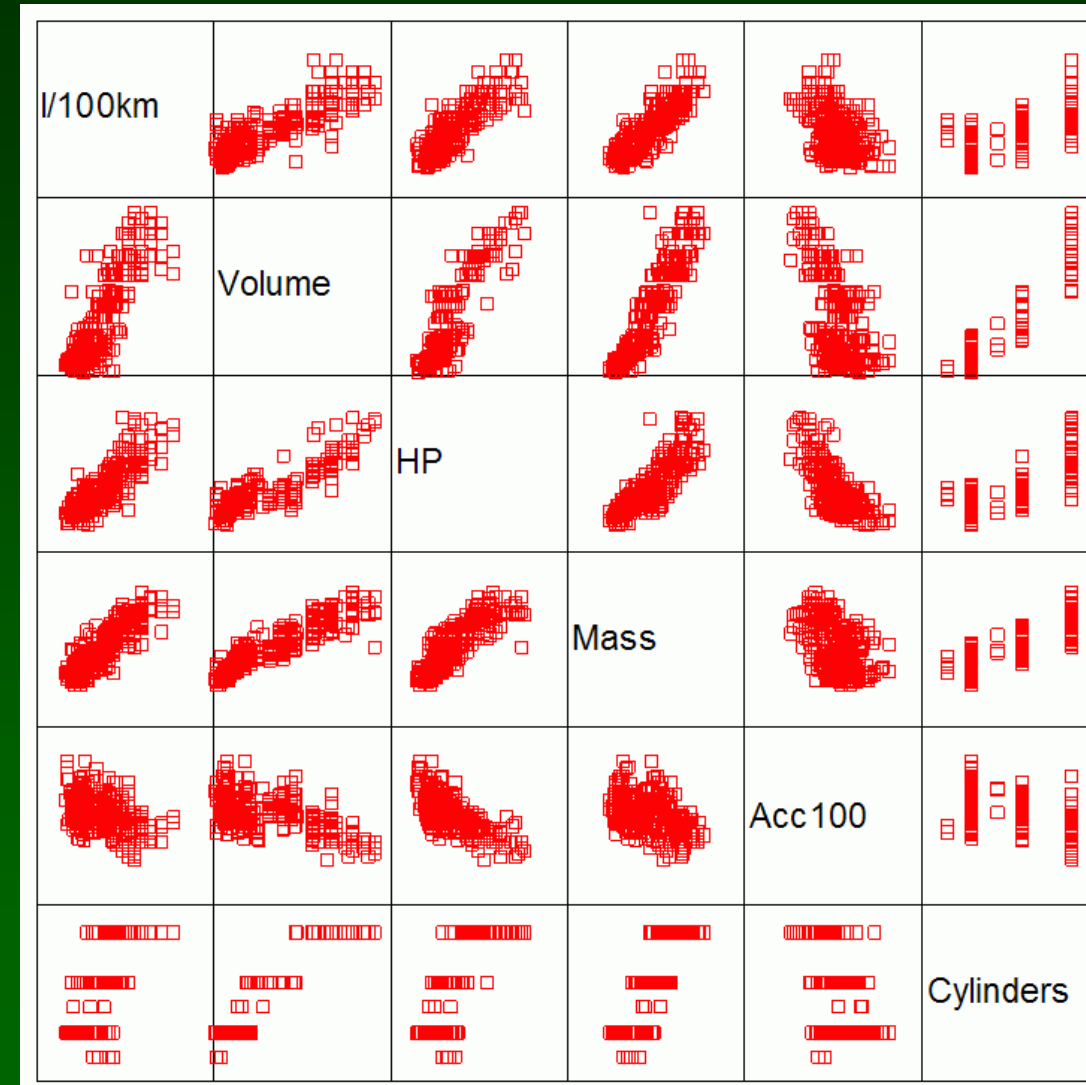
Samochody

Scaterogramy dla wszystkich par cech, dane dla samochodów z 3, 4, 5, 6 lub 8 cylindrami.

Dane punktowe są zbyt szczegółowe.
Interesują nas trendy, które można zaobserwować w funkcjach gęstości prawdopodobieństwa.
Skupienie wszystkich punktów, które są blisko siebie dla samochodów z N cylindrami.

Można to zrobić, dodając do każdego punktu szum gaussowski o rosnącej wariancji.

[Zobacz to na filmie.](#)



Grand Tour

Jeśli mamy wiele wymiarów może utworzyć projekcje zmieniając kierunki.
Grand Tour: implementacja w XGobi, XLispStat, ExplorN i innych programach.

Przykład: dane [7D w Grand Tour](#)

Widok sześciangu w 9D zwykle robi wrażenie chmury punktów Gaussa.

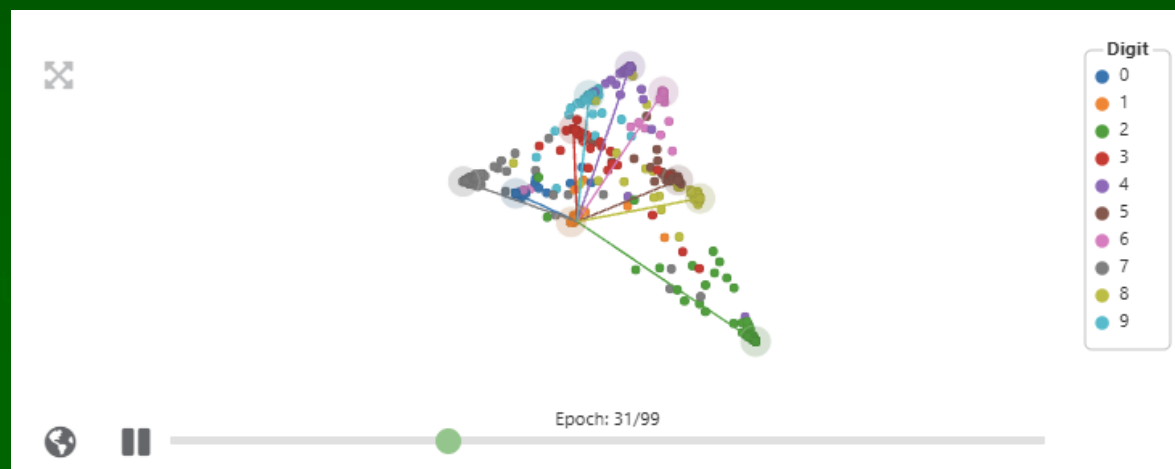
Wizualizacja aktywności [warstw sieci neuronowych](#) przez Grand Tour

Więcej przykładów:

[Tours for the dynamic visualization](#) of high-dimensional data, Ursula Laa

Data [tour w Python](#)

[Wiki Grand Tour](#)



Reprezentacje bezpośrednie: star

Zmiana liczby pracujących kobiet w poszczególnych stanach.

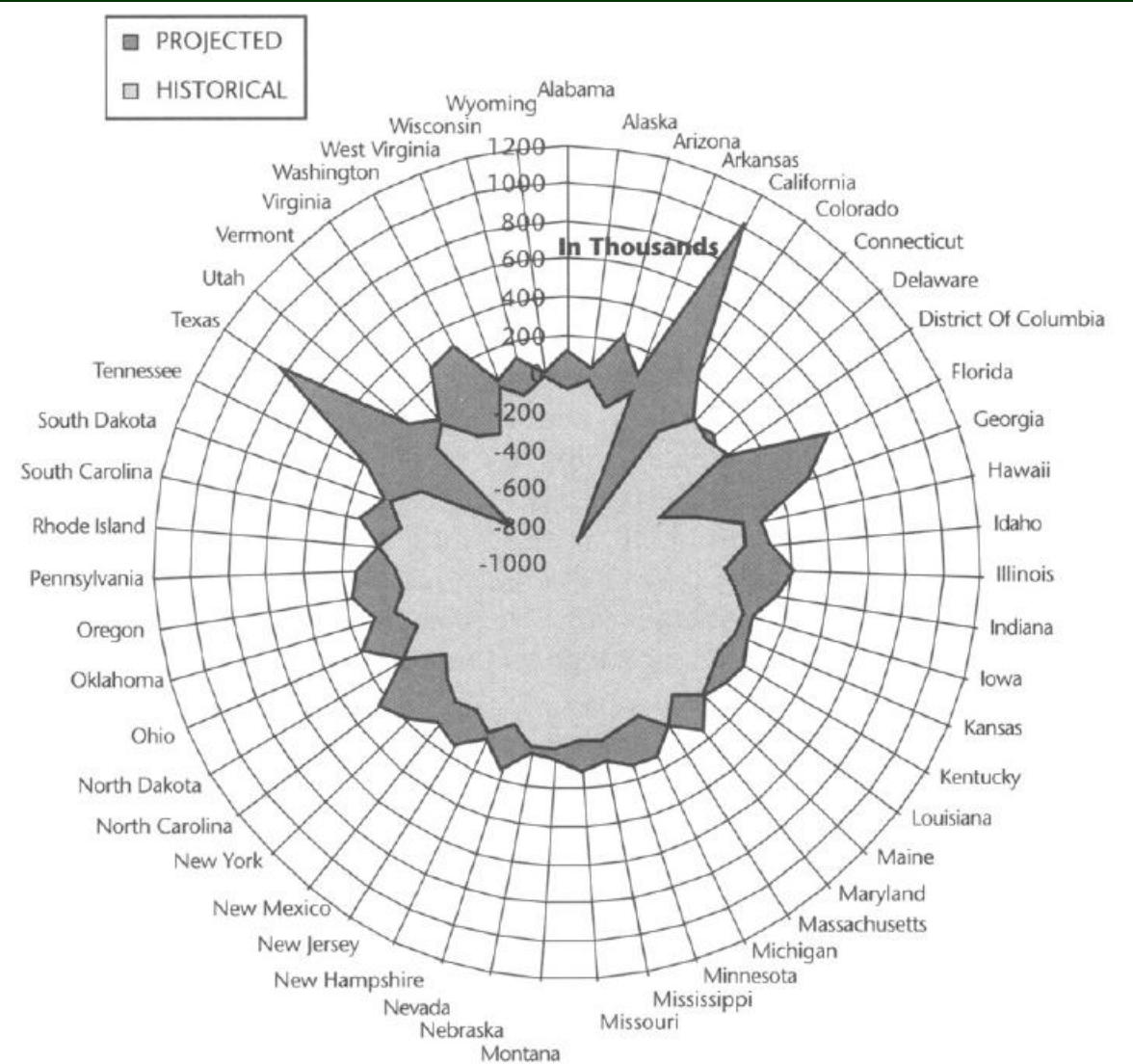


Figure 8.8: Projected versus historical working women population changes by state.

Reprezentacje bezpośrednie: ||

Współrzędne równoległe: zamiast prostopadłych narysujmy osie jako równoległe!

Popularna w bioinformatyce, wiele zastosowań w naukach technicznych.

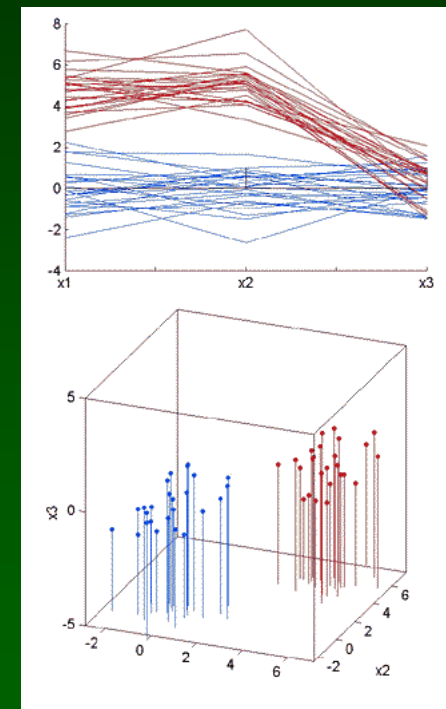
[Wiki page](#) + [software links](#)

Dwa klastry w 3D

Punkty w N-wymiarowej przestrzeni => linie o N segmentach.

Alfred Inselberg, książka [Parallel Coordinates: Visual Multidimensional Geometry and Its Applications](#) (Amazon)

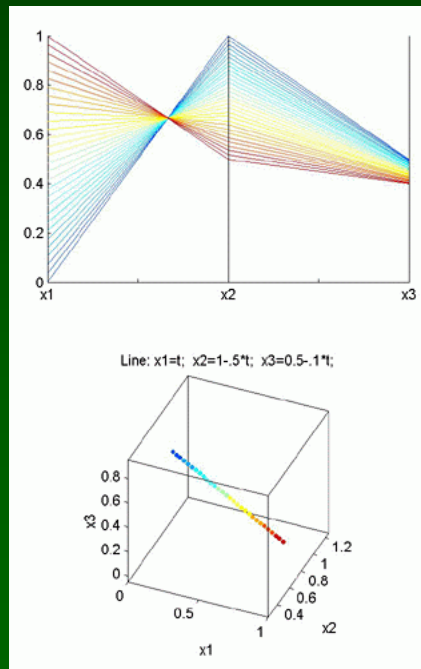
<https://people.ece.cornell.edu/land/PROJECTS/Inselberg/>



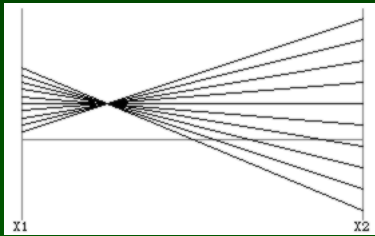
|| lines

Proste w wielu wymiarach

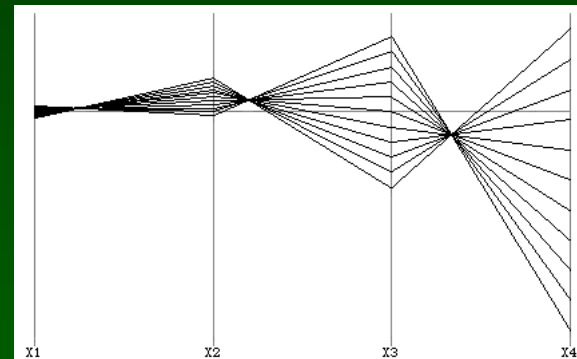
2D line



3D line



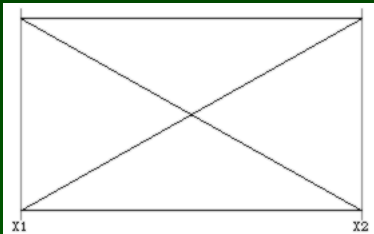
4D line



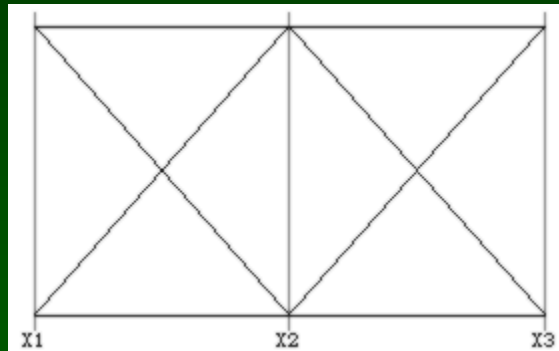
|| cubes

Hipersześciiany w współrzędnych równoległych:

