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.

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: Toniemy 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.

Porównanie czasów rozwoju ES

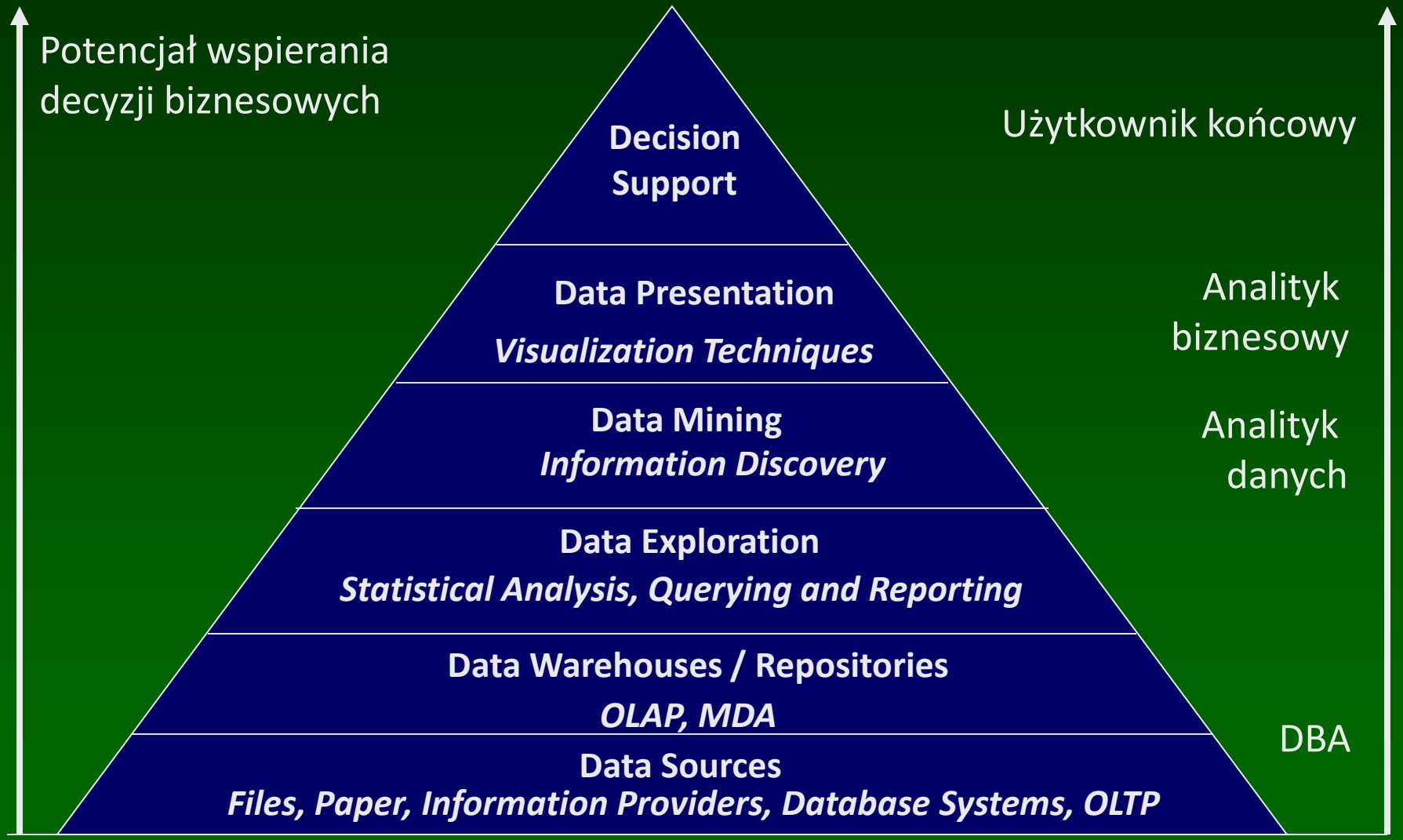
Nazwa	Typ	L. reguł	Czas (lat)	Poprawki (lata)
MYCIN	ES	1.000	100	wiele
XCON	ES	8.000	180	30
GASOIL	ML	2.800	1	0,1
BMT	ML	30.000	9	2

Dzięki ML drastycznie skracał się czas rozwoju złożonych systemów ekspertowych.
BMT Group Ltd (British Maritime Technology), sterowanie statkami.
To przyczyniło się do rozwoju data mining.

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

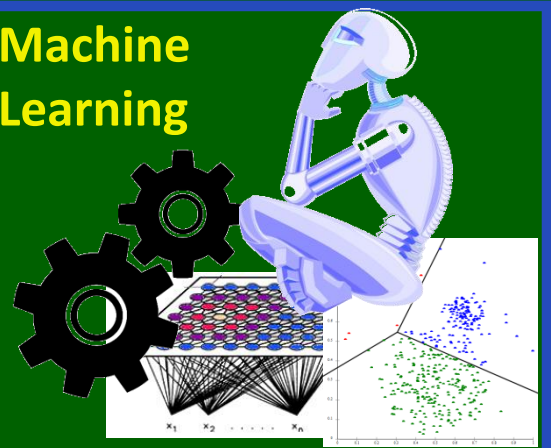
Environment

DATA

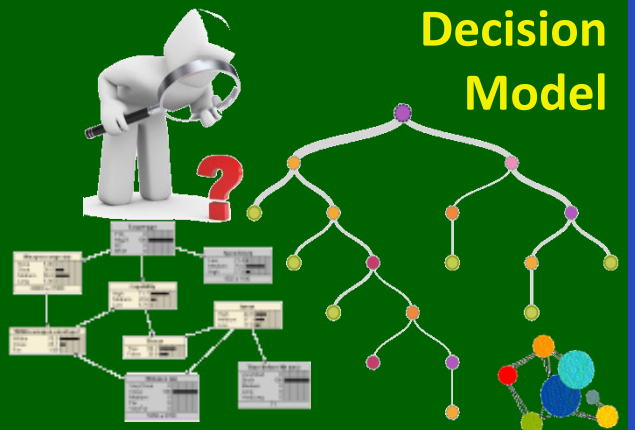
Decision

Data Mining &
Knowledge Discovery

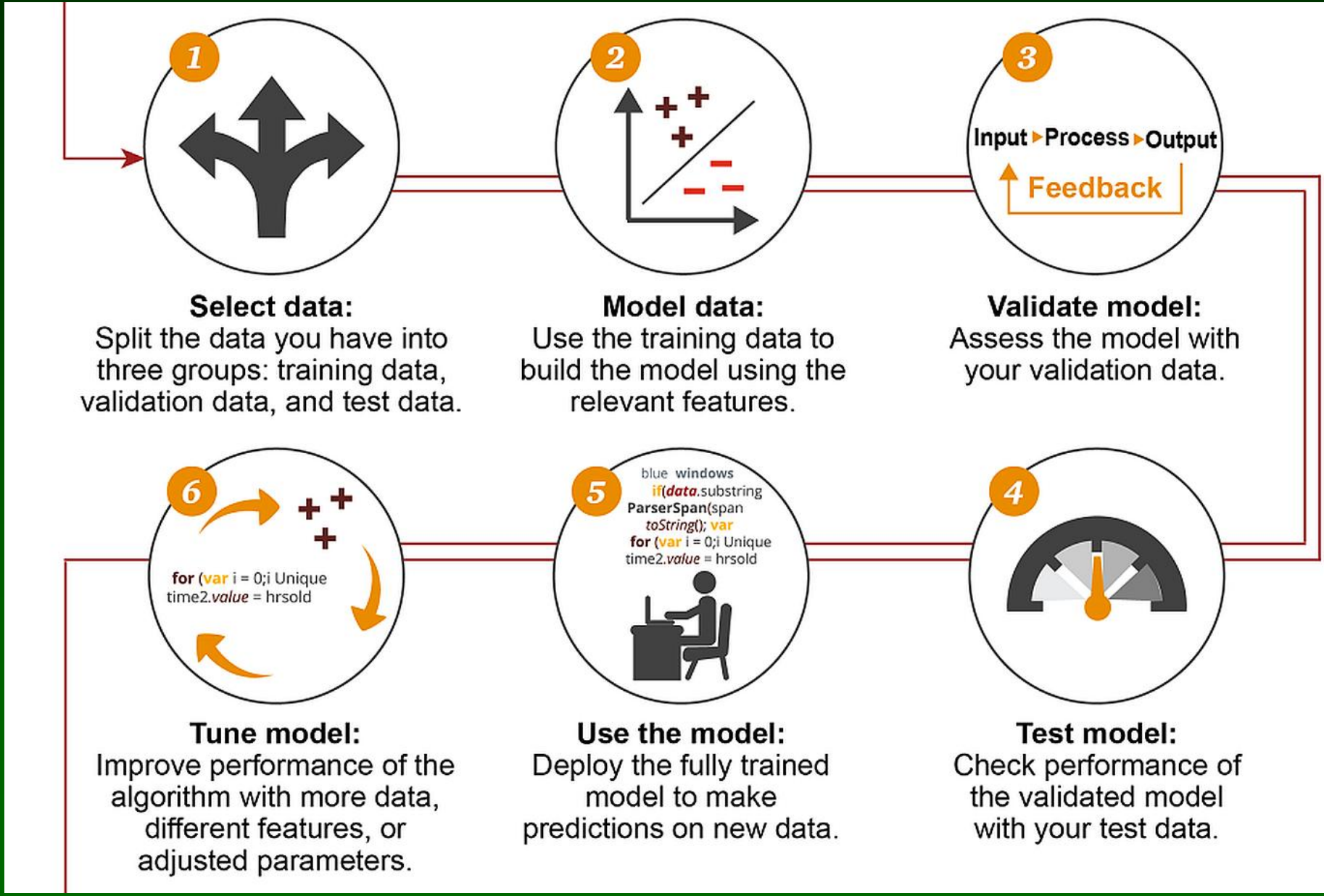
Machine
Learning



Decision
Model



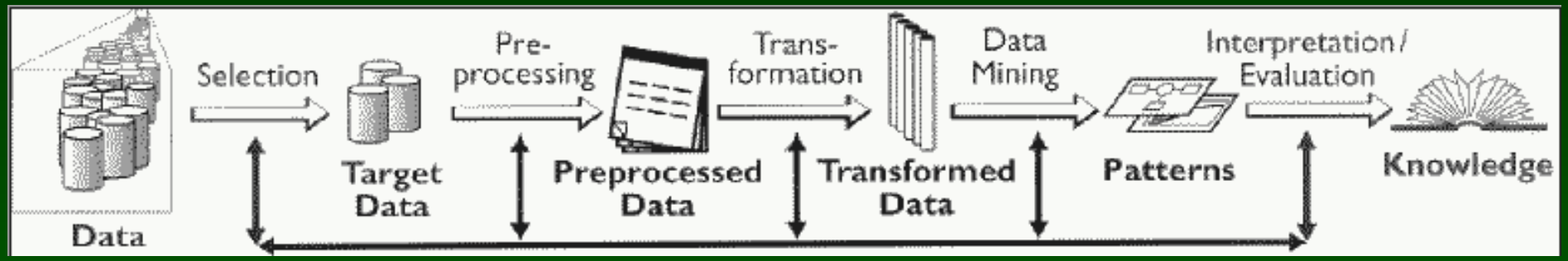
Proces analizy danych: Data Mining



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.

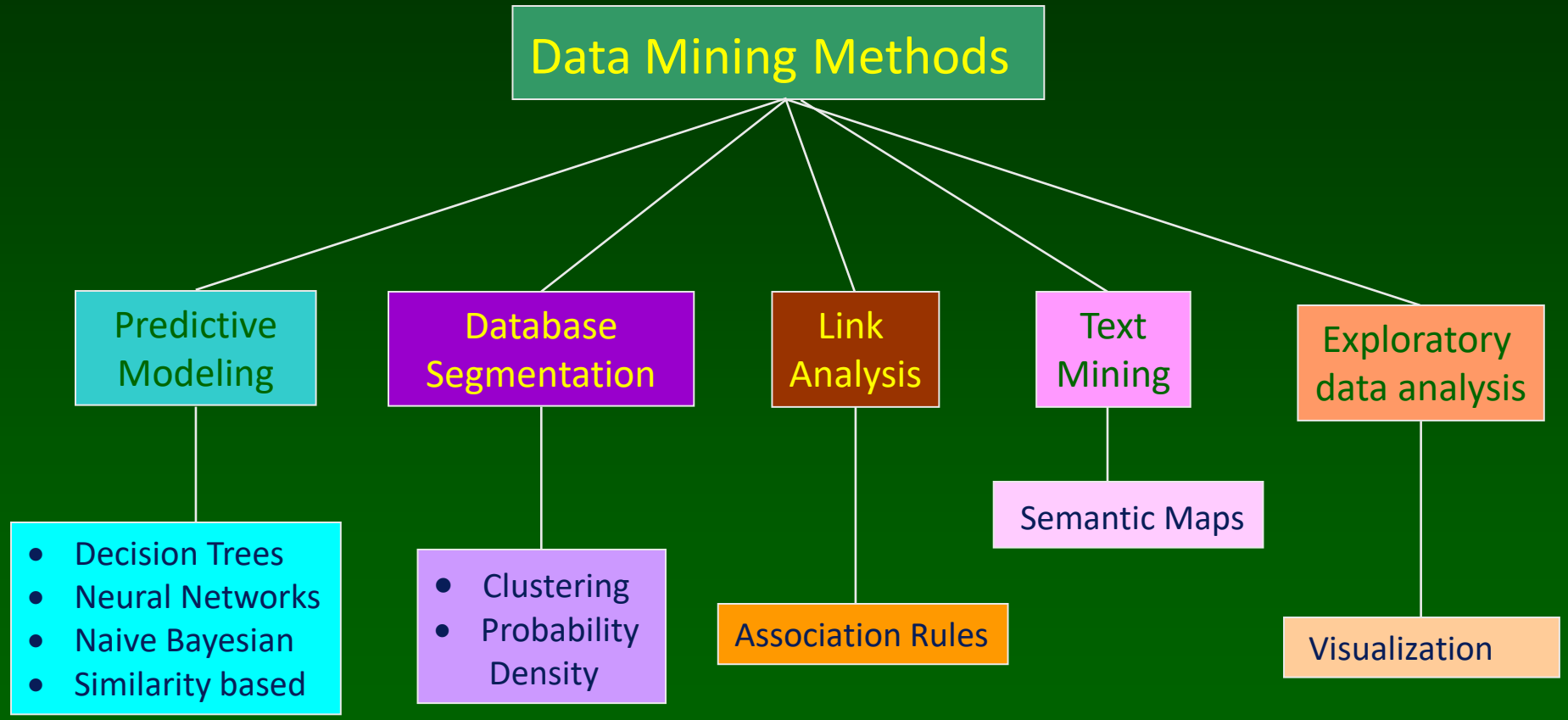
Pre-processing - wstępne przetwarzanie, czyszczenie danych (szum i wyrzutki), uzupełnianie braków, standaryzacja ...

Transformacja - do postaci akceptowalnej przez program.

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

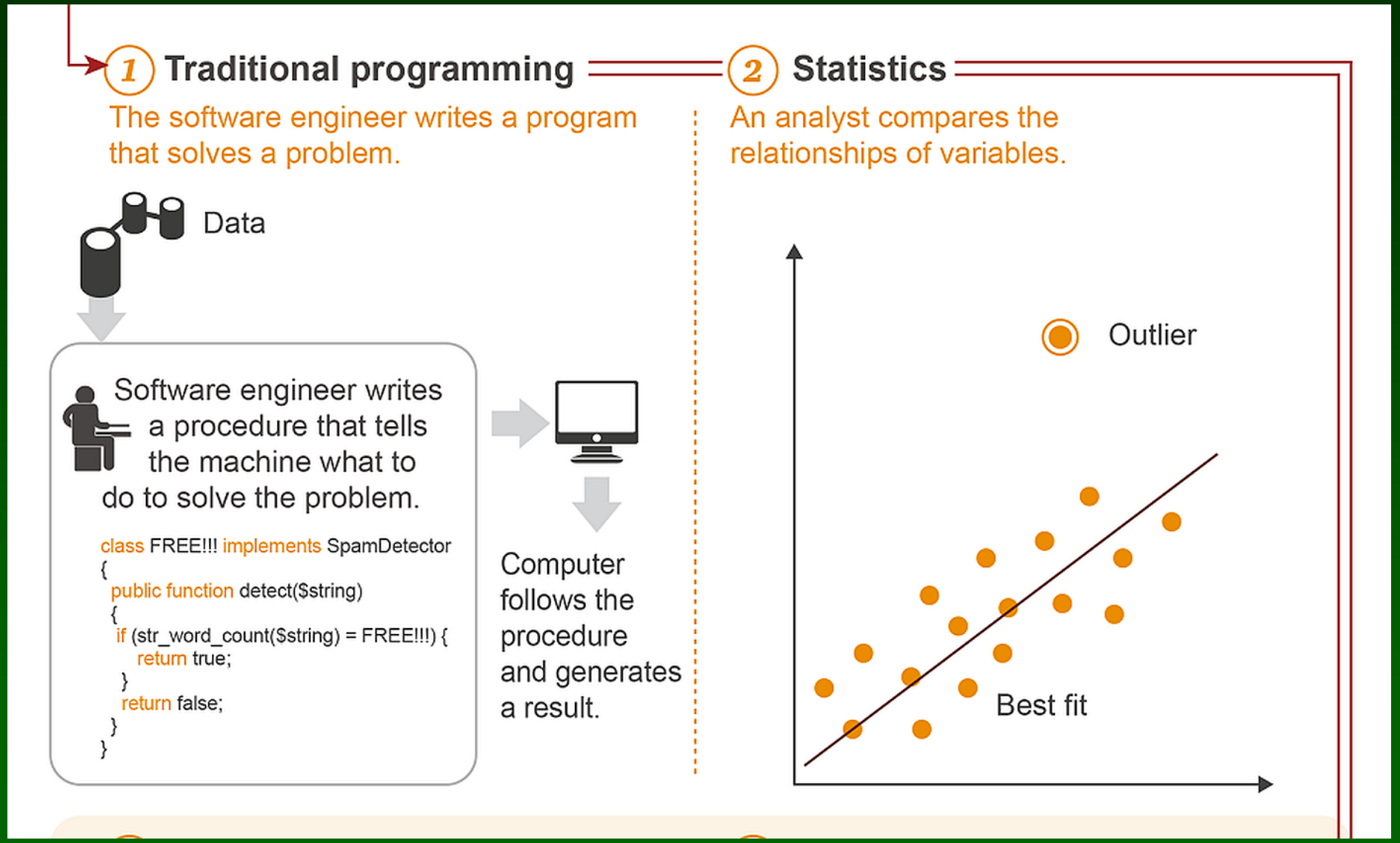
Tutorial na temat data mining.

Taksonomia metod Data Mining



Machine Learning

DM i statystyka



Klasy metod ML

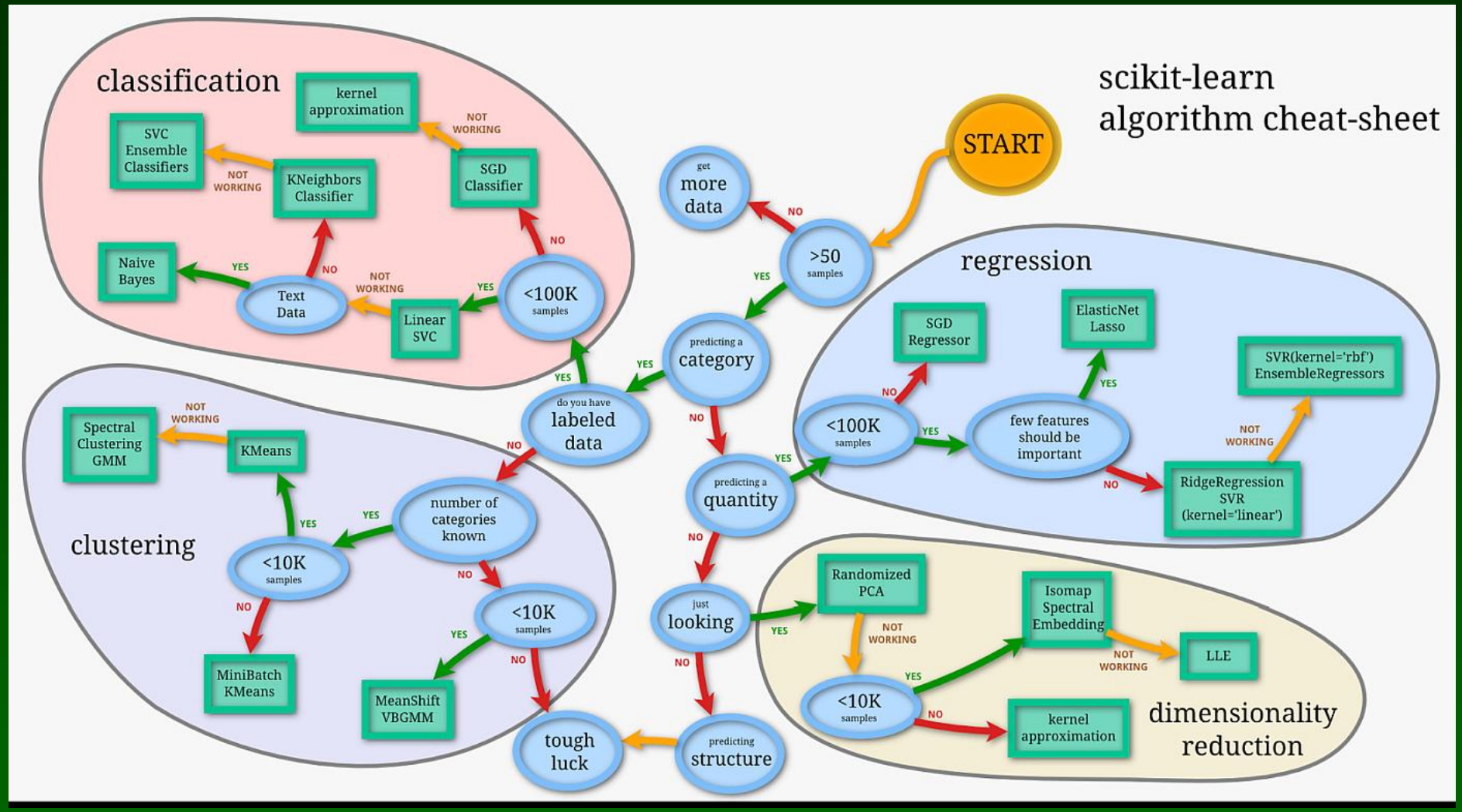
Powszechnie stosowane metody, ML w Python, tutoriale scikit-learn

- Drzewa decyzji/indukcja reguł.
- Logika rozmyta, systemy na regułach rozmytych.
- Rozumowanie oparte na analogiach, precedensach (case-based)
- Skojarzenia oparte na 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 ...

