

# 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

# Co będzie



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

# Uczenie maszynowe

- Machine learning (ML), to część sztucznej inteligencji, symbolicznej AI lub obliczeniowej (CI, computational intelligence) zajmującej się akwizycją wiedzy z danych.
- Uczenie maszynowe umożliwia mechanizację akwizycji wiedzy.
- 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.
- Wielkie zainteresowanie ML nastąpiło dzięki dobrym wynikom analizy obrazów w dekadzie 2010-20.
- 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 skraca się czas rozwoju złożonych systemów.  
BMT Group Ltd (British Maritime Technology), sterowanie statkami.

# Uczenie maszynowe

- System uczący się używa danych by utworzyć ich model, pozwalający na przewidywanie zachowania dla przyszłych danych.
- Uczenie: zmiany w systemie adaptującym się pozwalające mu w przyszłości działać bardziej efektywnie na zadaniach o podobnym lub analogicznym charakterze.
- Czasami wiedzę można wyrazić w postaci symbolicznej, ale często zdobywamy ją bezpośrednio z obserwacji za pomocą zmysłów.
- Ekspertci opisują sytuacje podając przykłady. W GOFAI był to głównie opis symboliczny ale to bardzo ograniczona metoda.
- Problemy często redukują się do klasyfikacji (diagnozy).
- Reguły potrzebne do wnioskowania otrzymane metodami ML mogą być lepsze niż reguły wydedukowane przez ludzi.
- ML oferuje szybszą drogę do tworzenia systemów eksperckich, a w skomplikowanych sytuacjach może to być jedyna droga.

# Motivation: data=>knowledge



- AI systems: “spoon-feeding” knowledge instead of autonomous learning. Meanwhile: computer memory increased, database and sensor technology created data explosion problem.
- Information repositories, data warehouses keep all kinds of data, internet became a source of data about people’s behavior.
- **We are drowning in data, but starving for knowledge!**
- Search for knowledge in data – use on-line analytical processing (OLAP) and machine learning techniques for data mining.
- Extract interesting knowledge (rules, regularities, patterns, constraints, correlations, summaries) from large databases.
- 1990-2000: data mining and data warehousing, multimedia databases, and Web databases started the “big data” movement.
- Data scientist – currently (2018) the hottest job!

# ML definition

- Computer programs implement algorithms, if the problem is precisely defined and algorithms are efficient there is no need to learn. Most software in our computers is like that.
- Machine Learning: learning from data, measurements, tests, signals, images and video to improve performance, creating new features, adapting to environment.
- ML: programming computers to optimize some measures of performance, using examples, errors, failures and successes.
- Learning is needed in many situations:
  - Humans have no, or only limited, experience.
  - Humans are unable to verbalize their experience.
  - Adaptation to individual cases is needed.
  - Situation is changing in an unpredictable ways.

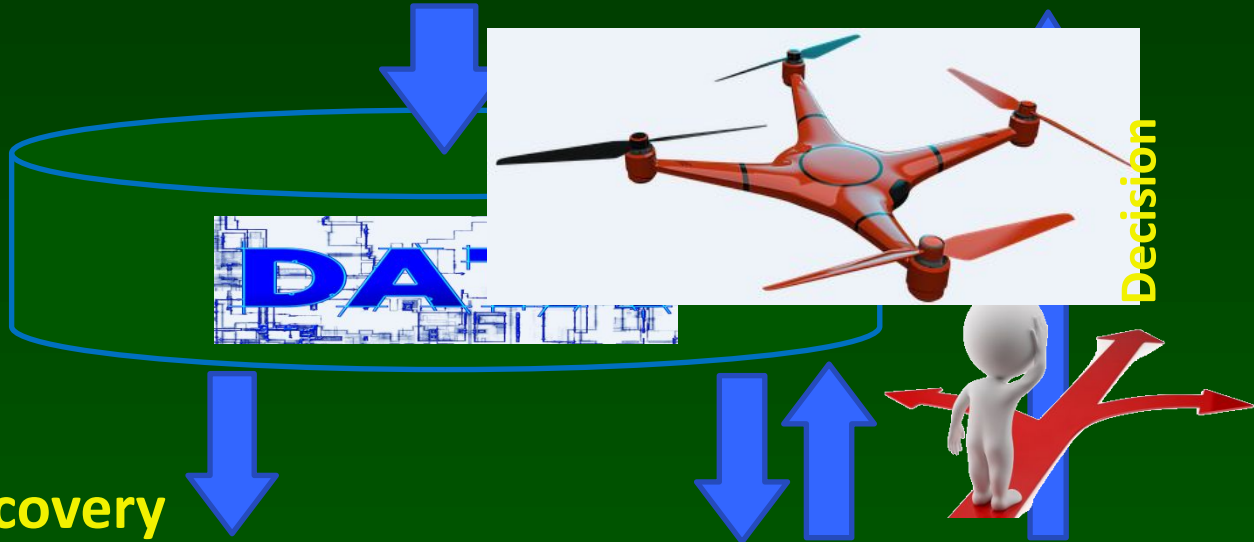




# Machine Learning Process

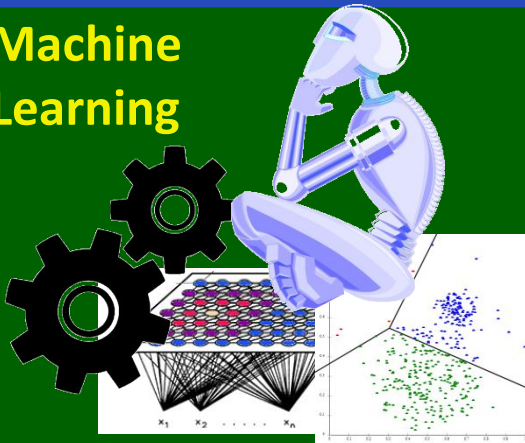


Environment

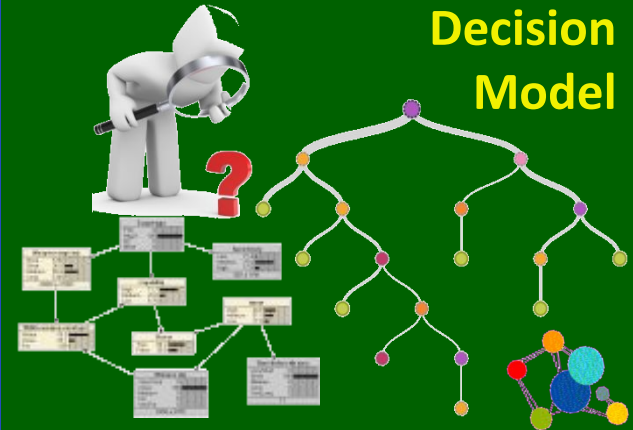


Data Mining & Knowledge Discovery

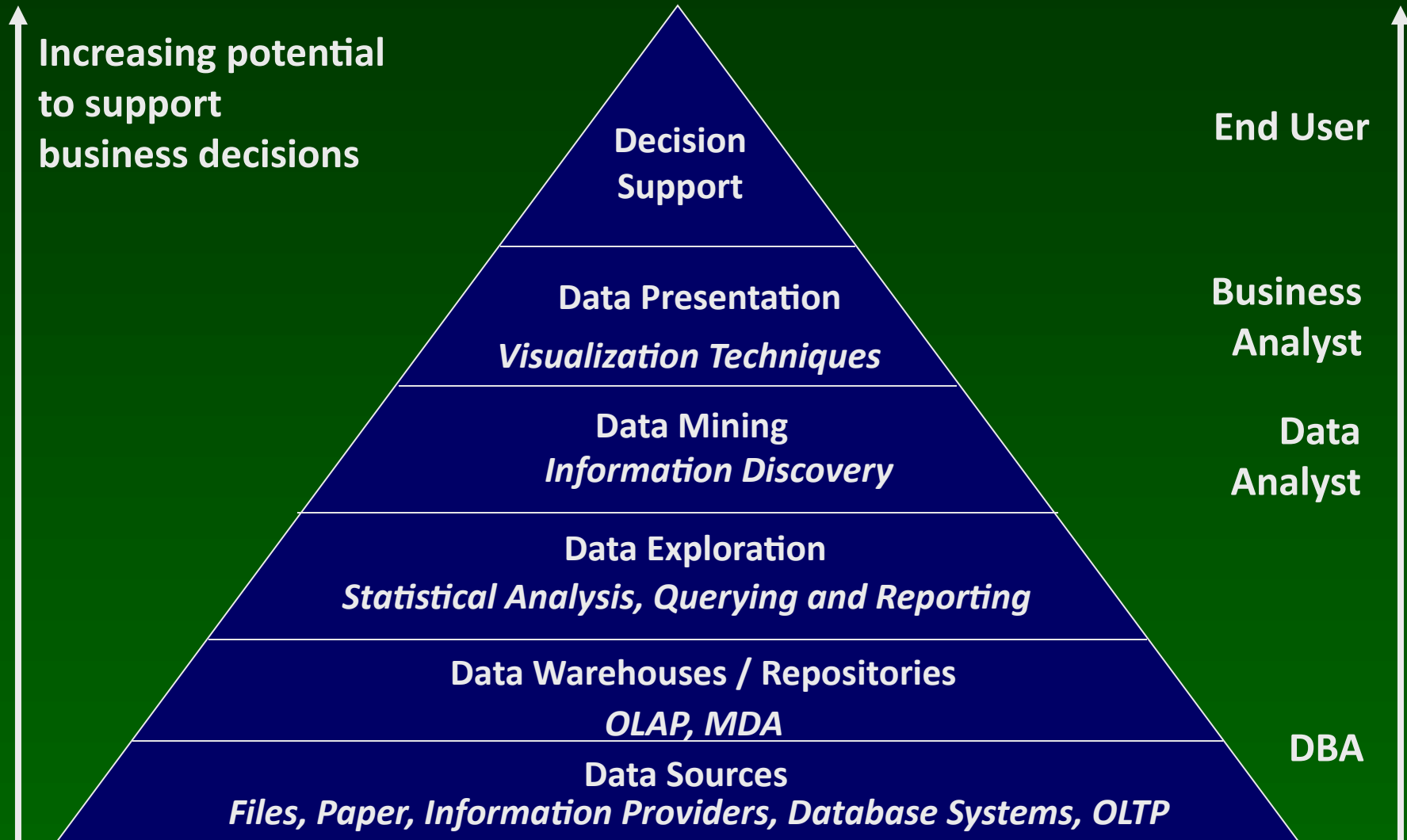
Machine Learning



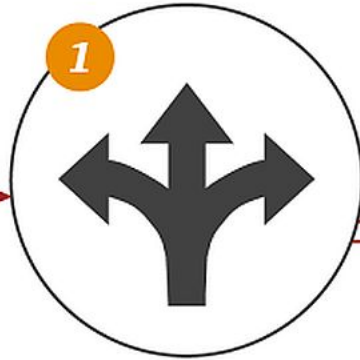
Decision Model



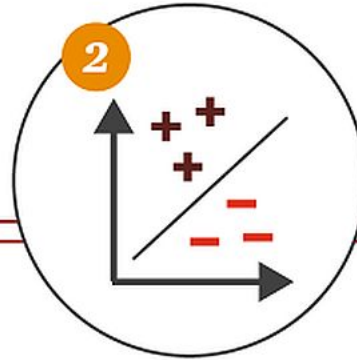
# Data Mining and Business Intelligence



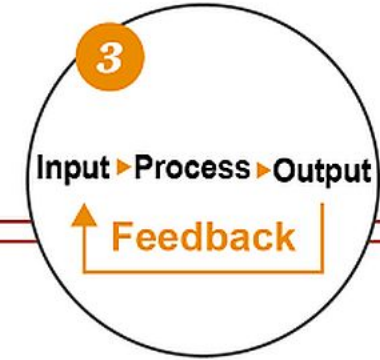
# Data Mining Process



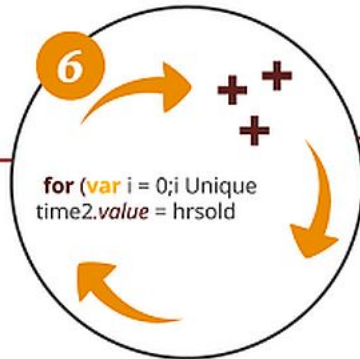
**1 Select data:**  
Split the data you have into three groups: training data, validation data, and test data.



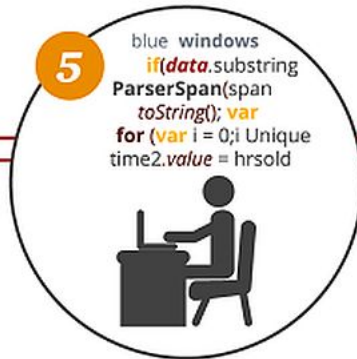
**2 Model data:**  
Use the training data to build the model using the relevant features.



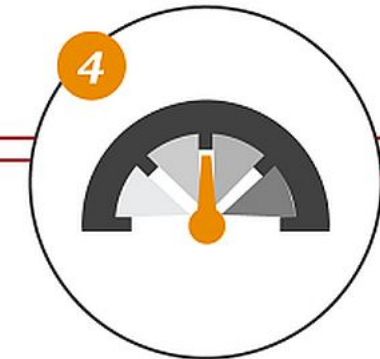
**3 Validate model:**  
Assess the model with your validation data.



**6 Tune model:**  
Improve performance of the algorithm with more data, different features, or adjusted parameters.



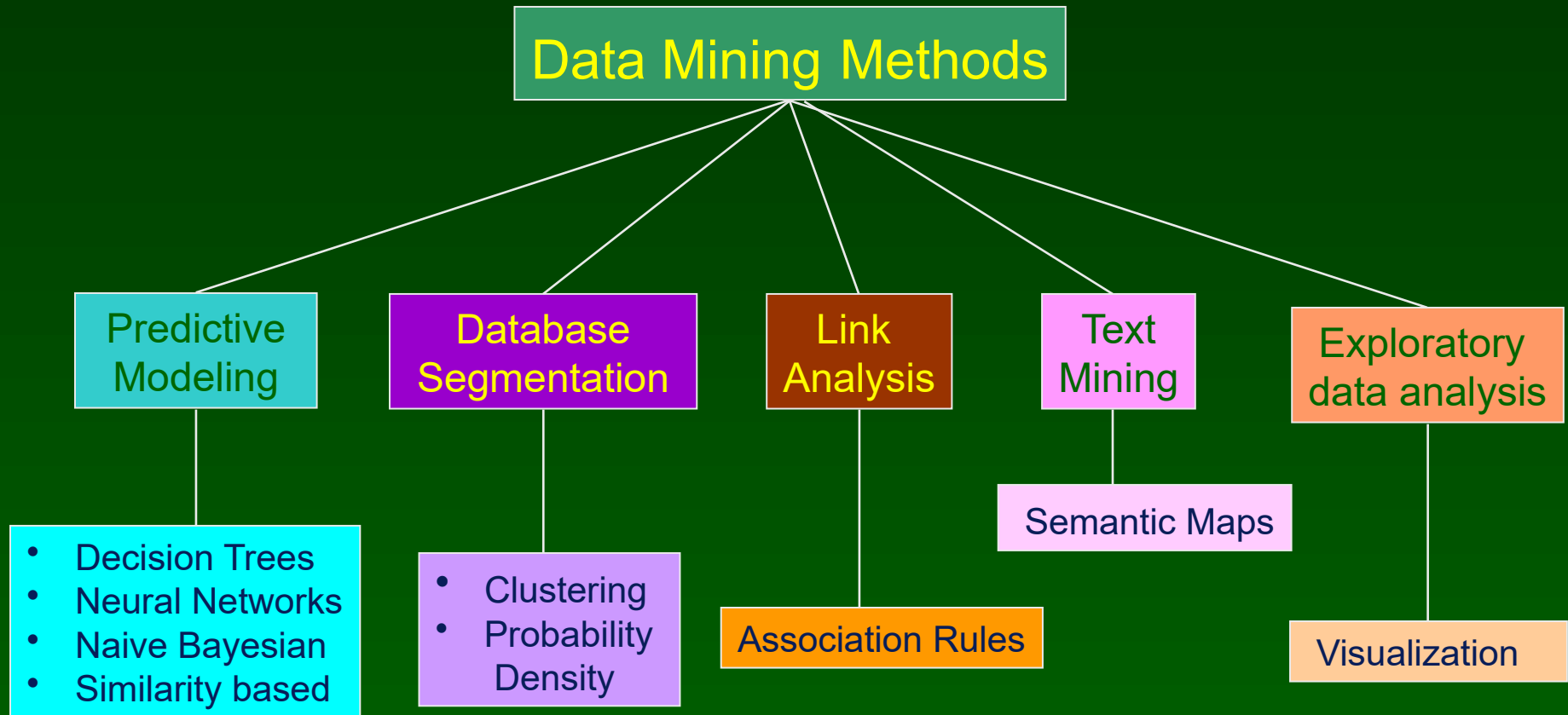
**5 Use the model:**  
Deploy the fully trained model to make predictions on new data.