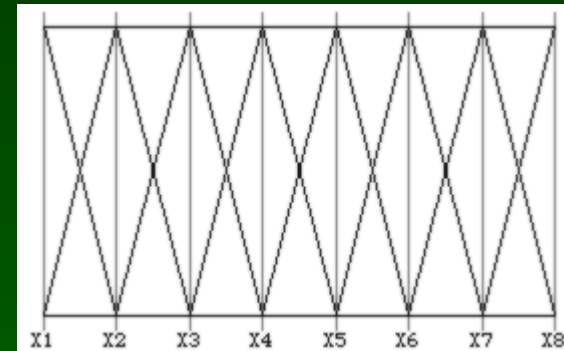
2D (kwadraty)



3D sześciiany: 8 vertices



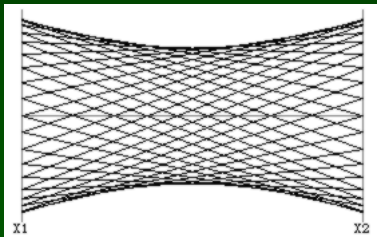
8D: 256 vertices



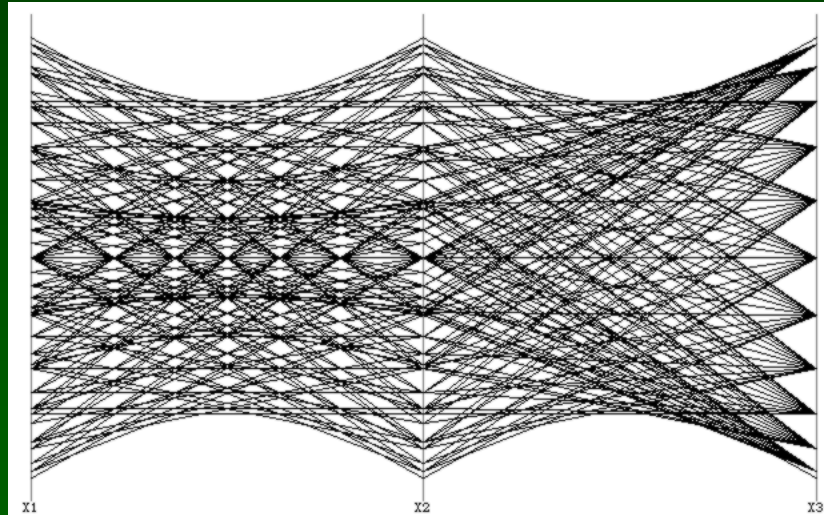
|| sfery

Hipersfery

2D (circle)



3D (sphere)

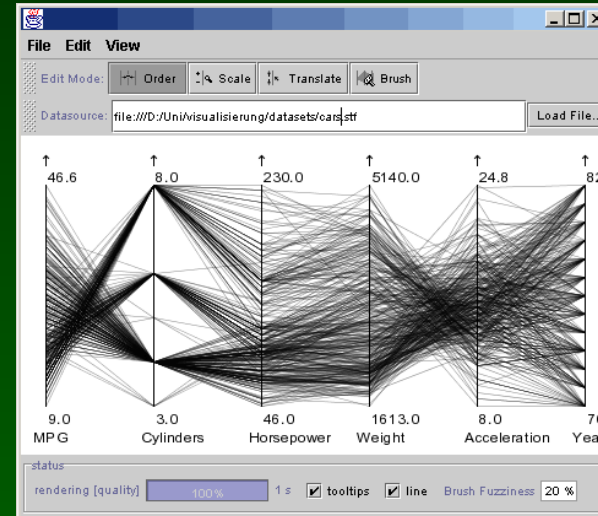
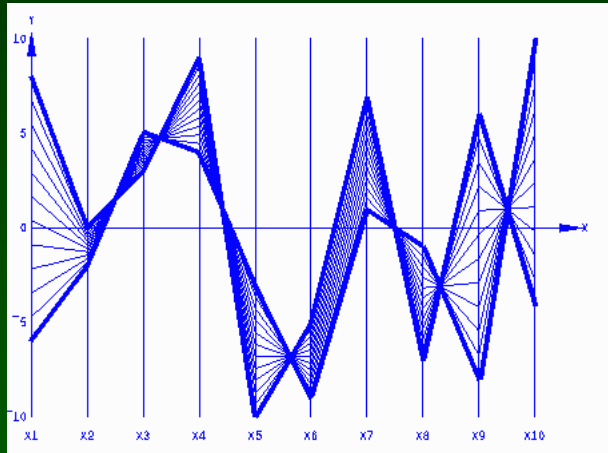


... 8D: ???

Try some other geometrical figures and see what patterns are created.

|| współrzędne

Reprezentacje w 10-D (x_1, \dots, x_{10}), Informacje na temat samochodów



Parallax software: <http://www.kdnuggets.com/software/parallax/>
IBM Visualization Data Explorer <http://www.research.ibm.com/dx/>
Parallel Coordinates module
<http://www.cs.wpi.edu/Research/DataExplorer/contrib/parcoord/>
[Financial analysis example](#)

MDS: idea

MDS, Multi-Dimensional Scaling, Sammon mapping
(Thorton 1954, Kruskal 1964, Sammon 1964)

Inspiracje głównie z psychologii lub psychometrii!

Problem: wizualna reprezentacja danych o podobieństwie/odmienności.

Informacje o klastrach w danych znajdują się w relacjach między maksimami gęstości prawdopodobieństwa znalezienia wektorów danych.

Wizualizacja danych wielowymiarowych prowadzi do pewnych zniekształceń topograficznych, więc wymagana jest miara tych zniekształceń.

Jak różnią się odległości w oryginalnej, d -wymiarowej przestrzeni i w przestrzeni docelowej, zwykle 2-wymiarowej?

Algorytm MDS

- Dane w przestrzeni cech $\mathbf{X} \in \mathbb{R}^d$, wektory $\mathbf{X}^{(i)}$, $i=1..n$ reprezentowane przez punkty \mathbf{Y} w przestrzeni docelowej, zwykle $\mathbf{Y} \in \mathbb{R}^2$.
- Odległości $R_{ij} = \|\mathbf{X}^{(i)} - \mathbf{X}^{(j)}\|$ pomiędzy $\mathbf{X}^{(i)}$ i $\mathbf{X}^{(j)}$ w przestrzeni cech \mathbb{R}^d ;
a odległości $r_{ij} = \|\mathbf{Y}^{(i)} - \mathbf{Y}^{(j)}\|$ w przestrzeni docelowej \mathbb{R}^2 .
- Szukamy odwzorowania $\mathbf{X} \rightarrow \mathbf{Y} = \mathbf{M}(\mathbf{X})$ które minimalizują jakiś globalny indeks zniekształcenia topografii danych, oparty na porównaniu R_{ij} i r_{ij} .

$$E(\mathbf{r}) = \sum_{i>j}^n (R_{ij} - r_{ij})^2 = \sum_{i>j}^n \left(R_{ij} - \sqrt{(Y_1^{(i)} - Y_1^{(j)})^2 + (Y_2^{(i)} - Y_2^{(j)})^2} \right)^2$$

$\mathbf{r} = \mathbf{r}(\mathbf{Y})$, więc $E(\mathbf{r})$ zależy od $2n$ parameterów adaptacyjnych \mathbf{Y} ;
3 są redundantne: $\mathbf{Y}^{(1)} = (0, 0)^T$, środek układu współrzędnych,
i jedna ze składowych $Y_1^{(2)} = 0$ ustalającą orientację tego układu.

Inicjalizacja \mathbf{Y} może być przypadkowa, albo korzystająca z PCA. Minimalizujemy $E(\mathbf{r}(\mathbf{Y}))$

Miary zniekształcenia

Każda funkcja nieujemna $f(R_{ij}, r_{ij})$ może być użyta.

$$E(\mathbf{r}; \mathbf{W}) = \sum_{i>j}^n W_{ij} (R_{ij} - r_{ij})^2 \geq 0$$

Wagi mogą zależeć od odległości, na przykład malejąc wykładniczo dla dużych odległości R_{ij} .

Można wprowadzić różne współczynniki normalizacyjne, ale te, które nie zależą od odległości R_{ij} , nie mają wpływu na mapę MDS.

$$A(\mathbf{r}) = \frac{\sum_{i>j}^n (R_{ij} - r_{ij})^2}{\sum_{i>j}^n R_{ij}^2 + \sum_{i>j}^n r_{ij}^2} \in [0,1]$$

Strata informacji;

0 = bez straty, doskonała reprezentacja,

1 = brak informacji, ex. $r_{ij}=0$

Miary zniekształceń topografii

Wskaźnik Stressa (Kruskal), czyli błąd bezwzględny – dominują duże odległości, zachowuje się ogólną strukturę skupisk.

$$S_1(\mathbf{r}) = \sum_{i>j}^n (R_{ij} - r_{ij})^2 \geq 0$$

Wskaźnik Sammona, czyli błąd pośredni – wpływ dużych odległości jest ograniczony.

$$S_2(\mathbf{r}) = \sum_{i>j}^n \frac{(R_{ij} - r_{ij})^2}{R_{ij}} \geq 0$$

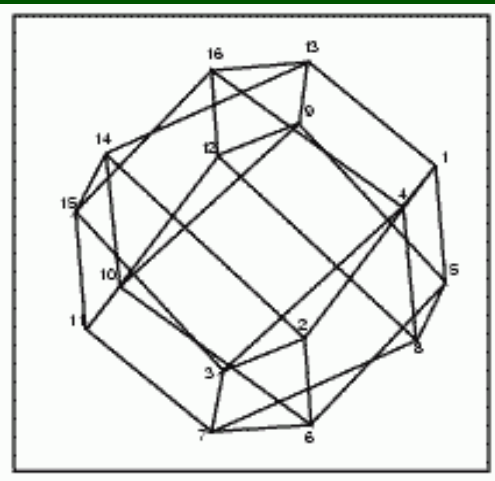
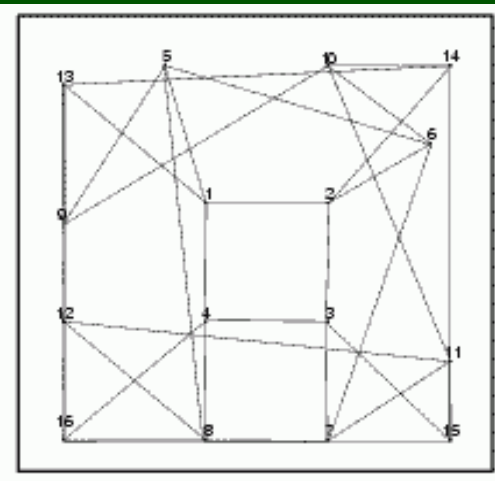
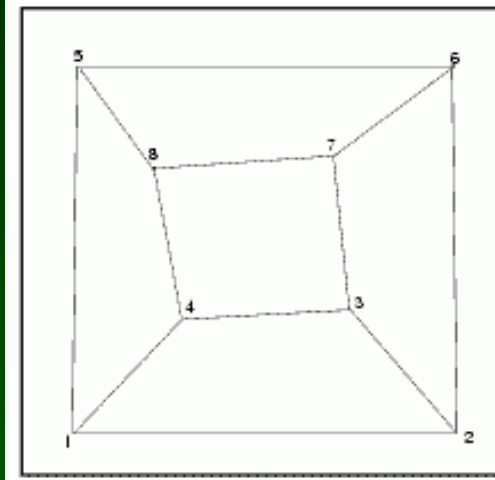
Błąd względny – wszystkie skale odległości traktowane są w ten sam sposób.

$$S_3(\mathbf{r}) = \sum_{i>j}^n (1 - r_{ij} / R_{ij})^2 \geq 0$$

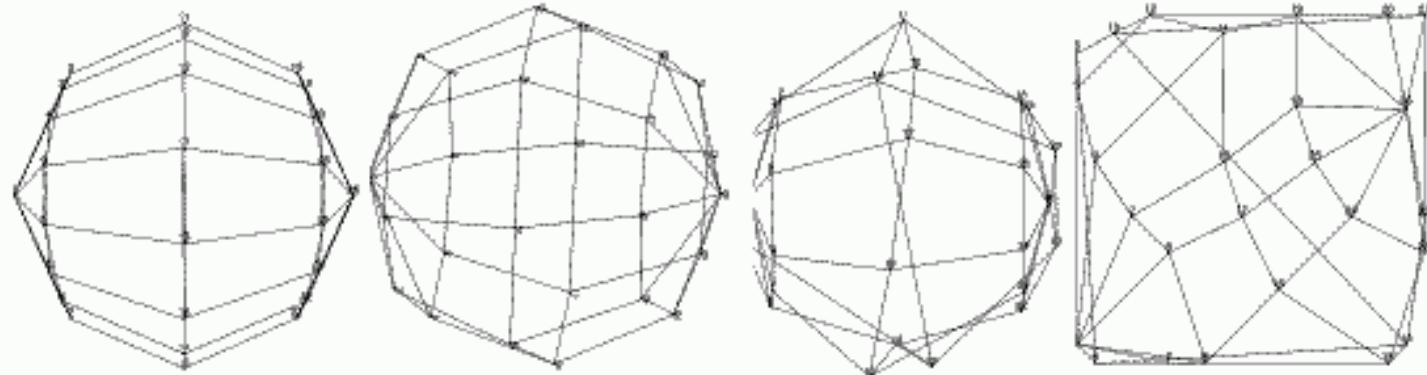
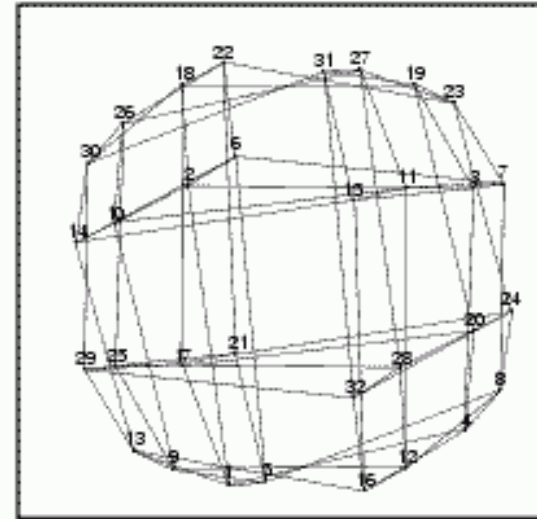
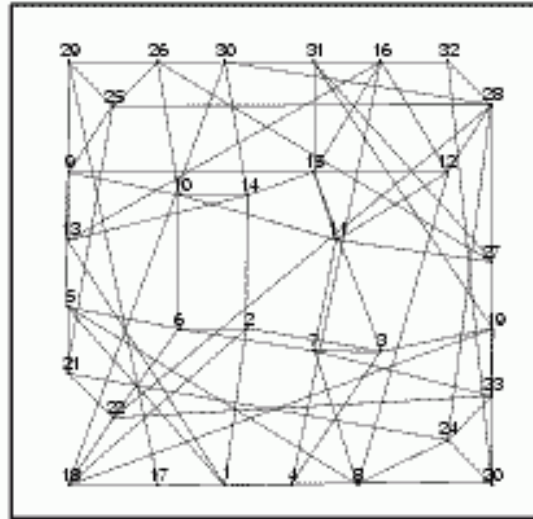
Współczynnik alienacji (Guttman-Lingoes) – podobny do błędu względnego, ale trudniejszy do zminimalizowania.