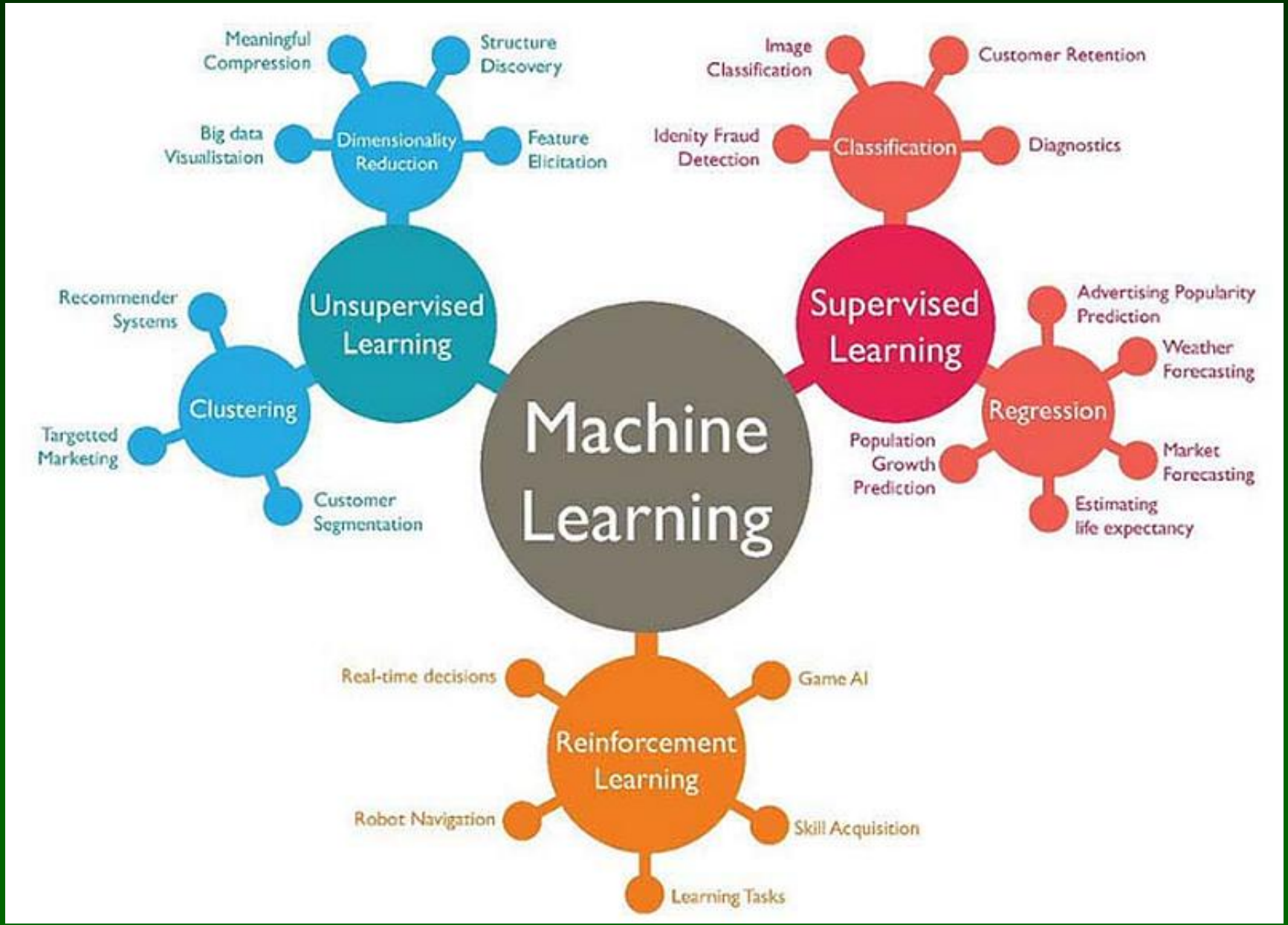
Metalearning - automatyczne tworzenie modeli.

Wybierz co chcesz zrobić.

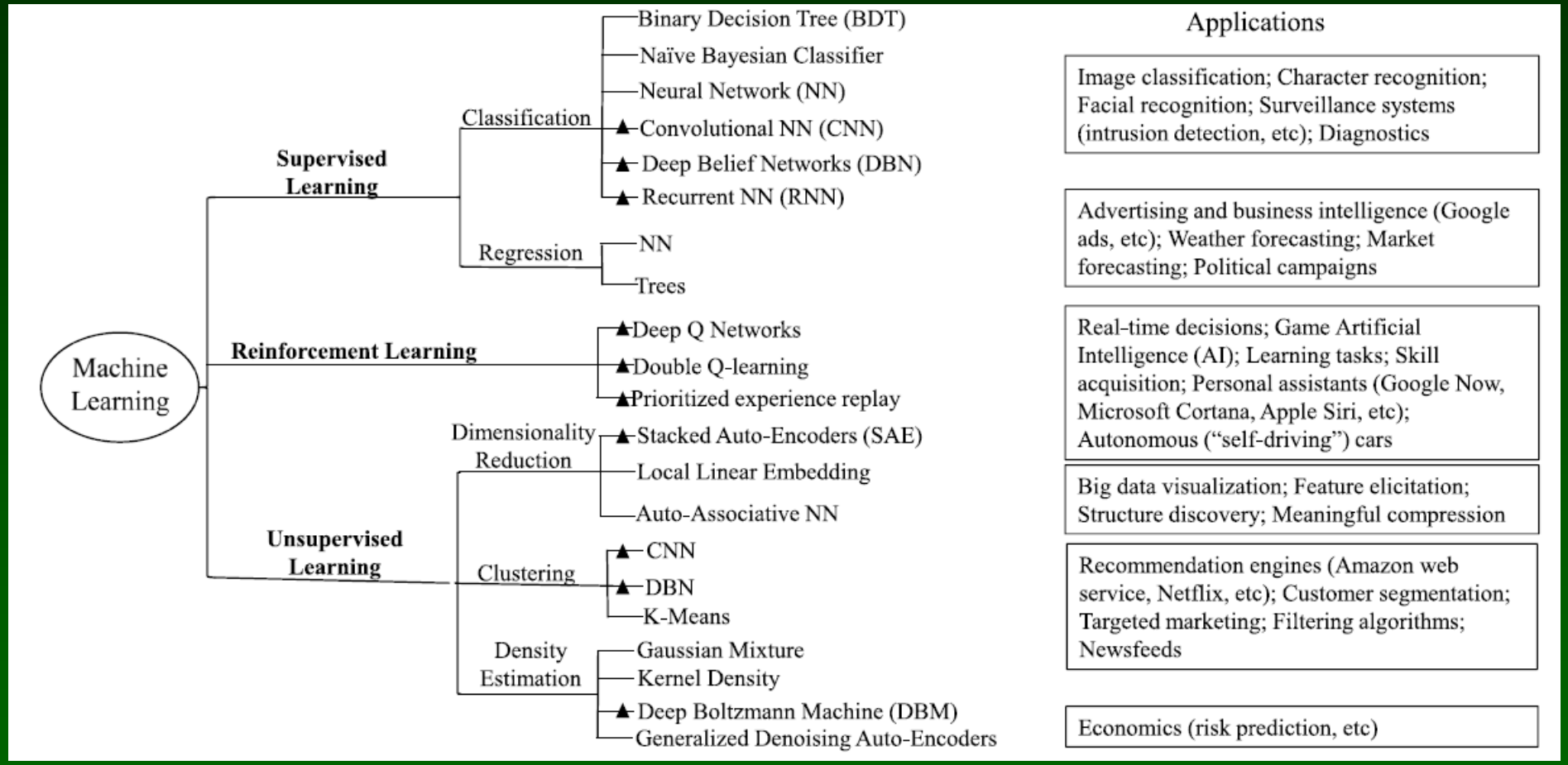


Scikit https://scikit-learn.org/stable/

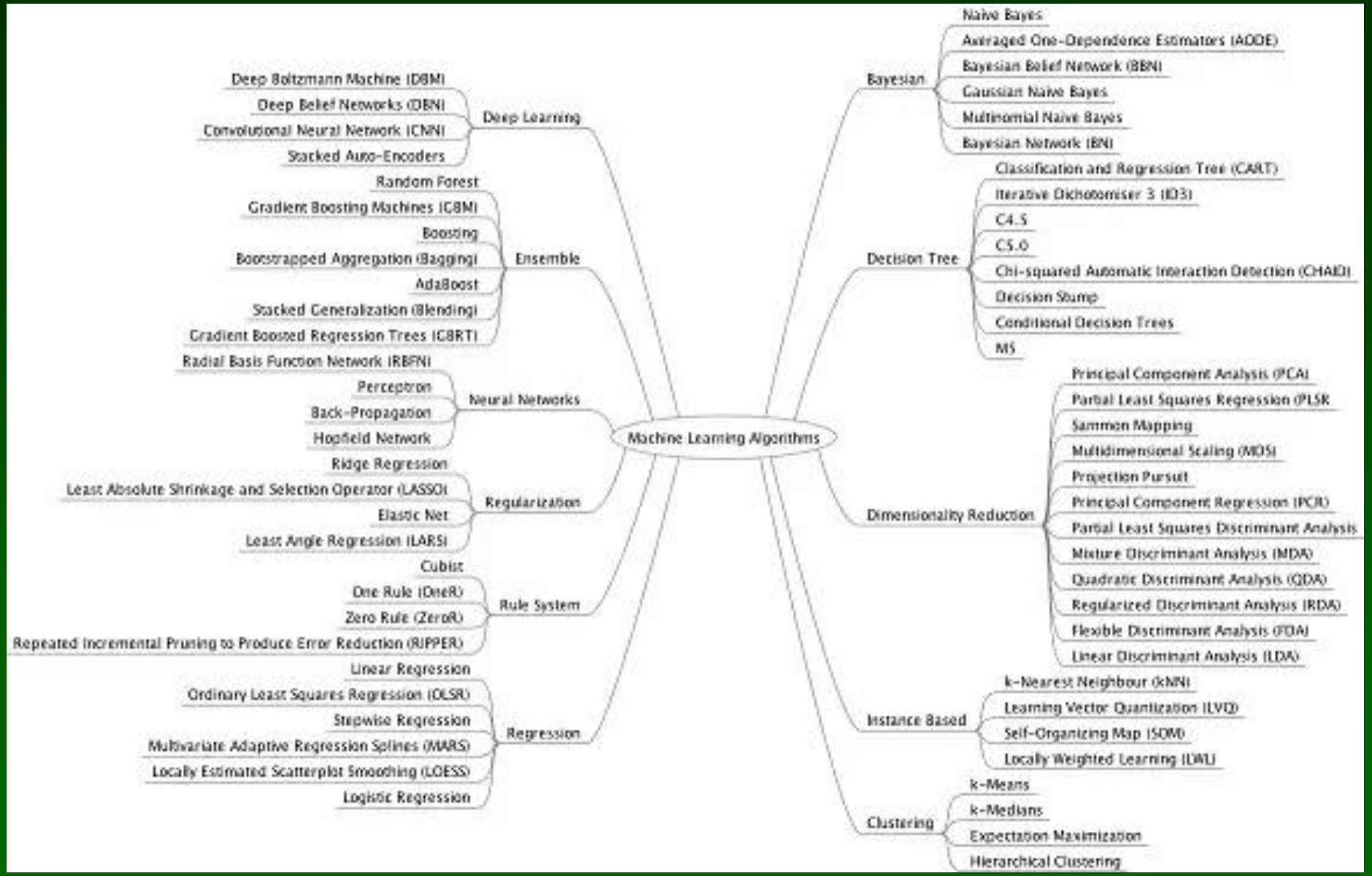
3 typy ML


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Subtypes of ML



Wiele rodzajów algorytmów 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.

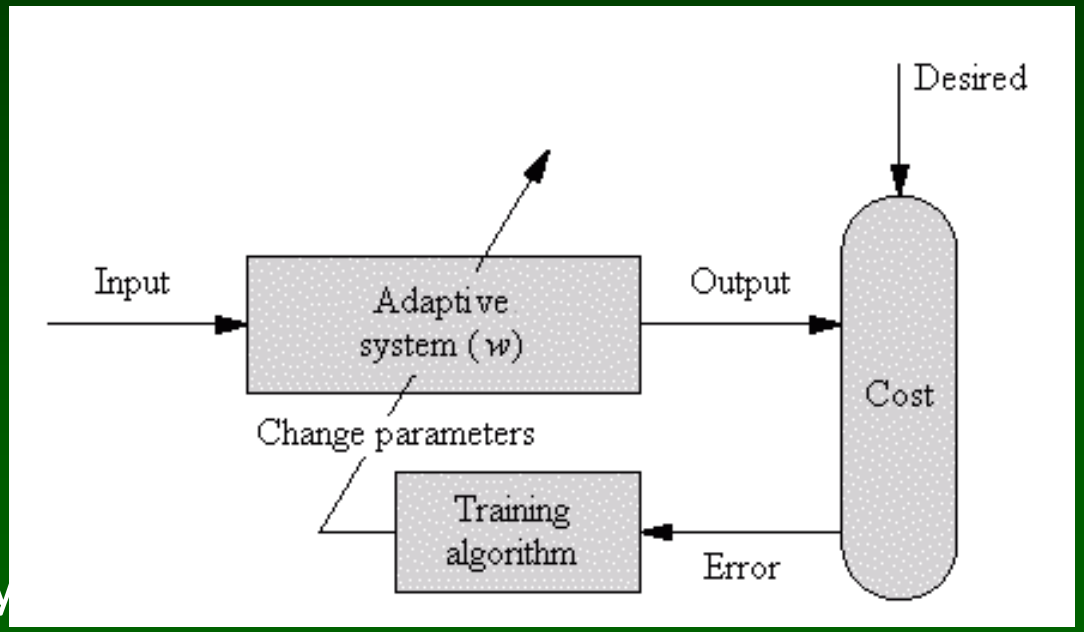
System A adaptujący swoje parametry W do danych.

Pożądane odpowiedzi Y(X)

Wyniki działania: Y'=A(X;W)

Funkcja błędu/kosztu E(Y,Y')

Algorytm minimalizujący błędy na danych treningowy



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

Uczenie bez nadzoru

Chcemy uprościć opis złożonych danych, bez nadzoru, spontaniczne.

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

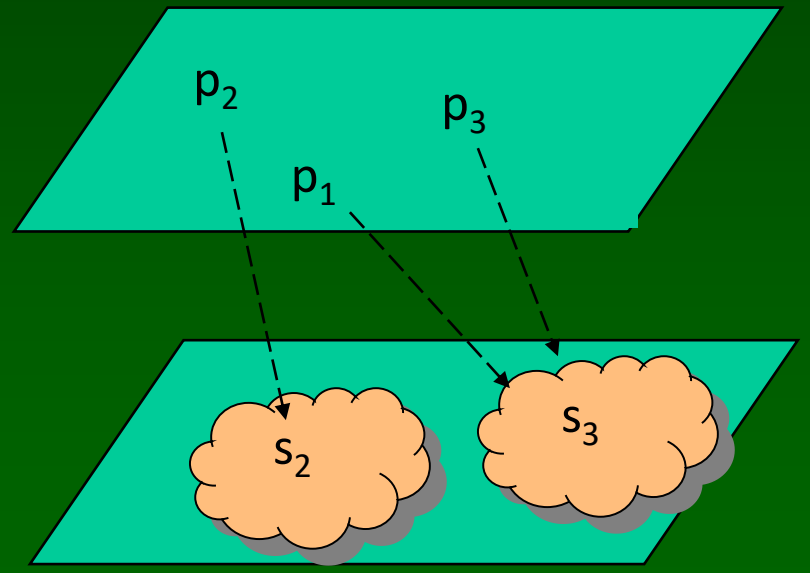
Miary podobieństwa $S(p_i,p_j)$

Algorytm uczący, analizujący rozkłady miar podobieństwa.

Tworzenie odróżnialnych skupień S_i

Funkcja jakości grupowania $E(S_i,S_j)$

Bioinfromatyka – 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:

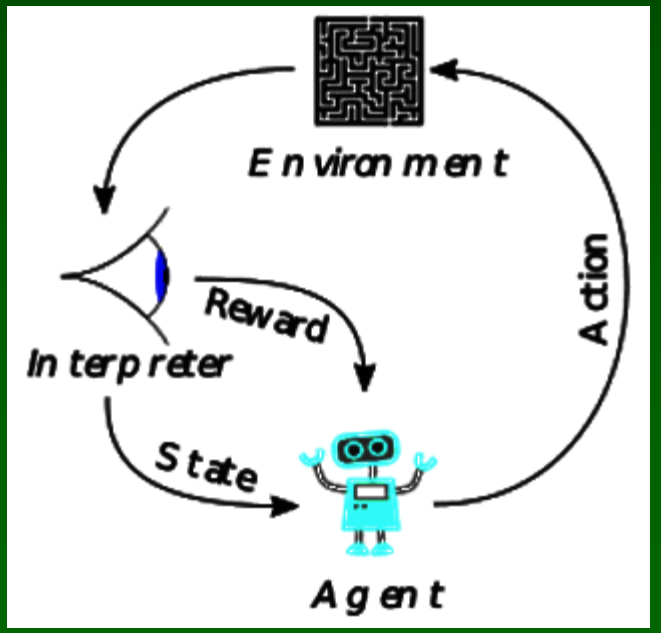
Opis stanów X w obserwowanym środowisku.

Opis akcji a, działania agenta A w stanie s.

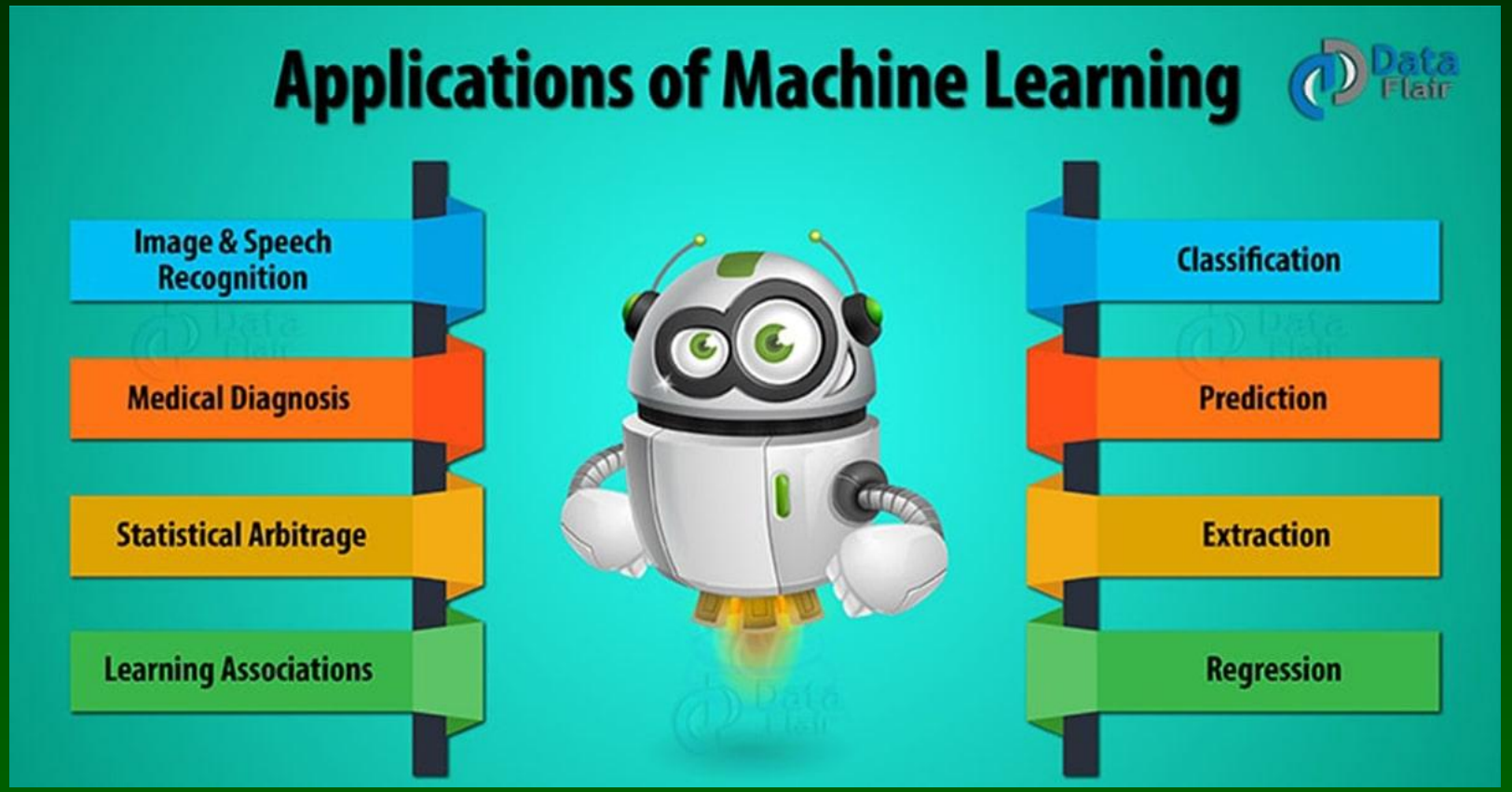
System adaptujący prawdopodobieństwa zmiany stanu s=>s’ pod wpływem akcji a.

Zmiana stanu środowiska i miara nagrody.

Algorytm tworzący plany zwiększające końcową nagrodę.