**4 Test model:**  
Check performance of the validated model with your test data.

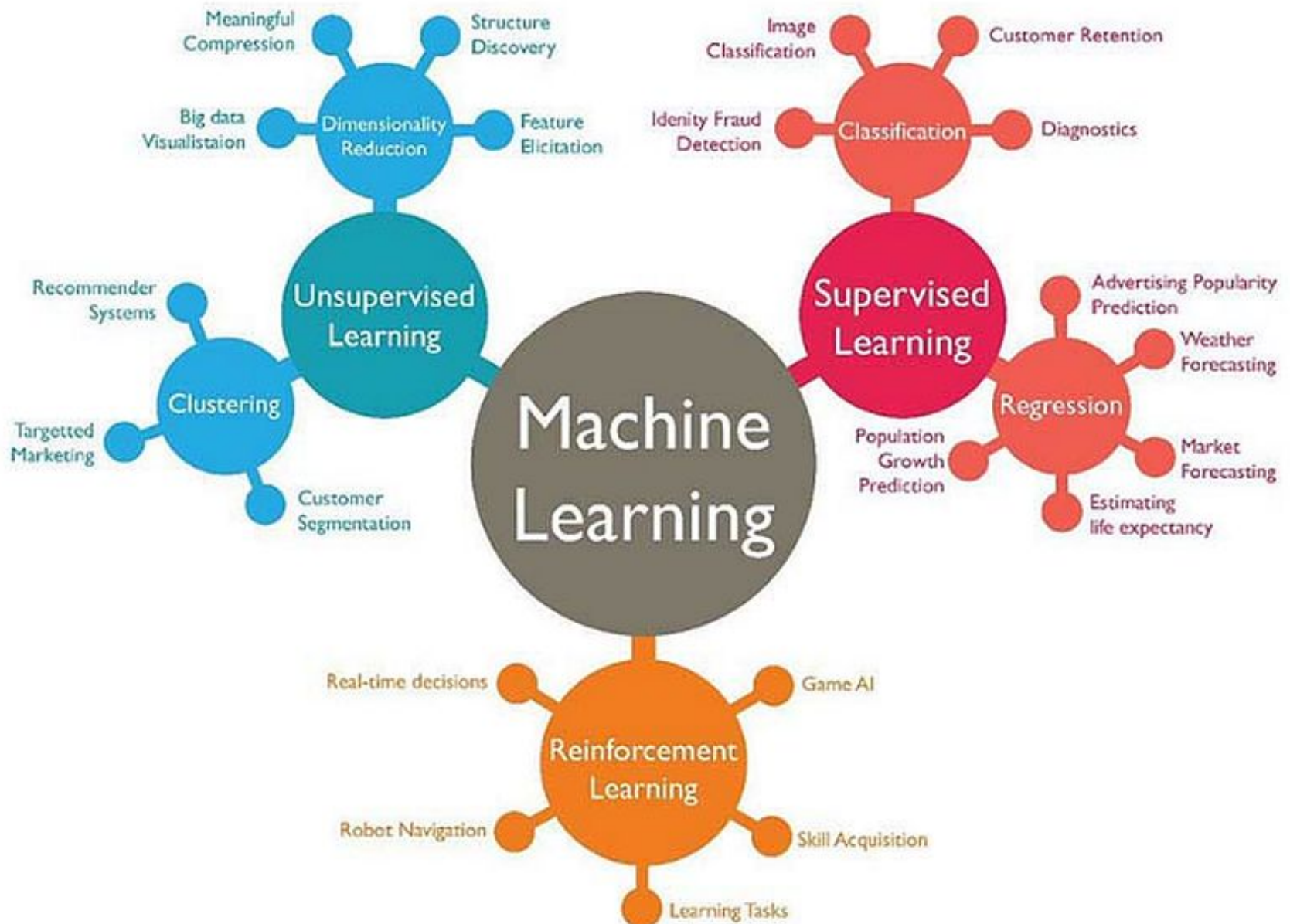
```
blue windows  
if(data.substring  
ParserSpan(span  
toString()); var  
for (var i = 0; i Unique  
time2.value = hrsold
```

# Taxonomy of Data Mining Methods

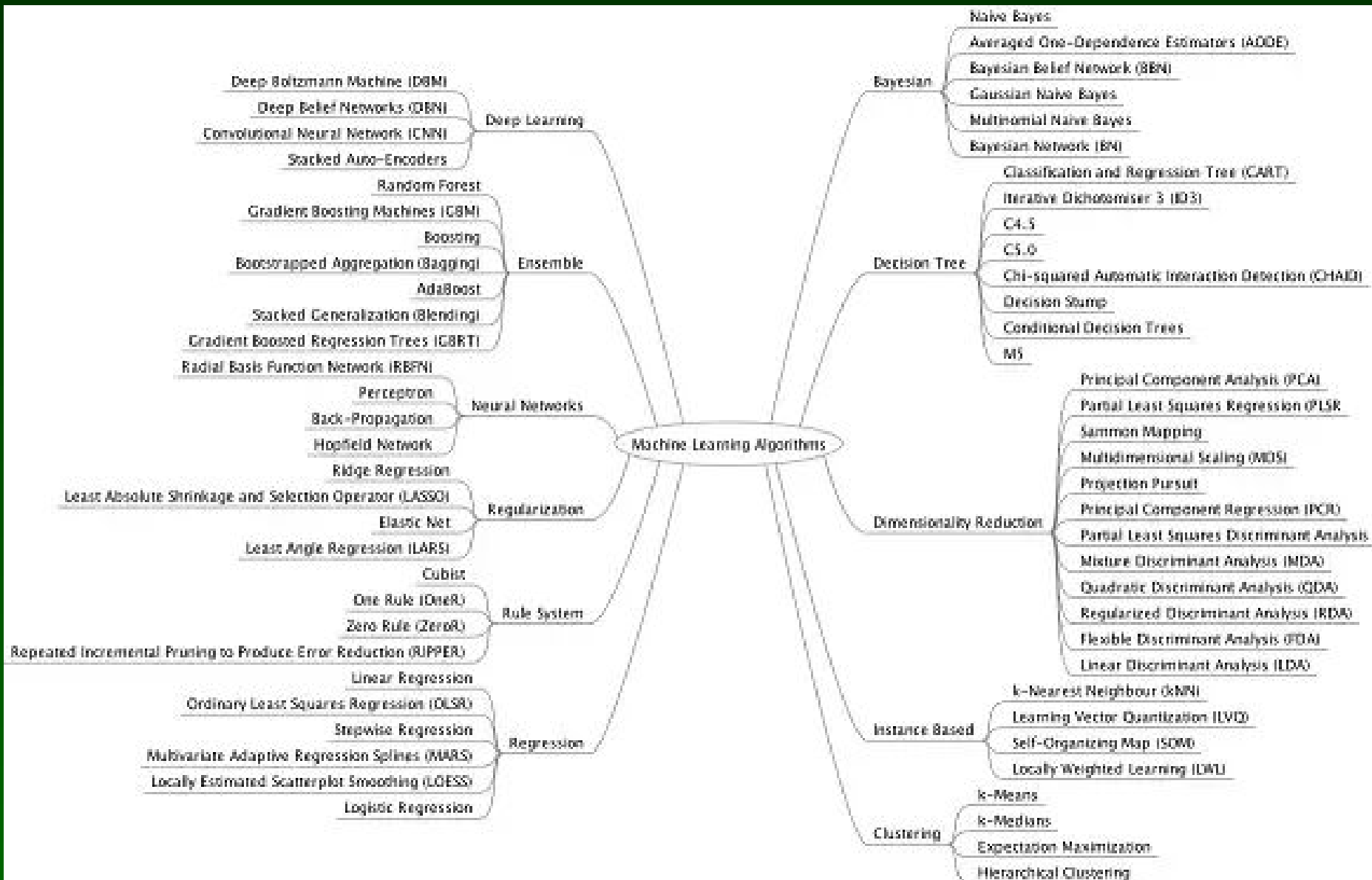


Machine Learning

# 3 branches of machine learning

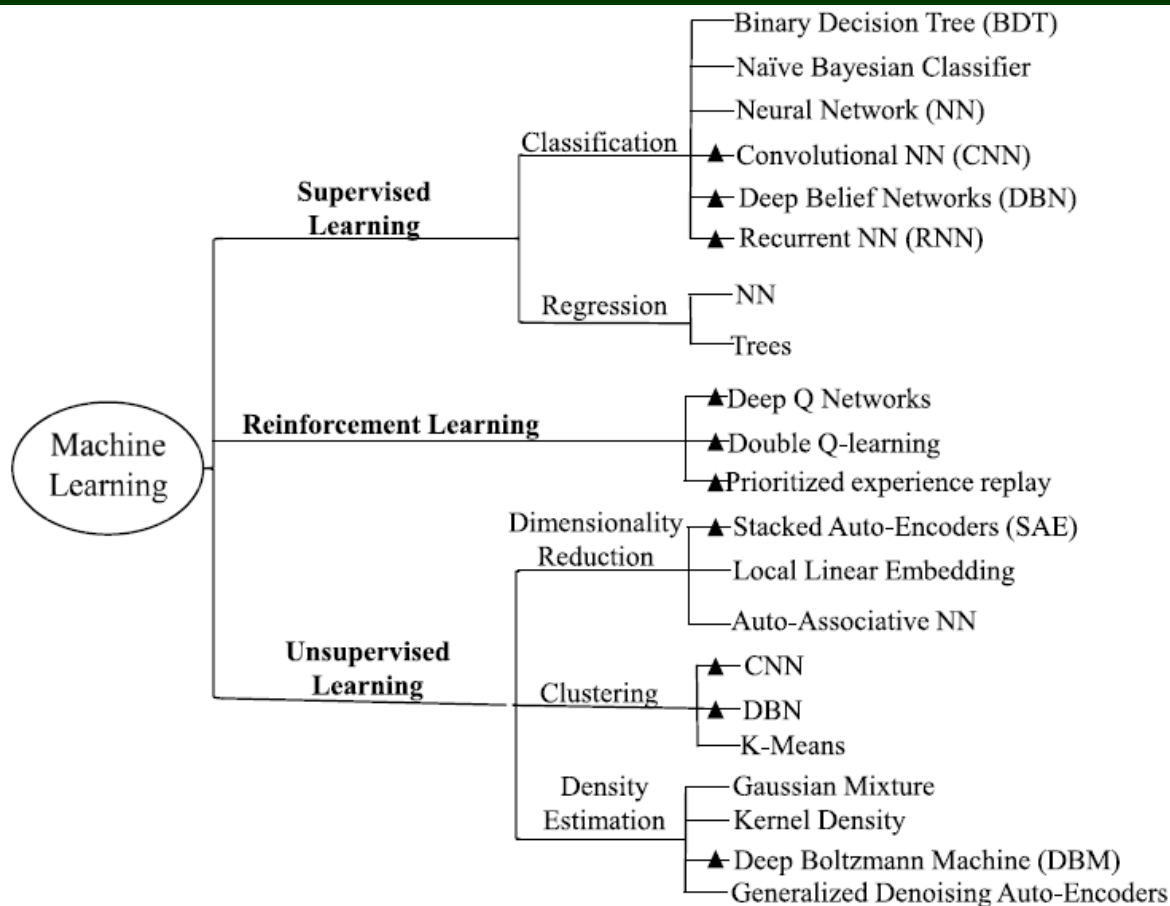


# Many types of ML algorithms





# Subtypes of ML



## Applications

Image classification; Character recognition; Facial recognition; Surveillance systems (intrusion detection, etc); Diagnostics

Advertising and business intelligence (Google ads, etc); Weather forecasting; Market forecasting; Political campaigns

Real-time decisions; Game Artificial Intelligence (AI); Learning tasks; Skill acquisition; Personal assistants (Google Now, Microsoft Cortana, Apple Siri, etc); Autonomous ("self-driving") cars

Big data visualization; Feature elicitation; Structure discovery; Meaningful compression

Recommendation engines (Amazon web service, Netflix, etc); Customer segmentation; Targeted marketing; Filtering algorithms; Newsfeeds

Economics (risk prediction, etc)

# Klasy metod ML

Powszechnie stosowane metody:

- Drzewa decyzji/indukcja reguł.
- Rozumowanie oparte na analogiach, precedensach (case-based)
- i oparte na pamięci (memory-based)
- SVM, Maszyny Wektorów Wsparcia (klasyfikacja, regresja).
- Sieci neuronowe w wielu odmianach.
- Logika rozmyta, systemy na regułach rozmytych.

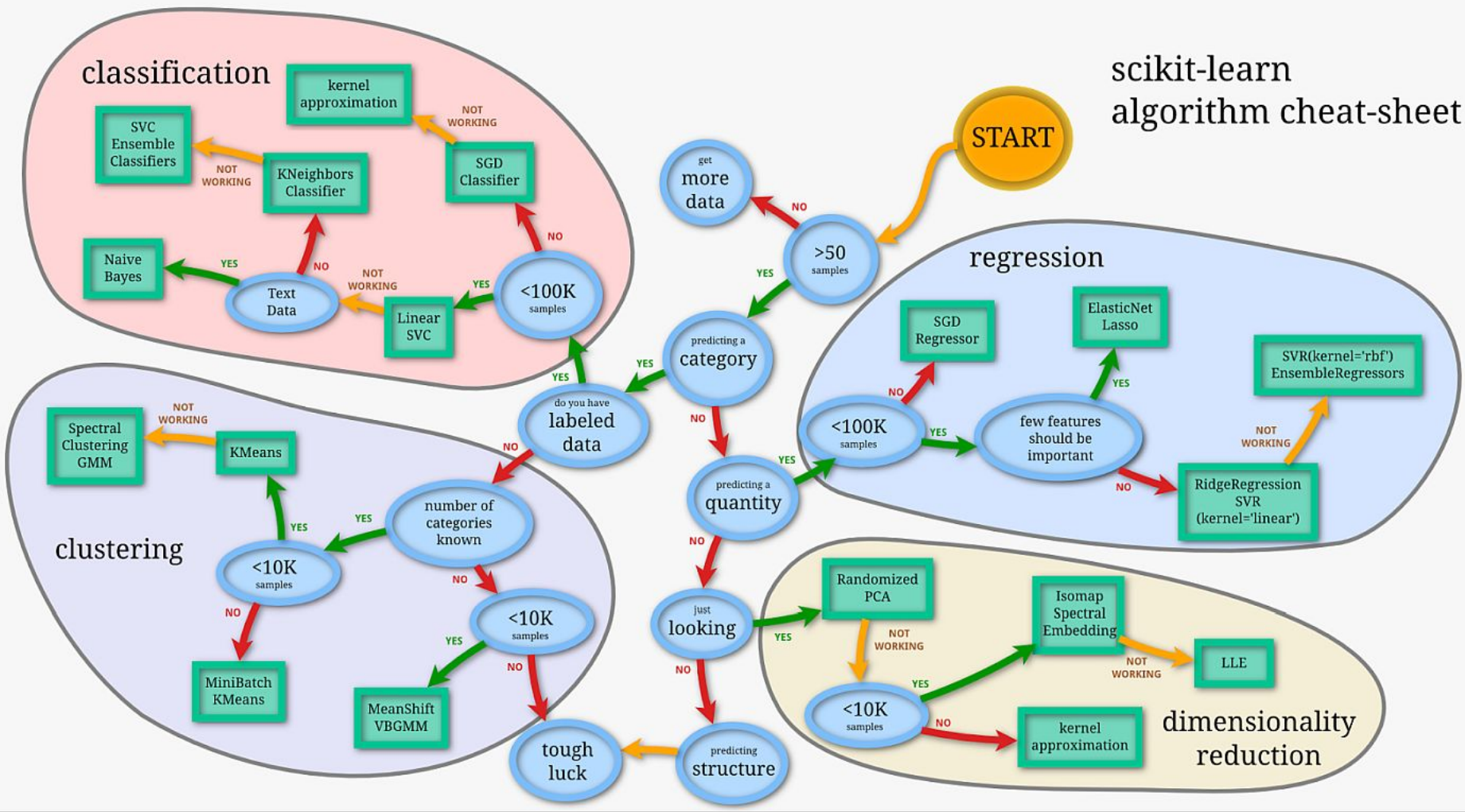
Pojawiają się:

- przestrzenie wersji (version spaces)
- algorytmy genetyczne
- sieci probabilistyczne (Baysowskie), Naiwny Bayes
- zbiory i logika przybliżona
- indukcyjne programowanie logiczne (ILP)
- uczenie ze wzmocnieniem (reinforcement)



# Choose your method

scikit-learn  
algorithm cheat-sheet



# 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.

System  $A$  adaptujący swoje  
parametry  $W$  do danych.

