Computer Vision Summer Camp

Introduction to Machine Learning

What is machine learning?

- In 1950 Alan Turing in his work “Computing Machinery and Intelligence”, considering the ability to build an artificial intelligence, concluded that it’s much easier to program a child's mind and then subject it to a period of education, that to try to build adults mind from scratch.

- Human learn from experience all the time.

What is machine learning?

"Field of study that gives computers the ability to learn without being explicitly programmed".

Arthur Samuel, 1959.

- Computer program is a set of precise instructions, prescribing machine what and when to do.
- There are tasks where it's easy to give such instructions, e.g.: array sorting, but there are also those that can barely be solved in such a way, e.g.: speech/image recognition, language translation, etc.

What is machine learning?

"A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *Q* if its performance at tasks in *T*, as measured by *Q*, improves with experience *E* ".

Tom M. Mitchell, 1997.

- In a digital world experience is represented as data.
- Machine learning is about obtaining knowledge/information from data.

Machine learning vs Data mining

- ML and DM are close and sometimes used interchangeably.

- Though ML focuses more on obtaining information that is useful for machine itself, i.e., for improving its performance on the task of interest.

- DM focuses on making data useful for the user,
i.e., helps to get insights into the processes described by data.

Types of ML problems

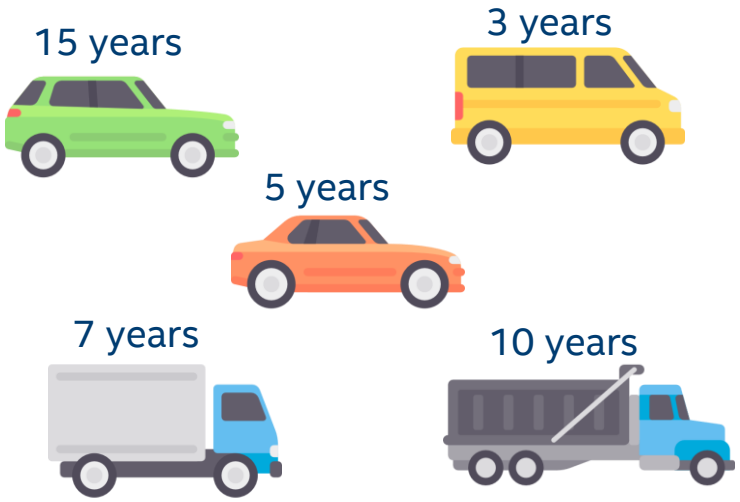
- Depending on the type of available data we consider different machine learning problems:
  - supervised learning,
  - reinforcement learning,
  - unsupervised learning,
  - etc.

Supervised learning

- The most “straightforward” way of learning.
- Assumes the presence of a ***teacher***,
  who/that knows answers to questions we ask.
- Let some process or thing be the subject of our interest.
- Some information about the subject is easy to get, some is difficult. We want to infer/predict information that is difficult to obtain,
  from the one that is easy to get.
- For that purposes we have a limited set of examples, where everything is known (given by a teacher), our target is to find regularities in data and generalize them to previously unseen examples.

Supervised learning. Example

- Let's imagine that we want to estimate car life time expectancy.
- We have a bunch of cars for that we know their actual life time.



For each car we collect information about:

- Engine type
- Transmission type
- Fuel consumption
- Travelled distance
- Climate control
- Color
- ...

Learn to predict the life time expectancy for an unseen car, given the same information about it.

Image credits: Icons made by Freepik from www.flaticon.com. Icons made by Those Icons from www.flaticon.com

Features

- We describe all subjects using the same set of *features*.
- Features may have different types:
  - Continuous, a real number (e.g.: fuel consumption, liters/km).
  - Categorical, an element of finite unordered set (e.g.: color, {red, green, blue}).
  - Binary, either zero or one (e.g.: air conditioner, {yes, no}).
  - Ordered, an element of finite ordered set (e.g.: number of gears, {4, 5, 6}).
- One feature is special – target feature that we want to predict.
  Depending on its type we distinguish:
  - *Regression* task, target feature is continuous.
  - *Classification* task, target feature is categorical.