Zastosowania ML



Obecnie ML do wszystkiego …

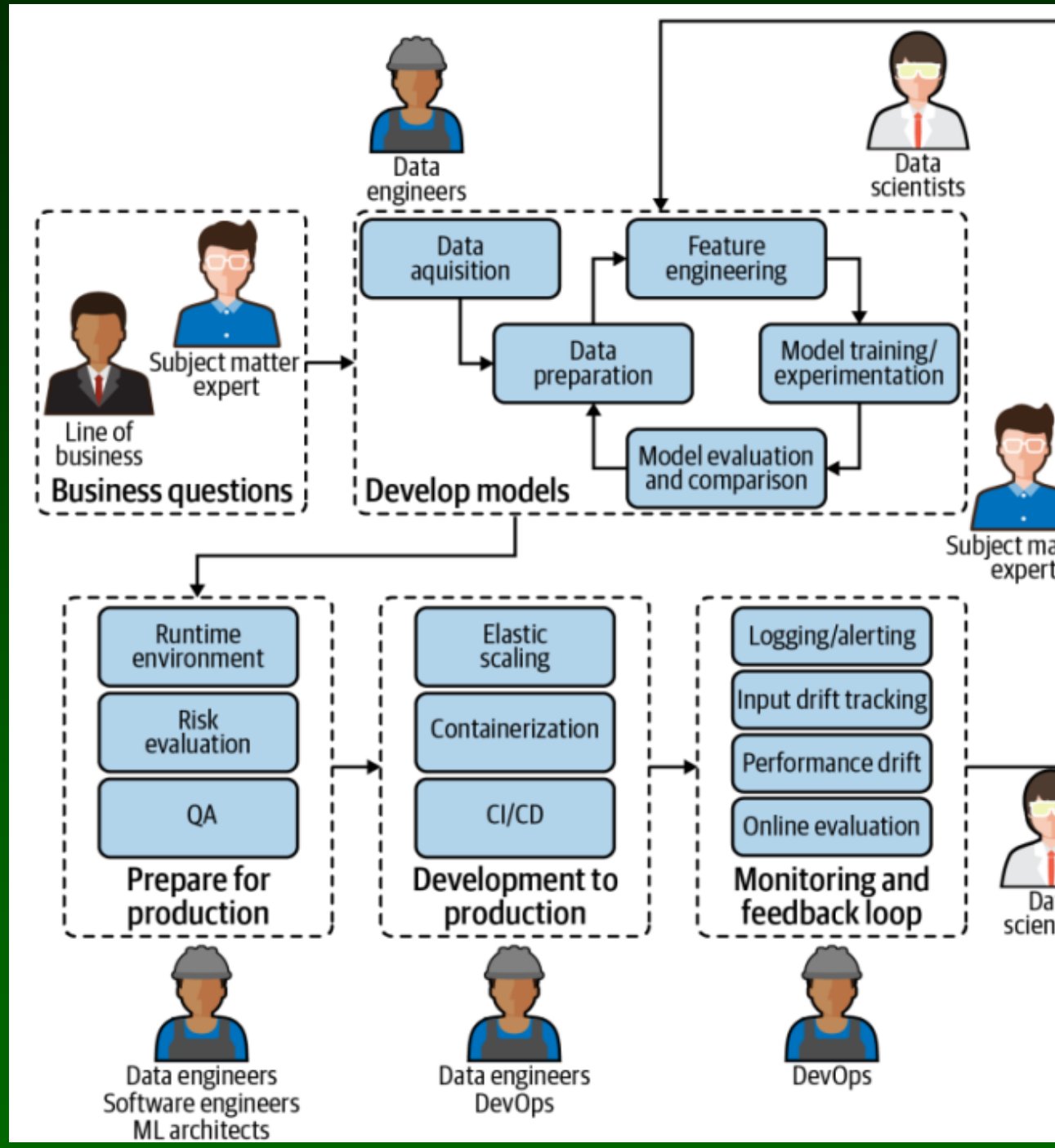
- ✓ Zintegrowane pakiety programów z elementami ML do data mining.
- ✓ Analiza obrazów i sygnałów.
- ✓ Sterowanie: automatyczny kierowca/pilot/czołg ...
- ✓ Analiza tekstów.

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.

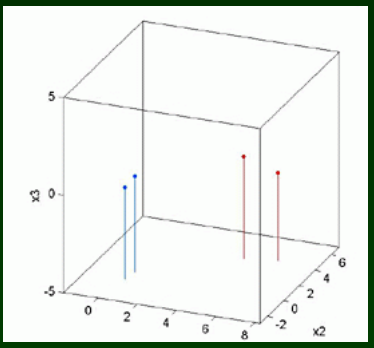
To problemy inżynieryjno-naukowo-biznesowe, a nie samo rozwijanie algorytmów. Analiza ryzyka, responsible AI



Exploracyjna analiza danych

Wiki

Projekcje 2D: skaterogramy



Najprostsze projekcje: użyj projekcji, wybierz tylko 2 cechy.

Przykład: cukier - próchnica zębów.

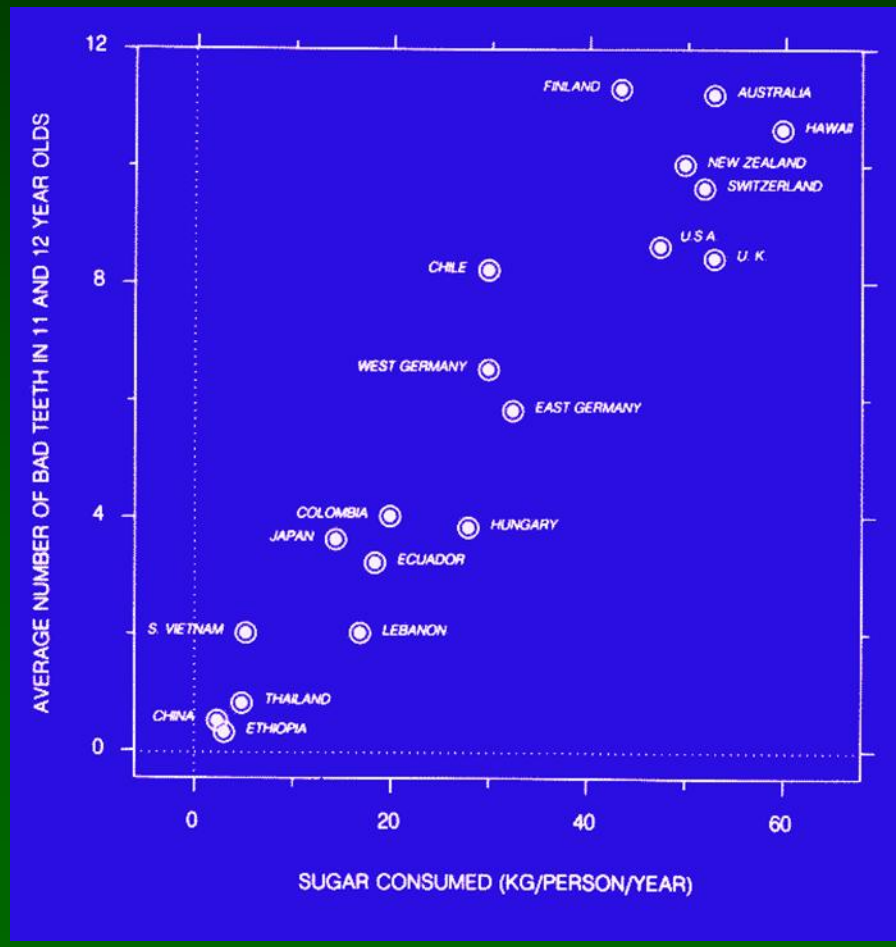
Dla d zmiennych mamy d(d-1)/2 podzbiorów 2D
i czasami wyświetlane są na jednym rysunku.
Każdy punkt 2D jest rzutem ortogonalnym ze wszystkich pozostałych d-2 wymiarów.

Czego szukać:

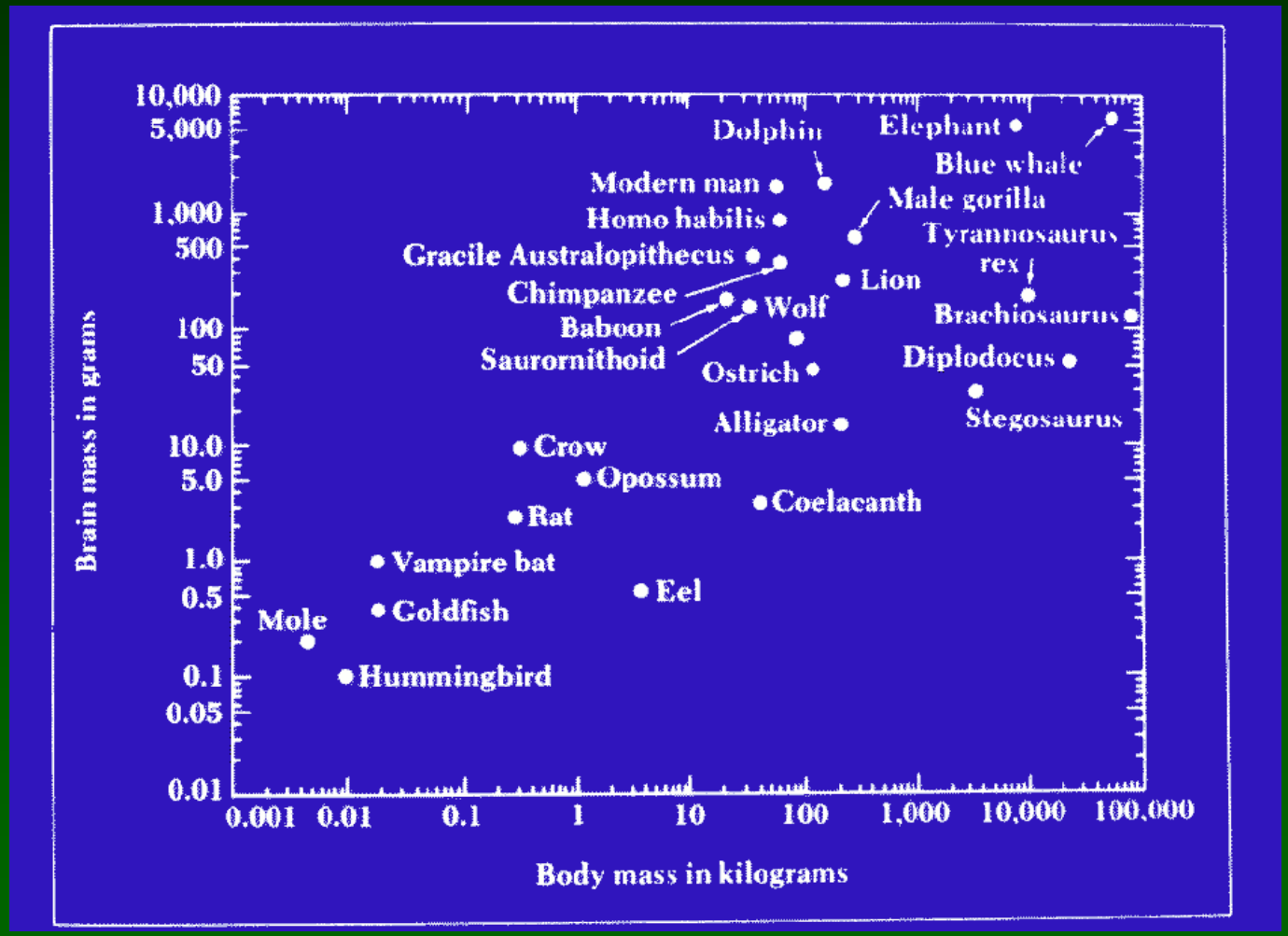
korelacje między zmiennymi,

grupowanie różnych obiektów.

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



Masa mózgu vs ciała



4 Gaussy w 4D

Rozkłady mają często postać Gaussów.

Skaterogramy danych 4D w dwóch wymiarach.

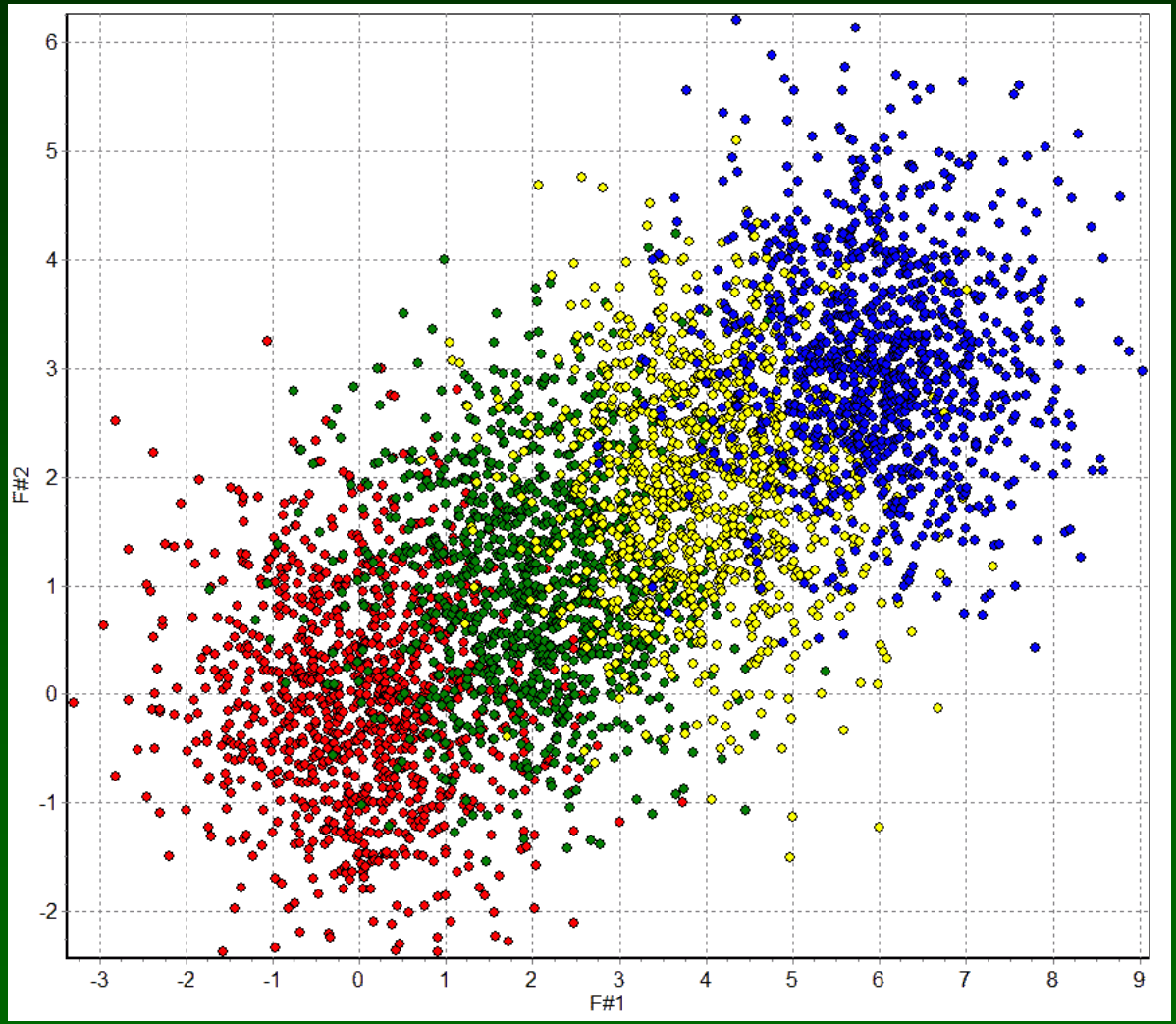
4 rozkłady w 4D, przesunięte względem siebie:

czerwony (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)

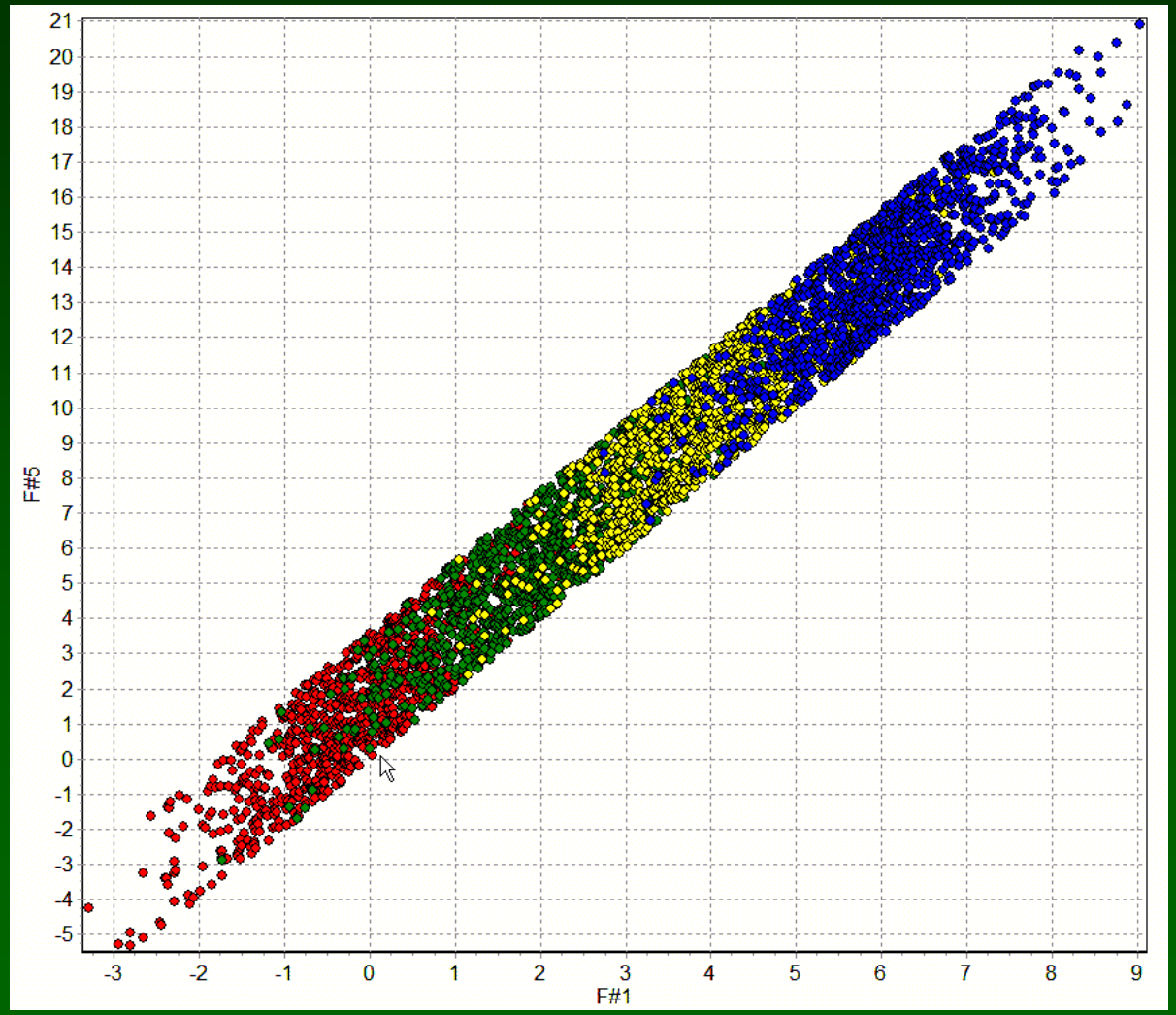


4 Gaussy w 8D, X_1 vs. X_5

O czym świadczy ten rysunek?

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

Jak te dodatkowe 4 cechy zostały utworzone?



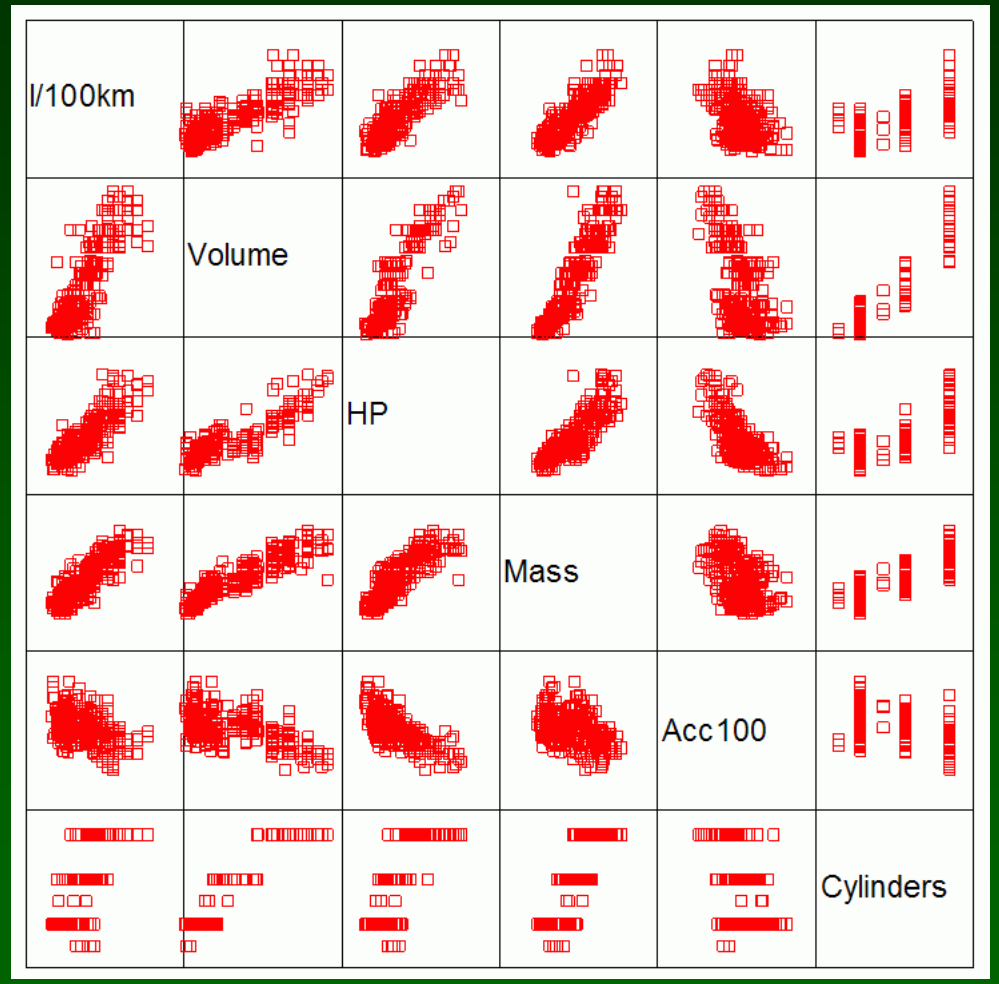
Samochody

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

Więcej przykładów:

http://www.public.iastate.edu/~dicook/JSS/paper/paper.html

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

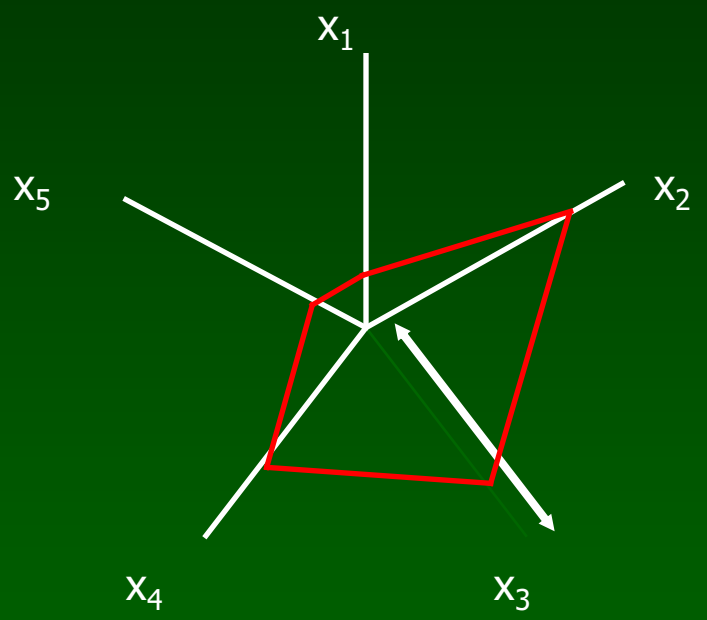
Opis Tourr w R data tour w Python

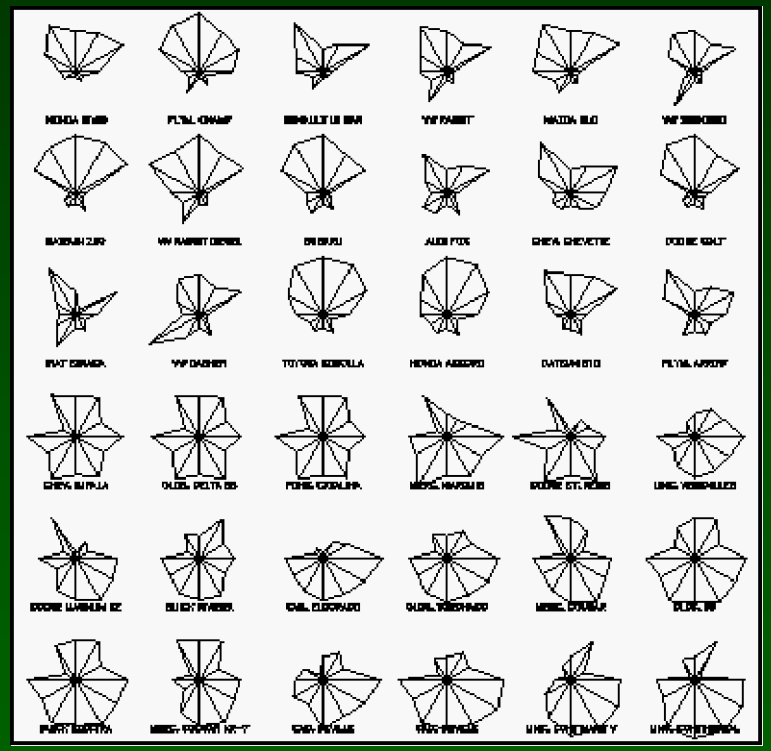
Wiki Grand Tour

Direct representation: star plots

Star Plots, radar plots:

represent the value of each component in a “spider net”.