$$E_a(\mathbf{r}) = \sum_{i>j}^n (1 - R_{ij} / r_{ij})^2 \geq 0$$

Hipersześciiany

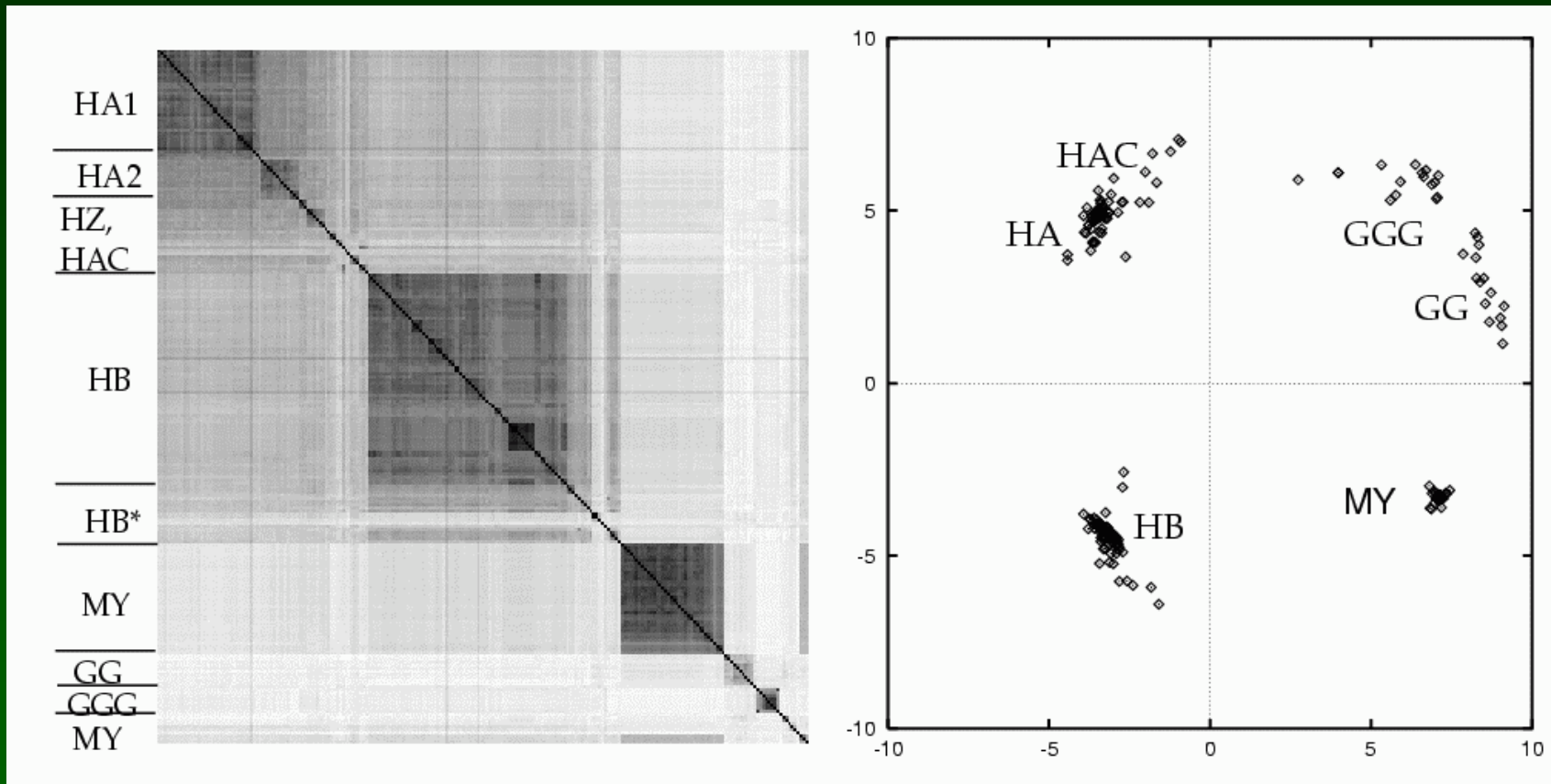


Hipersześciany 5D + sfery w 3D



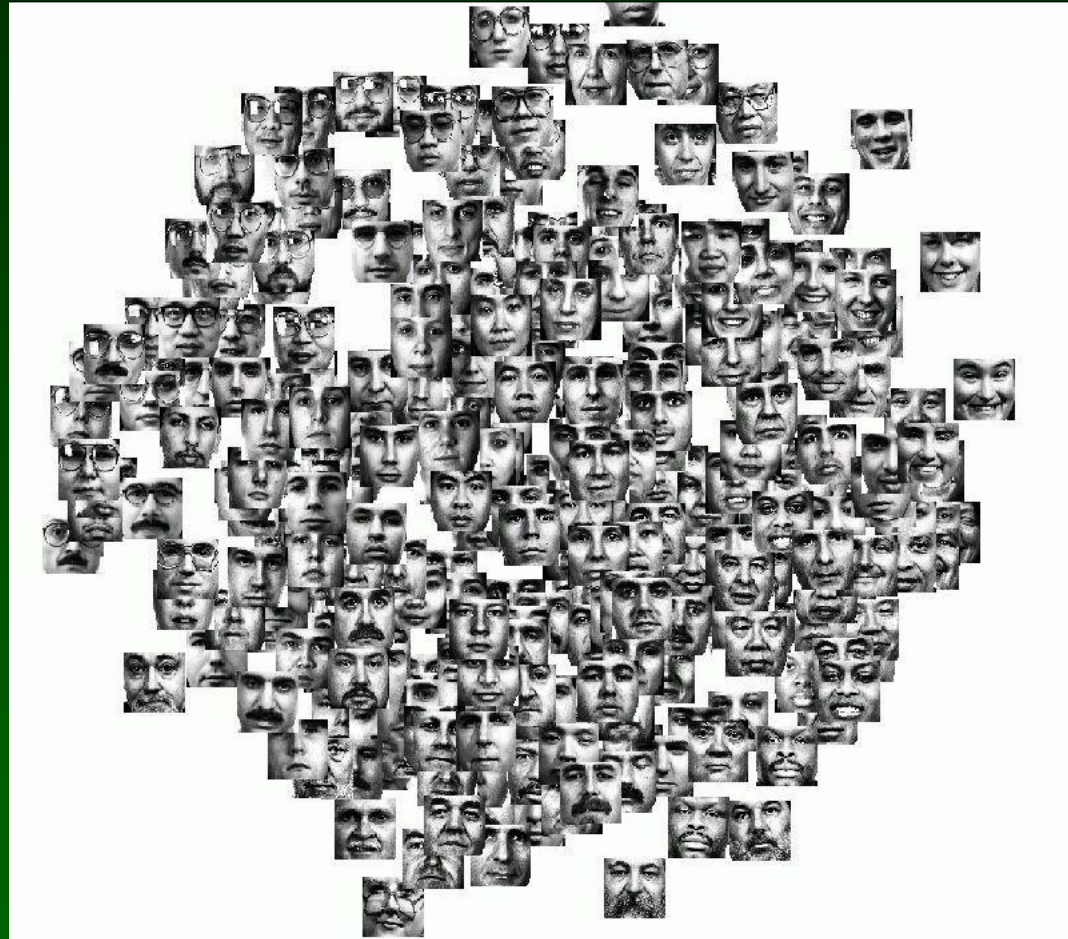
The two-dimensional representations of the 26 points on the sphere obtained by minimization of *S*, *E*, *A*, and by **SOFM** (left to right) with a 20 x 20 neurons map.

Sekwencje białek (Globiny)



226 sekwencji białkowych z rodziny globin; macierz podobieństwa wykazuje wysokie wartości podobieństwa (ciemne plamy) w obrębie podgrup, a analiza MDS ukazuje strukturę skupisk danych (Klock & Buhmann 1997).

Podobieństwo twarzy



300 twarzy, wyznaczono macierz podobieństwa i zastosowano mapowanie Sammona (Klock & Buhmann 1997).

Umap / tSNE

Popularne algorytmy do wizualizacji wielowymiarowych danych, dostępne w Scikit

Zalety: szybsze niż MDS.

Jak działa UMAP: topologiczna analiza danych i kompleksy sympleksyjne.

Szczegółowe omówienie UMAP

UMAP explorer interaktywne przykłady, UMAP zoo.

Plotly, UMAP i tSNE w Pythonie.

TSNE: t-distributed Stochastic Neighbor Embedding.

Perplexity – parametr kontrolujący liczbę sąsiadów w procesie redukcji wymiarowości.

Wprowadzenie do tSNE, porównanie z PCA. tSNE w Scikit.

tSNE explorer, przykłady w bioinformatyce.

Więcej narzędzi do wizualizacji

Statgraphics charting tools

Modeling and Decision Support Tools collected at the University of Cambridge:

Decision Support tools – representation aids

<http://www.visual-literacy.org/pages/documents.htm>

Książka: T. Soukup, I. Davidson, Visual Data Mining-Techniques and Tools for Data Visualization and Mining. Wiley 2002

Więcej takich narzędzi:

<https://is.umk.pl/~duch/projects/CI.html#vis>

Drzewa decyzji

function DT(E : zbiór przykładów) **returns** drzewo;

$T' :=$ buduj_drzewo(E);

$T :=$ obetnij_drzewo(T');

return T ;

function buduj_drzewo(E : zbiór przykładów) **returns** drzewo;

$T :=$ generuj_tests(E);

$t :=$ najlepszy_test(T, E);

$P :=$ podział E indukowany przez t ;

if kryterium_stopu(E, P)

then return liść(info(E))

else

for all E_j in P : $t_j :=$ buduj_drzewo(E_j);

return węzeł($t, \{(j, t_j)\}$);

DT do klasyfikacji

Testy: podział pojedynczej cechy, lub kombinacji
Atrybut={wartość_i} lub Atrybut < wartość_i

Kryteria: maksymalizacja ilości informacji,
maksymalizacja liczby poprawnie podzielonych obiektów,
„czystość” węzła.

Przycinanie: usunąć gałęzie, które zawierają zbyt mało przypadków
prostsze drzewo może lepiej generalizować
ocenić optymalną złożoność na zbiorze walidacyjnym.

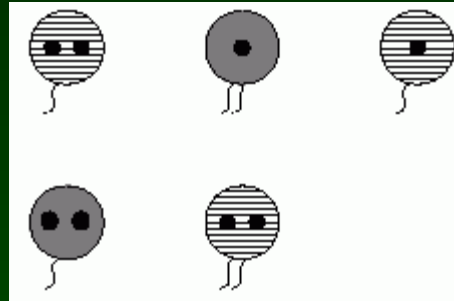
Kryterium stopu: osiągnięta dokładność podziałów,
zbyt złożone drzewo, za mało przypadków do podziału.

Popularne systemy: C4.5, CART – szybkie i często wystarczające.
SSV, Separability Split Criterion (K. Grąbczewski)

DT - przykład

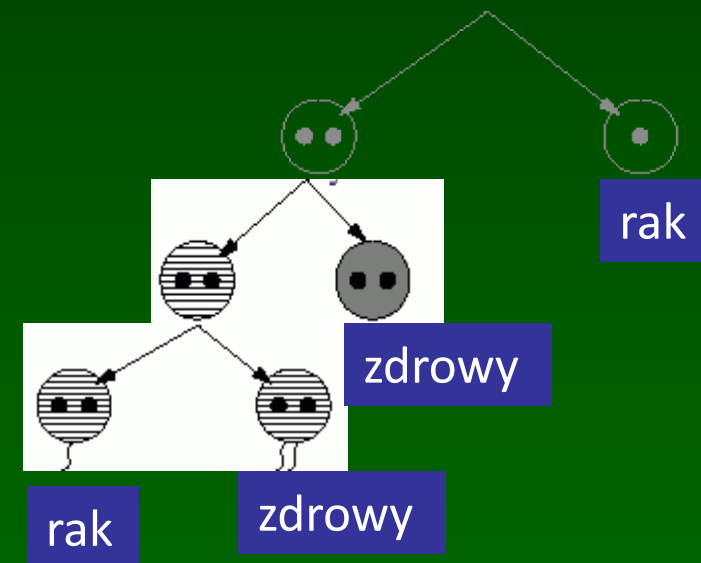
Klasy: {rakowe,

zdrowe}



Cechy: ciało komórki: {cienie, paski}

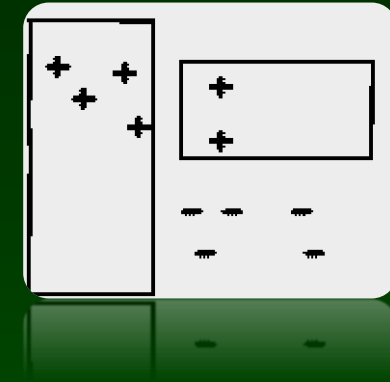
jądra: {1, 2}; ogonki: {1, 2}



Indukcja reguł

DT: tworzą reguły hierarchiczne

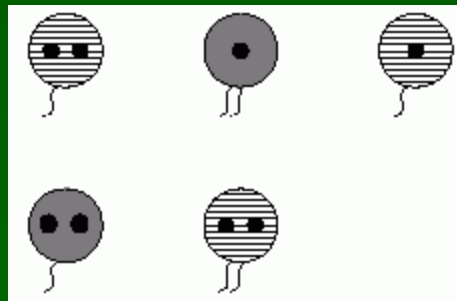
Utwórz reguły próbując pokryć obszar przestrzeni, w którym znajdują się wektory określonej klasy.



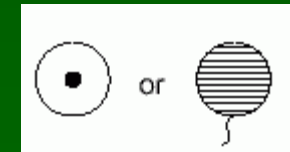
Zalety: łatwo zrozumiałe, dobra reprezentacja hipotez.

Wady: powolne działanie, algorytmy pokrycia NP-trudne

Przykłady: CN2, AQ-nn, GREEDY ...



Abstrahowanie od
szczegółów => rakowe
jeśli:



Ocena podobieństwa (similarity-based)

Pamiętaj przykłady i oceniaj podobieństwo; stosuj algorytmy „leniwe”, czyli oceniające sytuację dopiero wtedy, gdy znajdzie potrzeba.

Zalety: brak fazy uczenia, dobra dokładność jeśli dużo danych.

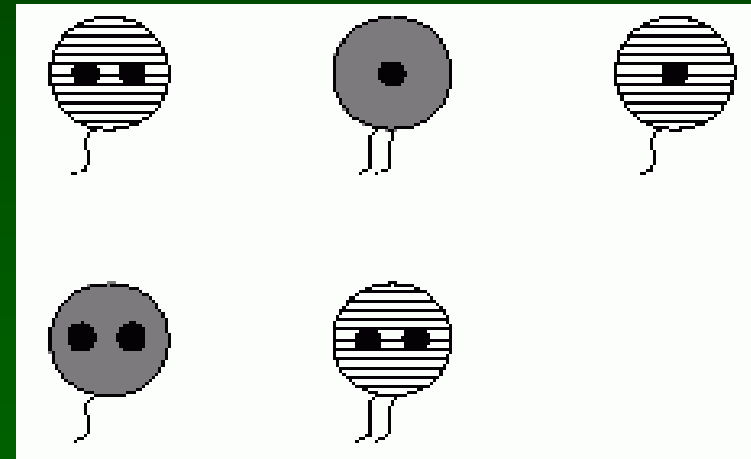
Wady: mogą wymagać dużo pamięci, konieczna selekcja atrybutów, konieczne dobranie odpowiednich funkcji podobieństwa, trudno zrozumiałe wyniki.