Features

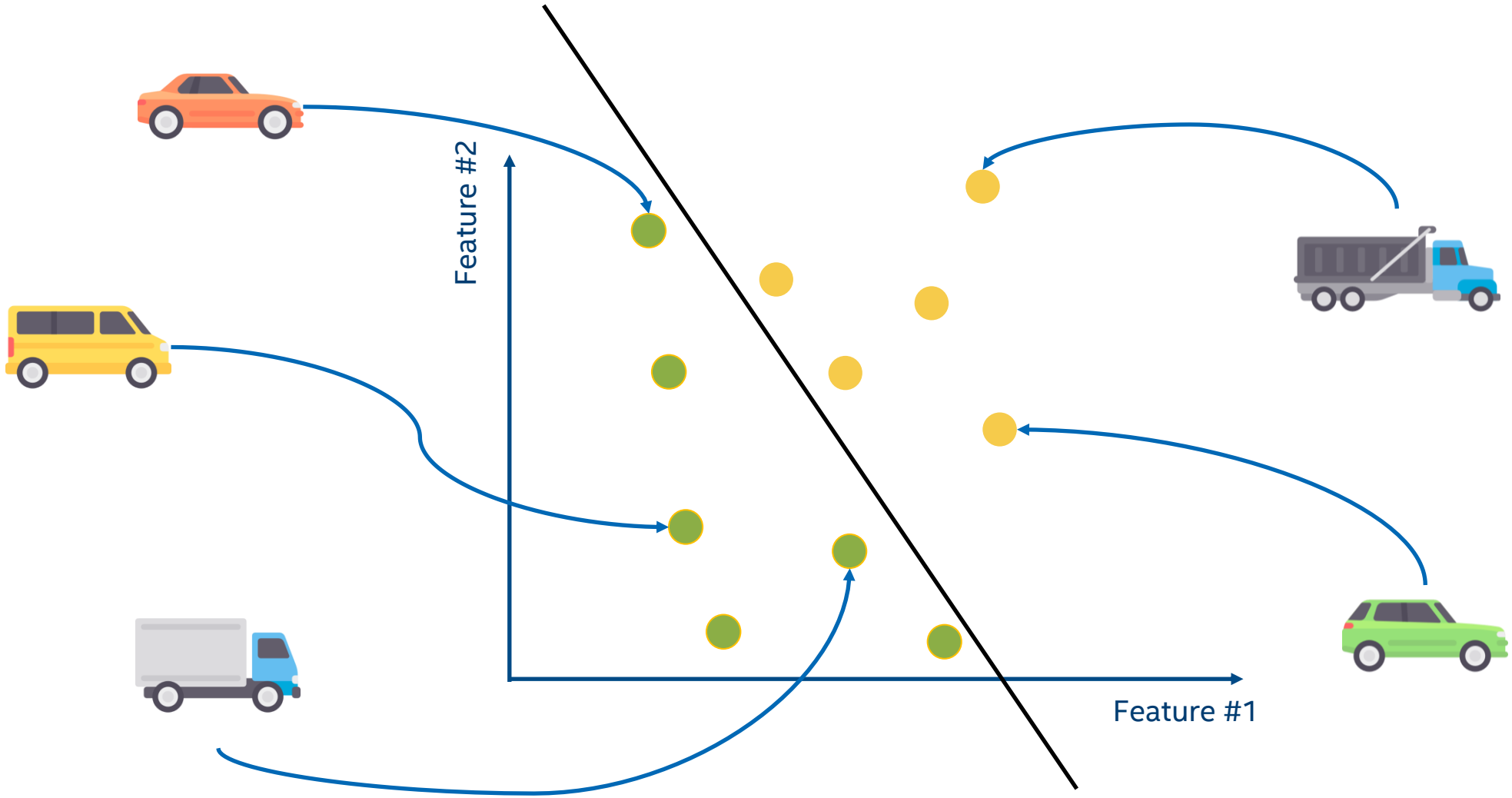