Useful to display single or a few vectors per plot, uses many plots.

Too many individual plots? Cluster similar ones, as in the car example

Direct representation: star

Working woman

population changes in different states, projections and reality.

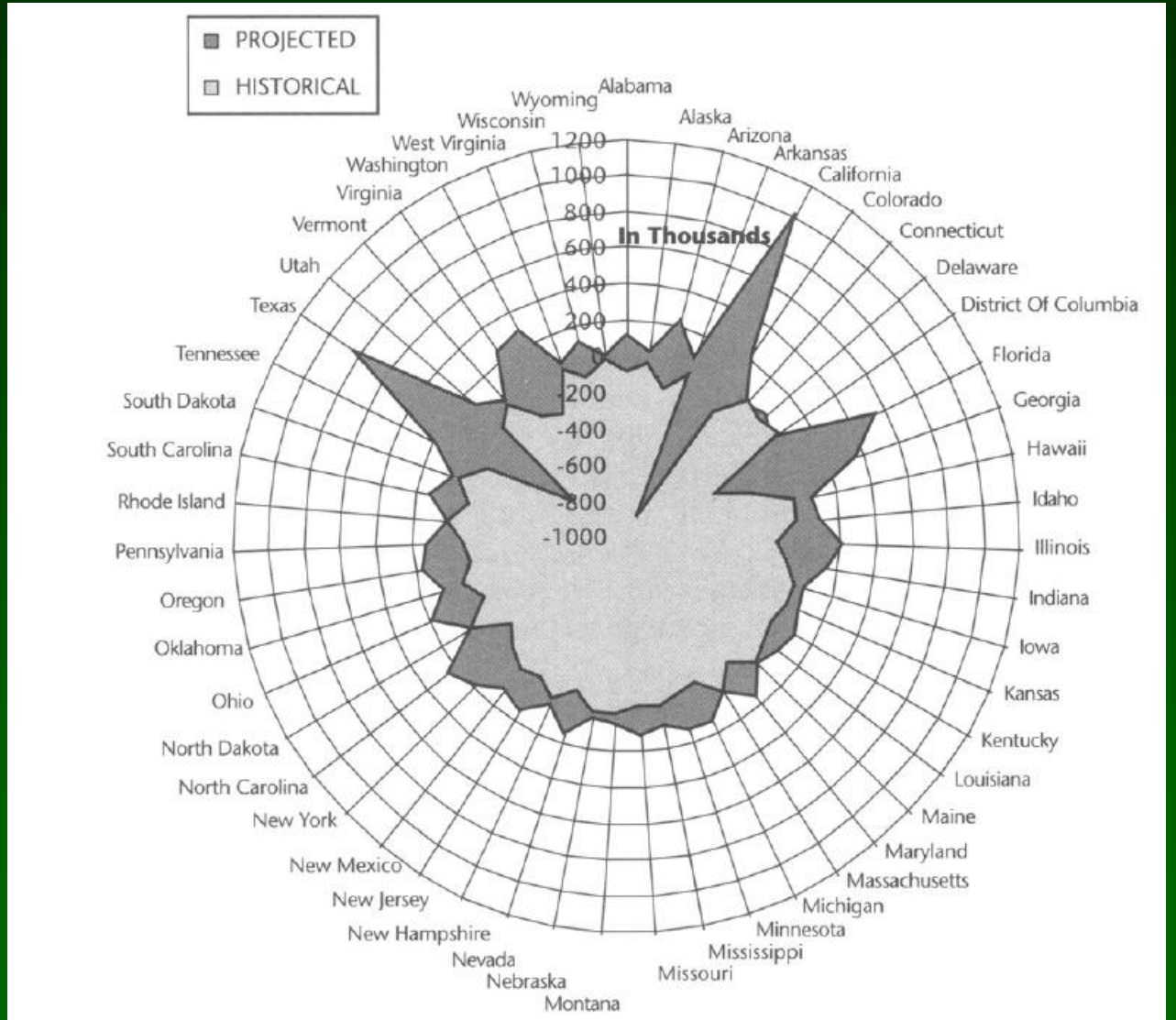


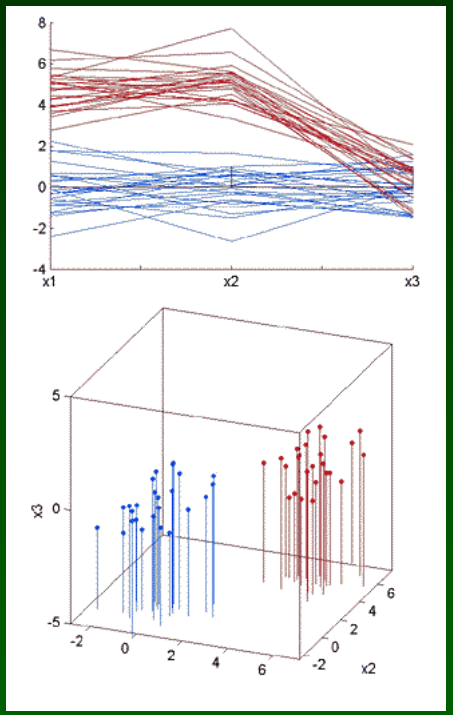
Figure 8.8: Projected versus historical working women population changes by state.

Direct representations: | |

Parallel coordinates: instead of perpendicular axes use parallel!

Many engineering applications, popular in bioinformatics.

Wiki page with software links



Two clusters in 3D

Instead of creating perpendicular axes put each coordinate on the horizontal x axis and its value on the vertical y axis.

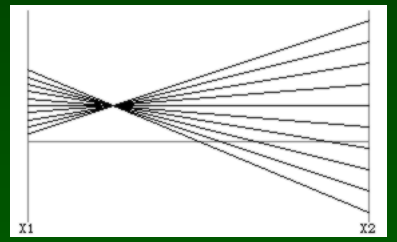
Point in N dim => line with N segments.

Alfred Inselberg, Parallel Coordinates: Visual Multidimensional Geometry and Its Applications (Amazon)
http://www.nbb.cornell.edu/neurobio/land/PROJECTS/Inselberg/

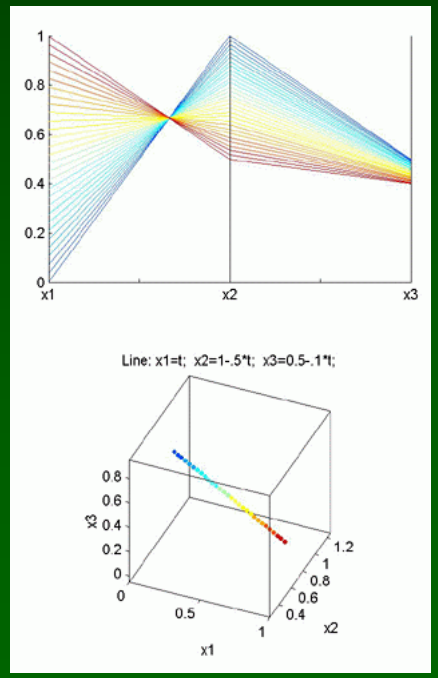
|| lines

Lines in parallel representation:

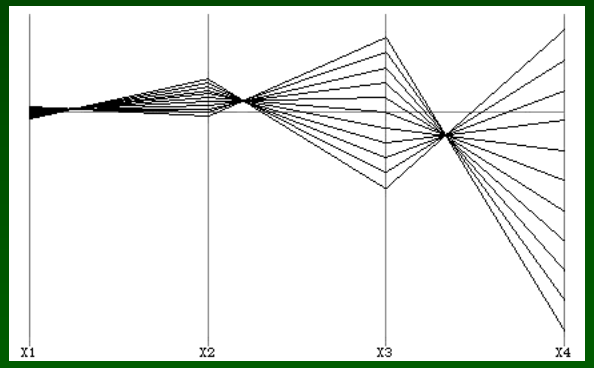
2D line



3D line



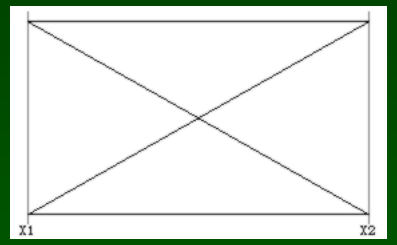
4D line



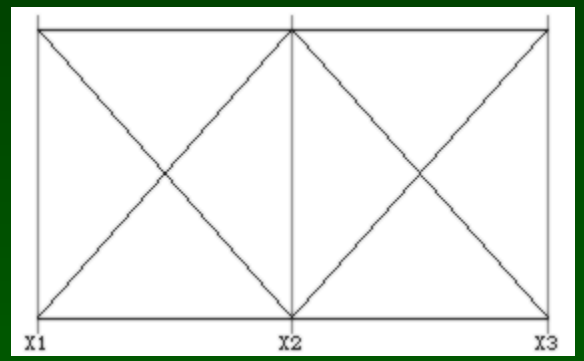
|| cubes

Hypercubes in parallel representation:

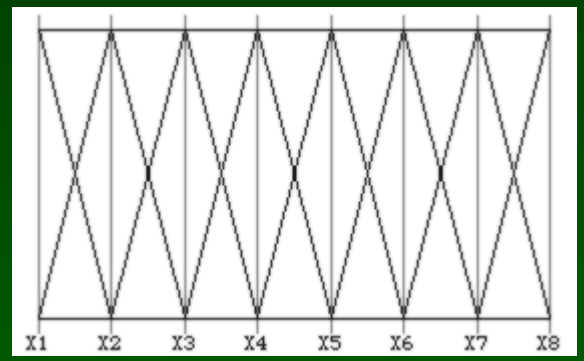
2D (square)



3D cube: 8 vertices



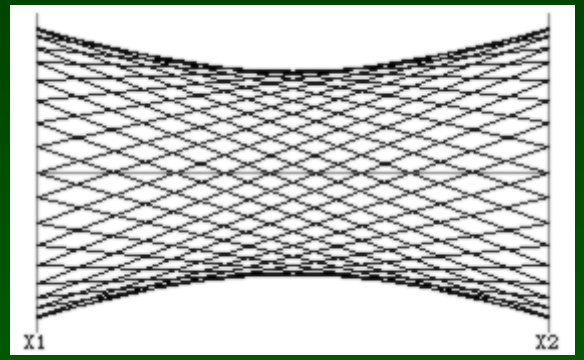
8D: 256 vertices



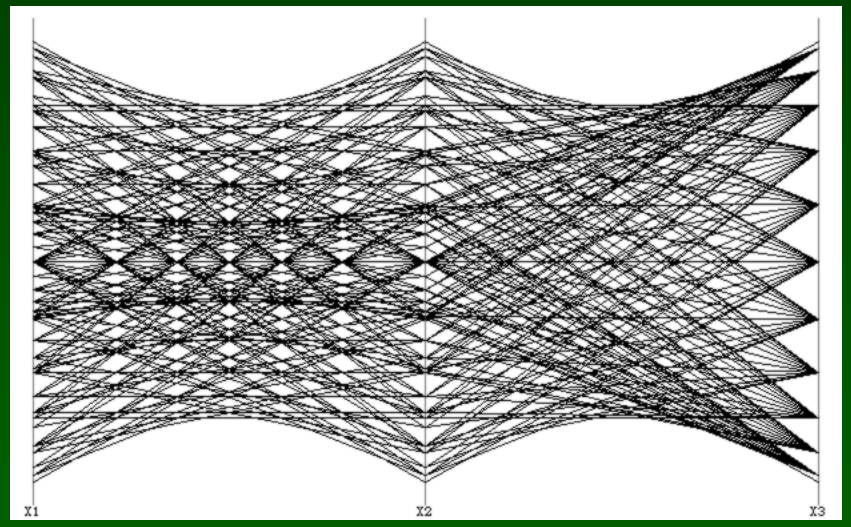
|| spheres

Hypercubes in parallel representation:

2D (circle)



3D (sphere)

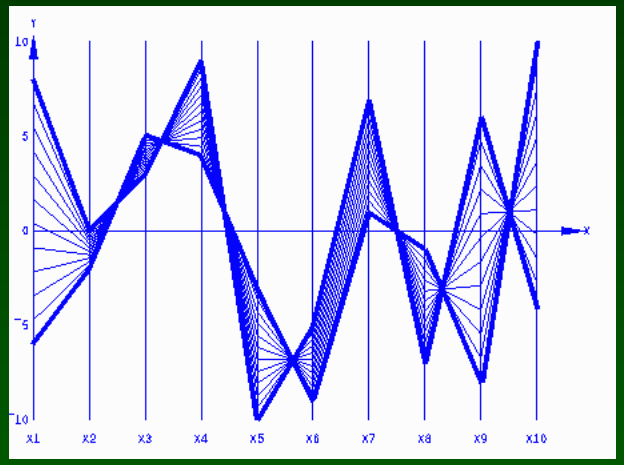


... 8D: ???

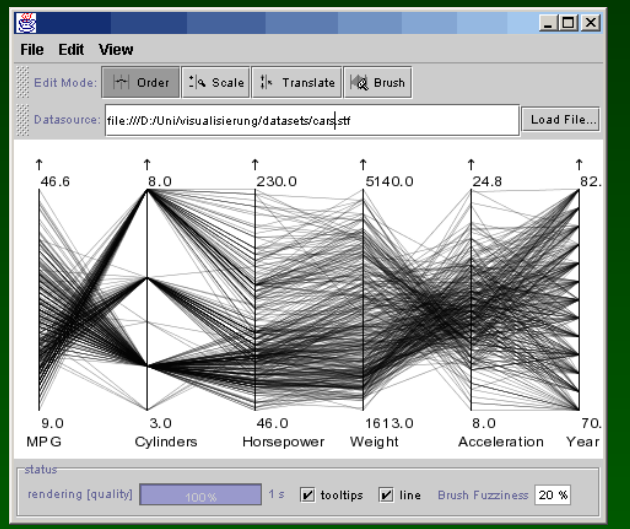
Try some other geometrical figures and see what patterns are created.

|| coordintates

Representation of 10-dim. line ($x_{1,}$,... x_{10}) t,



car information data



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 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?

MDS algorithm

- Data (feature) space $X \in R^d$, vectors $X^{(i)}$, i=1.. *n*
  represented by points Y in the target space, usually $Y \in R^2$.
- Distances $R_{ij} = ||X^{(i)}-X^{(j)}||$ between $X^{(i)}$ and $X^{(j)}$ in feature space R^d;

  distances $r_{ij} = ||Y^{(i)}-Y^{(j)}||$ in target space R^2.

- Find mapping $X \rightarrow Y=M(X)$ that minimizes some global index of topographical distortion, based on differences between R_{ij} and r_{ij}.

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

r=**r**(**Y**), so E(**r**) depends on 2*n* adaptive parameters **Y**;
3 are redundant: $Y^{(1)}=(0,0)^T$, choosing coordinate center,
and $Y_1^{(2)}=0$ choosing orientation of the coordinate center.

Initialize **Y** randomly or using PCA, minimize *E*(**r**(**Y**))

Miary zniekształcenia

Any non-negative function $f(R_{ij},r_{ij})$ may be used.

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

Weights may depend on the distance, for example decreasing exponentially for large R_{ij} distances.

Various normalization factors may be introduced, but those that do not depend on the r_{ij} distances have no influence on the MDS map.

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

Information loss;
0 = no loss, perfect representation,
1 = all lost, no info, ex. r_{ij}=0

Topographical distortion measures

Stress (Kruskal), or absolute error, large distances may dominate, preserves overall cluster structure.

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

Sammon measure, or intermediate error, contributions from large distances is reduced.

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

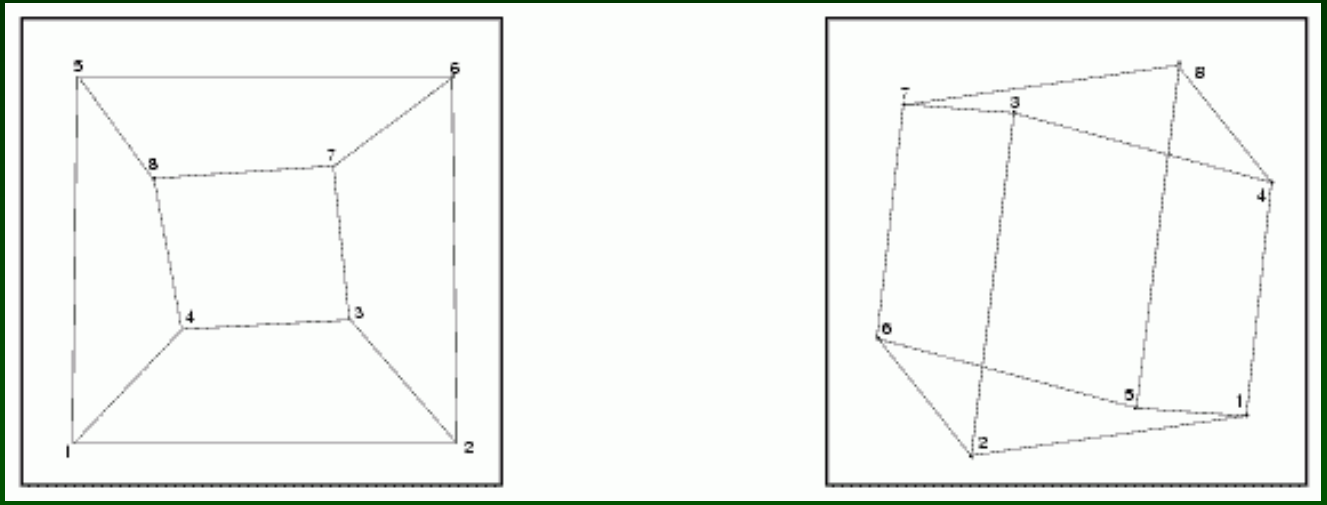
Relative error, all distance scales treated in the same way.

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

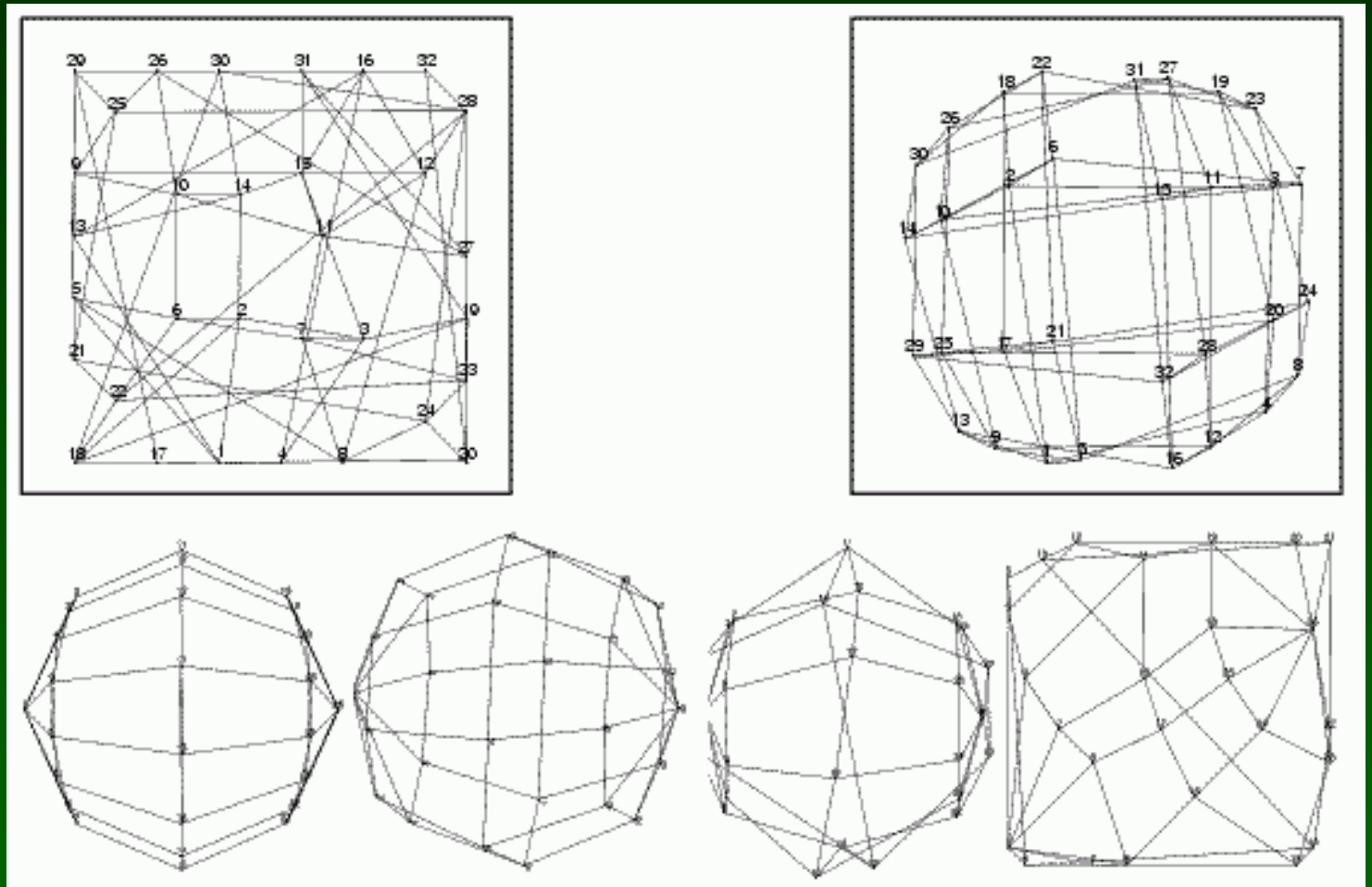
Alienation coefficient (Guttman-Lingoes) – similar to relative error, but more difficult to minimize.