Przykłady: kNN, IBn, MBR ...

Możliwości wyboru: cech, prototypów,

liczby uwzględnianych prototypów,

oceny wkładu prototypów ...



Sieci neuronowe

Inspiracja neurobiologiczna: jak robią to mózgi?

Używa:

Neuronów - prostych elementów przetwarzających sygnały.

Synaps - parametrów adaptacyjnych związanych z połączeniami określającymi siłę pobudzeń.

Wyrafinowanych algorytmów korekcji parametrów (uczenia).

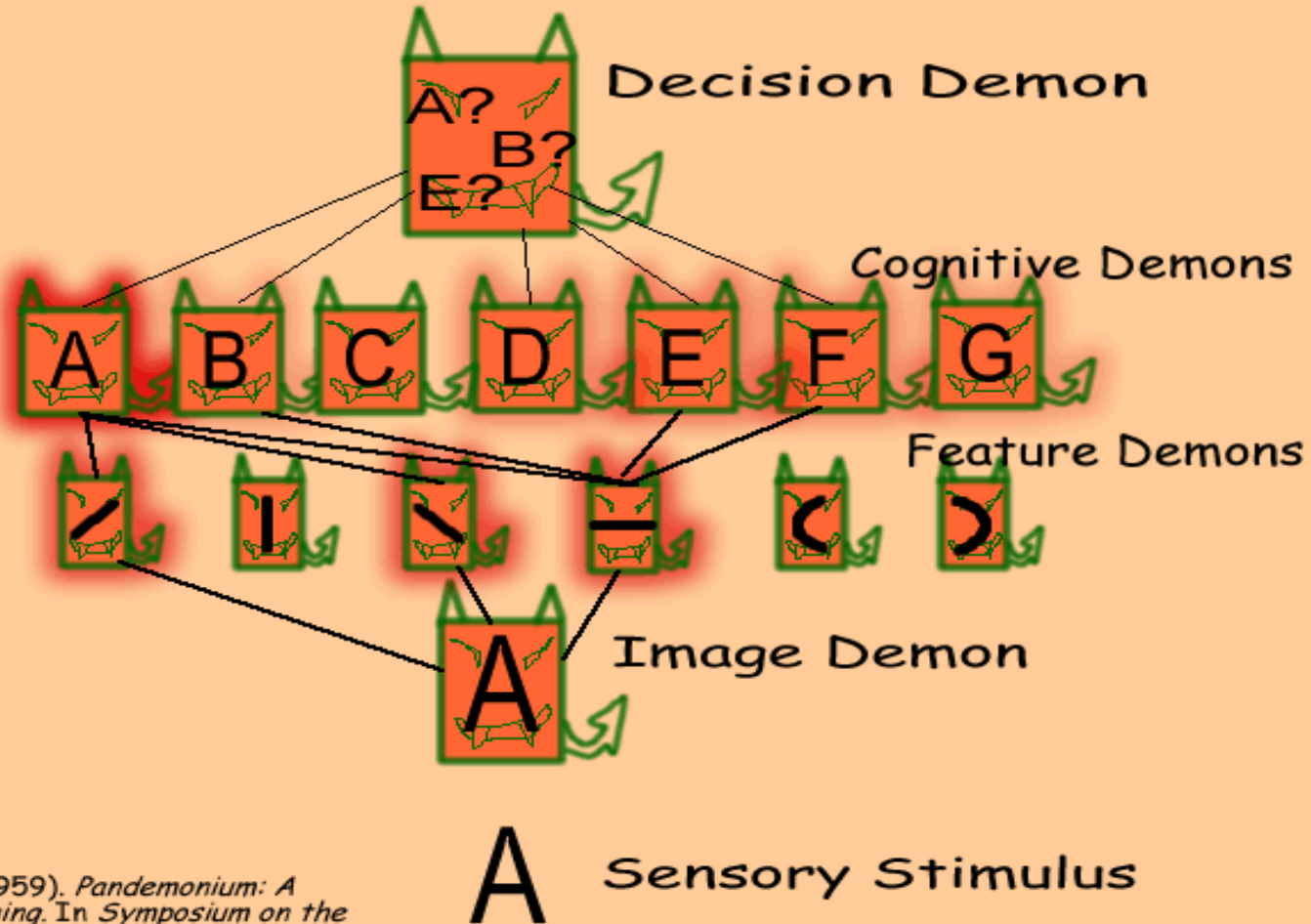
Funkcji kosztu/oceny jakości działania.

Wiele modeli: perceptrony wielowarstwowe (MLP), sieci RBF, samoorganizujące się sieci SOM ...

Zalety: uniwersalne, dużo symulatorów, odporne na szum w danych, wiele zastosowań i wariantów.

Wady: niektóre modele wolno się uczą, wymaga wielu danych, uczenie nie zawsze się kończy sukcesem, mają wiele parametrów, interpretacja jest trudna, mogą istnieć prostsze modele.

Selfridge's Model



Based on:

Selfridge, O. G. (1959). *Pandemonium: A paradigm for learning*. In *Symposium on the mechanization of thought processes* (pp. 513-526). London: HM Stationery Office.

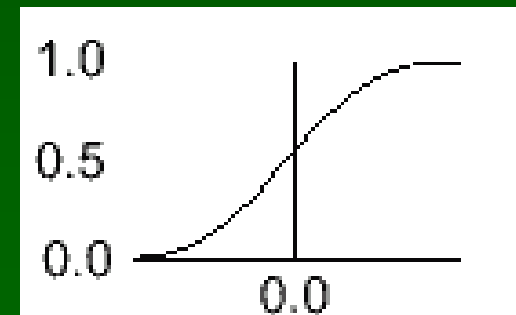
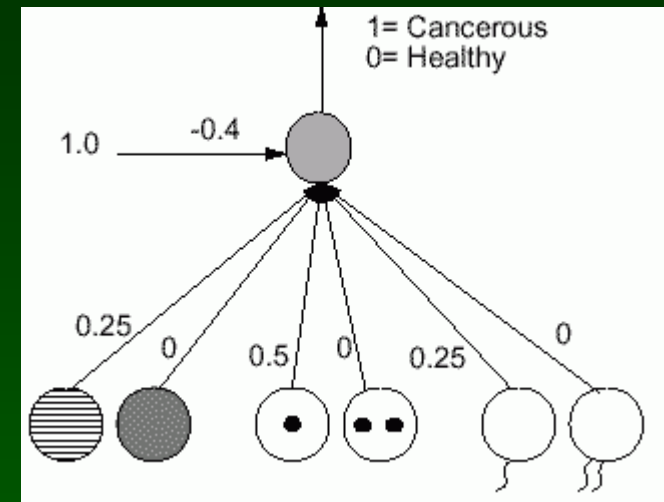
Sieci neuronowe - przykład

Wejścia: dane ciągłe lub dyskretne.
Neuron sumuje wartości sygnałów wejściowych określając pobudzenie.

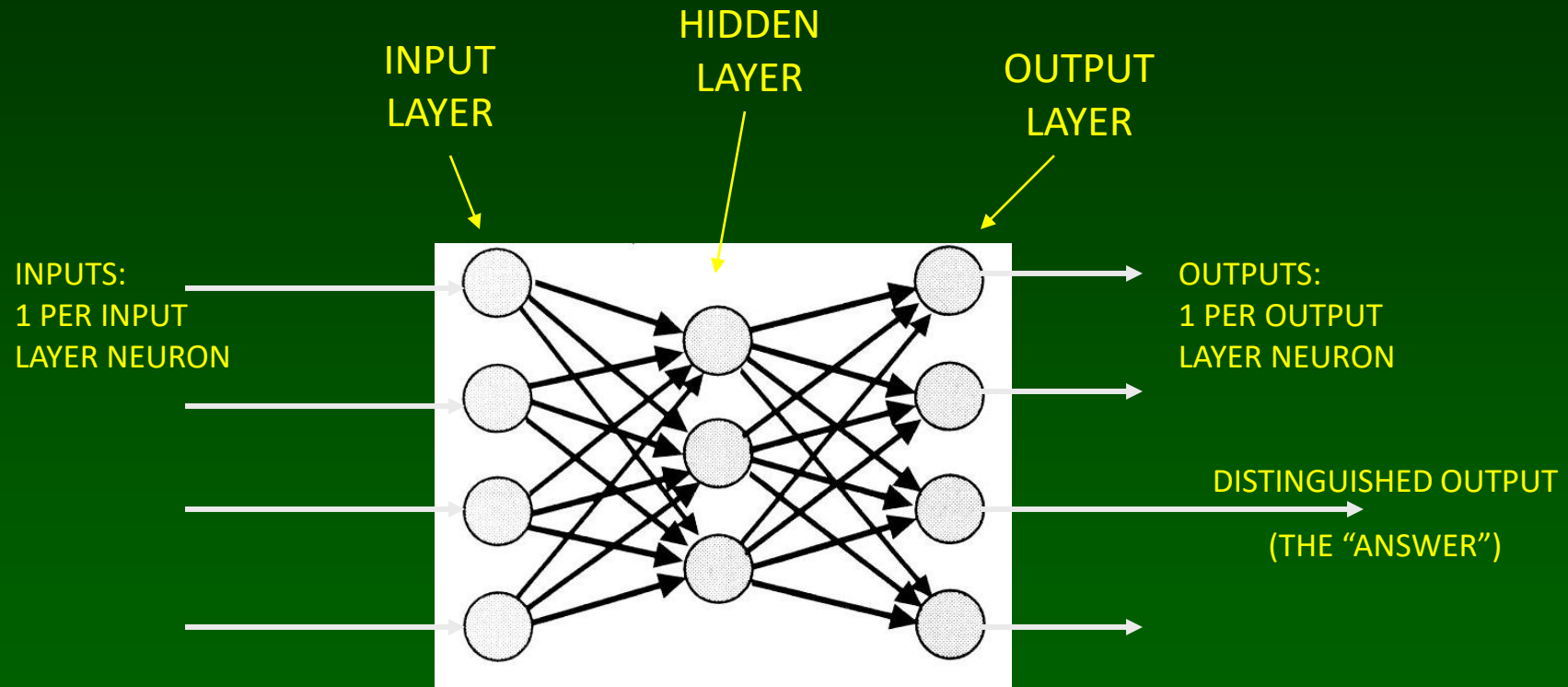
$$f\left(\sum_i W_i X_i - \theta\right)$$

$f(\cdot)$ - funkcja schodkowa lub sigmoidalna
 q - prób działania neuronu.

W sieciach neuronowych wiele takich neuronów połączonych jest ze sobą realizując dowolnie skomplikowane funkcje.