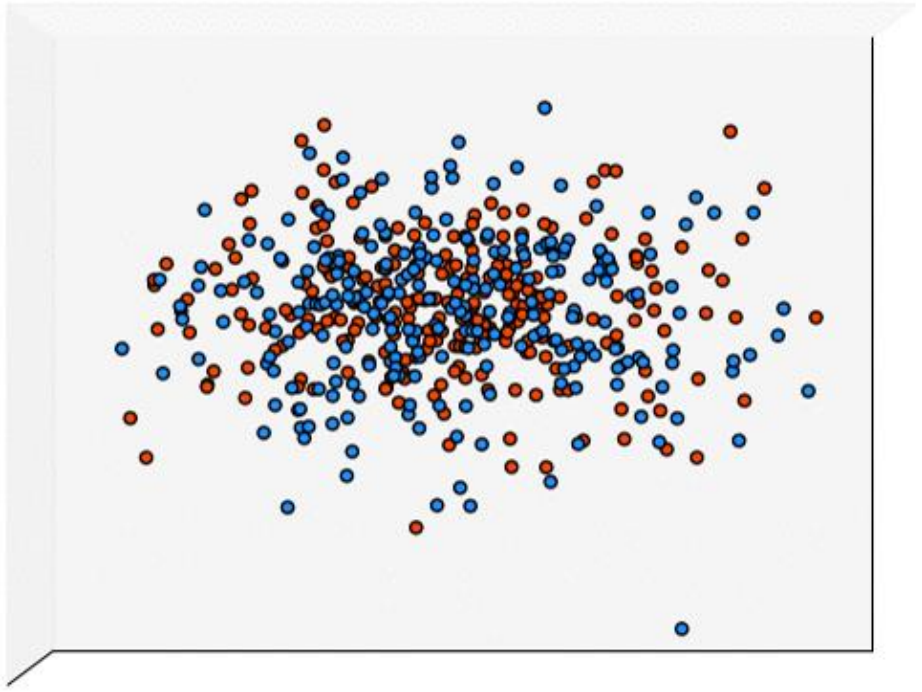
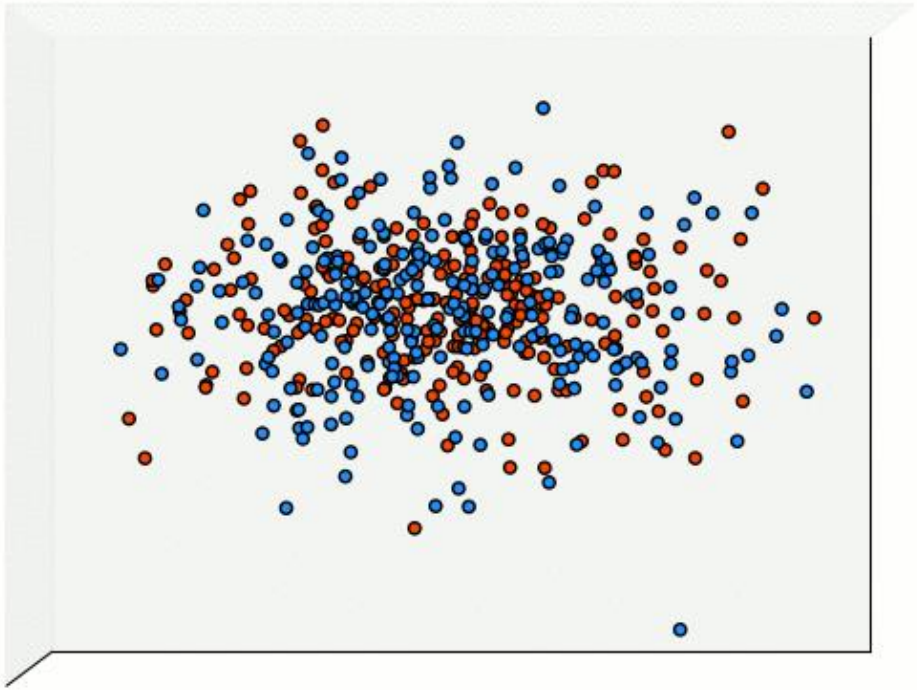