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

Hipersześciany

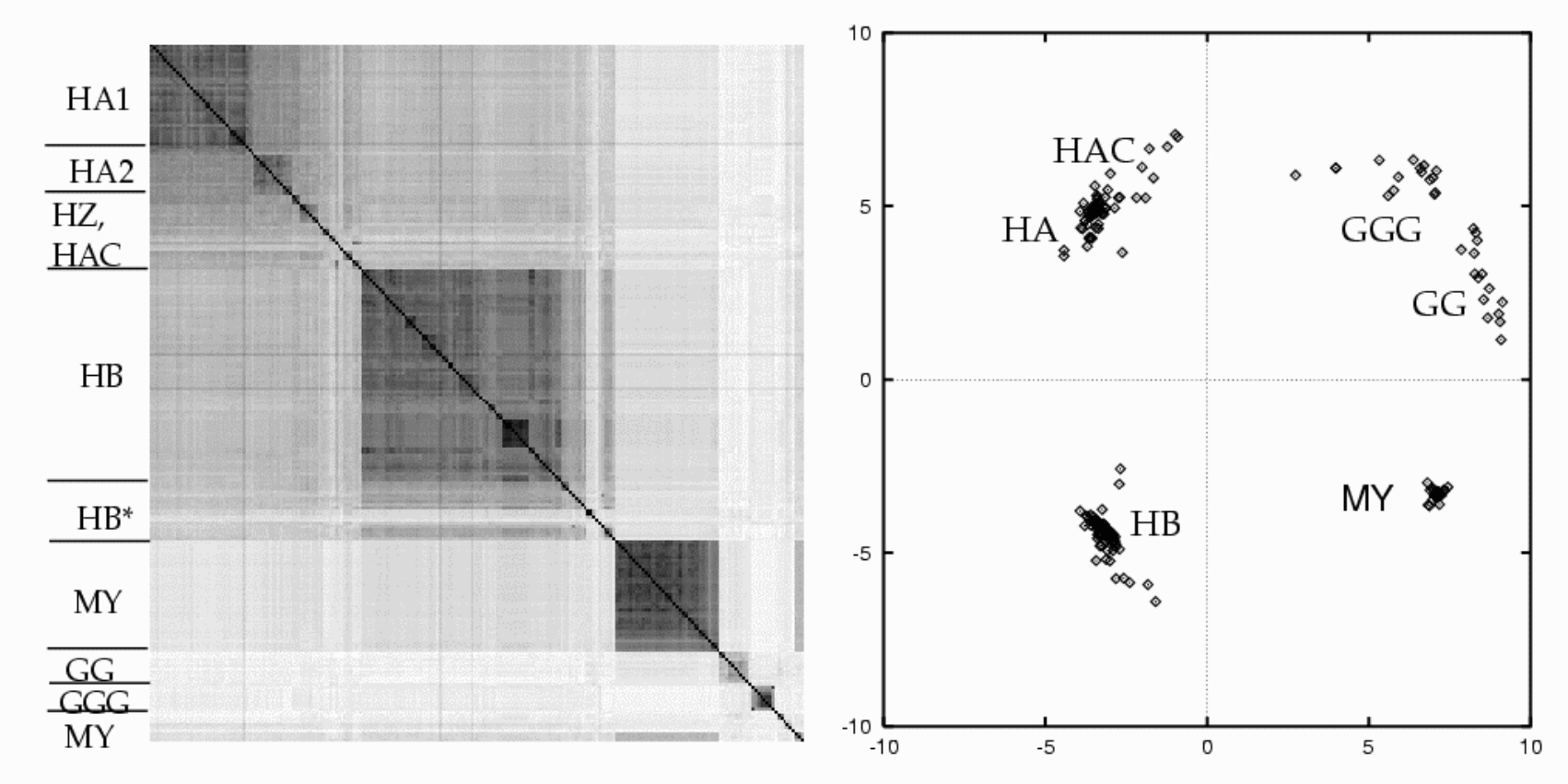


Hipercubes 5D + sphere in 3D



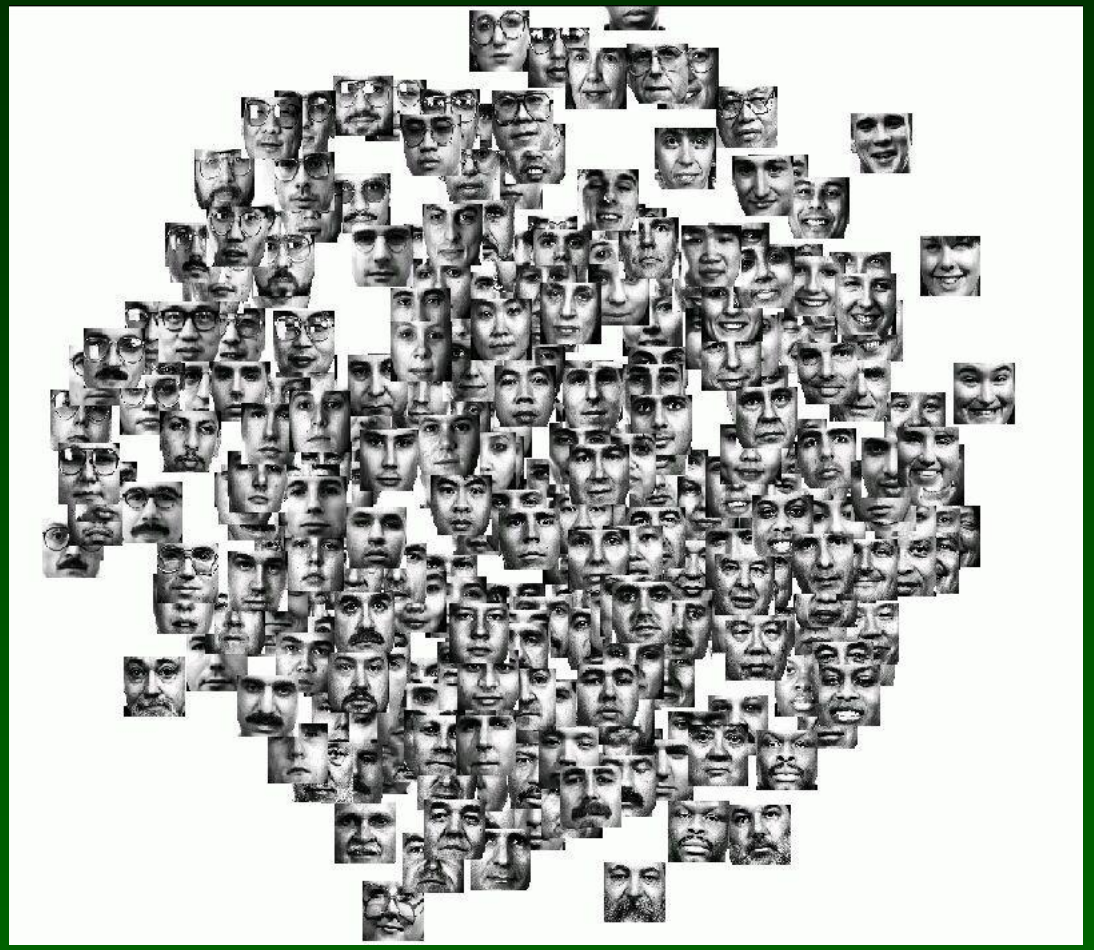
The two-dimensional representations of the 26 points on the sphere obtained by minimization of *S*, *E*, *A*, and by **SOFM** (left to right) with a 20 x 20 neurons map.

Sequences of the Globin family



226 protein sequences of the Globin family; similarity matrix shows high similarity values (dark spots) within subgroups,
MDS shows cluster structure of the data (from Klock & Buhmann 1997).

Similarity of faces



300 faces, similarity matrix evaluated and Sammon mapping applied (from Klock & Buhmann 1997).

Popularne algorytmy do wizualizacji wielowymiarowych danych, dostępne w Scikit

Zalety: szybsze niż MDS.

Jak działa UMAP: topologiczna analiza danych i kompleksy symplicjalne.

Szczegółowe omówienie UMAP

UMAP explorer interaktywne przykłady, UMAP z zoo.

Porównanie UMAP i tSNE w Pythonie.

UmapTSNE

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 bioinfromatyce.

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
Attrybut={wartość$_i$} lub Attrybut < wartość$_i$

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

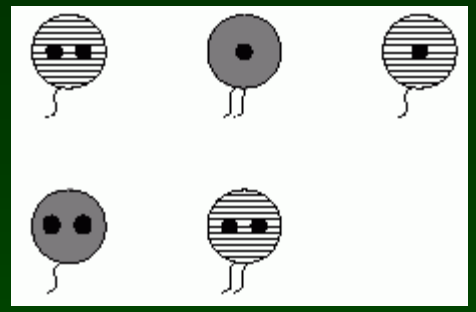
Przycinanie: usuń gałęzie, które zawierają zbyt mało przypadków
prostsze drzewo może lepiej generalizować
oceń optymalną złożoność na zbiorze walidacyjnym.

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

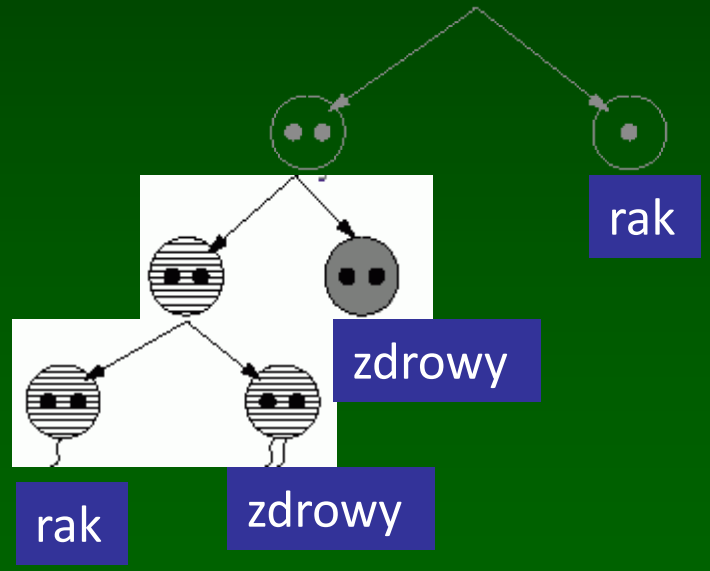
Klasy: {rakowe,



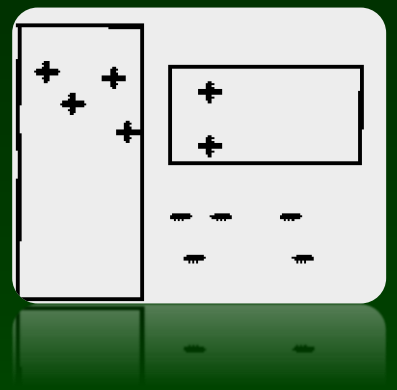
zdrowe}

Cechy: ciało komórki: {cienie, paski}

jądra: {1, 2}; ogonki: {1, 2}



Indukcja reguł



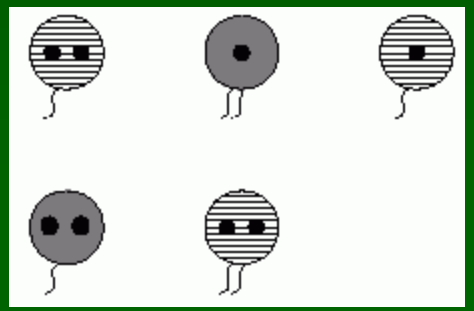
DT: tworzą reguły hierarchiczne

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

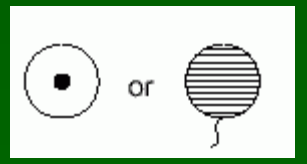
Zalety: łatwo zrozumiałe, dobra reprezentacja hipotez.

Wady: powolne działanie, algorytmy pokrycia NP-trudne

Przykłady: CN2, AQ-nn, GREEDY ...



Abstrahowanie od szczegółów => rakowe jeśli:



Przestrzenie wersji

Indukcja reguł/pojęć na podstawie analizy danych

Miejsce	Posiłek	Dzień	Koszt	Reakcja
DS. 1	śniadanie	Piątek	tanio	Tak
Kosmos	lunch	Piątek	drogo	Nie
DS. 1	lunch	Sobota	tanio	Tak
Bar mleczny	śniadanie	Niedziela	tanio	

W jakich warunkach mamy reakcje alergiczne?

Miejsce		Posiłek		Dzień		Koszt
3	X	3	X	7	X	2 = 126

Jeden z pierwszych algorytmów uczenia w AI.
Obecnie mało używane.

Ocena podobieństwa (similarity-based)

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

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

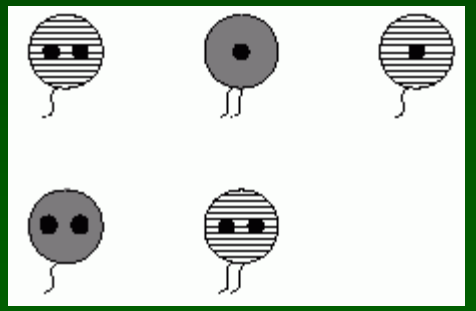
Wady: mogą wymagać dużo pamięci, konieczna selekcja atrybutów, konieczne dobranie odpowiednich funkcji podobieństwa, trudno zrozumiałe wyniki.

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).

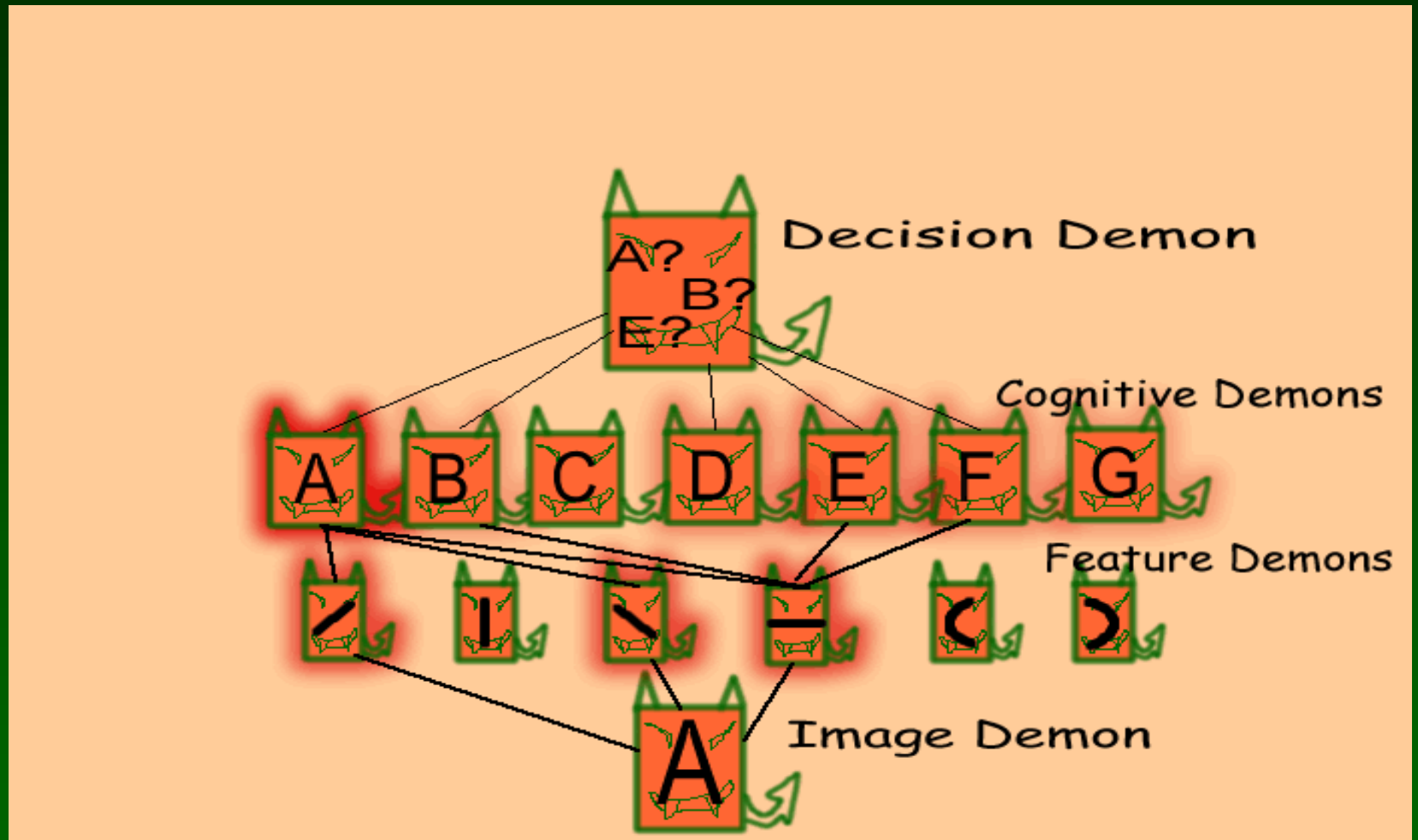
Funkcji kosztu/oceny jakości działania.

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



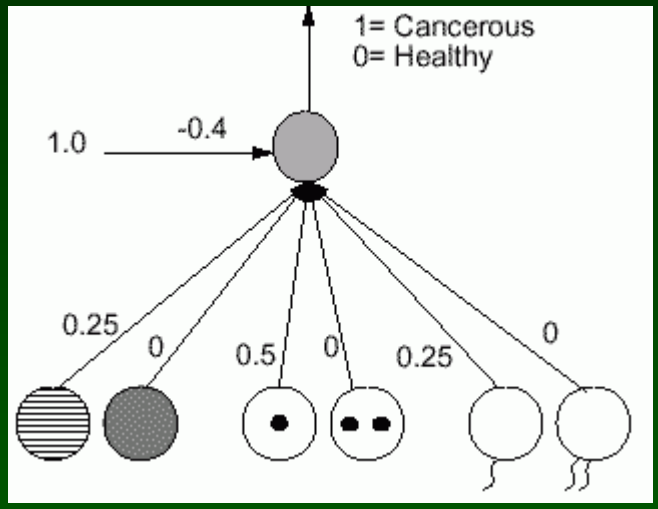
Based on:

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

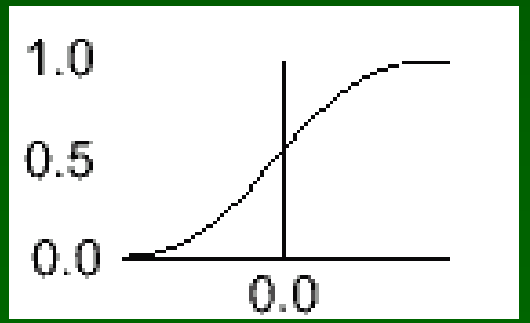
Wejścia: dane ciągłe lub dyskretne.
Neuron sumuje wartości sygnałów wejściowych określając pobudzenie.

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

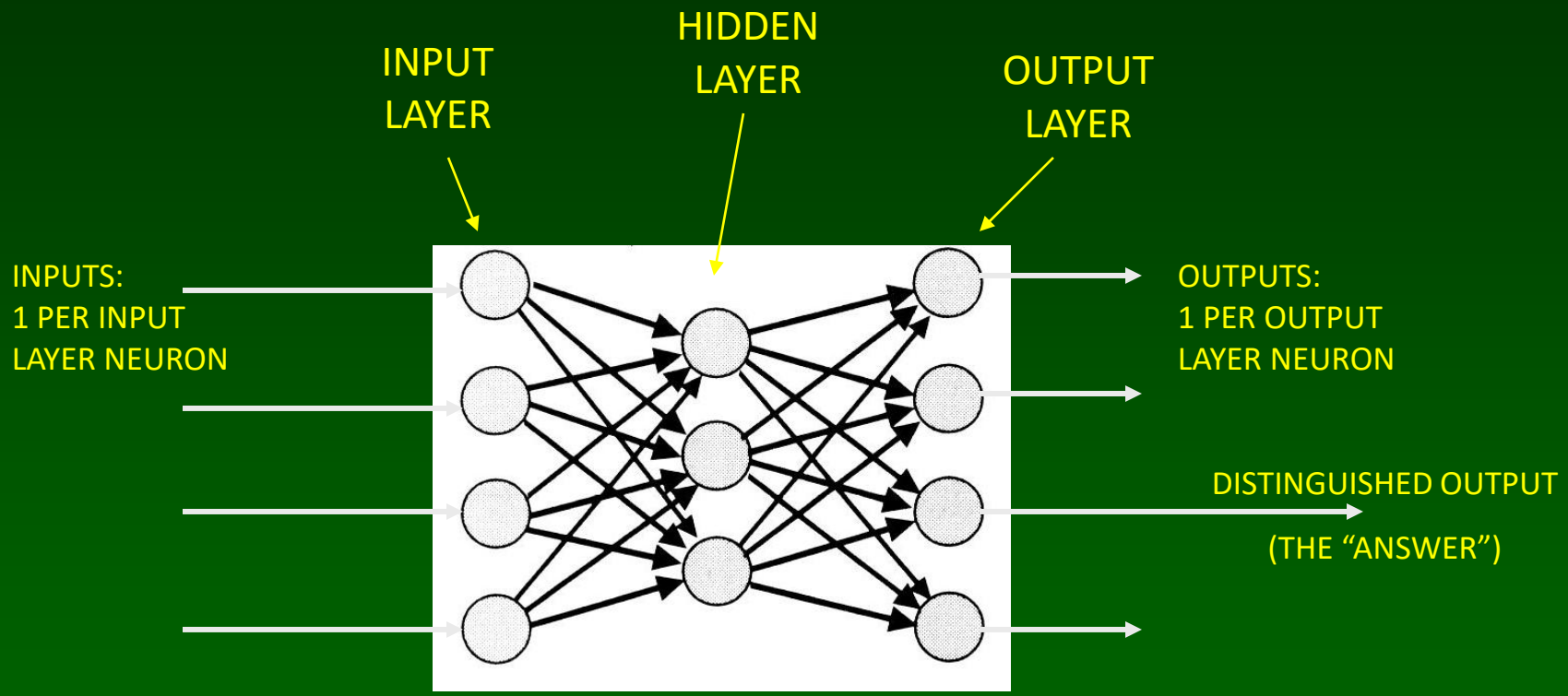


$f(\cdot)$ - funkcja schodkowa lub sigmoidalna
q - prób działania neuronu.