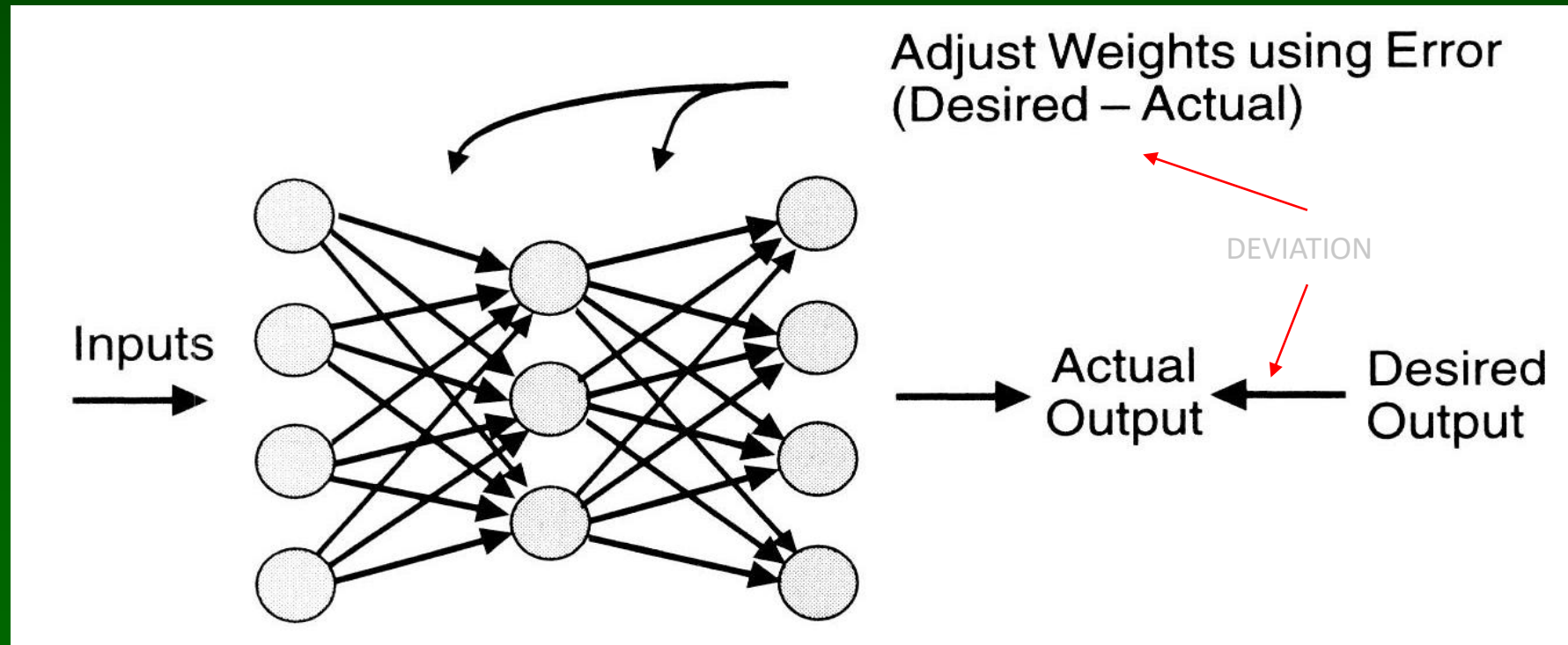
Sieć neuronowa



Wsteczna propagacja błędów

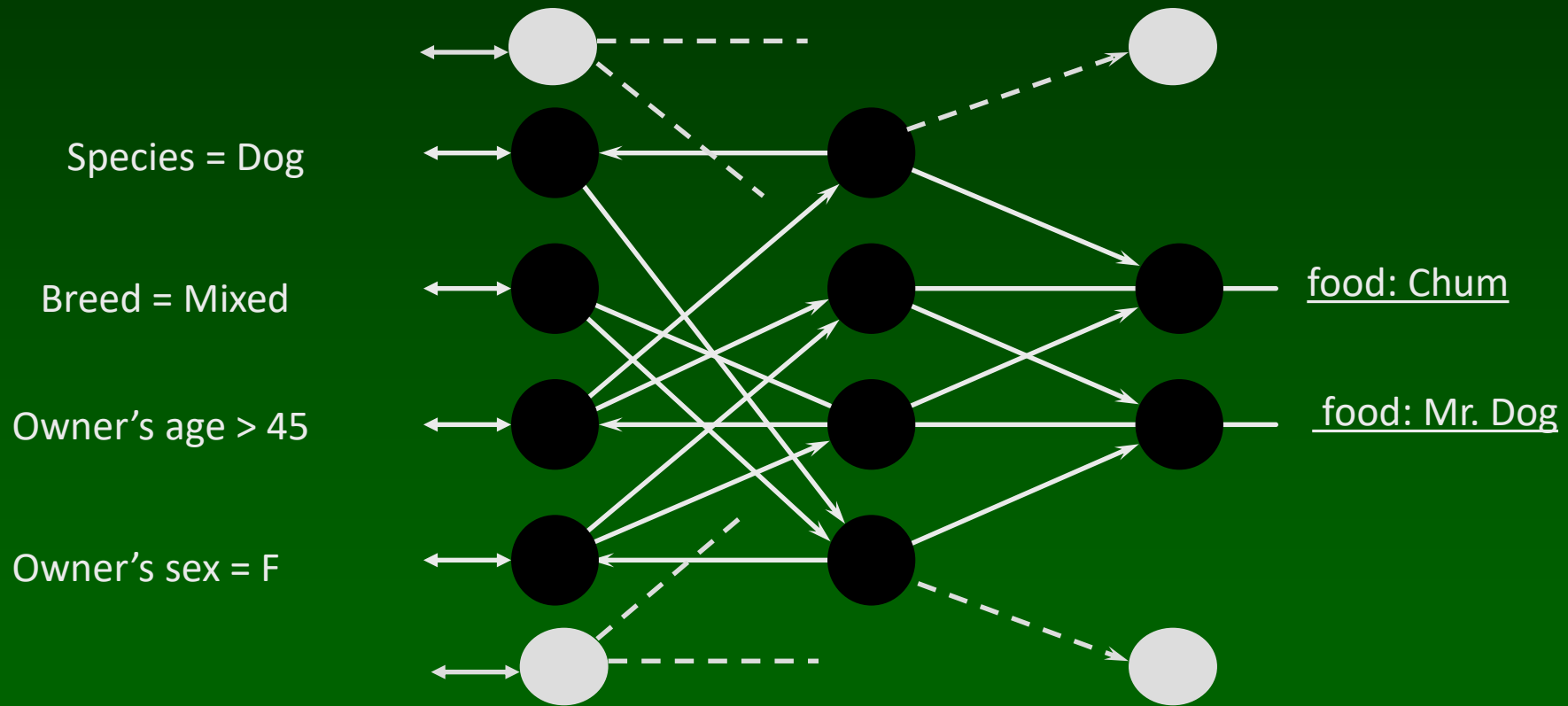
Algorytm rozpowszechniony od 1986 roku. Sieć ma wiele warstw elementów, zwanych neuronami.

1. Wiele przykładów (wektorów) na wejściu, znany wynik.
2. Różnica oczekiwanego wyniku od otrzymanego jest podstawą do korekty parametrów (wag).
3. Po nauczaniu, zwykle minimalizacją gradientu funkcji błęd, można ją stosować do nowych danych.
4. Uczenie nowych danych może popsuć działanie na wcześniejszych – katastroficzne zapominanie.

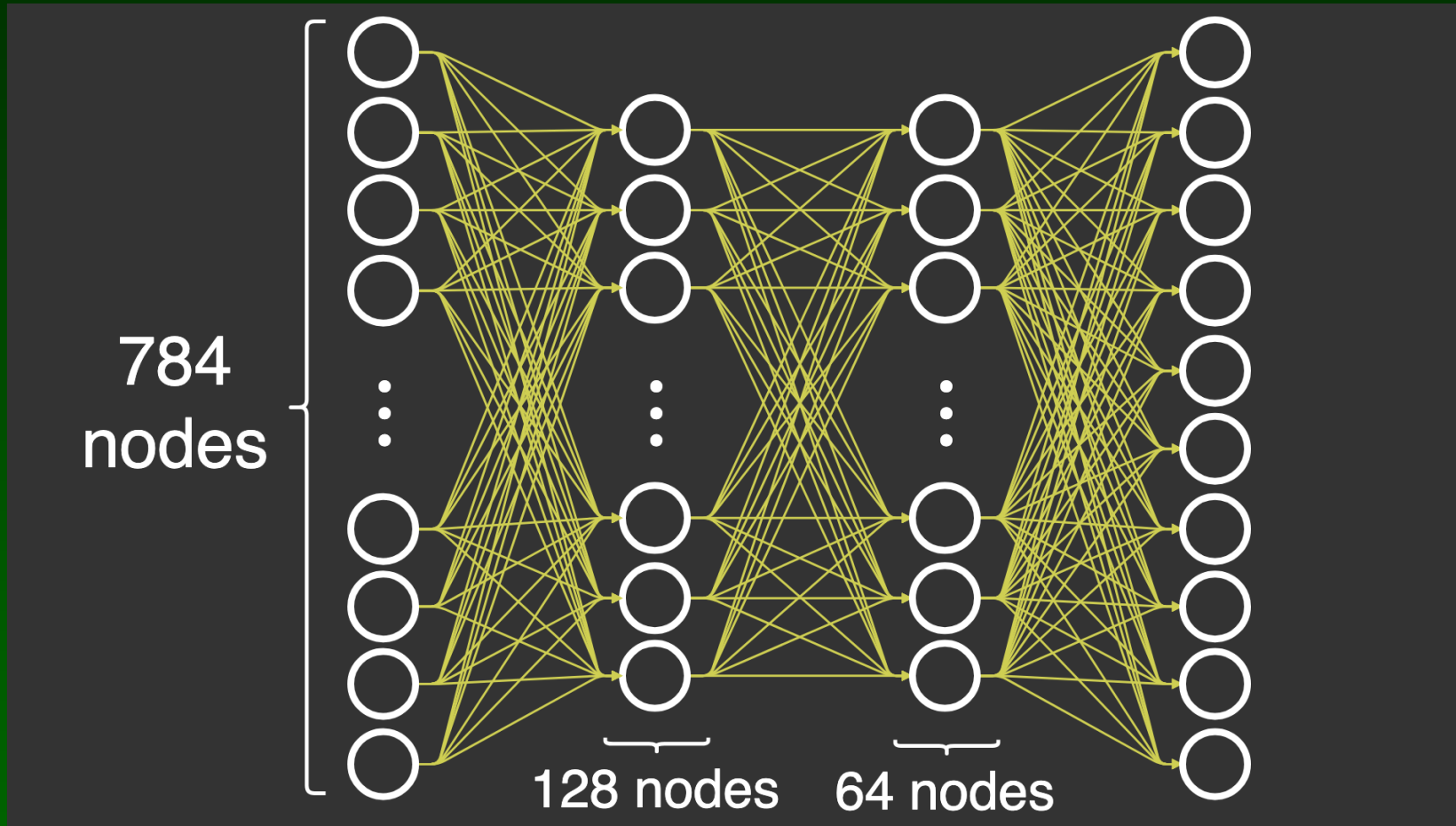


Klasyfikacja za pomocą sieci neuronowych

“Jakie czynniki wpływają na wybór ulubionej karmy psów?”



Rozpoznawanie cyfr



IBM developer: [Neural networks from scratch](#) in Python
Obrazy 28x28, wyjście 0, 1 ... 9.

Przykład: Grzyby



Baza danych dla grzybów:

Dane z podręcznika opisującego różne gatunki grzybów.

22 cechy i 3 klasy: jadalne, trujące, nie polecane.

4208 (51.8%) jadalnych, 3916 (48.2%) niejadalnych.

Cechy (nazwy oryginalne): 118 binarnych wartości.

cap shape (6, e.g.. bell, conical,flat...), cap surface (4), cap color (10), bruises (2), odor (9), gill attachment (4), gill spacing (3), gill size (2), gill color (12), stalk shape (2), stalk root (7, many missing values), surface above the ring (4), surface below the ring (4), color above the ring (9), color below the ring (9), veil type (2), veil color (4), ring number (3), spore print color (9), population (6), habitat (7).

Zadanie: zidentyfikować grzyby jadalne, znaleźć ważne cechy.

W książce wyraźnie napisano, że nie ma prostej reguły ...

Grzyby – przykładowe dane



Przykładowe dane:

M-1: edible,convex,fibrous,yellow,bruises,anise,free,crowded, narrow, brown, tapering,bulbous,smooth,smooth,white,white, partial,white, one, pendant,purple,several, woods

M-2: edible,flat,smooth,white,bruises,almond,free,crowded, narrow,pink,tapering,bulbous,smooth,smooth,white,white, partial,white,one,pendant,purple,several,woods

.....

M-8124: poisonous,convex,smooth,white,bruises,pungent,free,close, narrow, white, enlarging, equal, smooth, smooth, white, white, partial, white, one,pendant, black, scattered, urban

Grzyby - reguły



Grzyb jest jadalny jeśli:

$\text{odor} = (\text{almond} \vee \text{anise} \vee \text{none}) \wedge \text{spore_print_color} = \text{not.green}$

48 błędów, 99.41% prawidłowych

Grzyb nie jest jadalny jeśli: (używamy tylko 6 cech):

$R_1) \text{odor} = \text{not}(\text{almond} \vee \text{anise} \vee \text{none})$ 120 bł., 98.52%

$R_2) \text{spore_print_color} = \text{green}$ 48 bł., 99.41%

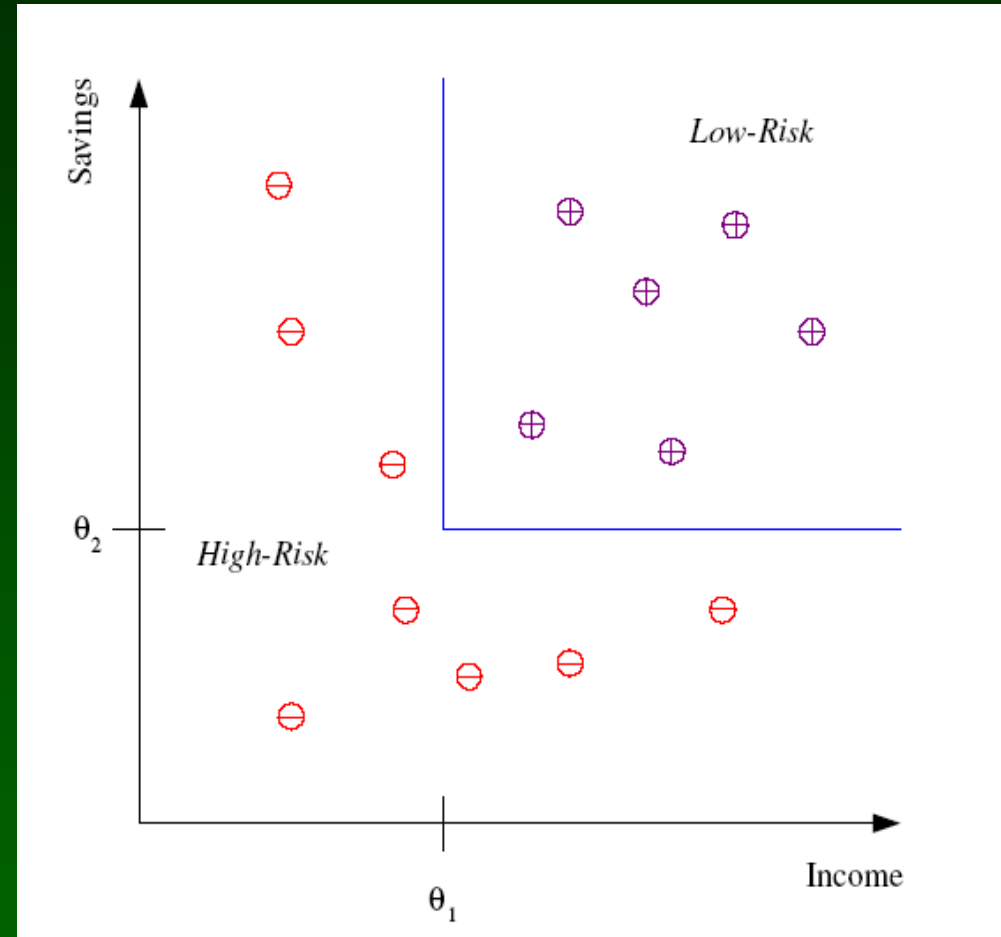
$R_3) \text{odor} = \text{none} \wedge \text{stalk_surface_below_ring} = \text{scaly} \wedge \text{stalk_color_above_ring} = \text{not.brown}$
8 bł., 99.90%

$R_4) \text{IF habitat} = \text{leaves} \wedge \text{cap_color} = \text{white}$ 0 błędów!

Teraz widać, dlaczego węch jest tak ważny dla zwierząt.

Klasyfikacja

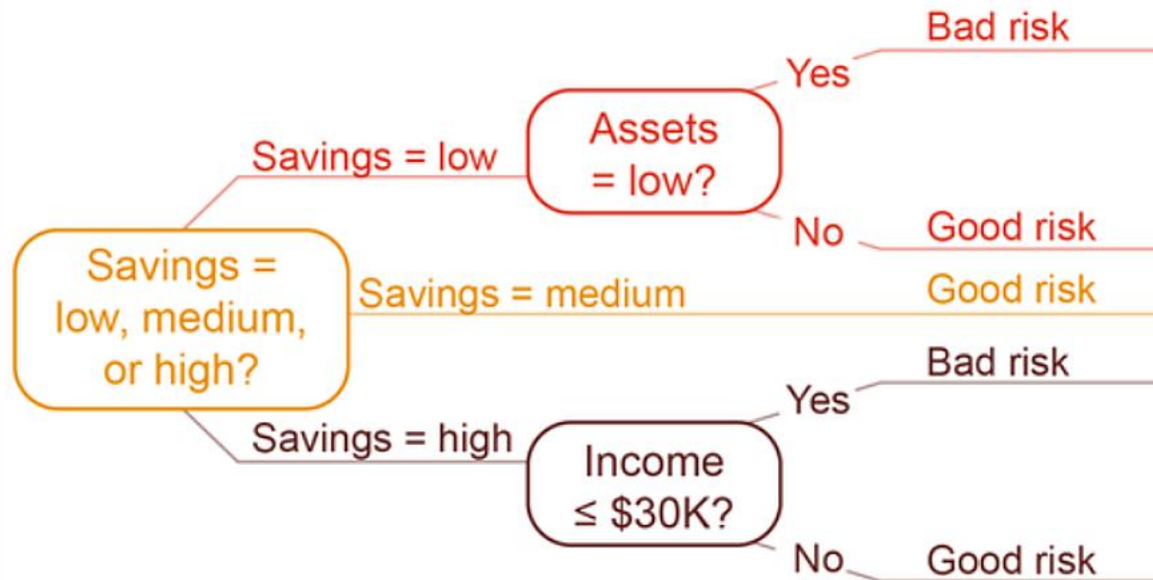
- Przykład: oceny zdolności kredytowej
- Rozróżnijmy klientów **low-risk** i **high-risk** na podstawie *income* i *savings*



Metody: Drzewa decyzji

Decision trees

Decision tree analysis typically uses a hierarchy of variables or decision nodes that, when answered step by step, can classify a given customer as creditworthy or not, for example.



Advantages

Decision trees are useful when evaluating lists of distinct features, qualities, or characteristics of people, places, or things.