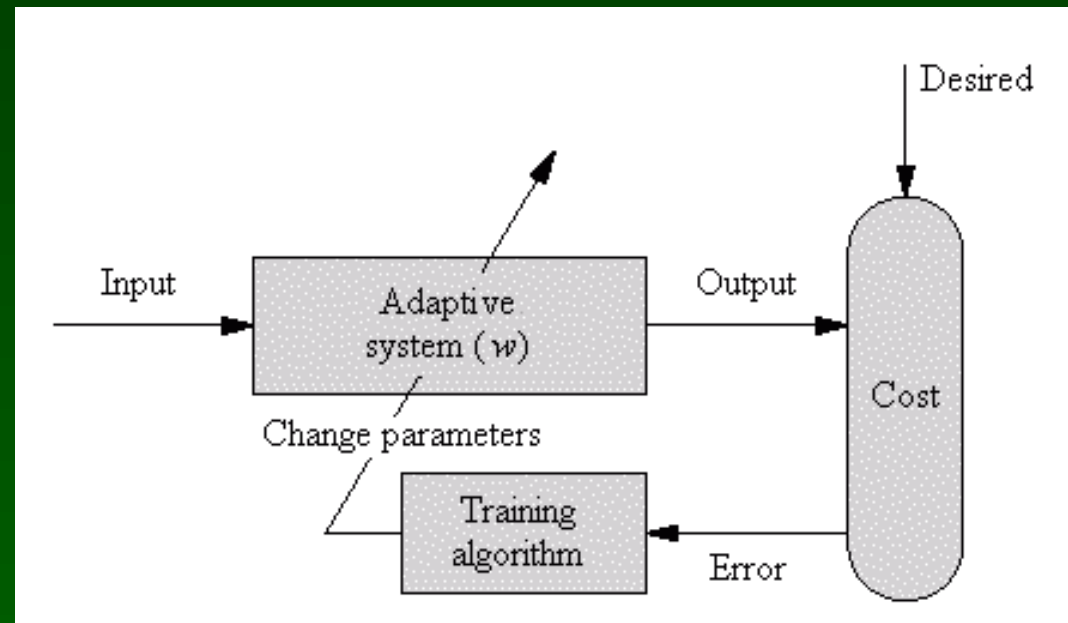
Pożądanane odpowiedzi  $Y(X)$

Wyniki działania:  $Y' = A(X; W)$

Funkcja błędu/kosztu  $E(Y, Y')$

Algorytm minimalizujący  
błędy na danych

treningowych  $A(X; W) \approx Y(X)$



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

# Supervised Learning

- Learn heteroassociations correcting errors: inputs-outputs,  $X \Leftrightarrow Y$ , for example image  $\Leftrightarrow$  label, category.
- Needed: database of labeled cases.
- Use an adaptive system  $A(X;W)$ , with parameters adjusted to data.
- Predict labels for new cases  $Y=A(X;W)$ , correct errors.

Knowledge extraction:

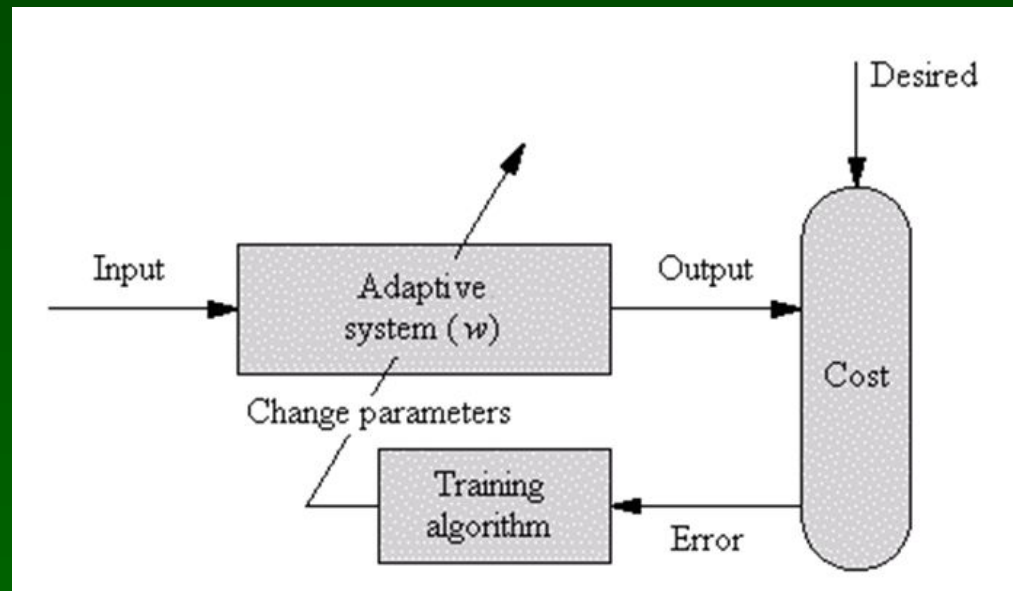
summarize databases by rules.

Information compression: rule is simpler than the data it explains.

Outlier detection: Exceptions not covered by the rule, e.g. fraud.

$A(X;W)$  may not be easy to understand – black box.

School-like learning.



# 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  $p_i \in P$

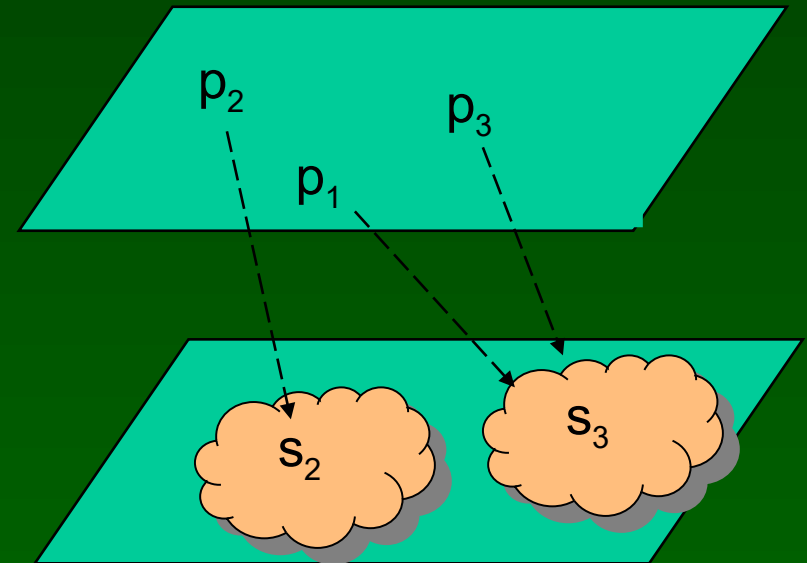
System adaptujący się

Miary podobieństwa  $S(p_i, p_j)$

Algorytm uczący, analizujący rozkłady miar podobieństwa.

Tworzenie odróżnialnych skupień  $S_i$

Funkcja jakości grupowania  $E(S_i, S_j)$



# Unsupervised Learning

Babies learn in spontaneous, unsupervised way, from sensory information:

Prioprioception => movement control,

Sounds => speech and object recognition,

Sights => visual recognition, face recognition etc.

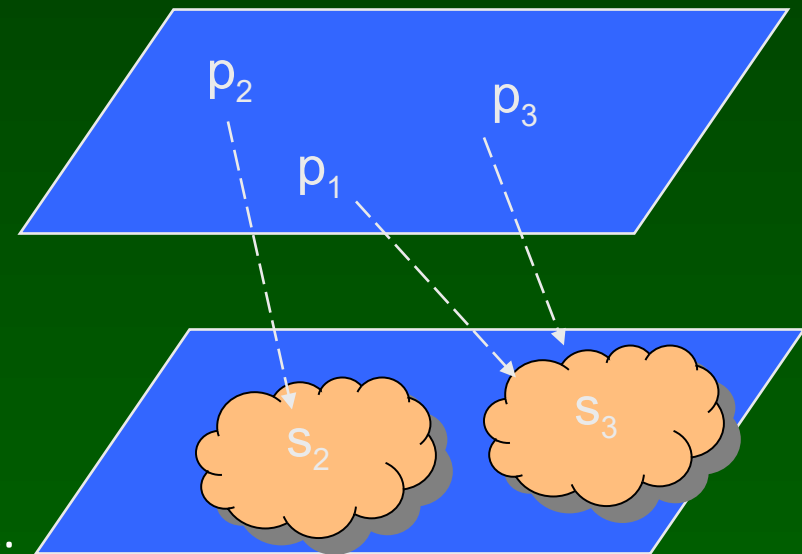
How do we do it? We have no clue, just hearing, seeing, moving ...

There are no labels.

- Clustering: grouping similar instances
- Discovering patterns in signals.

Example applications:

- Bioinformatics: learning motifs.
- Signal analysis: EKG, EEG, fMRI, music ...
- Customer segmentation, .



# Uczenie zachowań

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 zachowań. Np. gry z przeciwnikiem: przegrana lub wygrana dopiero na końcu partii.

Potrzebny jest:

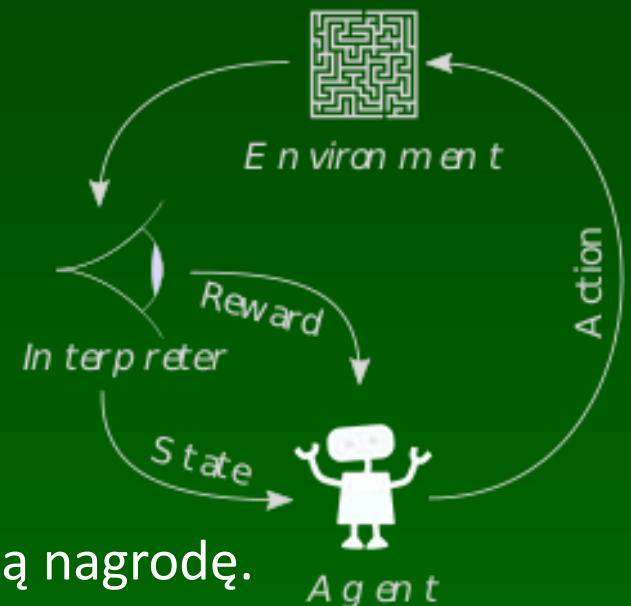
Opis stanów  $X$  w obserwowanym środowisku.

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.



# Reinforcement Learning



Learning from experiences, strategies, wisdom, without corrections after each action.

- Learning a policy: A sequence of outputs.
- No supervised output, only delayed reward.
- Credit assignment problem: which actions to take?
- Game playing, robots in a maze.
- Multiple agents, partial observability, ...





# Drzewa decyzji

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

$T' :=$  buduj\_drzewo( $E$ );

$T :=$  obetnij\_drzewo( $T'$ );

**return**  $T$ ;

**function** buduj\_drzewo( $E$ : zbiór przyk.) **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<sub>i</sub>} lub Atrybut < wartość<sub>i</sub>

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

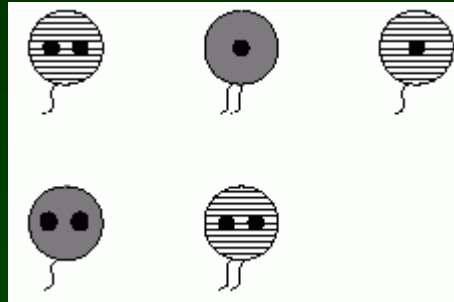
**Przycinanie:** usunąć gałęzie, które zawierają zbyt mało przypadków  
prostsze drzewo może lepiej generalizować  
ocenić optymalną złożoność na zbiorze walidacyjnym.

**Kryterium stopu:** osiągnięta dokładność podziałów,  
zbyt wiele gałęzi.

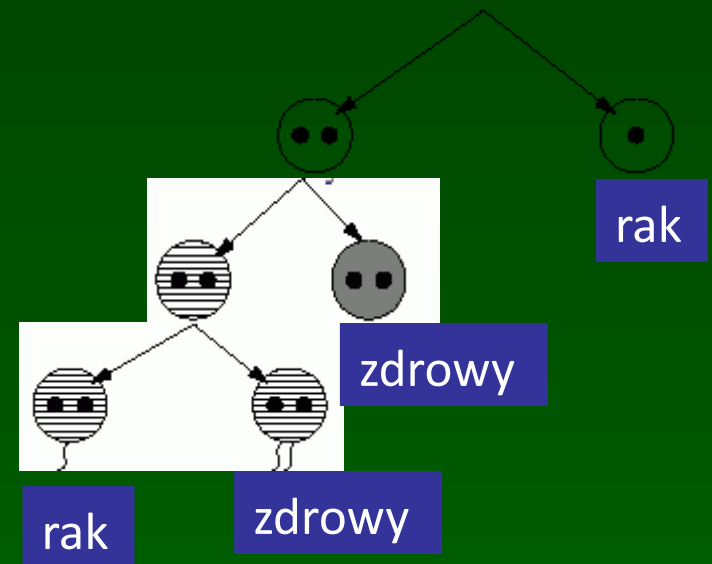
**Popularne systemy:** C4.5, CART – szybkie i często wystarczające.  
SSV, Separability Split Criterion (K. Grąbczewski)

# DT - przykład

Klasy: {rakowe,  
zdrowe}



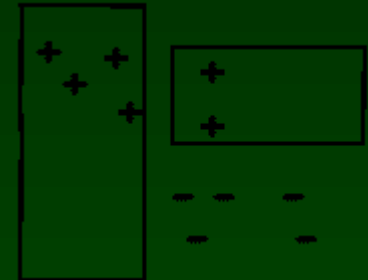
Cechy: ciało komórki: {cienie, paski}  
jądra: {1, 2}; ogonki: {1, 2}



# Indukcja reguł

DT: tworzą reguły hierarchiczne

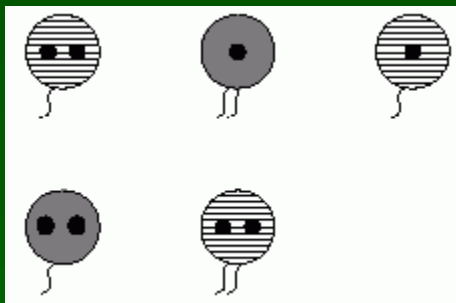
Utwórz reguły próbując pokryć obszar przestrzeni, w którym znajdują się wektory określonej klasy.