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



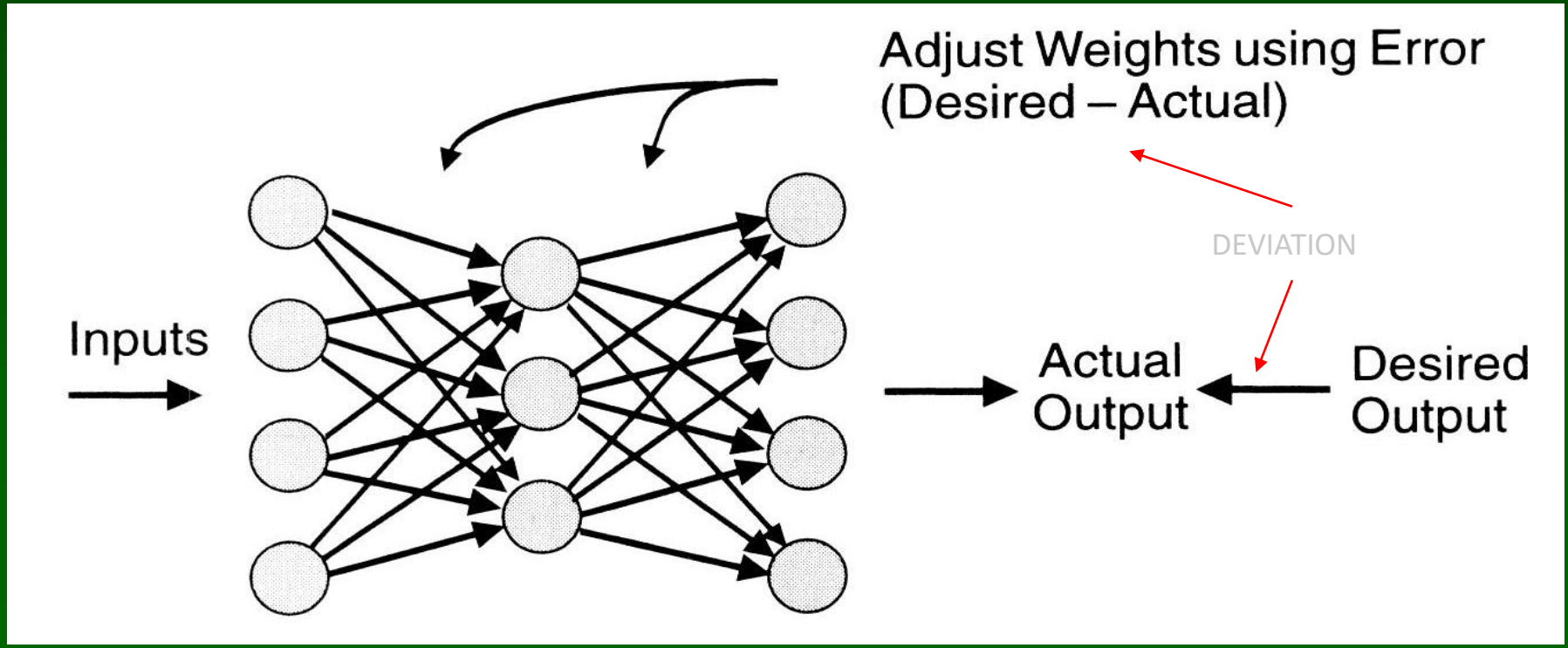
Neural Networks



Wsteczna propagacja błędów

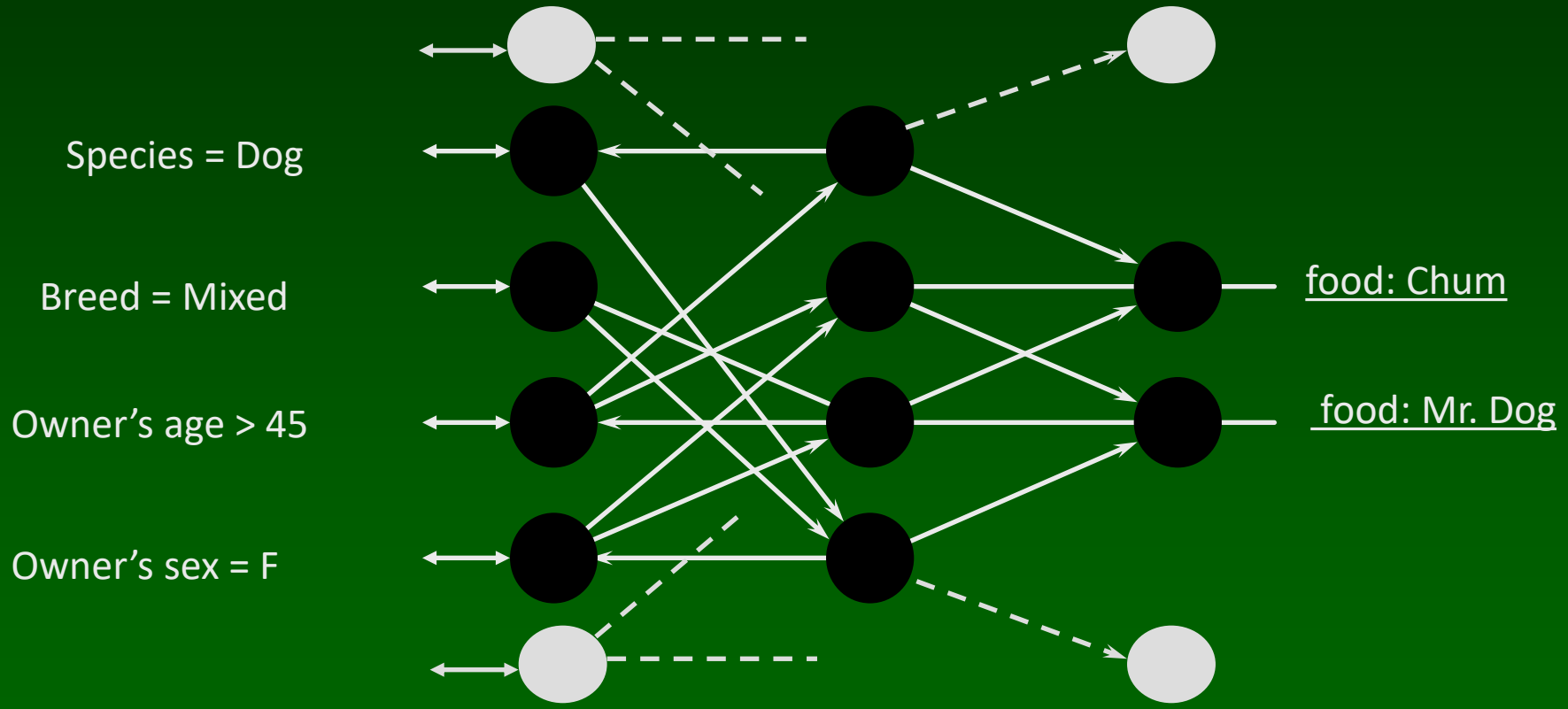
Algorytm rozpowszechniony od 1986 roku. Sieć ma wiele warstw elementów, zwanych neuronami.

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



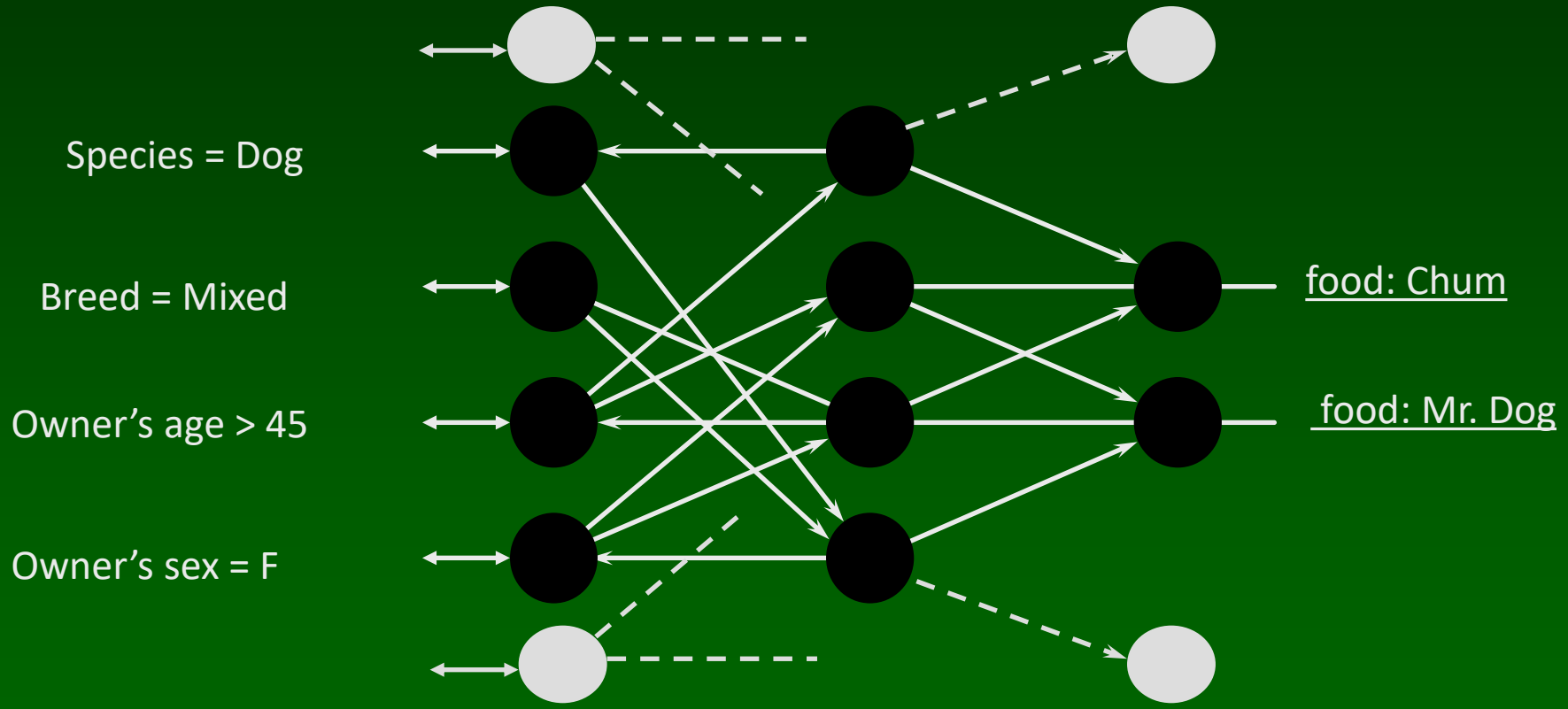
Klasyfikacja za pomocą sieci neuronowych

“Jakie czynniki wpływają na wybór ulubionej karmy psów?”

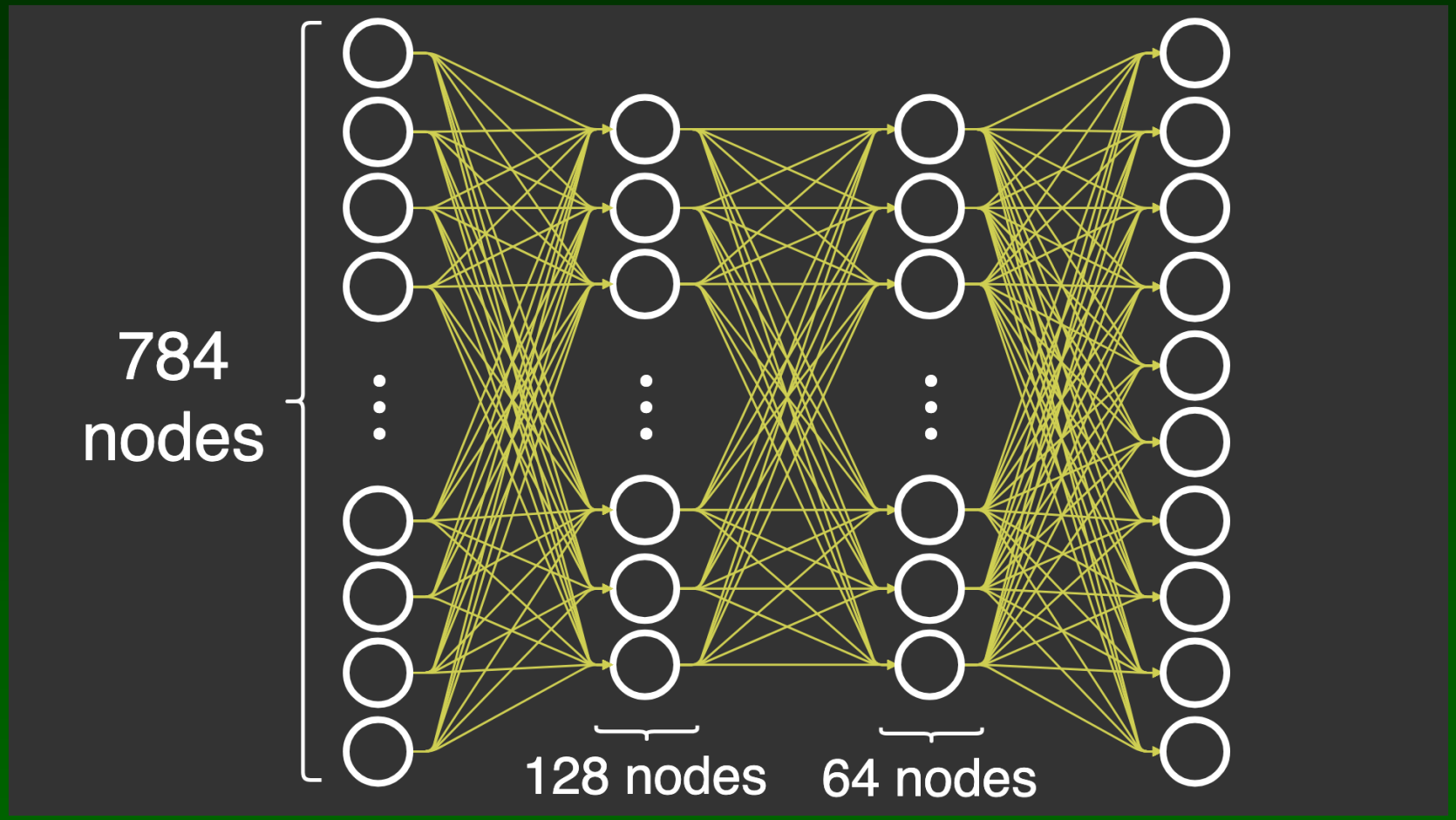


Klasyfikacja za pomocą sieci neuronowych

“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:

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

48 błędów, 99.41% prawidłowych

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

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

R_2) spore_print_color = green 48 bł., 99.41%

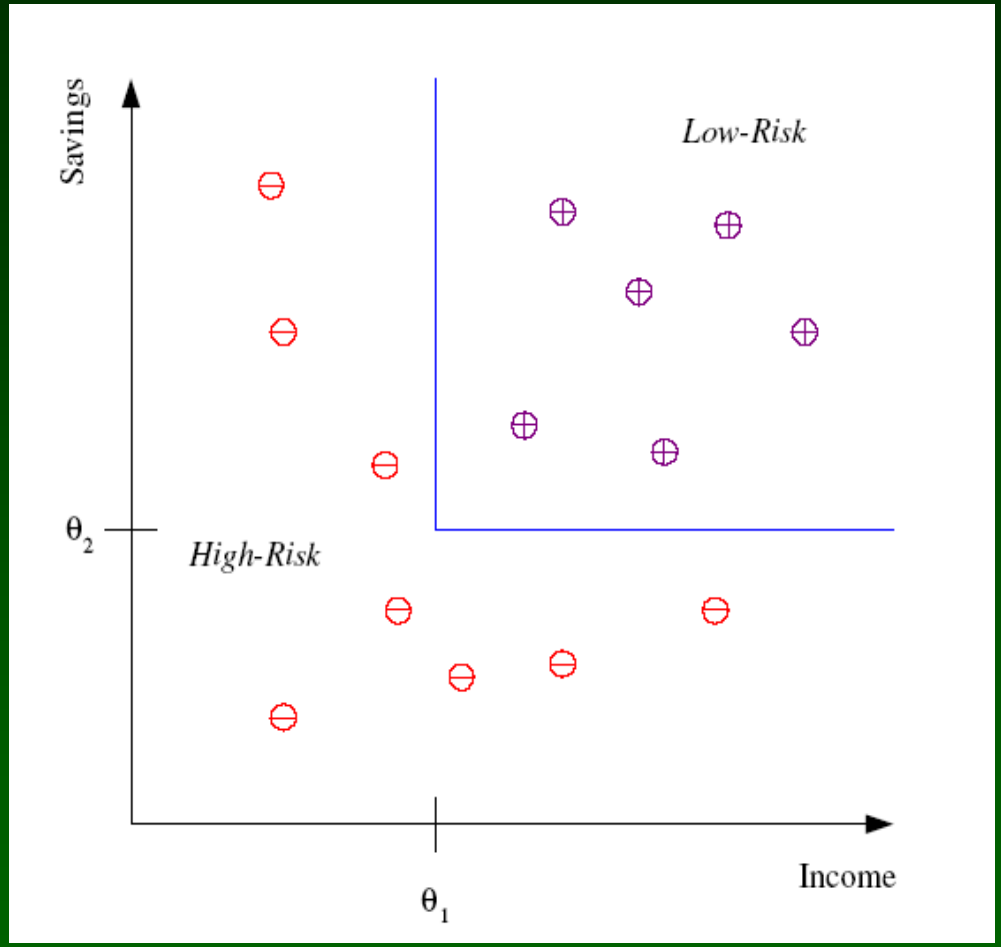
R_3) odor = none $\wedge$ stalk_surface_below_ring = scaly $\wedge$ stalk_color_above_ring = not.brown
8 bł., 99.90%

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

Teraz widać, dlaczego węch jest tak ważny dla zwierząt.