Use cases

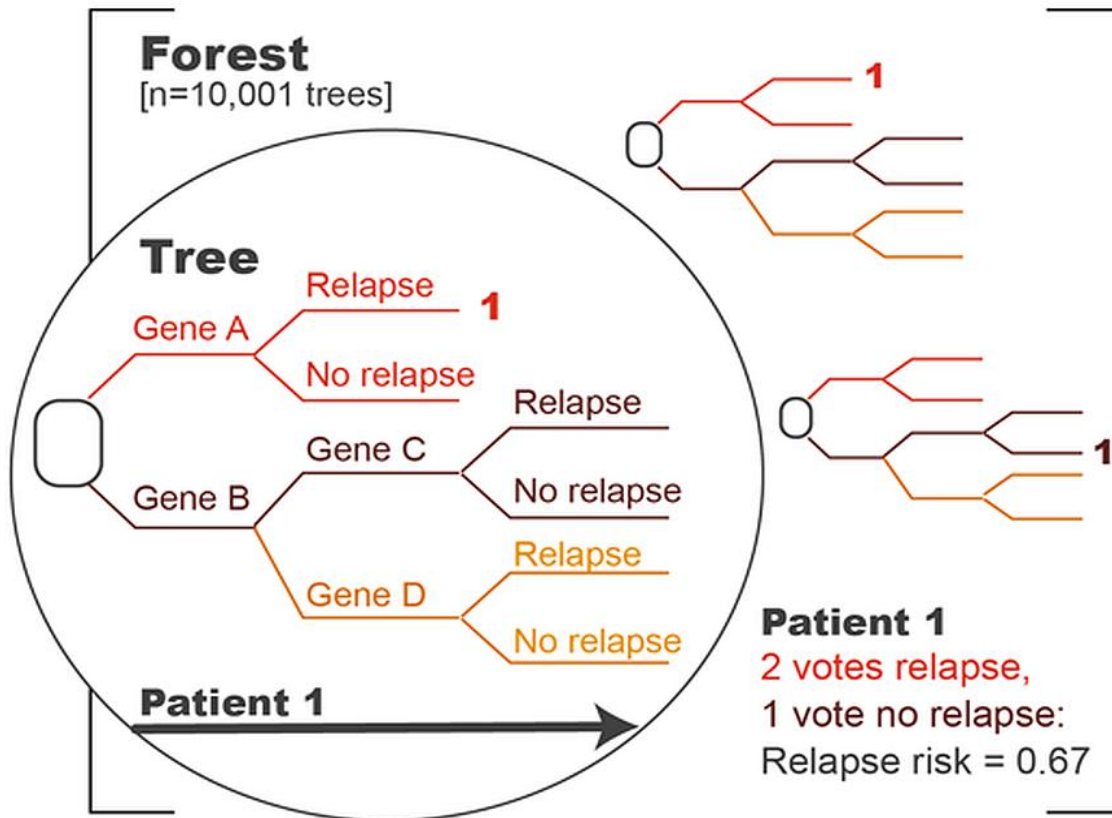
Rule-based credit risk assessment, horse race performance prediction

Source: Daniel T. Larose and Chantal D. Larose, *Data Mining and Predictive Analytics*, 2nd Edition, John Wiley & Sons, 2015

Methody: Random Forest

Random forest

Random forest algorithms improve the accuracy of decision trees by using multiple trees with randomly selected subsets of data. This example reviews the expression levels of various genes associated with breast cancer relapse and computes a relapse risk.



Advantages

Random forest methods prove useful with large data sets and items that have numerous and sometimes irrelevant features.

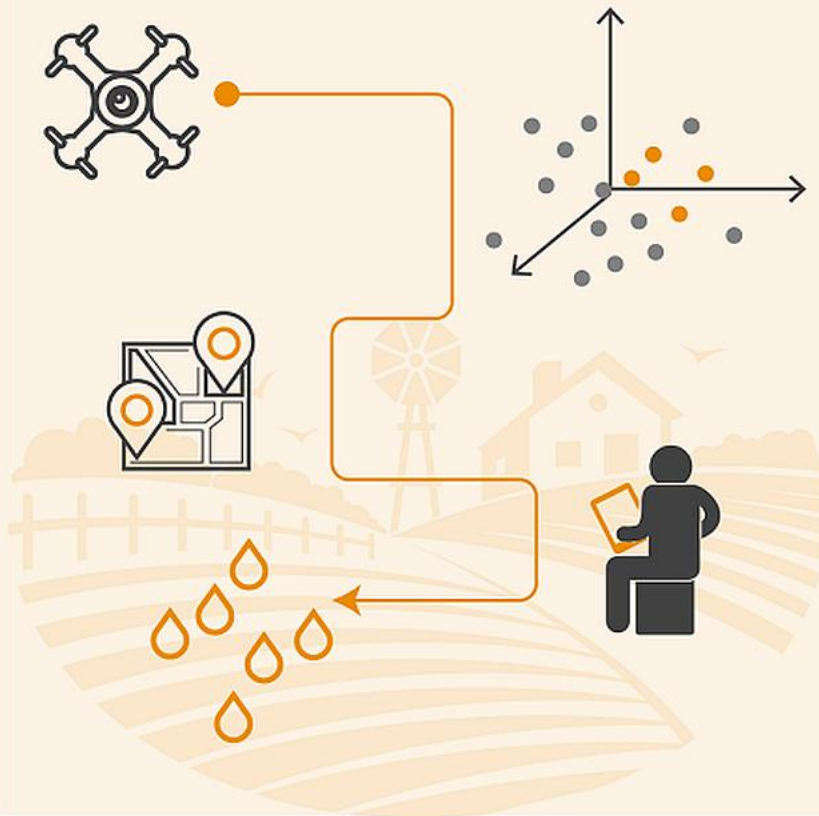
Use cases

Customer churn analysis, risk assessment

DM & Uczenie maszynowe

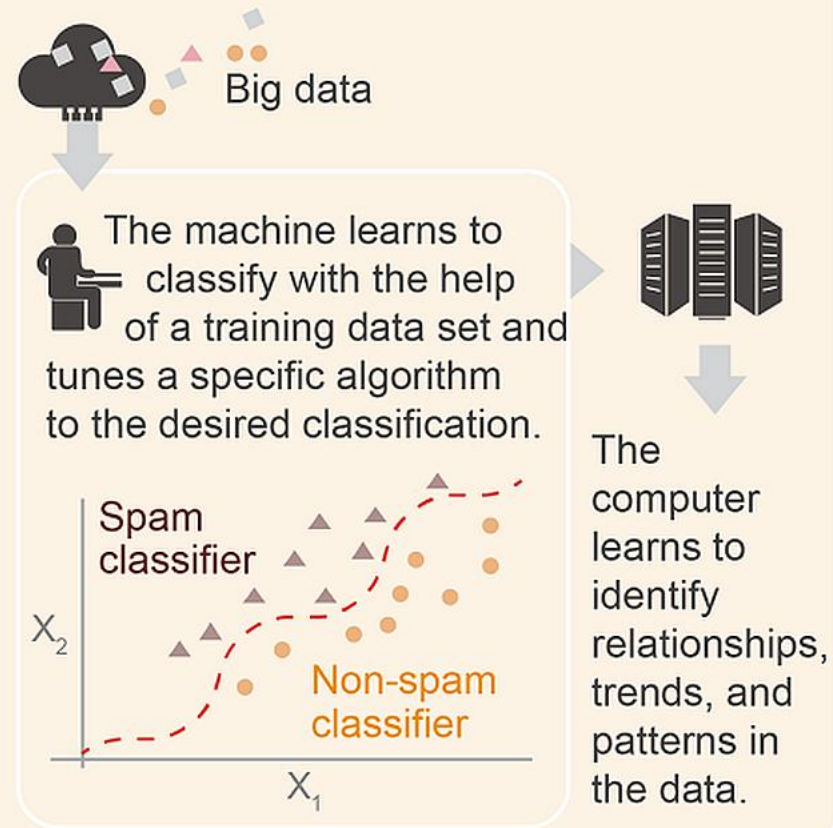
4 Intelligent apps

Intelligent apps leverage the outputs of AI, as in this precision farming example that uses drone-based data collection.



3 Machine learning

A data scientist uses a training data set to teach the computer what to do, and the system carries out the tasks.



Zastosowania ML

Here are just a few of the many ways we've put machine learning to work. How will your company use it?



Rapid 3D mapping and modeling

For a railway bridge reconstruction, PwC data scientists and domain experts applied machine learning to data captured from drones. The combination enabled precise monitoring and quick feedback on work in progress.



Enhanced profiling to mitigate risks

To detect insider trading, PwC combined machine learning with other analytic techniques to develop more comprehensive user profiles and gain deeper insight into complex suspicious behaviors.



Predicting the top performers

PwC used machine learning and other analysis to evaluate the potential of different horses running in the Melbourne Cup.



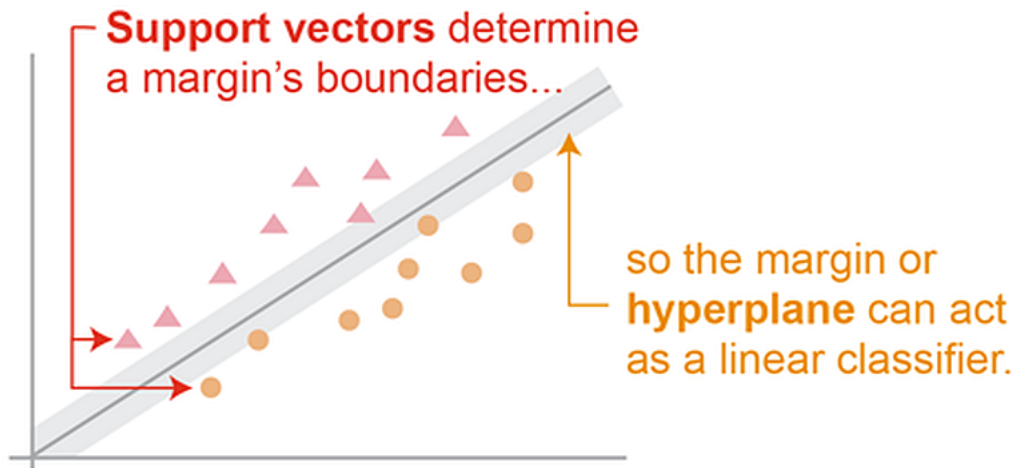
© 2016 PwC. All rights reserved. PwC refers to the US member firm or one of its subsidiaries or affiliates, and may sometimes refer to the PwC network. Each member firm is a separate legal entity. Please see www.pwc.com/structure for further details. This content is for general information purposes only, and should not be used as a substitute for consultation with professional advisors.

pwc.com/NextinTech

Metody: SVM

Support vector machines

Support vector machines classify groups of data with the help of hyperplanes.



Source: Matthew Kelly, *Computer Science: Source*, 2010

Advantages

Support vector machines are good for the binary classification of X versus other variables and are useful whether or not the relationship between variables is linear.

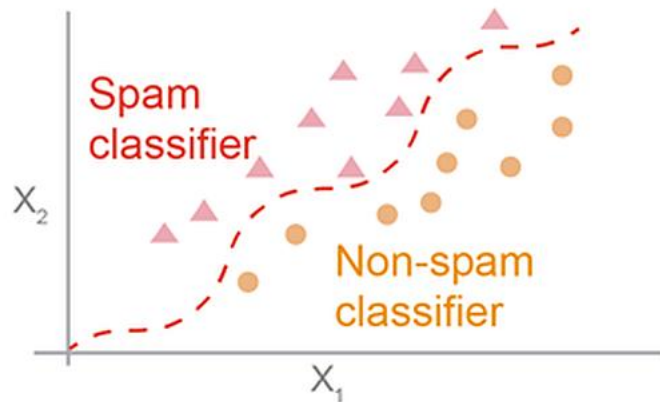
Use cases

News categorization, handwriting recognition

Metody: Regresja nieliniowa

Regression

Regression maps the behavior of a dependent variable relative to one or more independent variables. In this example, logistic regression separates spam from non-spam text.



Advantages

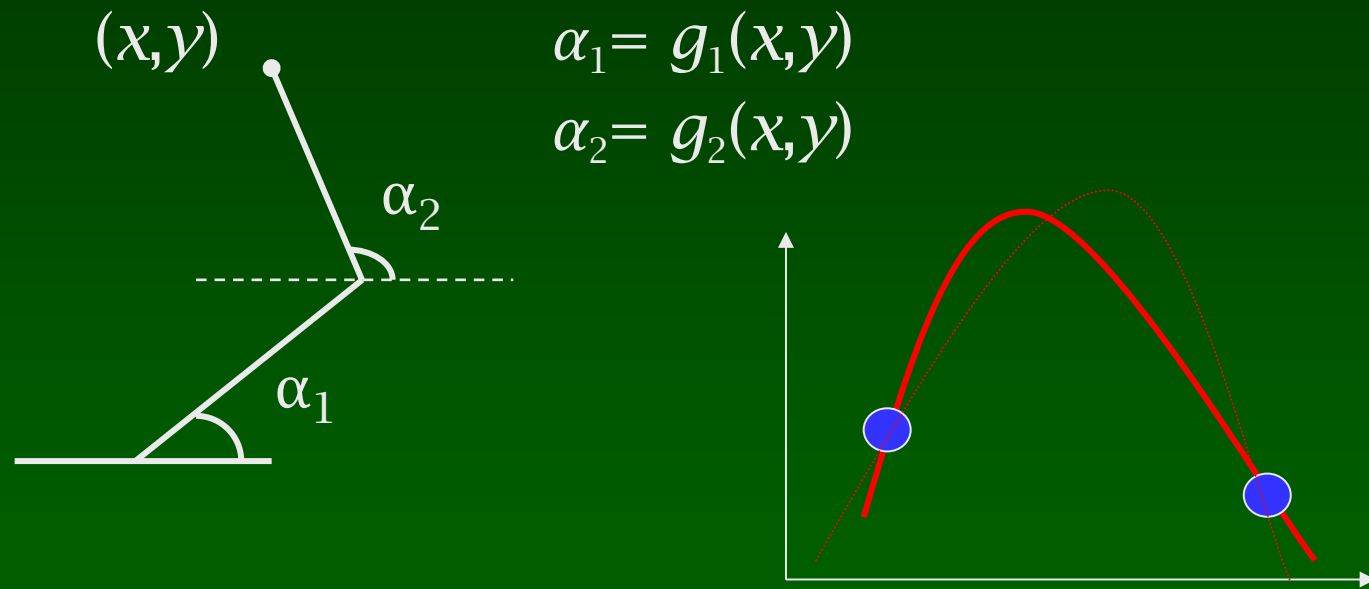
Regression is useful for identifying continuous (not necessarily distinct) relationships between variables.