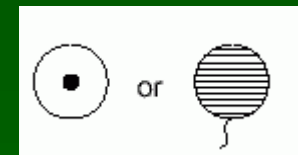
Zalety: łatwo zrozumiałe, dobra reprezentacja hipotez.

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
Kosmoslunch		Piątek	drogo	Nie
DS.1	lunch	Sobota	tanio	Tak
Bar mleczny	śniadanie	Niedziela	tanio	Nie
DS.1	śniadanie	Niedziela	drogo	Nie

W jakich warunkach mamy reakcje alergiczne?

$$\begin{array}{ccccccccc} \text{Miejsce} & \text{Posiłek} & \text{Dzień} & \text{Koszt} & & & & & \\ 3 & \times & 3 & \times & 7 & \times & 2 & = & 126 \end{array}$$

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

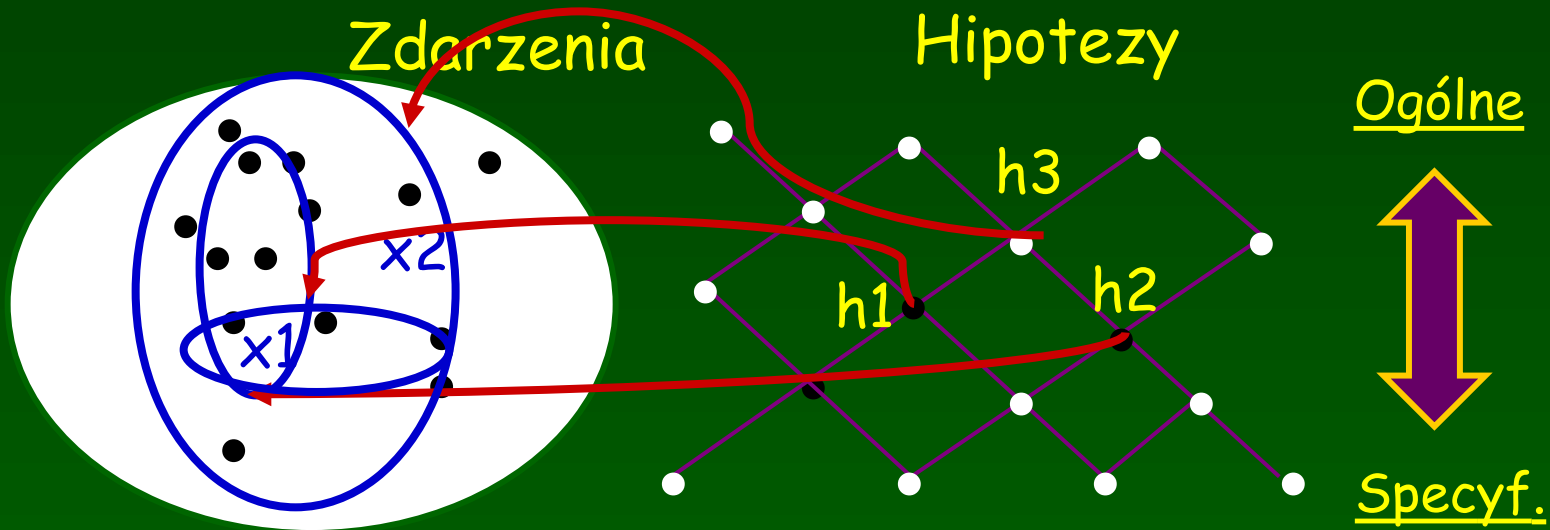
# VS - hipotezy

Mamy funkcję logiczną: Reakcja(Miejsce, Posiłek, Dzień, Koszt)

Znamy jej wartości dla kilku przykładów.

Zadanie: zrekonstruować najprostszą funkcję.

VS: koncepcje mapujemy na hipotezy o różnym stopniu ogólności.



# VS - hipotezy

$x_1 = \langle \text{DS1, lunch, Poniedziałek, tanio} \rangle$

$x_2 = \langle \text{Kosmos, lunch, Sobota, drogo} \rangle$

Hipoteza  $\langle ?, ?, ?, ? \rangle$

Hipoteza  $\perp$

$h_3 = \langle ?, \text{lunch}, ?, ? \rangle$       b. ogólna

$h_1 = \langle ?, \text{lunch}, ?, \text{tanio} \rangle$

$h_2 = \langle ?, \text{lunch}, \text{Sobota}, ? \rangle$

Koncepcje: od najbardziej szczegółowych do całkowicie ogólnych.

Kolor  $\rightarrow \{\text{Kolor czysty, kolor mieszany}\} \rightarrow \{ \{\text{cz, z, n}\}, \{\text{p, b, ...}\} \}$

Indukcyjne uczenie koncepcji: Jeśli koncepcja dobrze przybliży poszukiwaną funkcję na dużej liczbie przykładów to będzie dobrym przybliżeniem również dla nowych przykładów.

# VS - algorytm naiwny

Inicjalizacja:  $h := \perp$

For każdego **pozytywnego** przykładu  $x$  w  $D$  Do:

If  $h$  nie pokrywa  $x$ :

Zamień  $h$  na minimalne uogólnienie

$h$  które pokrywa  $x$

Return  $h$

Zalety: nie musi pamiętać poprzednich przykładów, prosty.

Wady: może być kilka minimalnych generalizacji  $\Rightarrow$  wiele kombinacji; nie wykrywa sprzeczności w danych, nie wykrywa ograniczeń języka hipotez  $H$  nie pozwalających nauczyć się danej koncepcji.

# Ocena podobieństwa

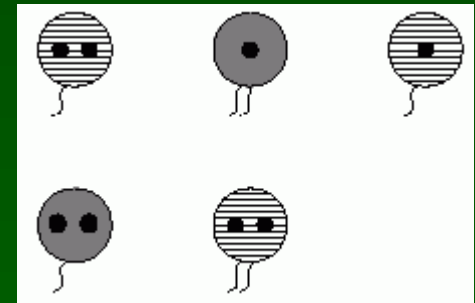
Pamiętaj przykłady i oceniaj podobieństwo; stosuj algorytmy „leniwe”, czyli oceniające sytuację dopiero wtedy, gdy zajdzie potrzeba.

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 ...

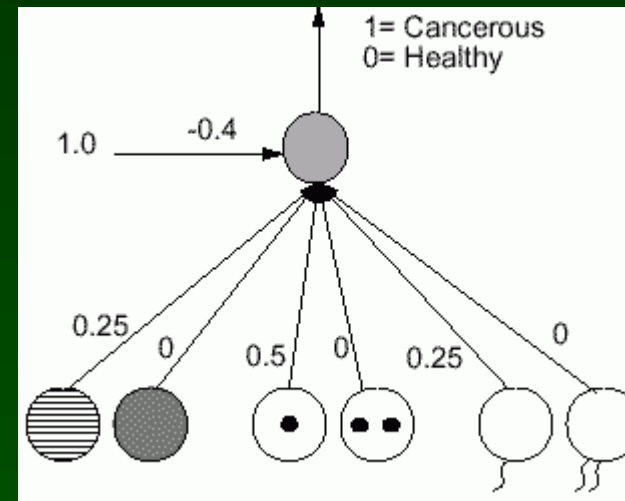
Zalety: uniwersalne, dużo symulatorów, odporne na szum w danych, wiele zastosowań i wariantów.

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.

# Sieci neuronowe - przykład

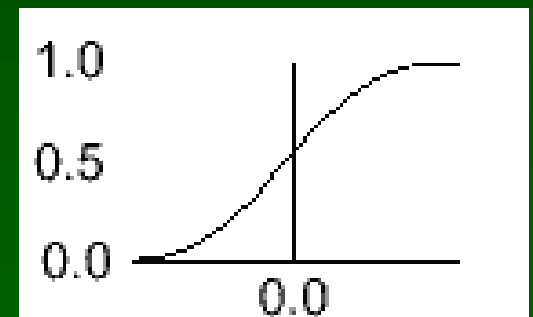
Wejścia: dane ciągłe lub dyskretne.  
Neuron sumuje wartości sygnałów wejściowych określając pobudzenie.

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



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

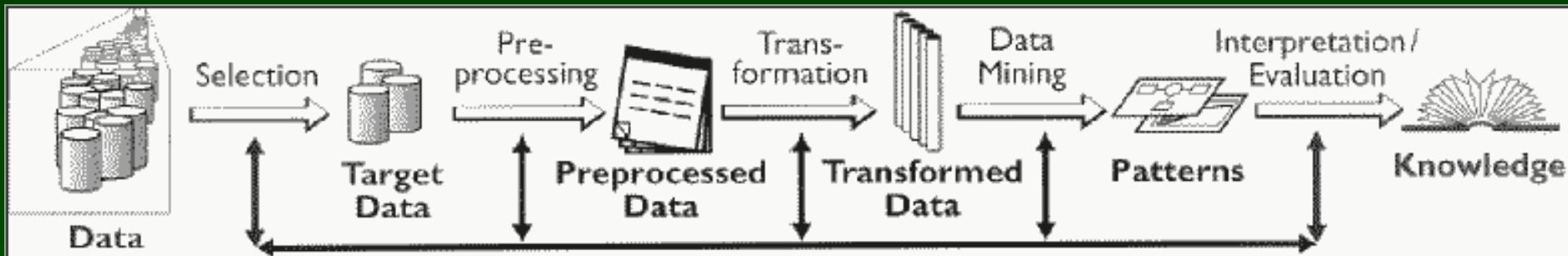
W sieciach neuronowych wiele takich neuronów połączonych jest ze sobą realizując dowolnie skomplikowane funkcje.



# 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

# 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%

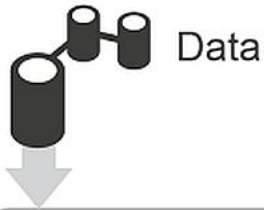
$R_4) \text{IF habitat} = \text{leaves} \wedge \text{cap\_color} = \text{white}$       nie ma błędów!

Teraz widać, dlaczego węch jest tak ważny dla zwierząt.

# DM & statistics

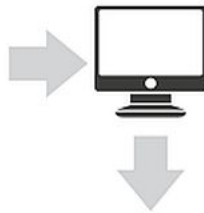
## 1 Traditional programming

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.

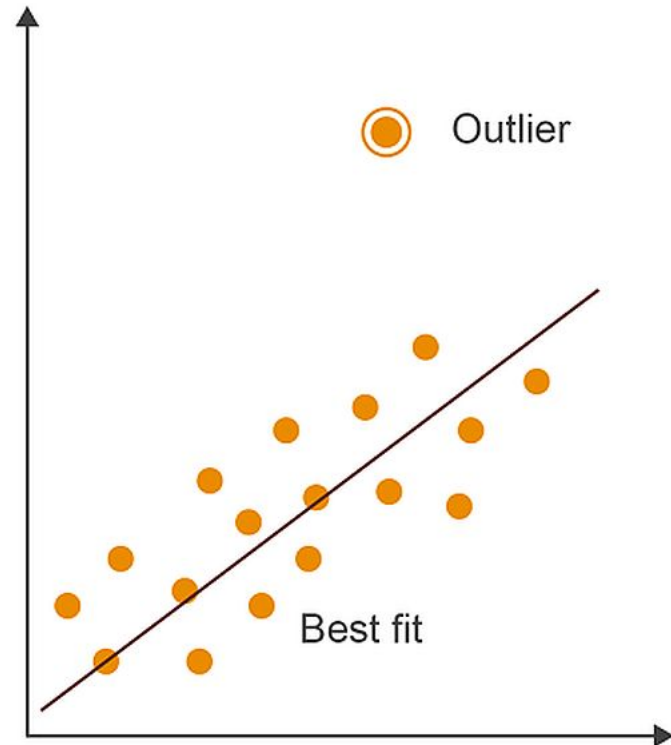
```
class FREE!!! implements SpamDetector
{
    public function detect($string)
    {
        if (str_word_count($string) = FREE!!!) {
            return true;
        }
        return false;
    }
}
```



Computer follows the procedure and generates a result.

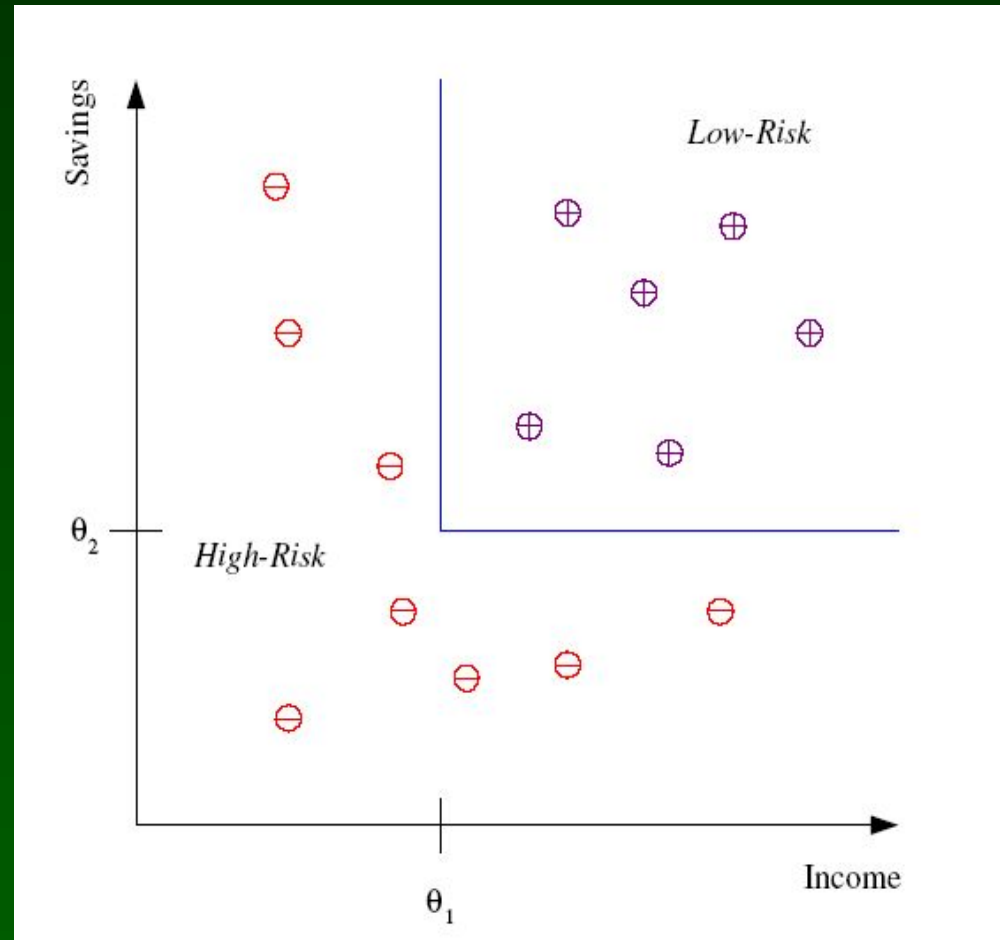
## 2 Statistics

An analyst compares the relationships of variables.