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*

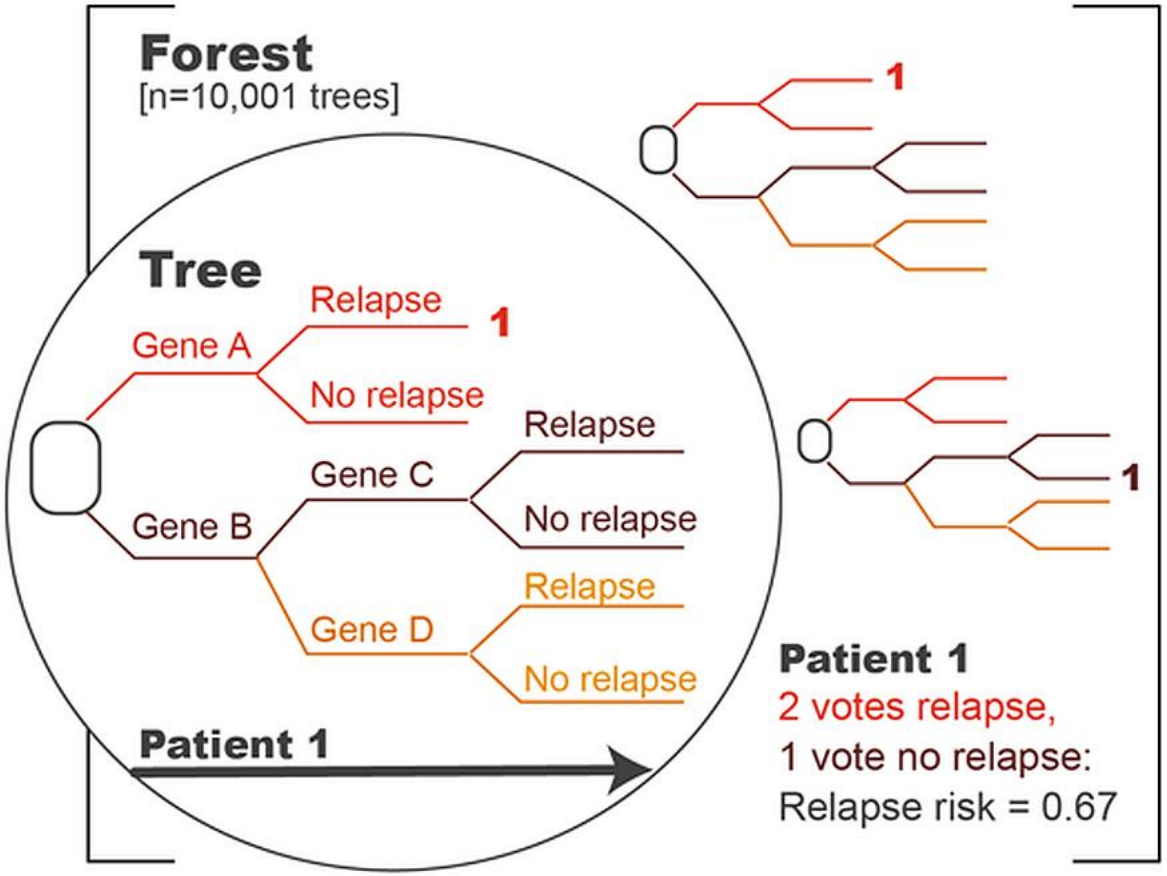


Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

Methods: Random Forest

Random forest

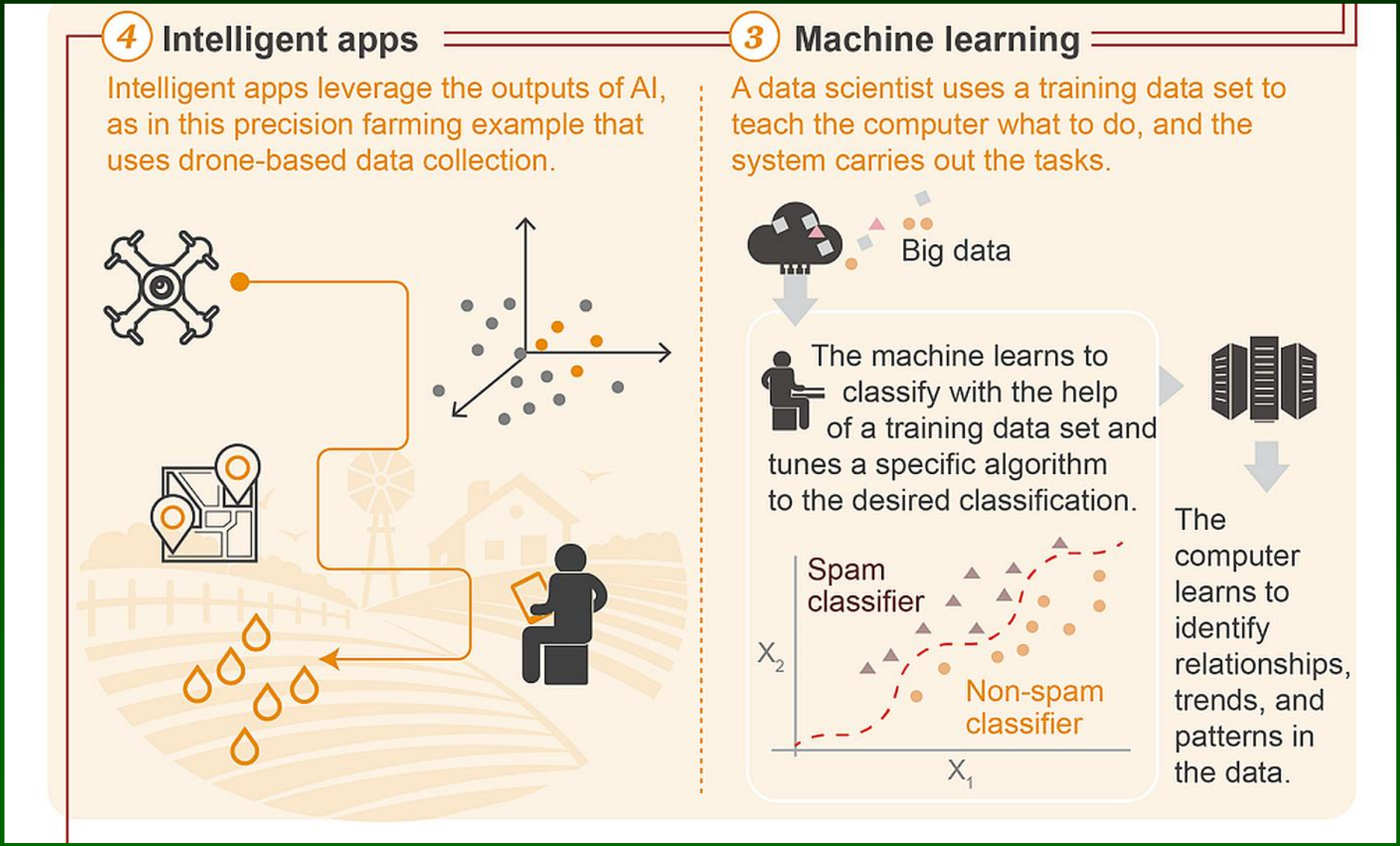
Random forest algorithms improve the accuracy of decision trees by using multiple trees with randomly selected subsets of data. This example reviews the expression levels of various genes associated with breast cancer relapse and computes a relapse risk.



Advantages	Use cases
Random forest methods prove useful with large data sets and items that have numerous and sometimes irrelevant features.	Customer churn analysis, risk assessment

Source: Nicolas Spies, Washington University, 2015

DM & Machine Learning



ML applications

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.

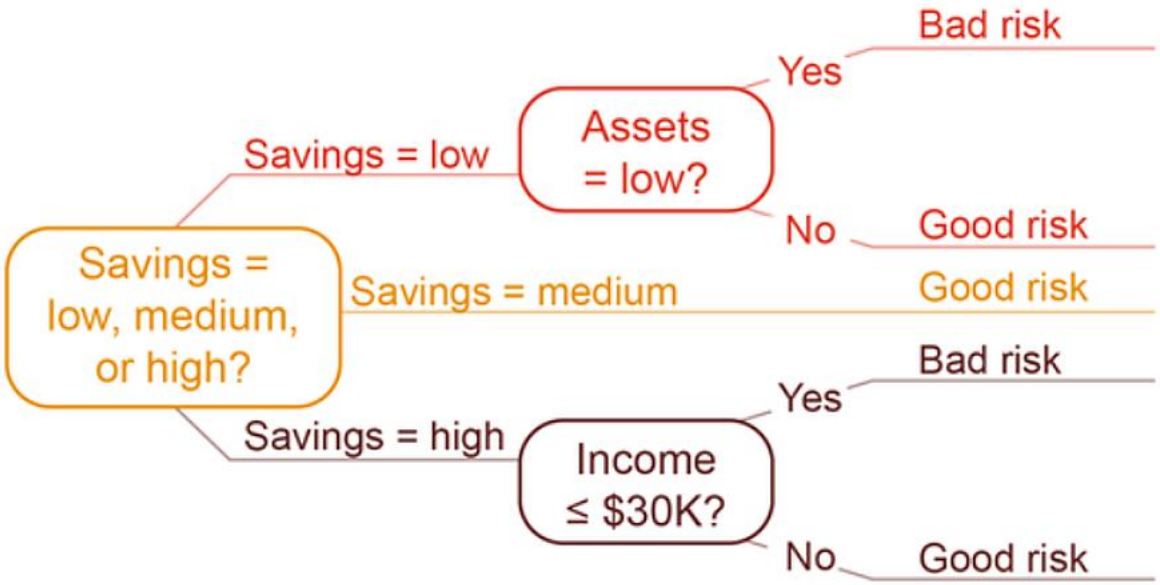
pwc.com/NextinTech

pwc

Methods: Decision Trees

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.



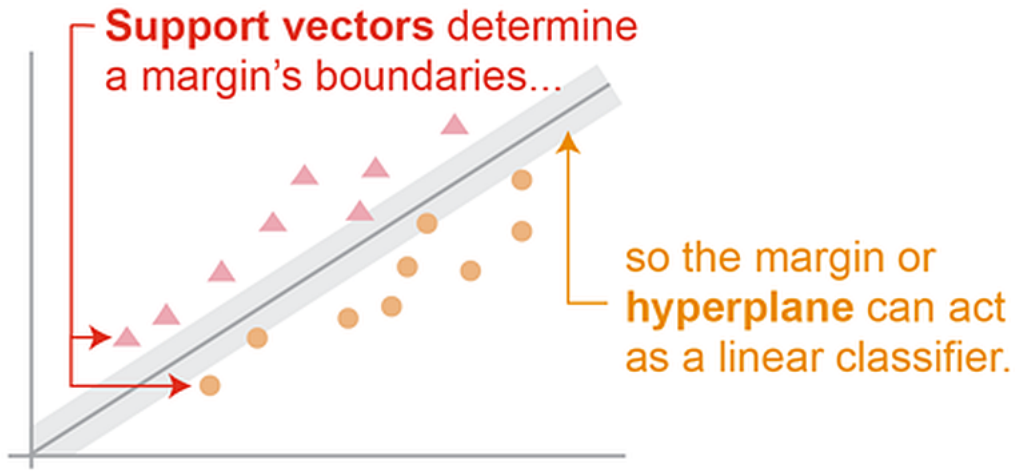
Advantages	Use cases
Decision trees are useful when evaluating lists of distinct features, qualities, or characteristics of people, places, or things.	Rule-based credit risk assessment, horse race performance prediction

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

Methods: SVM

Support vector machines

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



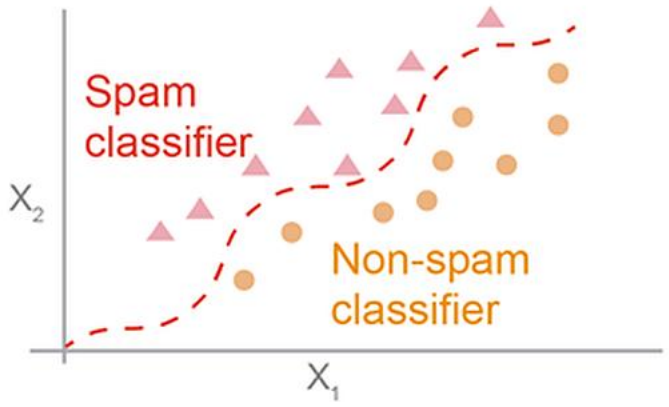
Advantages	Use cases
Support vector machines are good for the binary classification of X versus other variables and are useful whether or not the relationship between variables is linear.	News categorization, handwriting recognition

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

Methods: Regression

Regression

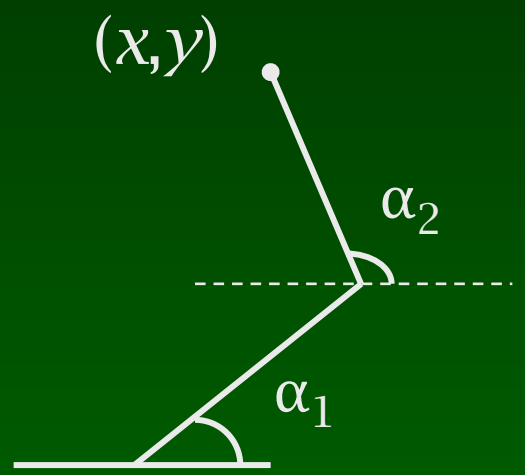
Regression maps the behavior of a dependent variable relative to one or more dependent variables. In this example, logistic regression separates spam from non-spam text.



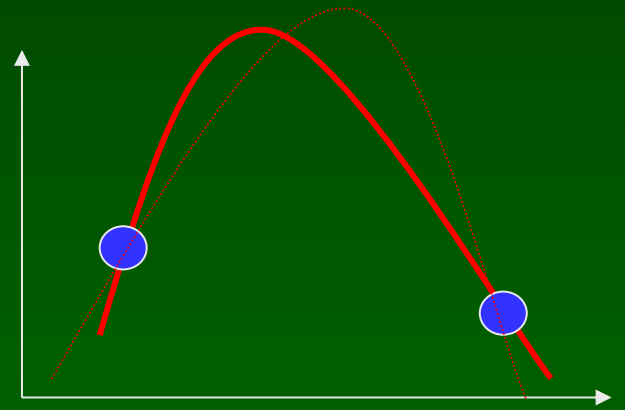
Advantages	Use cases
Regression is useful for identifying continuous (not necessarily distinct) relationships between variables.	Traffic flow analysis, email filtering

Regression Example

- Navigating a car: Angle of the steering wheel (CMU NavLab)
- Kinematics of a robot arm



$$\alpha_1 = g_1(x,y)$$
$$\alpha_2 = g_2(x,y)$$

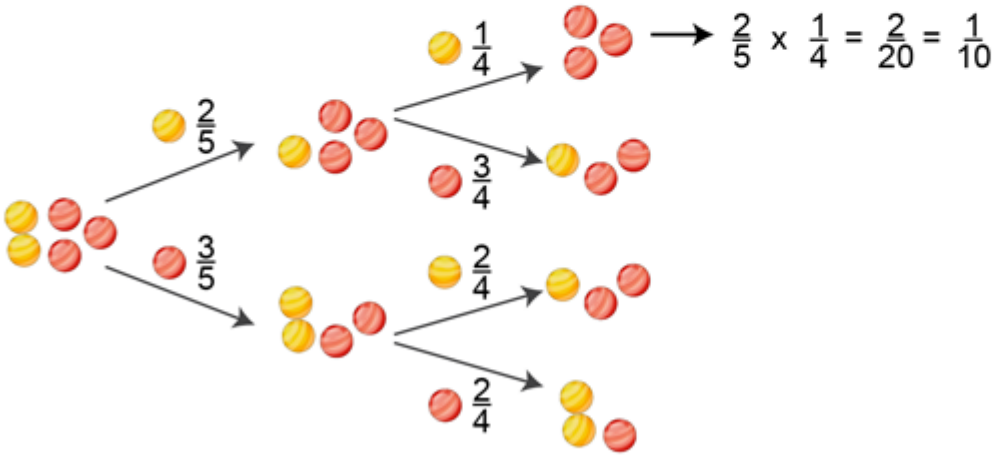


Control of cars or robots require calculation of continuous parameters. Linear regression is commonly used in social sciences, but the challenge is to predict complex nonlinear trajectories with many parametrers.

Methods: Probabilistic NB

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.



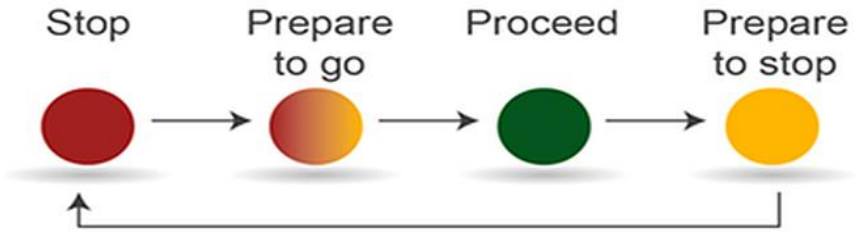
Advantages	Use cases
Naive Bayes methods allow the quick classification of relevant items in small data sets that have distinct features.	Sentiment analysis, consumer segmentation

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

Metody: Ukryte modele Markowa

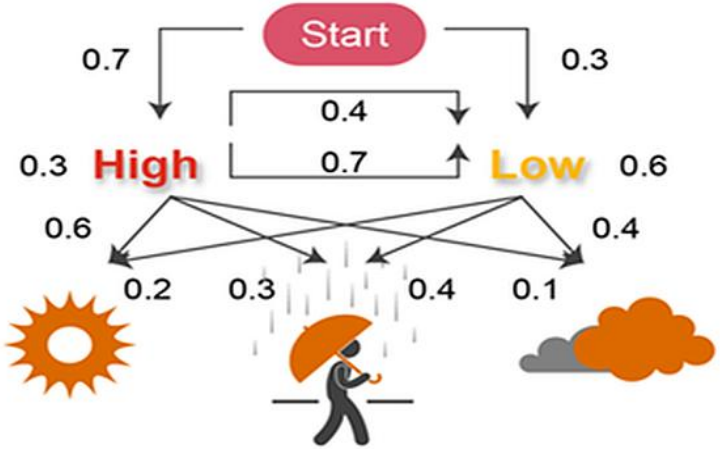
Hidden Markov models

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

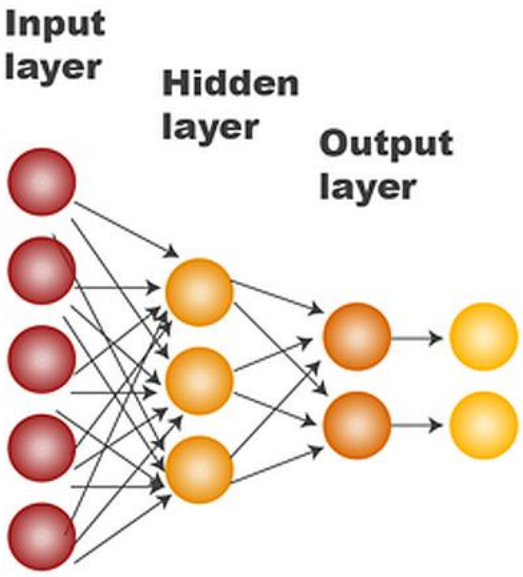
Advantages	Use cases
Tolerates data variability and effective for recognition and prediction.	Facial expression analysis, weather prediction

Metody: Sieci neuronowe

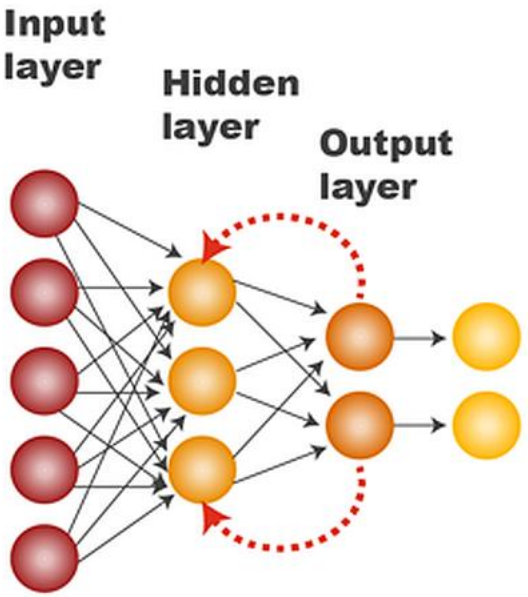
Recurrent neural networks

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



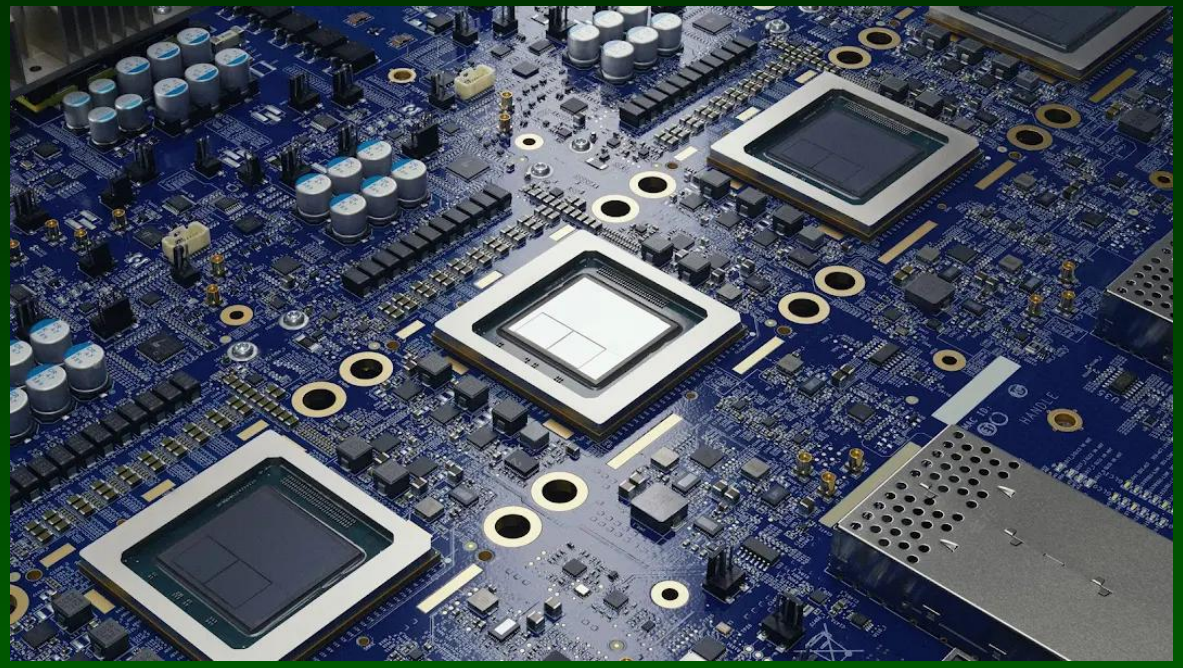
Advantages	Use cases
Recurrent neural networks have predictive power when used with large amounts of sequenced information.	Image classification and captioning, political sentiment analysis

Source: Joseph Wilks, 2012

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.

AlphaChip



AI has accelerated and optimized chip design, and its superhuman chip layouts are used in hardware around the world. Such layouts were used in the last three generations of Google's custom AI accelerator, the Tensor Processing Unit (TPU). AI is here a partner helping humans.

Mirhoseini, A., Goldie, A., Yazgan, M. *et al.* A graph placement methodology for fast chip design. *Nature* **594**, 207–212 (2021), Addendum 26.09.2024

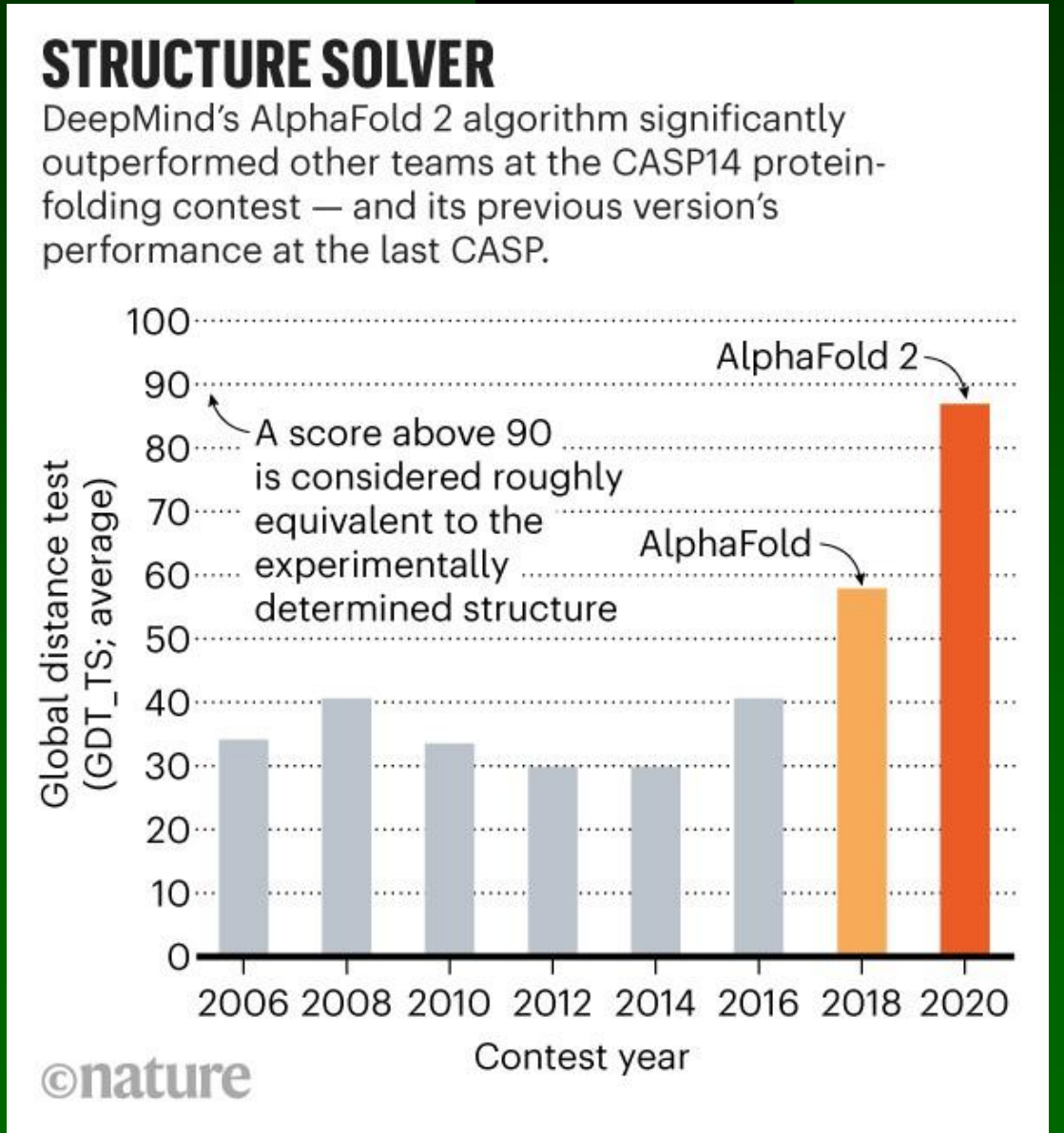
Zwijanie białek

AlphaFold 2 wykorzystując głębokie uczenie przewiduje ponad 2/3 struktur białek z dokładnością równoważną eksperymentalnej!