Use cases

Traffic flow analysis, email filtering

Przykład regresji

- Prowadzenie samochodu: kąt obrotu kierownicy (CMU NavLab)
- Kinematyka ramienia robota

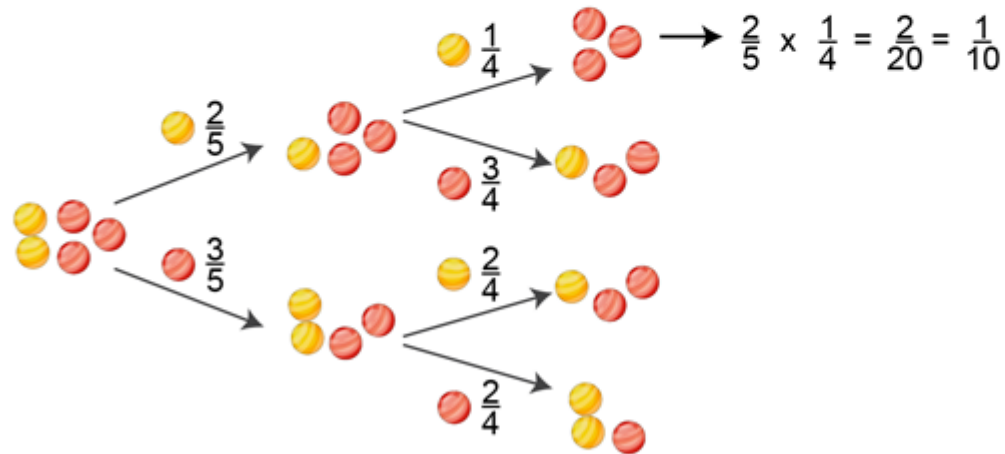


Sterowanie samochodami lub robotami wymaga obliczania parametrów w czasie rzeczywistym. Regresja liniowa jest powszechnie stosowana w naukach społecznych, jednak wyzwaniem jest przewidywanie złożonych, nieliniowych trajektorii z wieloma parametrami.

Metody: NB, Naive Bayes

Naive Bayes classification

Naive Bayes classifiers compute probabilities, given tree branches of possible conditions. Each individual feature is “naive” or conditionally independent of, and therefore does not influence, the others. For example, what’s the probability you would draw two yellow marbles in a row, given a jar of five yellow and red marbles total? The probability, following the topmost branch of two yellow in a row, is one in ten. Naive Bayes classifiers compute the combined, conditional probabilities of multiple attributes.



Source: Rod Pierce, et al., *MathIsFun*, 2014

Advantages

Naive Bayes methods allow the quick classification of relevant items in small data sets that have distinct features.

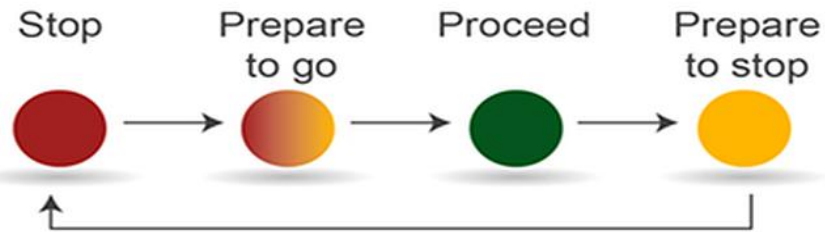
Use cases

Sentiment analysis, consumer segmentation

Metody: Ukryte modele Markowa

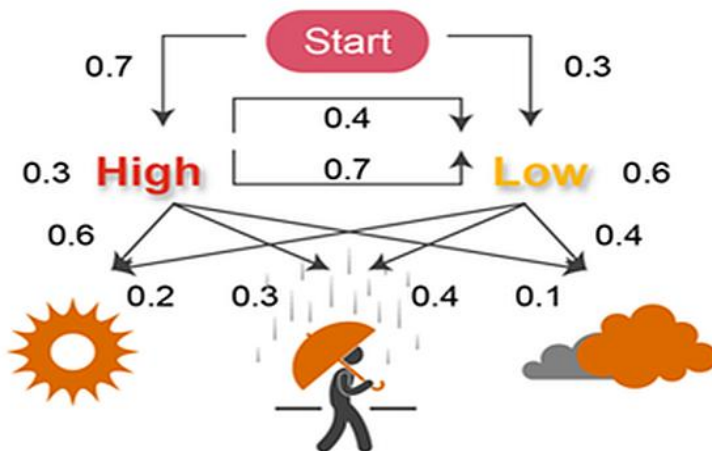
Hidden Markov models

Observable Markov processes are purely deterministic—one given state always follows another given state. Traffic light patterns are an example.



Source: Derek Kane, 2015

Hidden Markov models, by contrast, compute the probability of hidden states occurring by analyzing observable data, and then estimating the likely pattern of future observation with the help of the hidden state analysis. In this example, the probability of high or low pressure (the hidden state) is used to predict the likelihood of sunny, rainy, or cloudy weather.



Source: Leonardo Guizzetti, 2012

Advantages

Tolerates data variability and effective for recognition and prediction.

Use cases

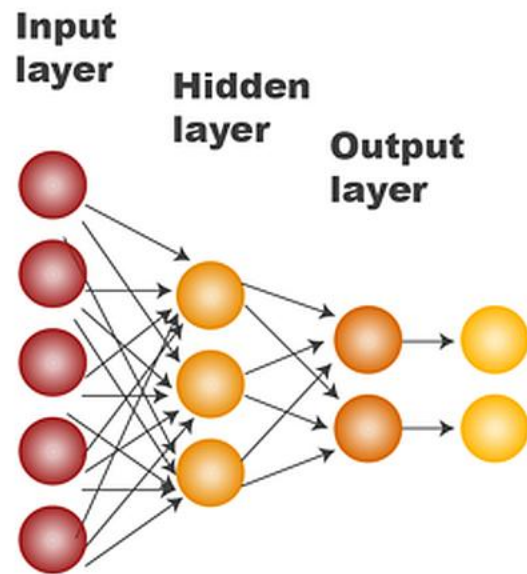
Facial expression analysis, weather prediction

Metody: Sieci neuronowe

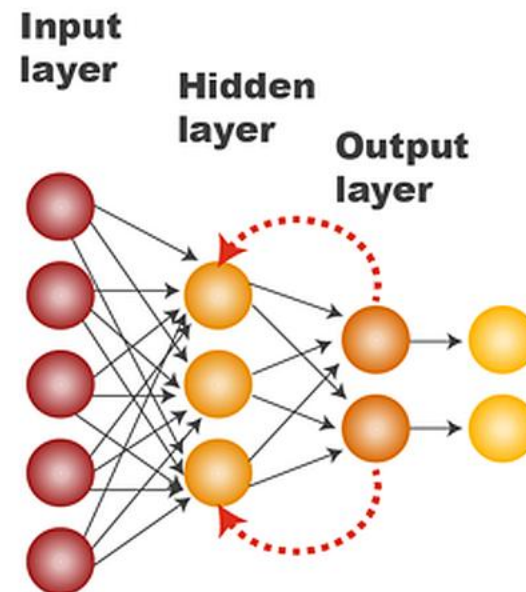
Recurrent neural networks

Each neuron in any neural network converts many inputs into single outputs via one or more hidden layers. Recurrent neural networks [RNNs] additionally pass values from step to step, making step-by-step learning possible. In other words, RNNs have a form of memory, allowing previous outputs to affect subsequent inputs.

Non-recurrent feed-forward neural network



Recurrent neural network—includes loops



Advantages

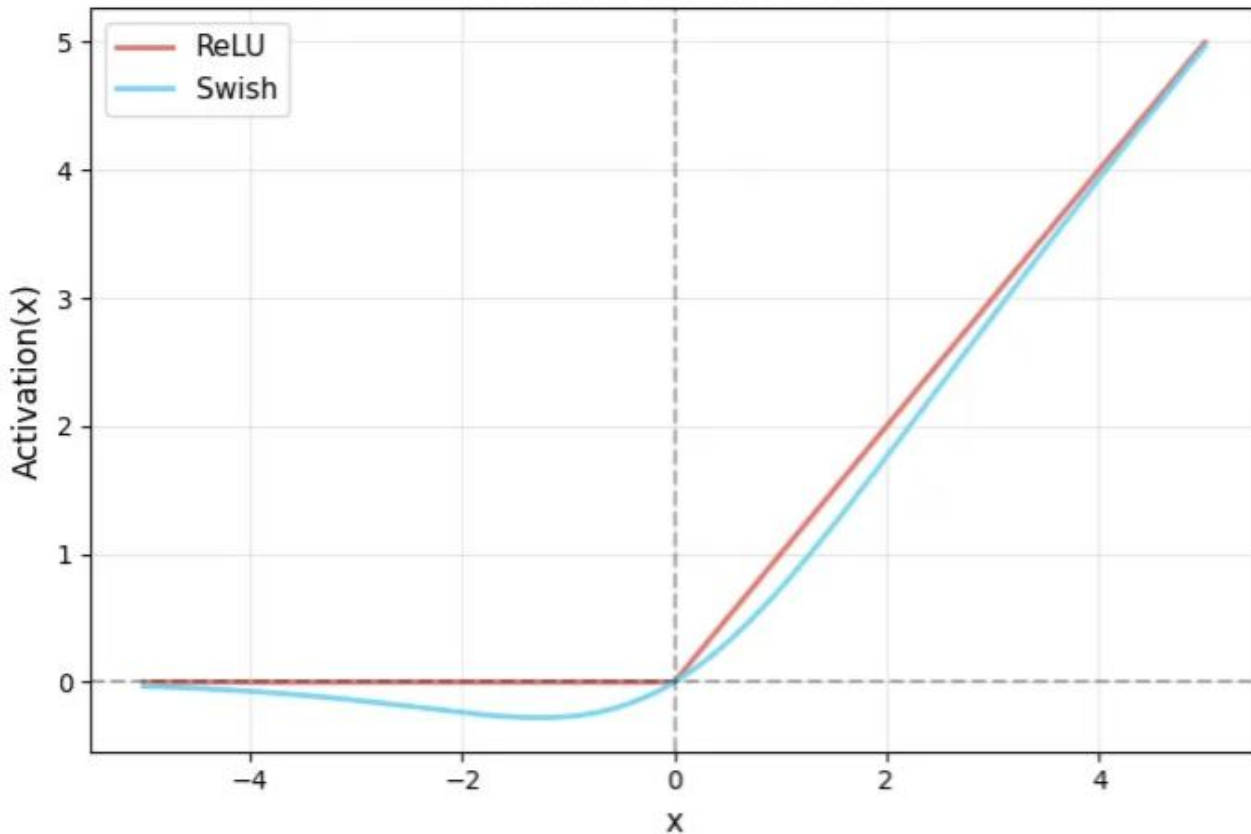
Recurrent neural networks have predictive power when used with large amounts of sequenced information.

Use cases

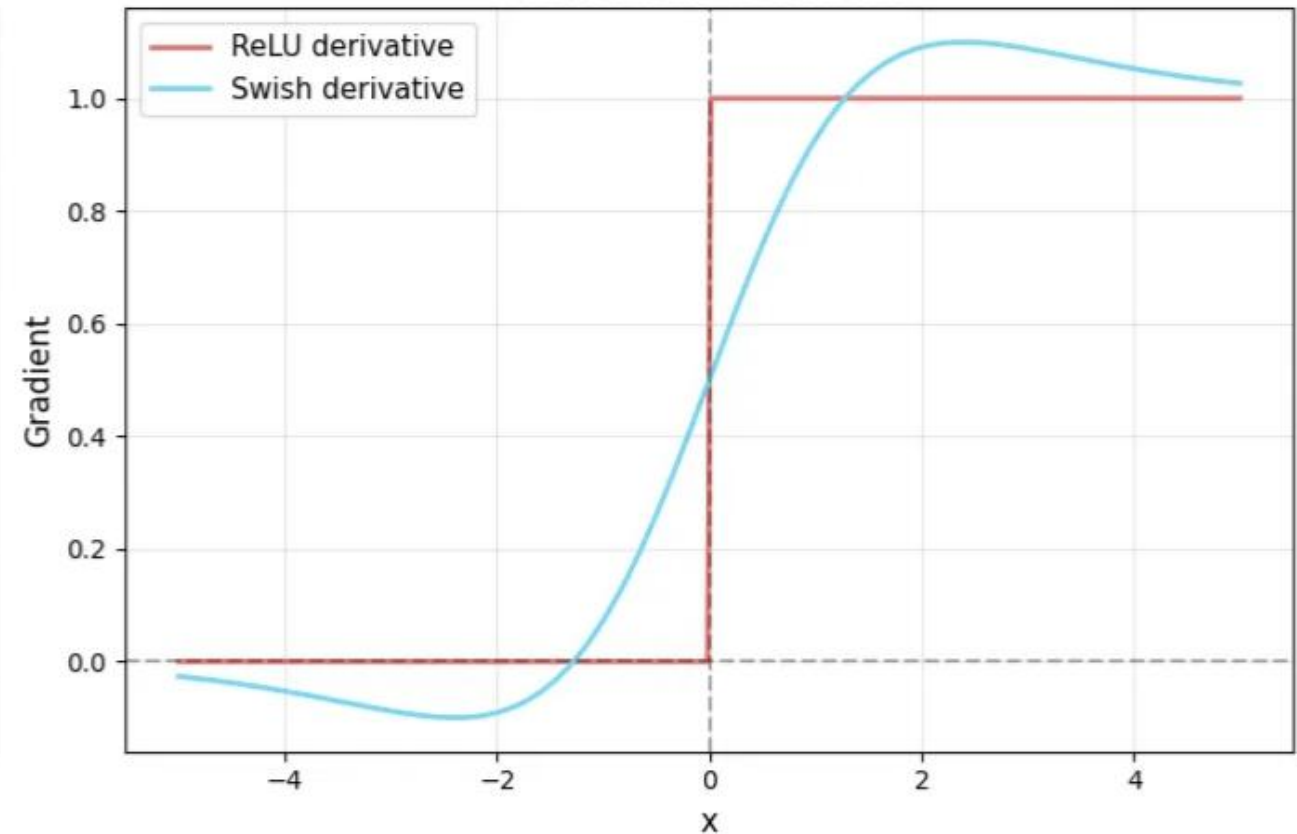
Image classification and captioning, political sentiment analysis