# Classification

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



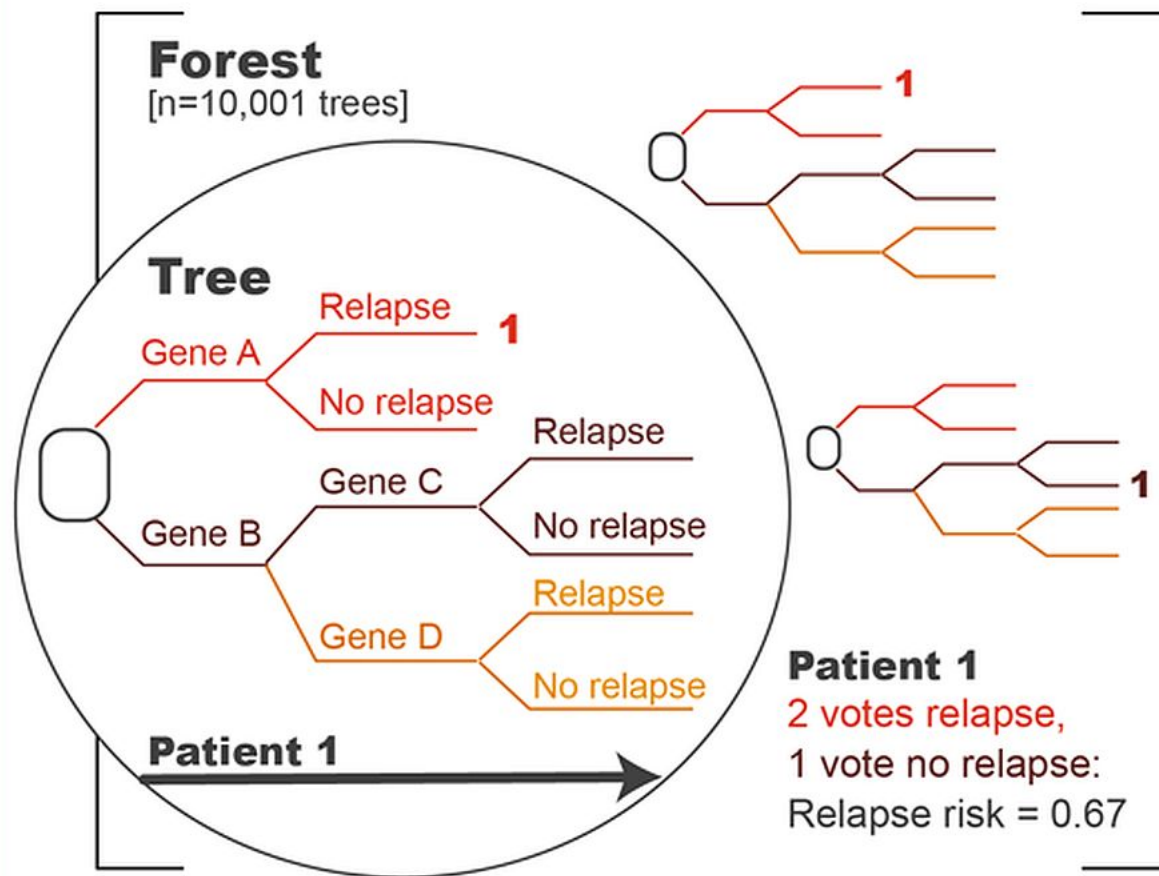
Discriminant: IF *income*  $>$   $\theta_1$  AND *savings*  $>$   $\theta_2$   
THEN **low-risk** ELSE **high-risk**



# 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

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 & Machine Learning

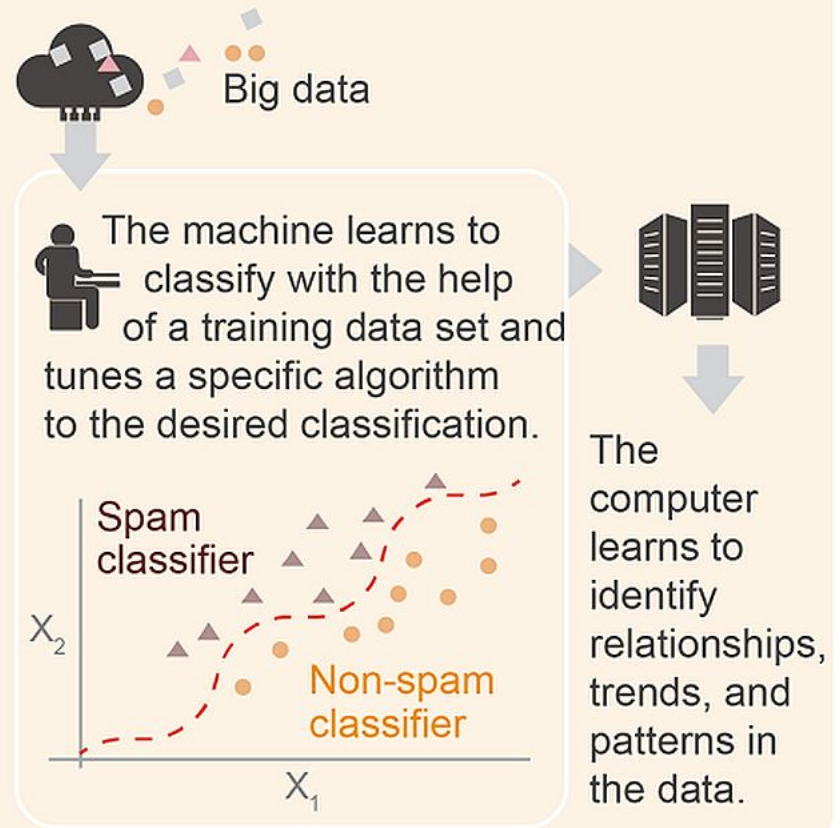
## 4 Intelligent apps

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.



# 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.



© 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](http://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.

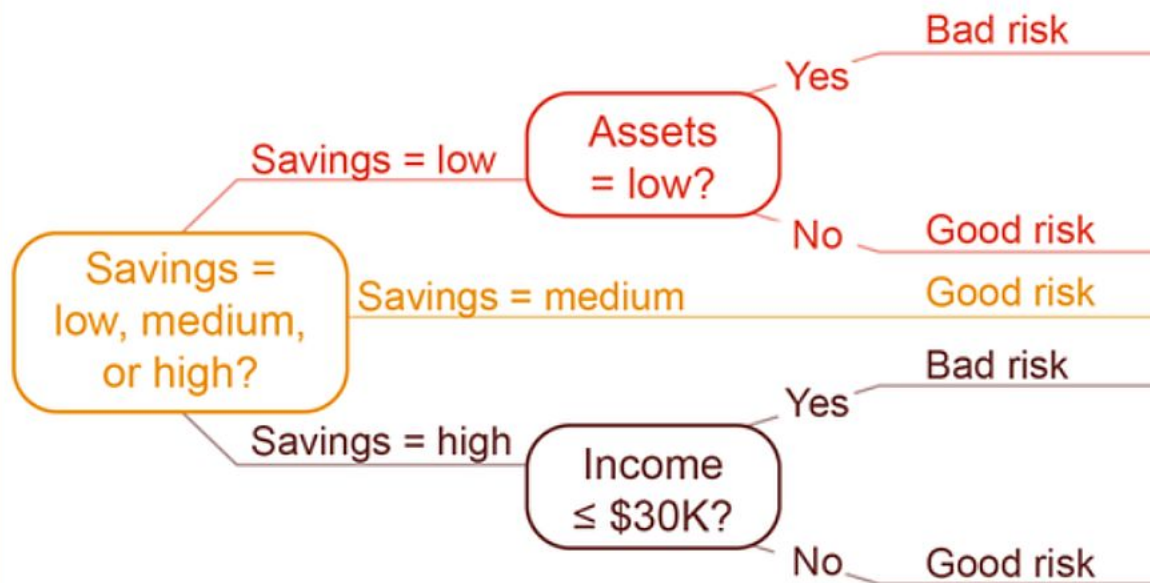
***[pwc.com/NextinTech](http://pwc.com/NextinTech)***



# 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

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

### Use cases

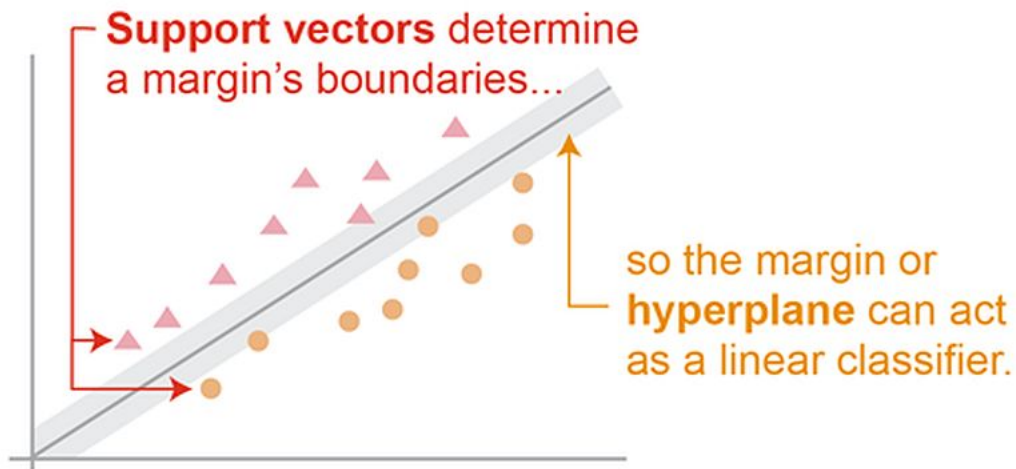
Rule-based credit risk assessment, horse race performance prediction

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

# Methods: 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.

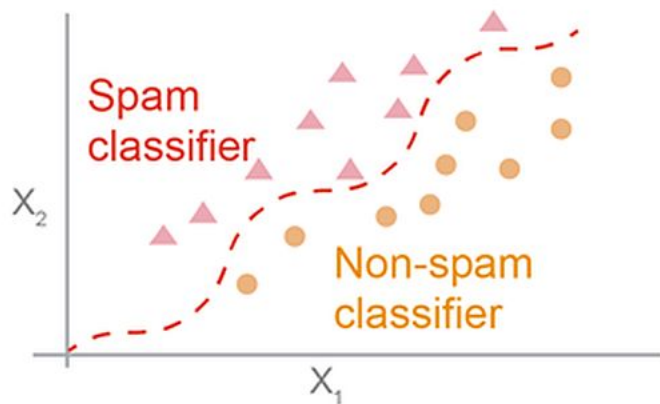
### Use cases

News categorization, handwriting recognition

# Methods: Regression

## 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

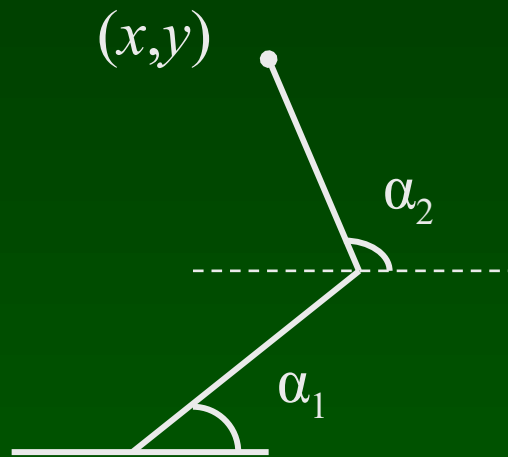
Regression is useful for identifying continuous (not necessarily distinct) relationships between variables.

### Use cases

Traffic flow analysis, email filtering

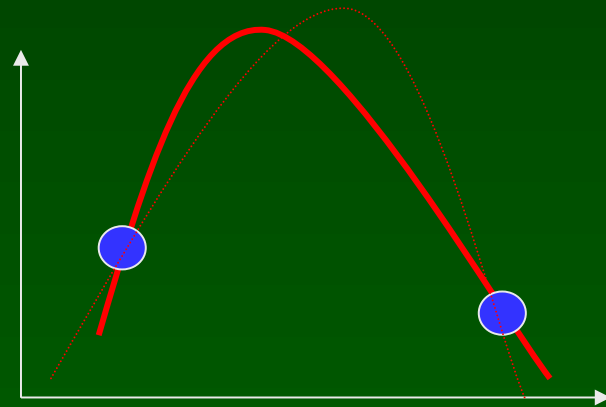
# Regression Example

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



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

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

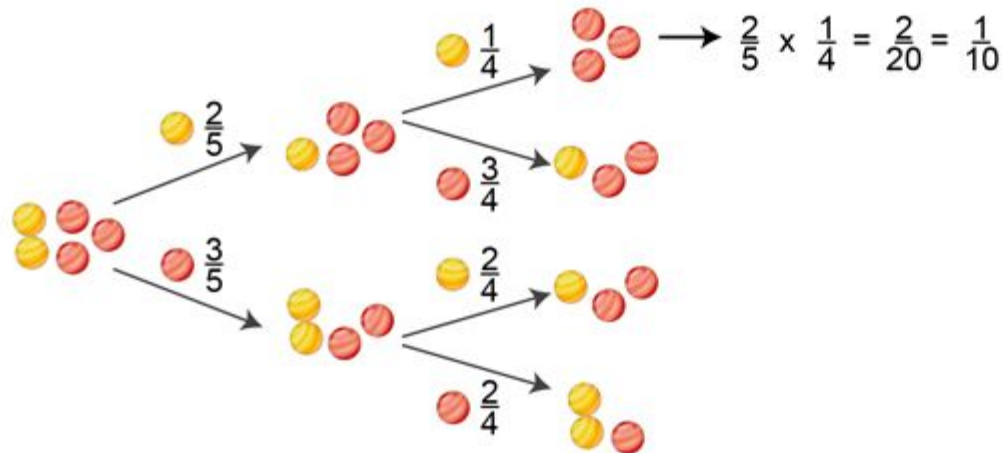


Control of cars or robots require calculation of continuous parameters. Linear regression is commonly used in social sciences, but the challenge is to predict complex nonlinear trajectories with many parameters.