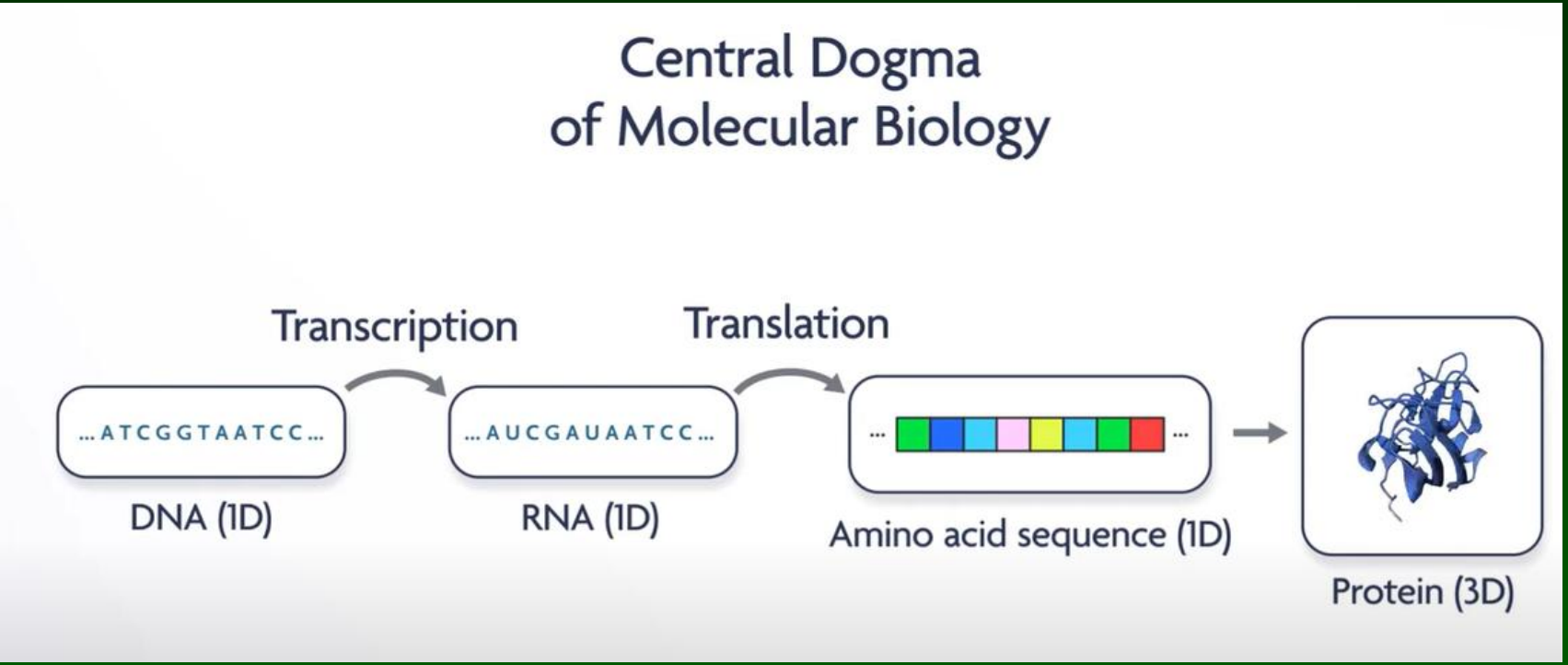
Nature, 30.11.2020

Rozpoznawanie struktur + uczenie się +wnioskowanie.

Przewidywanie struktur białek na podstawie sekwencji aminokwasów jest podstawą poszukiwania białek i projektowania leków o pożądanych własnościach.

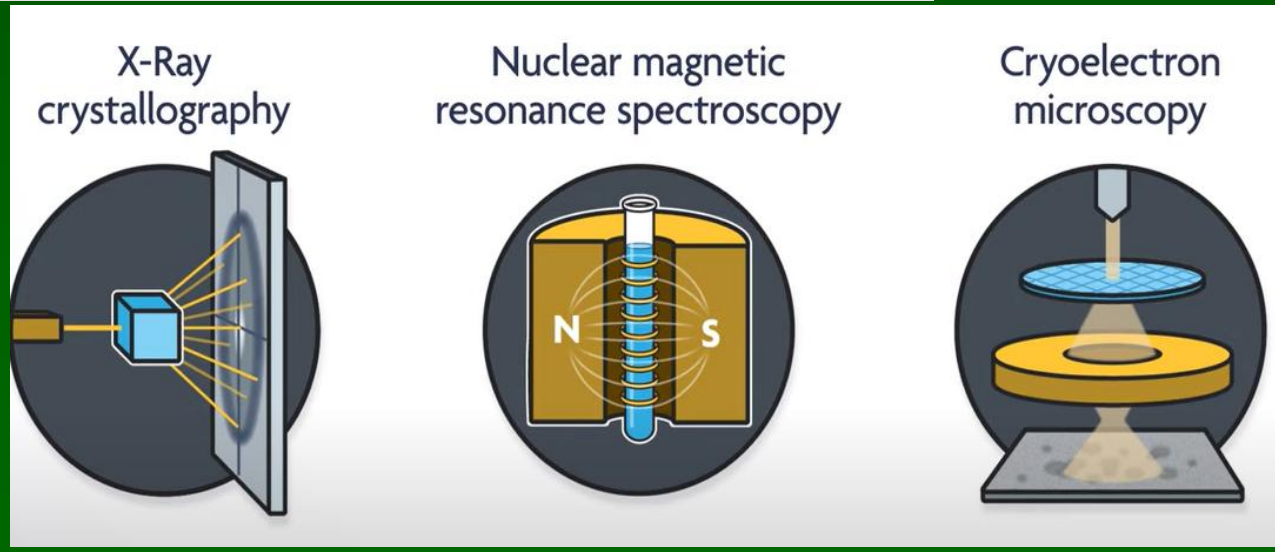


AI i zwijanie białek

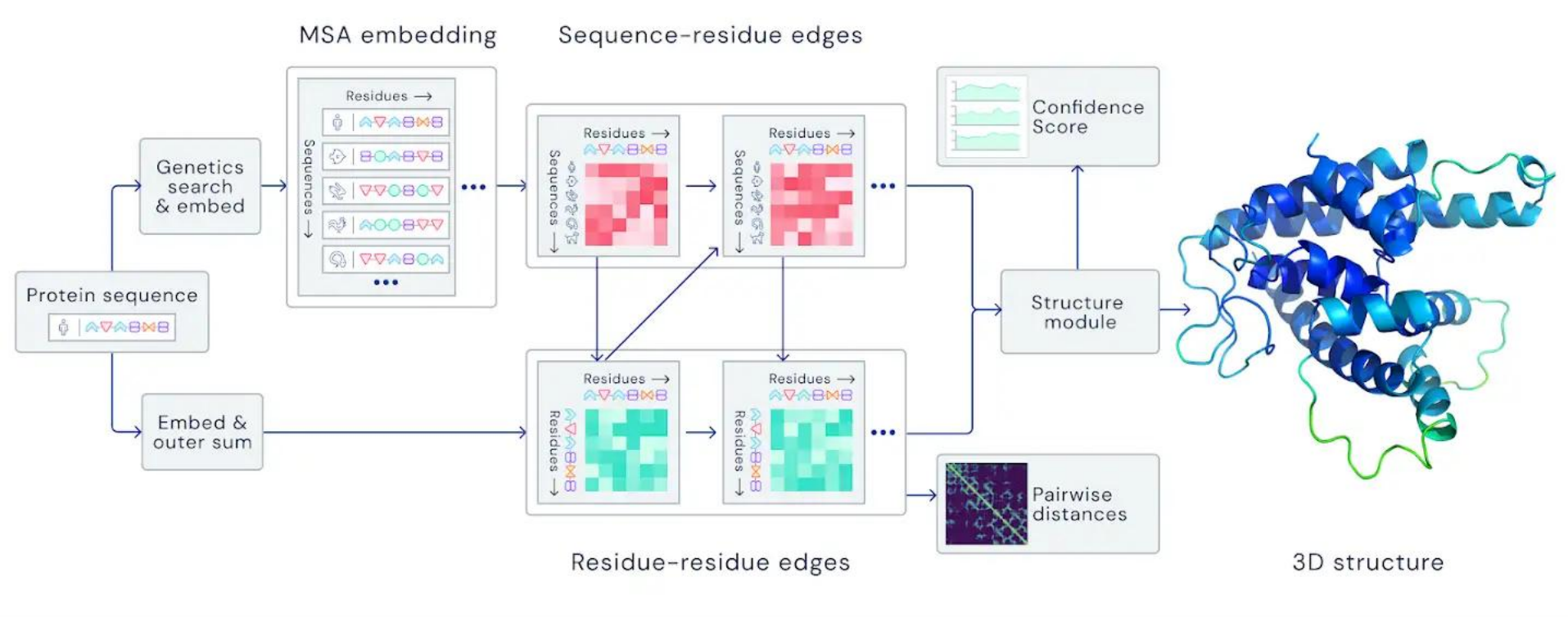


R. B. Altman, A Holy Grail — The Prediction of Protein Structure, New England J. of Medicine **389** (2023)

Google Deep Mind AlphaFold 2 has reached accuracy comparable to experiments on over 100.000 proteins.



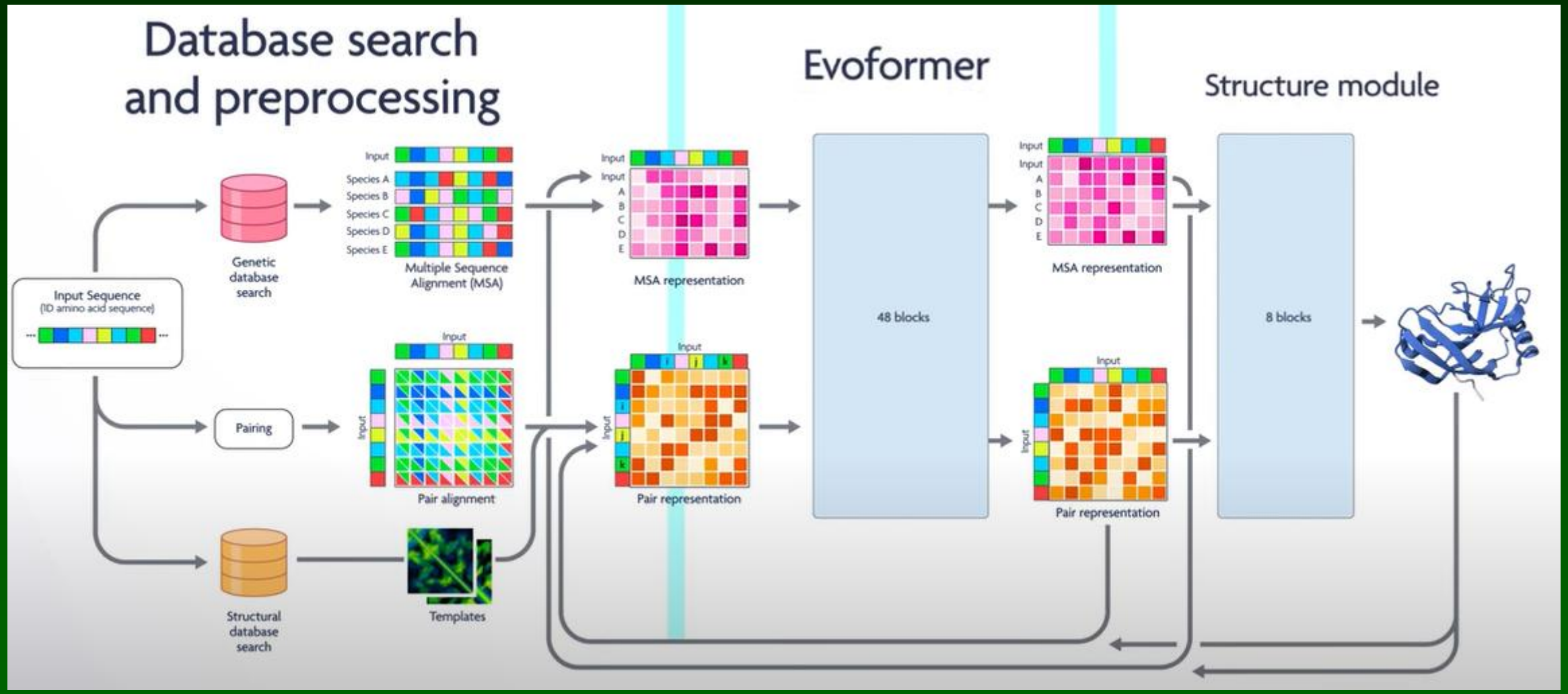
AlphaFold



Program trenowany na 170.000 struktur białek w otwartej bazy przez kilka tygodni na 100 GPU. Ważna jest metoda numerycznej reprezentacji symbolicznych danych (embedding), czyli łańcuchów aminokwasów.

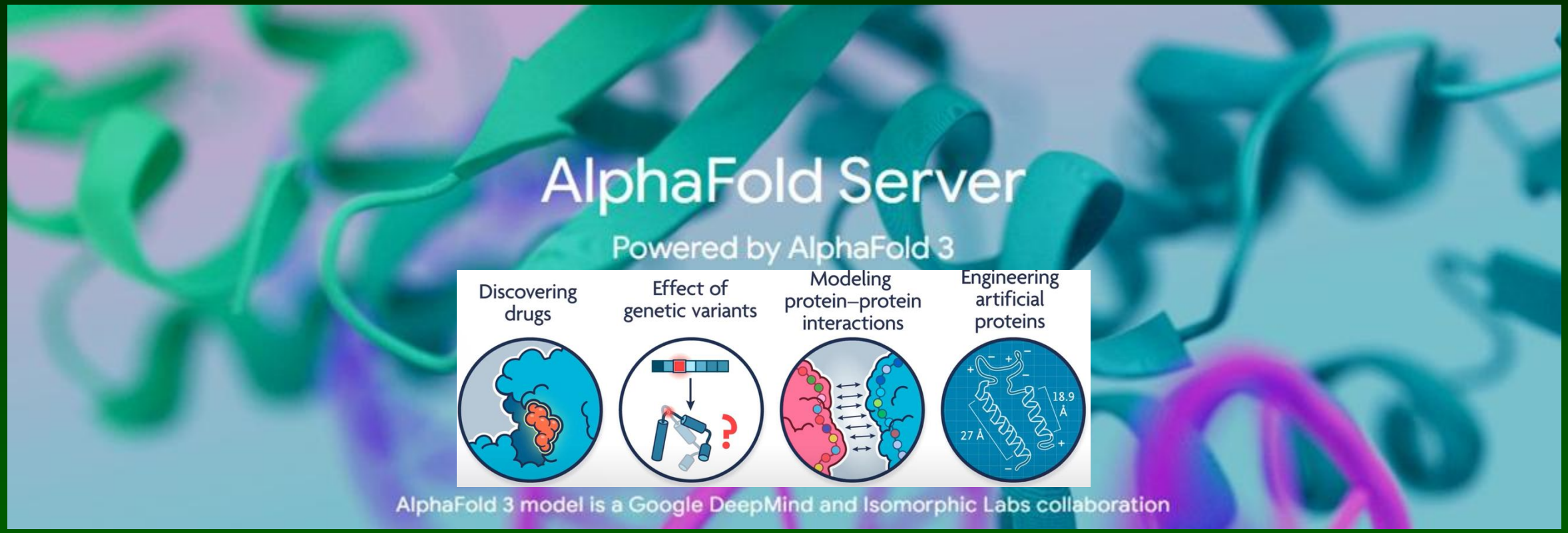
Szukanie parametrów sieci w tym przypadku oparte jest na minimalizacji funkcji błędu. 7/2022 DeepMind opublikował struktury 220 mln białek około miliona gatunków, Meta Metagenomic atlas ponad 600 mln struktur, 15 mld par.

AlphaFold architecture



AlphaFold require close collaboration with protein experts to use evolutionary information from genetic databases, find similar sequences, and compare pairing matrix to experimentally derived structures. Evoformer evaluates importance of positions in sequences and in sequences of similar species.

AlphaFold 3



J. Jumper + 32 coauthors + Demis Hassabis, Highly accurate protein structure prediction with AlphaFold, Nature **596**, 583 (2021). Nobel z Chemii, 2024.
Znacznie ulepszona wersja: J. Abramson + 56 coauthors, Accurate structure prediction of biomolecular interactions with AlphaFold 3, Nature **630**, 493 (2024).

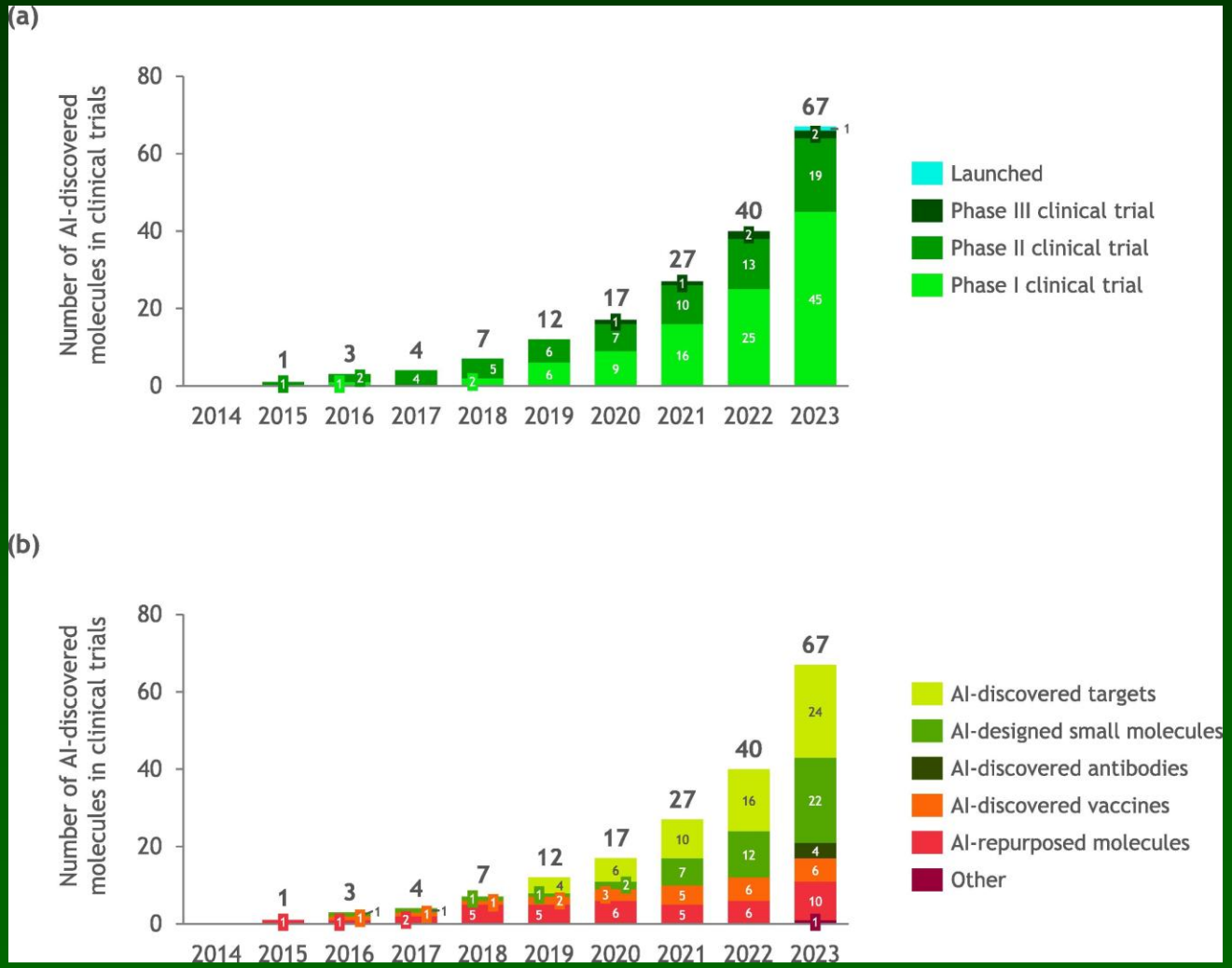
AlphaFold 3 predicts the structure and interactions of all of life's molecules, AlphaProteo generates novel proteins for biology and health accelerating research in nearly every field of biology/molecular medicine.

AI-discovered drugs in clinical trials

Number of molecules discovered by AI-first Biotechs that have entered clinical trials. Includes molecules that were partnered with pharma companies and excludes COVID-19-related molecules.

(a) AI molecules by clinical Phase.

(b) AI-discovered molecules by mode-of-discovery.



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