Funkcje w NN

Activation Functions

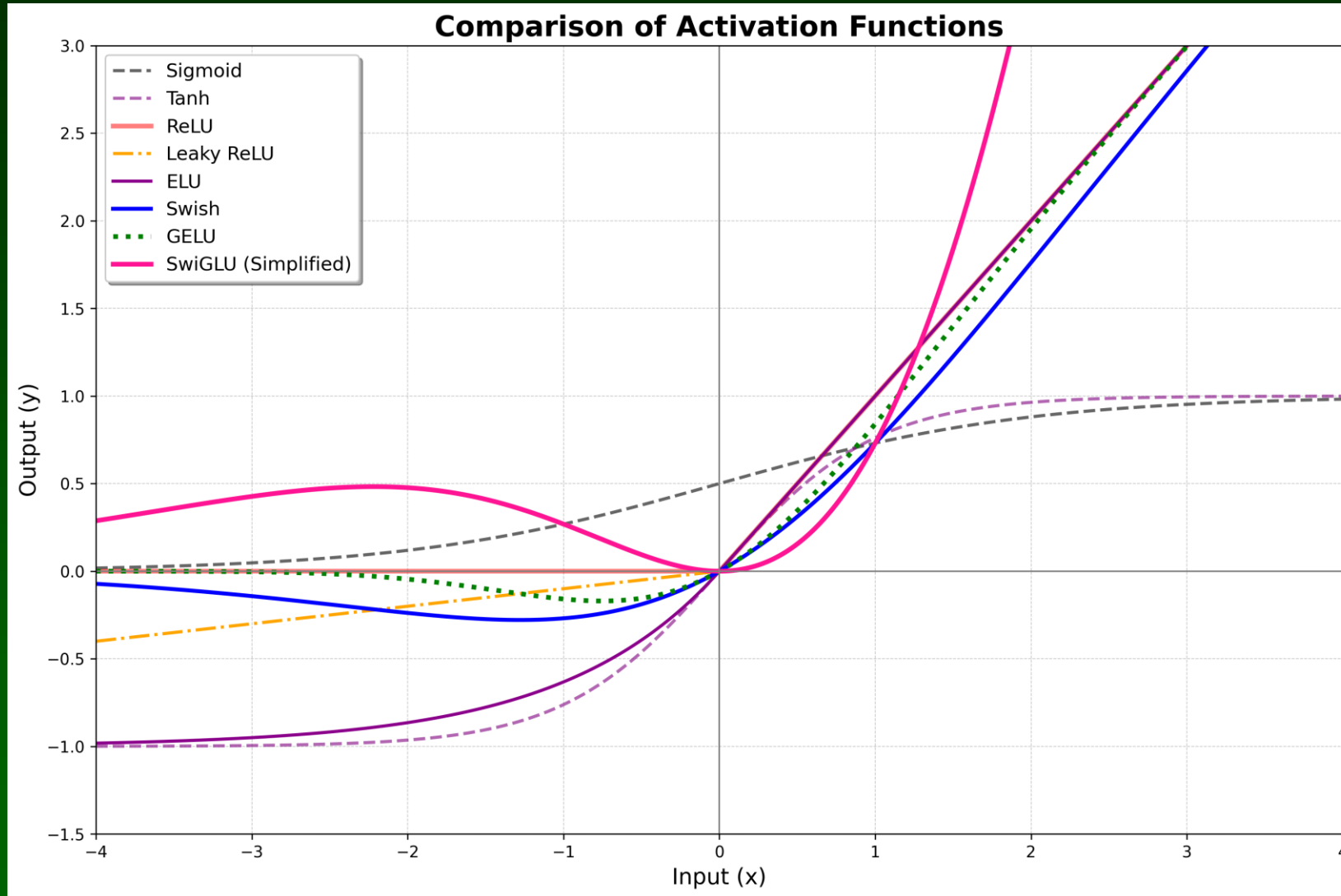


Activation Gradients

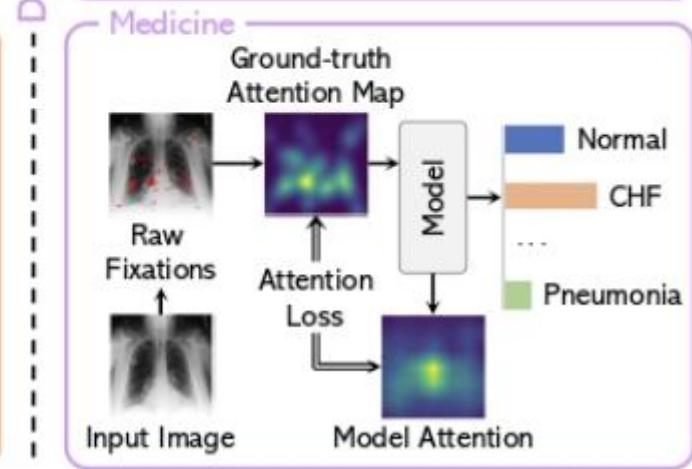
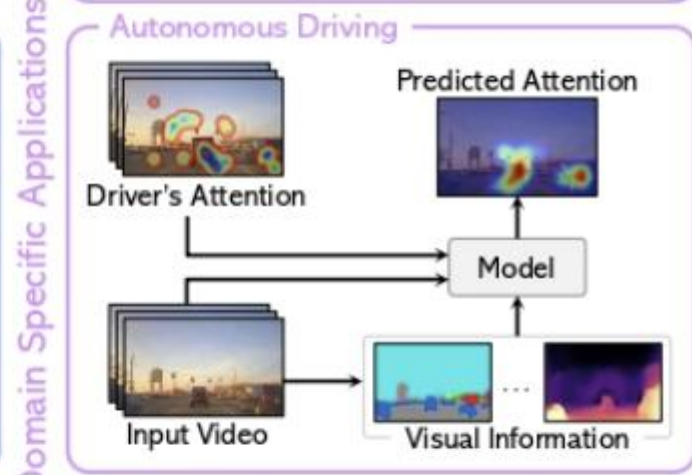
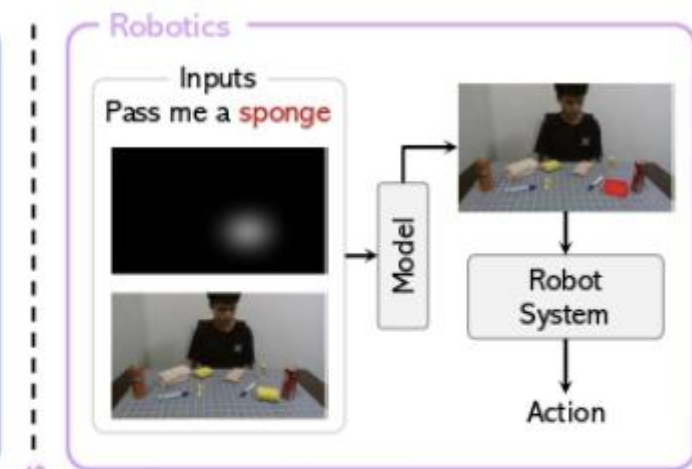
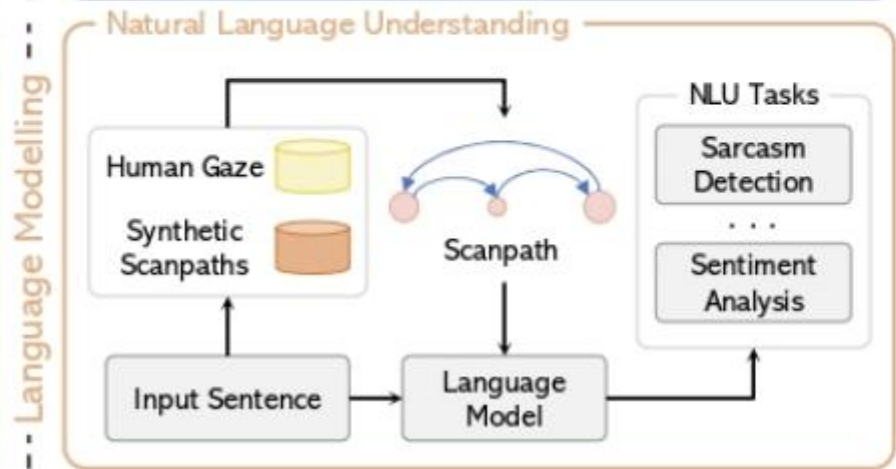
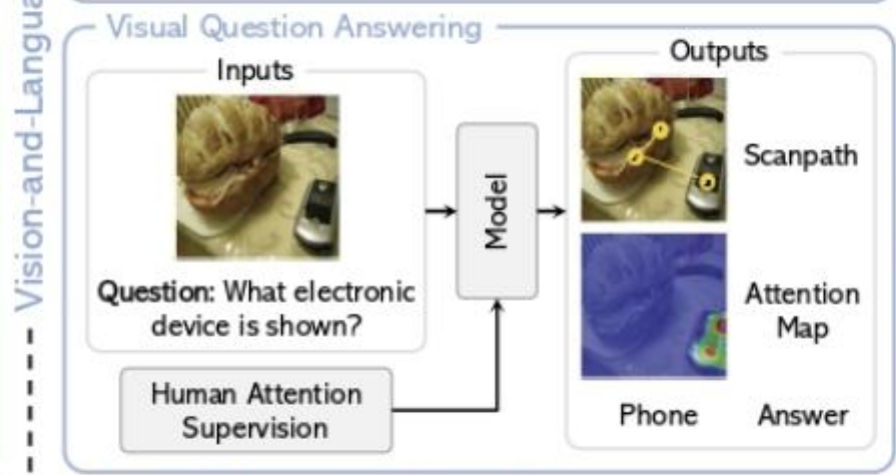
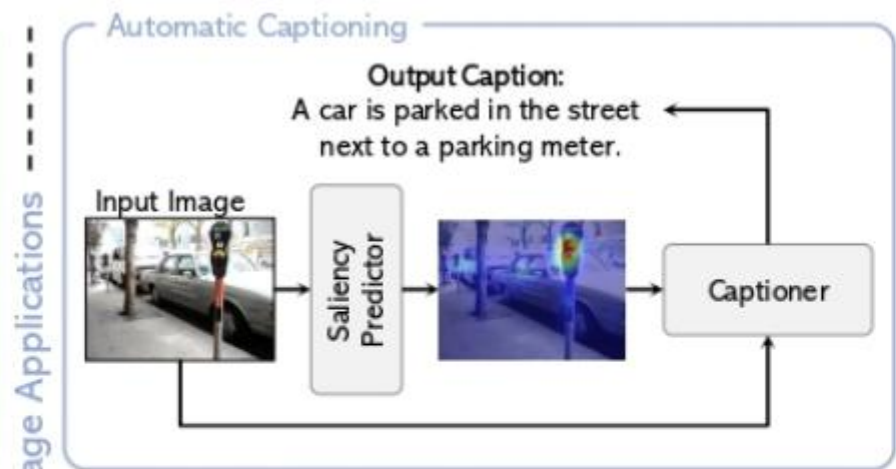
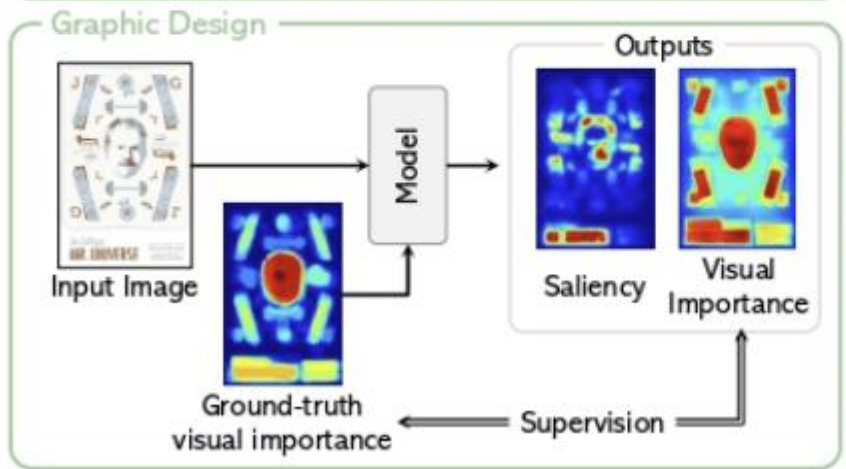
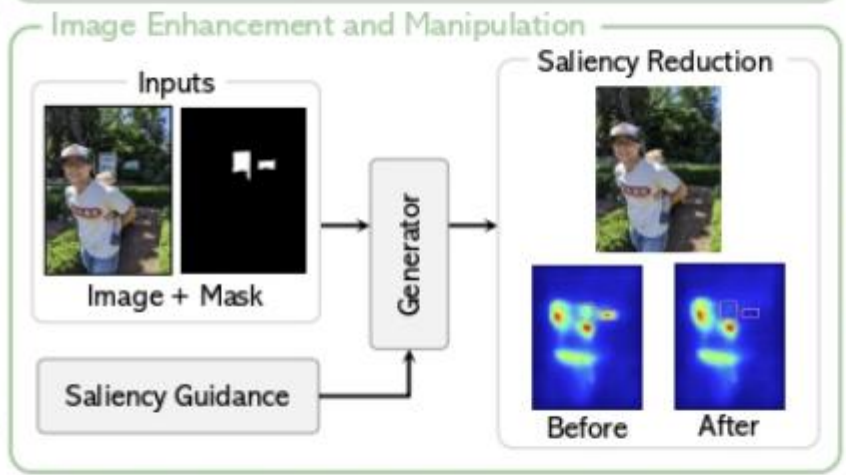
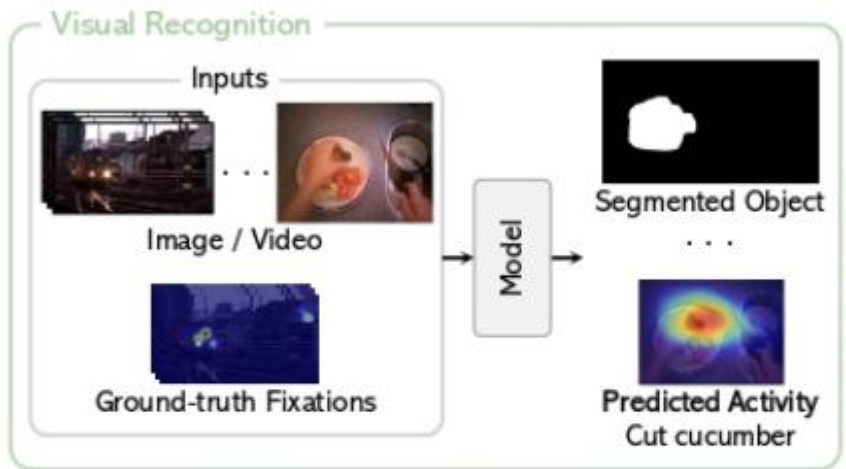


SwiGLU znajduje zastosowanie w wielu najnowocześniejszych modelach: LLaMA / LLaMA 2 / LLaMA 3 – modele językowe Meta na licencji open source; PaLM / PaLM 2 – modele LLM Google, Gemini; OpenAI, DeepSeek, Falcon, Mistral. Nie próbujemy wyjaśniać, dlaczego te architektury wydają się działać; ich sukces, podobnie jak wszystko inne, przypisujemy boskiej łaskowości 😊. Nie do końca ...

Funkcje w NN



Różne funkcje używane przez sieci neuronowe w modelach LLM



Vision-and-Language Applications

Domain Specific Applications

Language Modelling

Przykłady ciekawych aplikacji AI

- Forbes: [27 Incredible Examples of AI and Machine Learning](#) in practice: [Hello Barbie](#), [Coca-Cola](#) bot, [Heineken](#) bot.
- [Botsify](#) - bot assistants, from FAQ bots to tutoring bots.
- [Mika](#) - AI math tutor for higher education.
- [Snatchbot](#) for teachers, for classroom.
- [Ozobot](#) that can teach lessons to individuals about coding.
- [Chef Watson](#) from IBM
- [Project Malmo](#), AI in virtual reality.
- Google [Semantris](#) (word association);
- [Talk to Books](#), [Cyborg Writer](#),
- [Best Github projects](#) in data science and machine learning.
- Image restoration. deblurring, sharpening: [Topaz Labs](#).
- Shared autonomy: [3rd wave AI](#) (industrial)
- [Human-machine collaboration](#) in scientific research.

Repozytoria ML

- [OpenAI-cookbook](#)
- [Hugging Face](#) AI community models, datasets, and applications
- [Futurepedia](#) AI tools, agents, tutorials, innovations.
- [Papers with code](#)
- [There's An AI For That](#) - latest AI tools, by the #1 AI aggregator.
- [Data Science Central](#): online community for data science and AI.
- [Ozobot](#) that can teach lessons to individuals about coding.
- [Best Github projects](#) in data science and machine learning.
- [Human-machine collaboration](#) in scientific research.

- Deep Learning automated <https://www.youtube.com/@Deepia-ls2fo>
- [Scikit](#)