# 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

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

### Use cases

Sentiment analysis, consumer segmentation

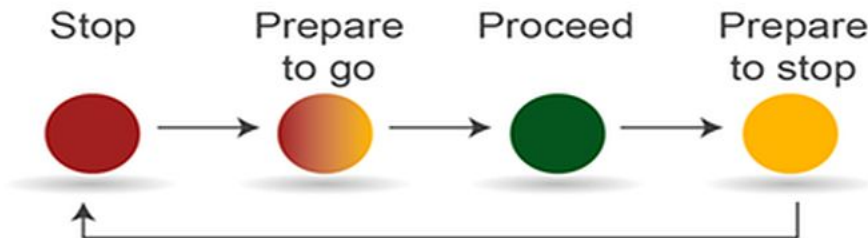
Source: Rod Pierce, et al., *MathIsFun*, 2014



# Methods: Hidden Markov

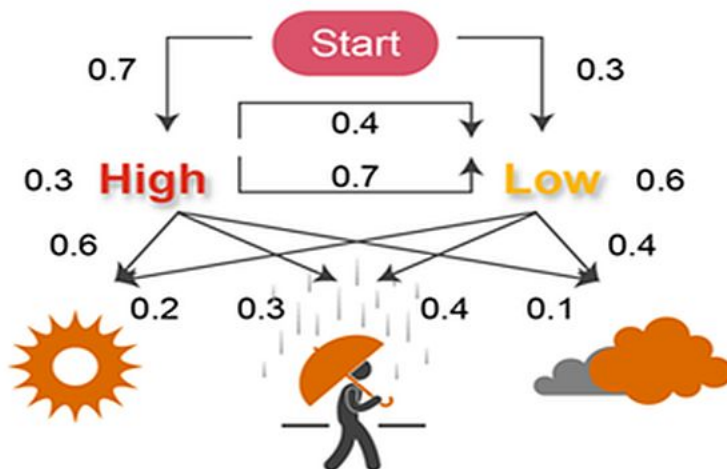
## Hidden Markov models

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

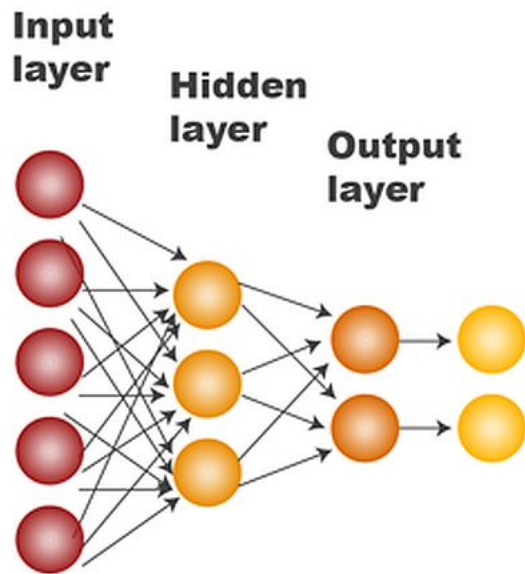
Facial expression analysis, weather prediction

# Methods: Neural Networks

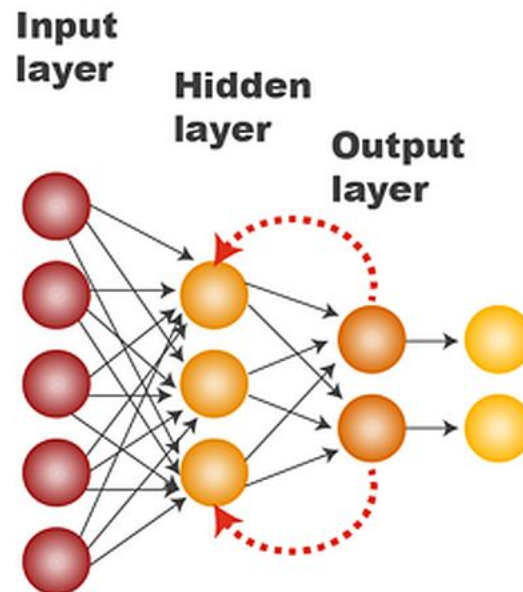
## Recurrent neural networks

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



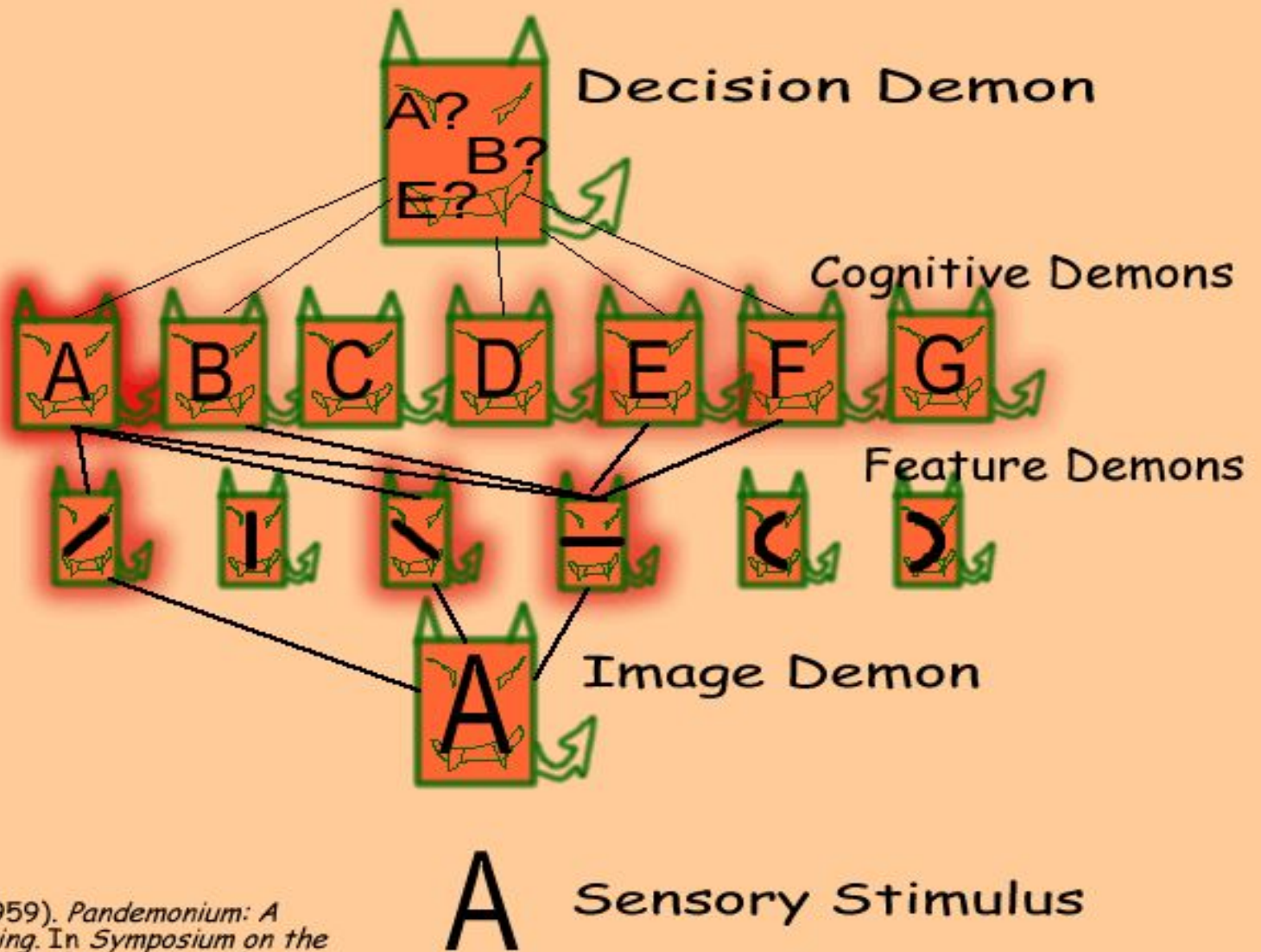
Advantages

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

Use cases

Image classification and captioning, political sentiment analysis

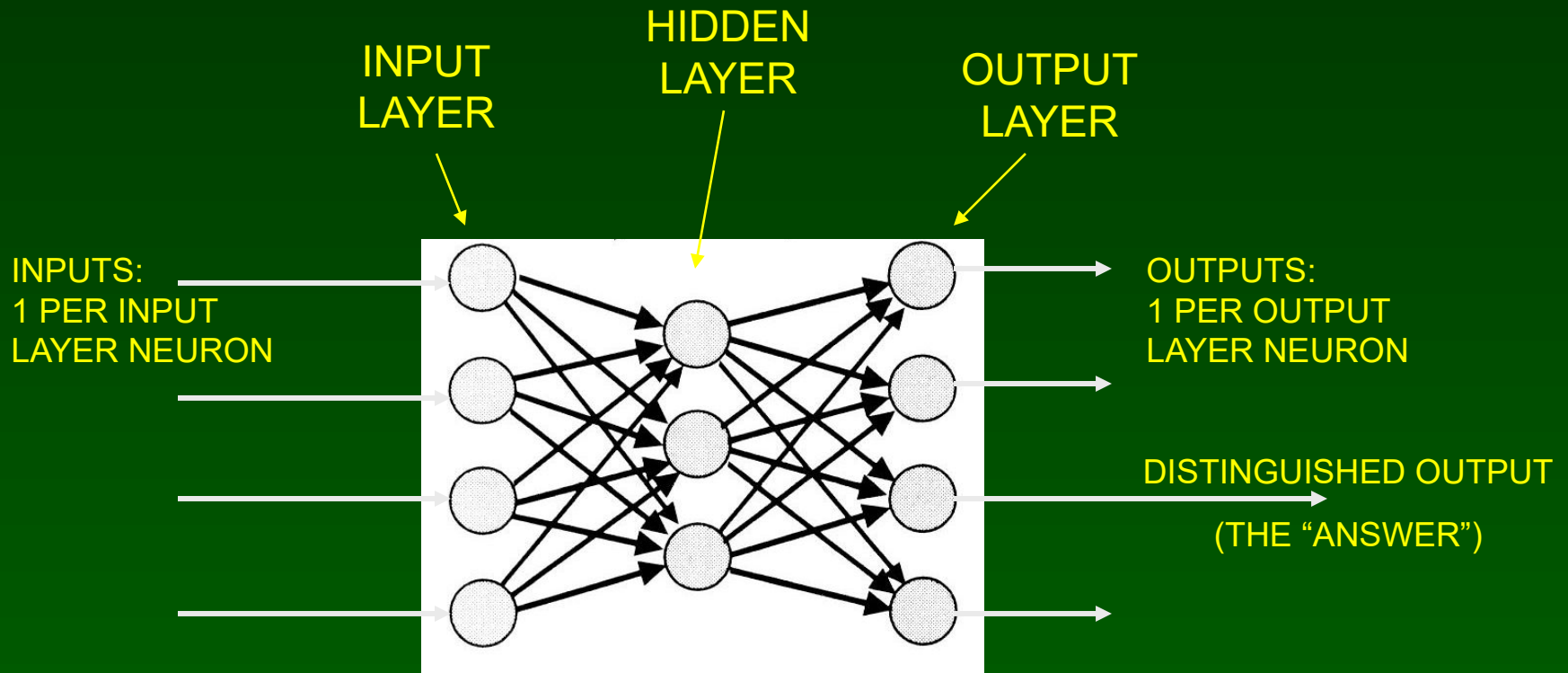
Source: Joseph Wilks, 2012



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.

# Neural Networks

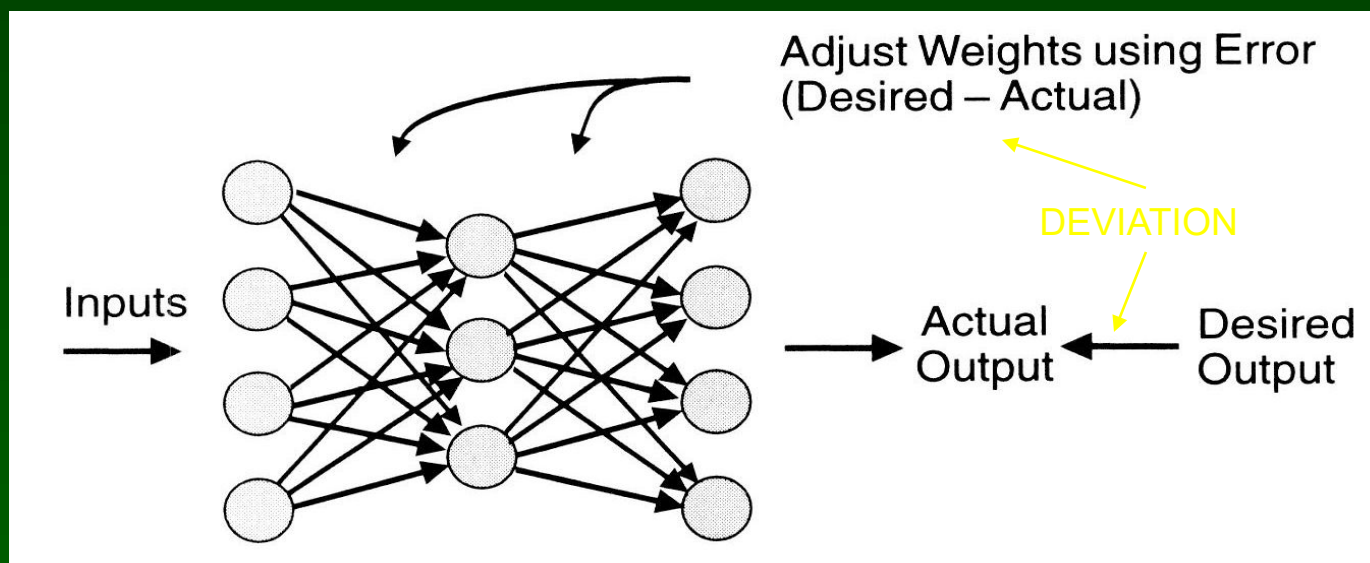




# Neural Networks

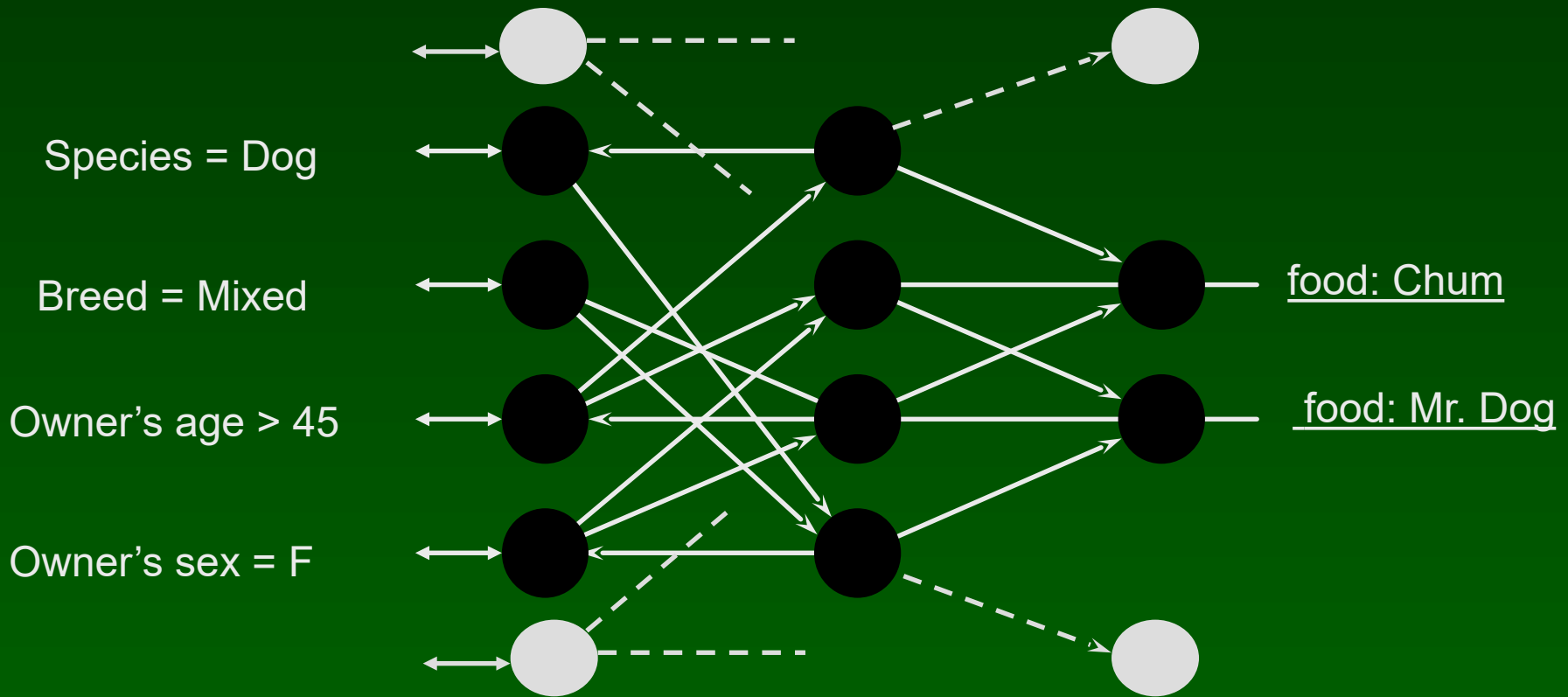
## Learning through back-propagation

1. Network is trained by giving it many inputs whose output is known
2. Deviation is “fed back” to the neurons to adjust their weights
3. Network is then ready for live data



# Neural Network Classification

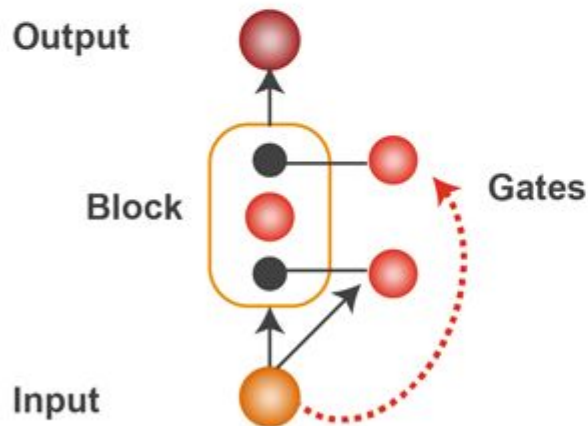
*“Which factors determine a pet’s favorite food?”*



# Methods: LSTM

## Long short-term memory & gated recurrent unit neural networks

Older forms of RNNs can be lossy. While these older recurrent neural networks only allow small amounts of older information to persist, newer long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks have both long- and short-term memory. In other words, these newer RNNs have greater memory control, allowing previous values to persist or to be reset as necessary for many sequences of steps, avoiding “gradient decay” or eventual degradation of the values passed from step to step. LSTM and GRU networks make this memory control possible with memory blocks and structures called gates that pass or reset values as appropriate.



Source: Genevieve Orr, et al., Willamette University, 1999

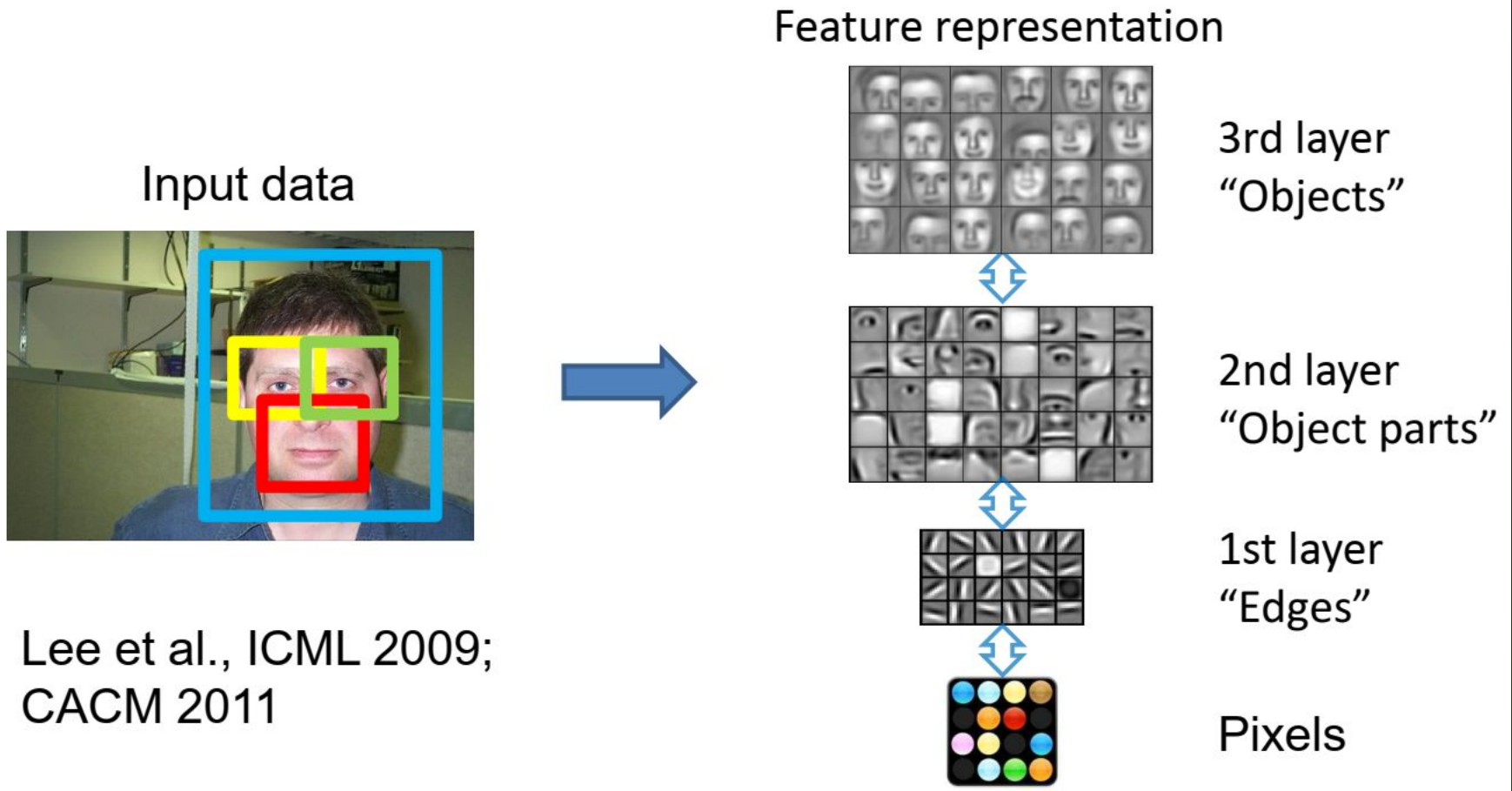
### Advantages

Long short-term memory and gated recurrent unit neural networks have the same advantages as other recurrent neural networks and are more frequently used than other recurrent neural networks because of their greater memory capabilities.

### Use cases

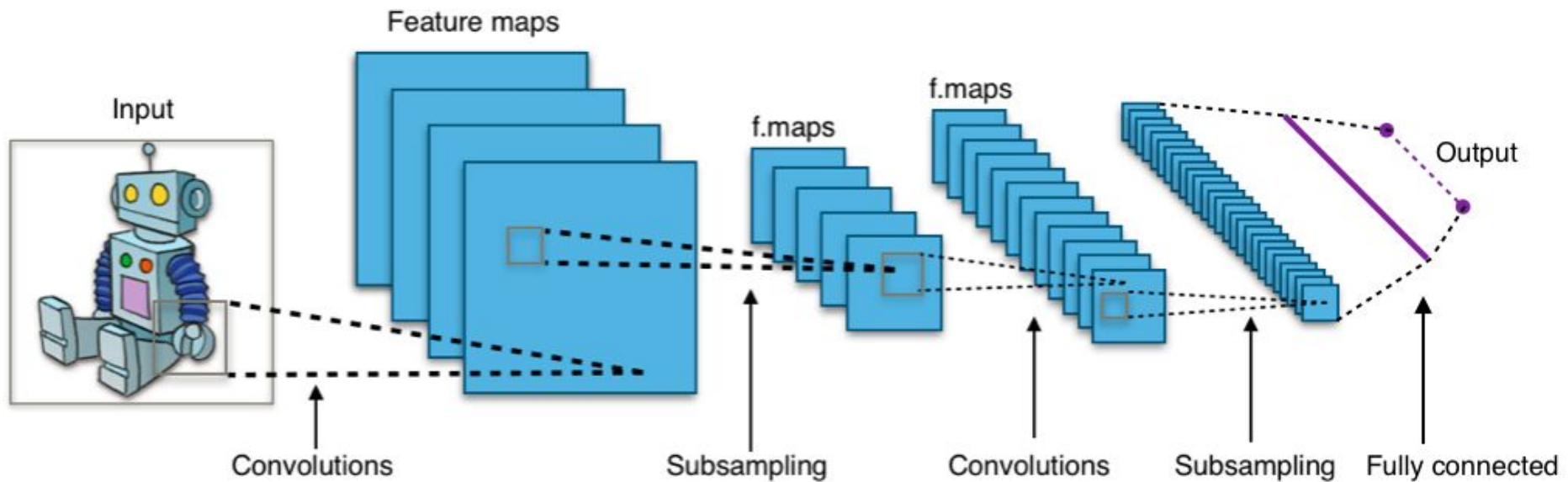
Natural language processing, translation

# Methods: NN for images





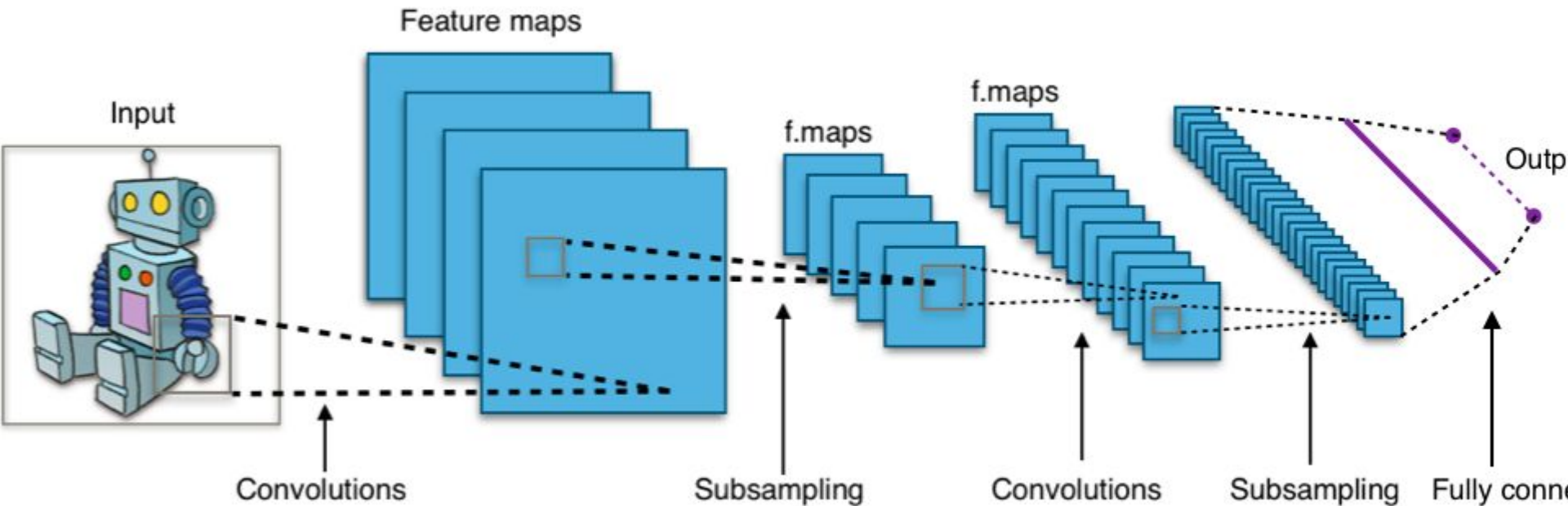
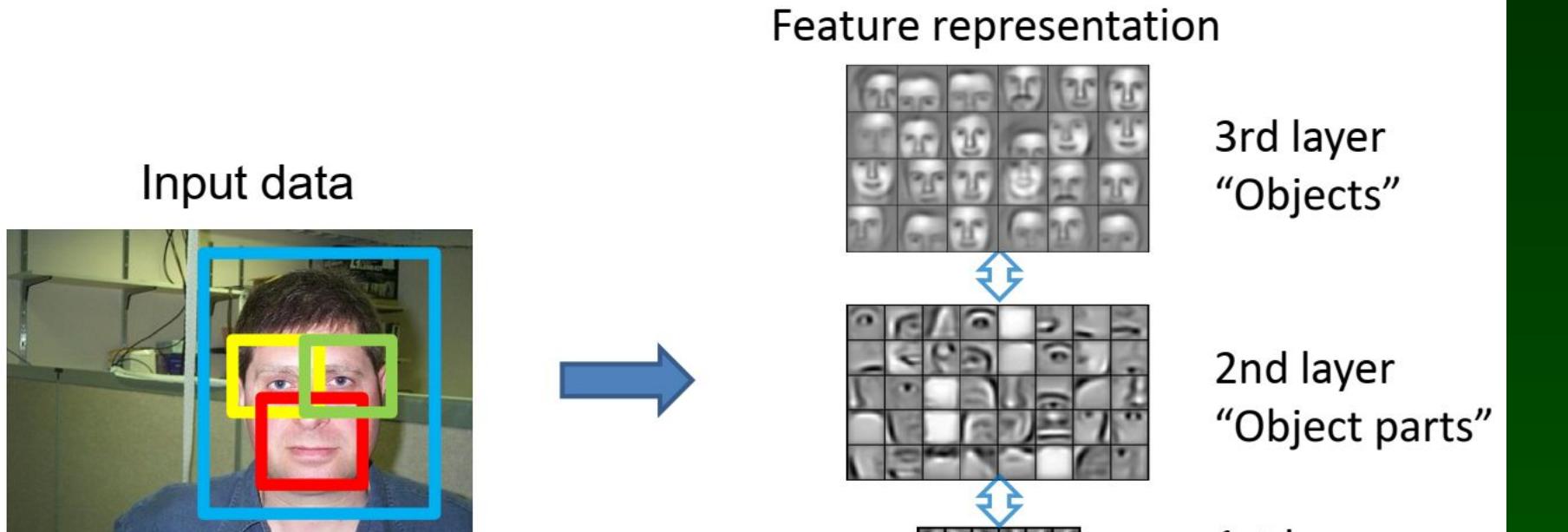
# Methods: Deep NN for images



Small patches of image (ex. 5x5 pixels) are filtered through receptive fields in convolution operation, creating elementary features, combined together using sampling, and again filtered to generate more complex features.

Finally generated features are analyzed to determine image category.

# Methods: NN for images

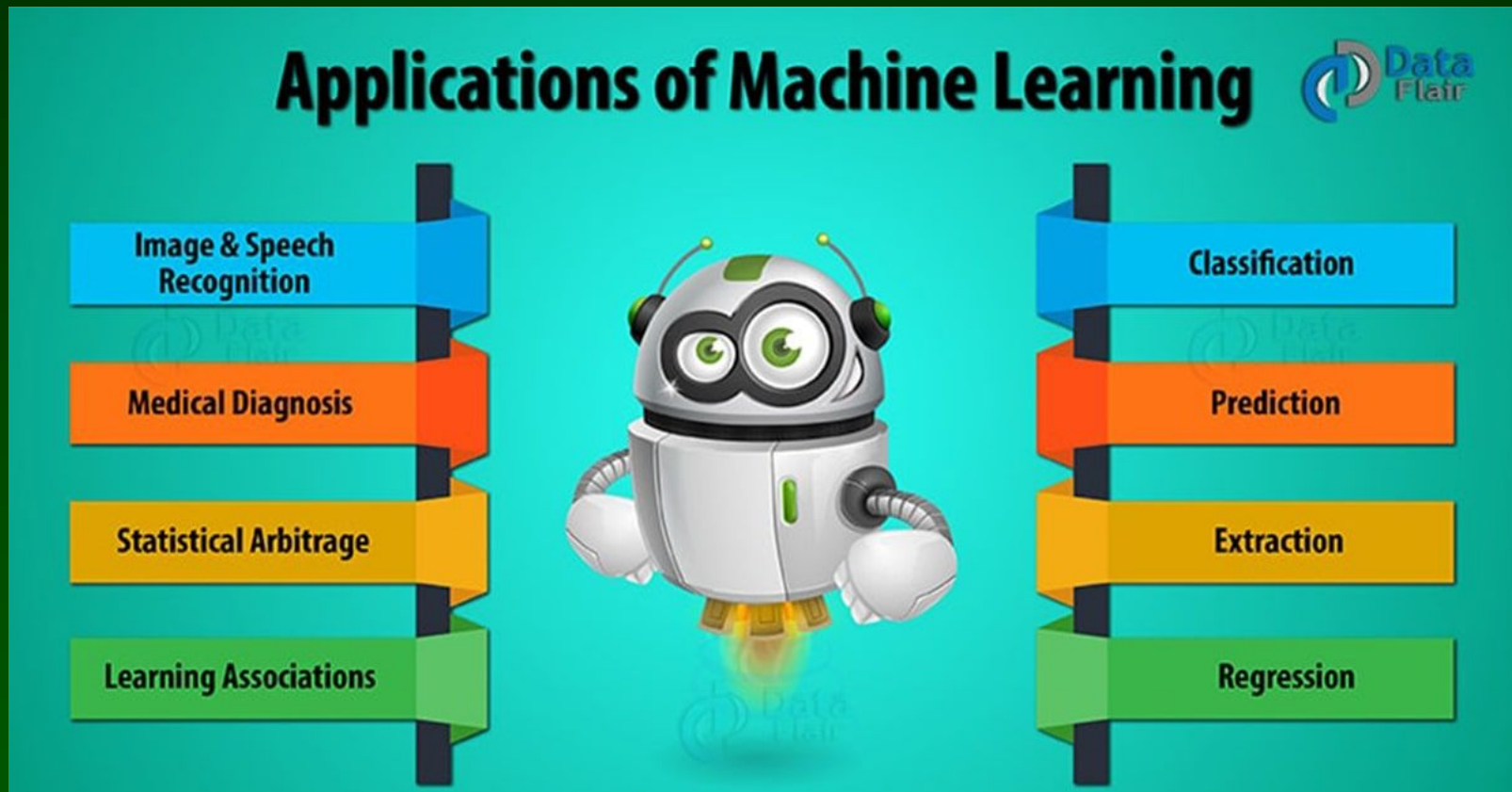


# Superresolution





# 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.

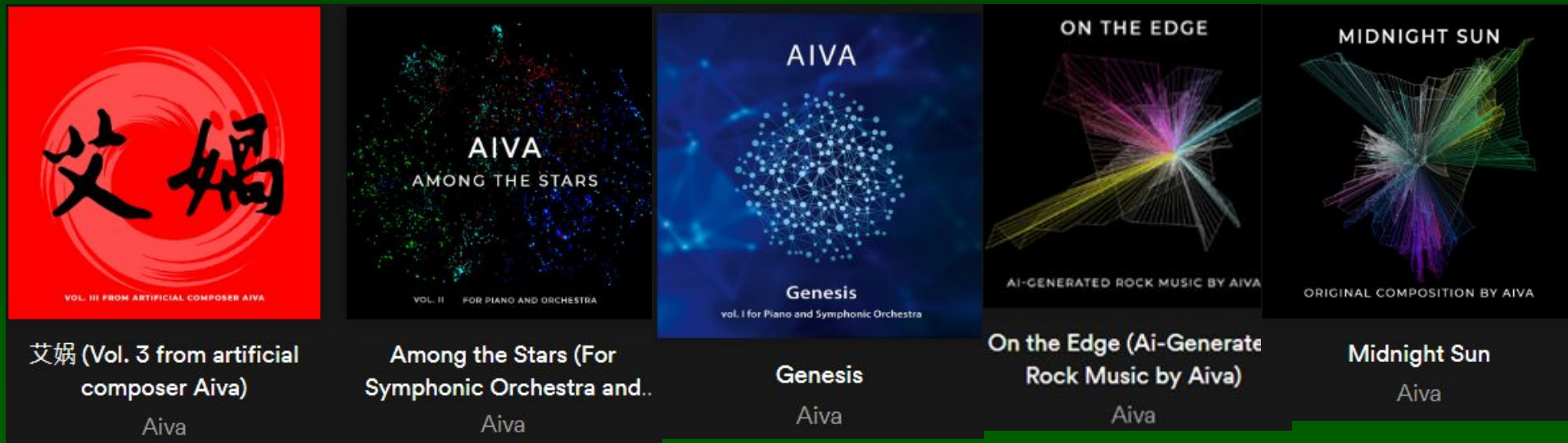
# AI Virtual Artist

[AIVA](#) – AI Virtual Artist, member of author's rights society ([SACEM](#)), 206 works.

[AIVA YouTube](#) channel, Youtube „[Letz make it happen](#)“, Op. 23

[SoundCloud](#) channel

[Spotify](#) and [Apple](#) channel



# Deep NN for protein folding



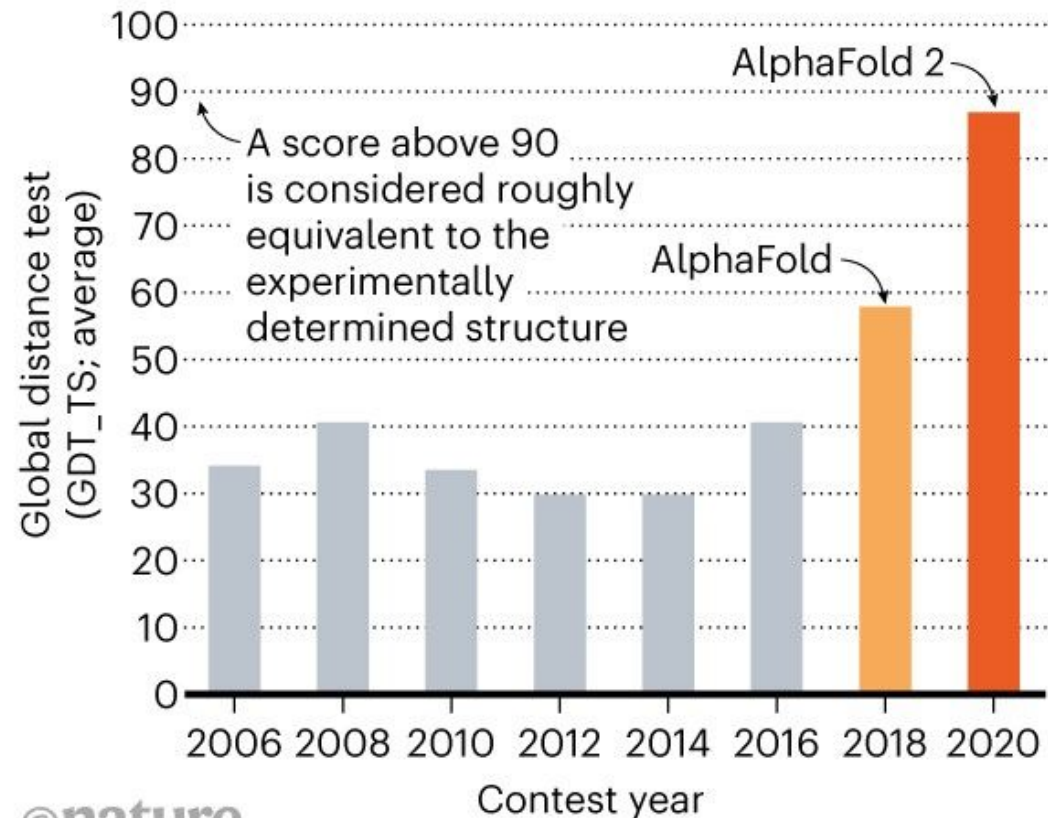
The ability to accurately predict protein structures from their amino-acid sequence will vastly accelerate efforts to understand the building blocks of cells, and enable quicker and more advanced drug discovery.

AlphaFold using deep neural networks has made great improvement, 2/3 of predicted structures is equivalent to experimental!

[Nature, 30.11.2020](#)

## STRUCTURE SOLVER

DeepMind's AlphaFold 2 algorithm significantly outperformed other teams at the CASP14 protein-folding contest — and its previous version's performance at the last CASP.

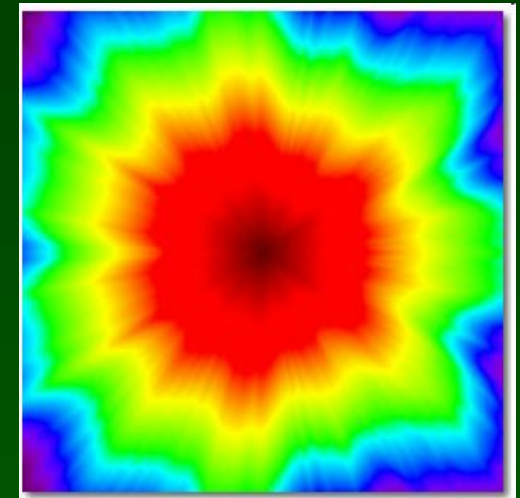
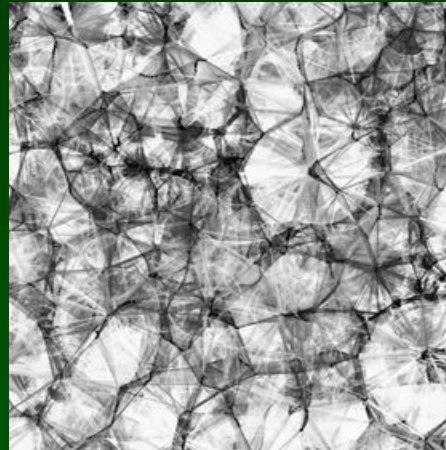


# Generative art

„Generative art” in Wikipedia

[http://en.wikipedia.org/wiki/Generative\\_art](http://en.wikipedia.org/wiki/Generative_art)

<http://www.random-art.org>



<http://www.solaas.com.ar/dreamlines/p5/>

<http://www.jhlab.com/java/art.html>

Yuki singing animated avatar, first digital idol from Japan!





# Aaron Painter

Harold Cohen has asked about minimal requirements that should be fulfilled to call something art painting, and created in 1973 [Aaron](#) robotic artist.





# Aaron examples



Meeting On Gauguin's Beach,  
Oil 1992



Aaron, with Decorative Panel, 1988

Basic principles of composition + randomization. [See more here.](#)

# Aaron in museums



Liberty & Friends, 1985  
Ink



Mural, Capitol  
Children Museum 1980



*i*

N

•

•

•

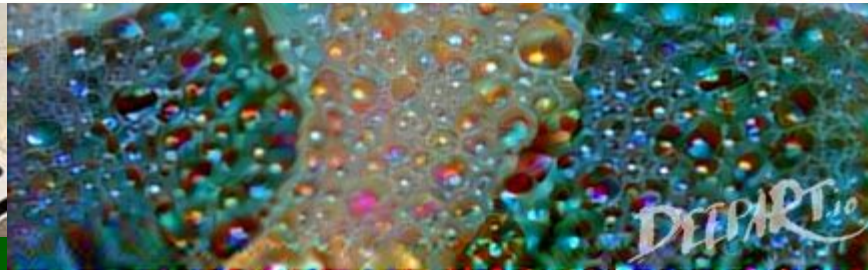
•



ed

f

# Deep Art



<https://deepart.io/latest/> Deep Neural Networks, Deep Dream, LA Gatys, AS Ecker, M Bethge, A Neural Algorithm of Artistic Style (2015)



# Examples of Deep Art



<https://deepart.io/latest/> Deep Art,  
LA Gatys, AS Ecker, M Bethge, A Neural Algorithm of Artistic Style (2015)



[Google Deep Dream/Deep Style i Generator,](#)  
LA Gatys, AS Ecker, M Bethge, A Neural Algorithm of Artistic Style (2015)

# Generative music

„Popcorn”, Hot Butter hit from 1969,  
Created through computer analysis of various melodies.  
Computer helping human musicians.



Musicians learn by hearing and analyzing.

Why computers should not do the same? With access to all music in the world?

Algorithmic music: John Cage, Iannis Xenakis ...

[http://en.wikipedia.org/wiki/Algorithmic\\_music](http://en.wikipedia.org/wiki/Algorithmic_music)

Lucasfilm Games' gra Ballblazer (1982) używała komputerowej improwizacji jazzowej, Xbox improwizuje na zadane tematy.

Generative music: Brian Eno started to experiment in 1975!

[http://en.wikipedia.org/wiki/Generative\\_music](http://en.wikipedia.org/wiki/Generative_music)

Examples of generative music:

<http://jmusic.ci.qut.edu.au/jmMusic.html>





# From images to music

Can music be composed converting images?

Example: finish “metacomposer” Lauri Gröhn:

[Synesthesia Software Music](#)

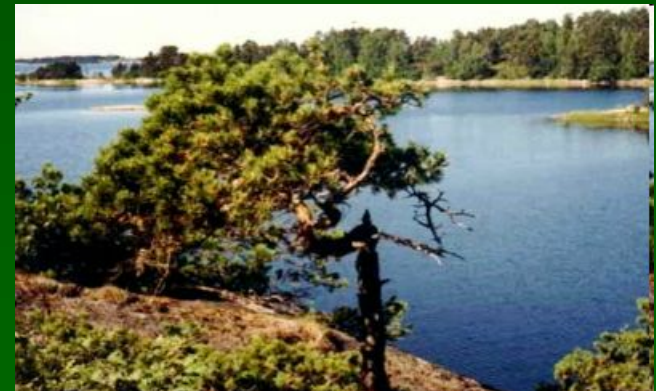
„I want to show that in future computers may compose as well as people”.



[Turku Archipelago](#), Fog, Sakura – interesting compositions!

Can computer really „feel” music?

[Greta ist trying...](#)





# Flow Machines

by Sony CSL

<http://www.flow-machines.com>

Software that learns.  
What is programmed  
is only the motivation  
to learn!

Learning to Create  
(Lrn2Cre8) EU FP7  
Project

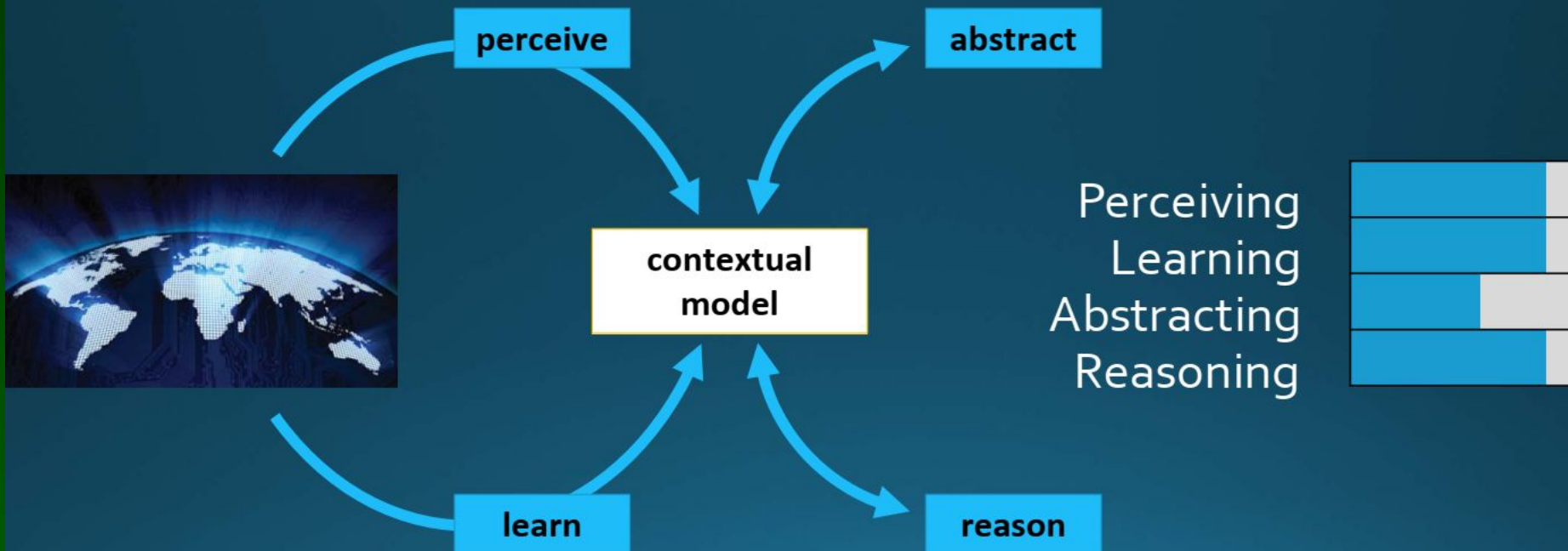
Daddy's Car: a song composed by Artificial Intelligence - in the style of the Beatles



flowmachines  
flowmachines

# Third wave of AI

## The third wave of AI



Final frontier: building models of objects and situations is the next step. GAN, Generative Adversarial Networks, one network creates false examples distorting learning data, another network learns to distinguish them from natural ones.