Image credits: Icons made by Freepik from www.flaticon.com.

Features



Features



Features

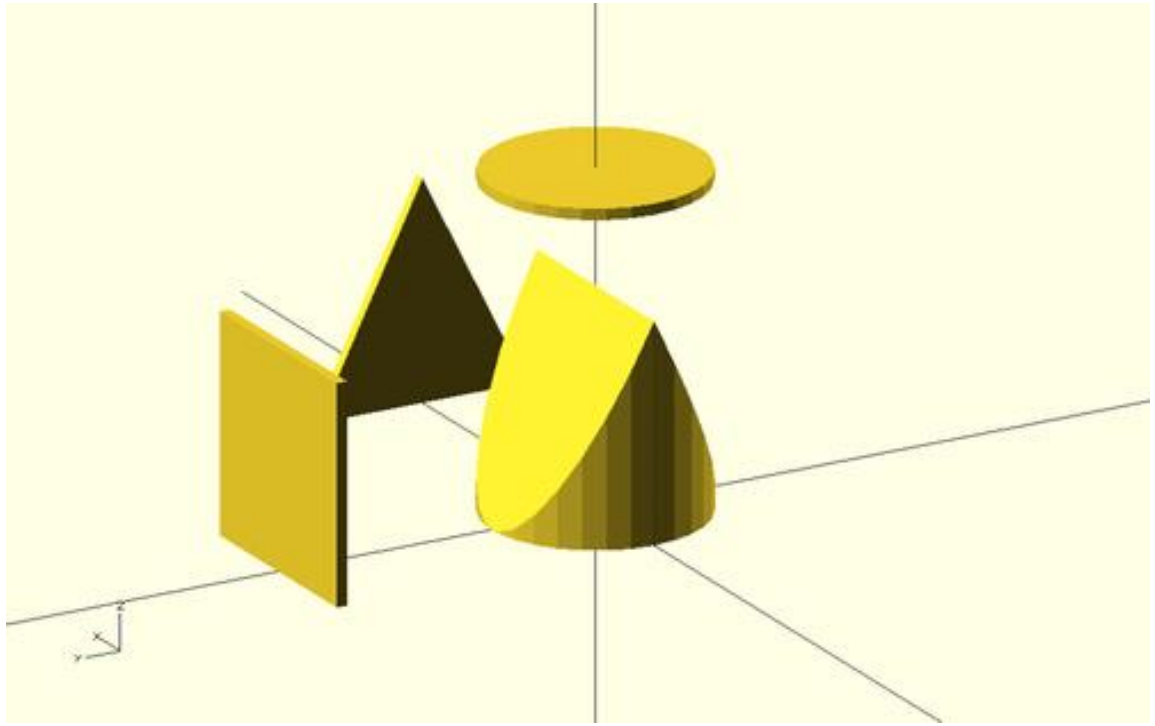


Image credits: http://kitwallace.tumblr.com/post/103975175234/george-harts-circle-triangle-square-puzzle

Supervised learning

Mathematical definition:

Defined:

- Space of viable input values $\mathcal{X}$.
- Space of viable output (target) values $\mathcal{Y}$.

Given:

- Dataset $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$, $|\mathcal{D}| = N$.

Target:

- Find a function (a.k.a. model) $h: \mathcal{X} \to \mathcal{Y}$ that is a good predictor of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$.

Feature space

- $x = (x_1, x_2, \ldots, x_d) \in \mathcal{X}$,
  where x_j is a feature that describes some subject property.
- $\mathcal{X} = \mathrm{P}_1 \times \mathrm{P}_2 \times \cdots \times \mathrm{P}_d$,
  where P_j is a set of viable values of feature j.
- P_j (and consequently $\mathcal{X}$) may have a difficult structure:
  - $\mathrm{P}_j = \mathbb{R}$;
  - $|\mathrm{P}_j| < \infty$ and P_j is ordered;
  - $|\mathrm{P}_j| < \infty$ and P_j is unordered;
  - etc.

Training and testing

- One should clearly distinguish between:
  - *Training* (a.k.a. learning or model fitting) phase.
    - Here we select the best model h^*.
    - As a rule, is extremely computation (time) and memory consuming part especially in the context of deep learning. Often takes place in datacenters.
  - *Testing* (a.k.a. prediction or inference or deployment or scoring) phase.
    - Here we evaluate a fixed function h^* in (arbitrary) points x.
    - Require much more light-weight computations but should work on embedded devices with limited computing power, memory and energy consumption and often in real-time.

Algorithms and models

- Supervised learning algorithm is defined by:
  - Class of functions $\mathcal{H}$ to search h in.
    - Broder family of functions increases the chance to have a good approximation in but makes it difficult to find it.
  - Quality metrics $\mathcal{Q}$ (or loss $\mathcal{L}$).
    - One should formalize what it means that one function is better than another and qualify the difference.
  - Optimization algorithm.
    - Defines a search procedure in $\mathcal{H}$ w.r.t. $\mathcal{Q}$.
    - We want to maximize $\mathcal{Q}$ (or minimize loss $\mathcal{L}$) on all viable data, which is not available or intractable.

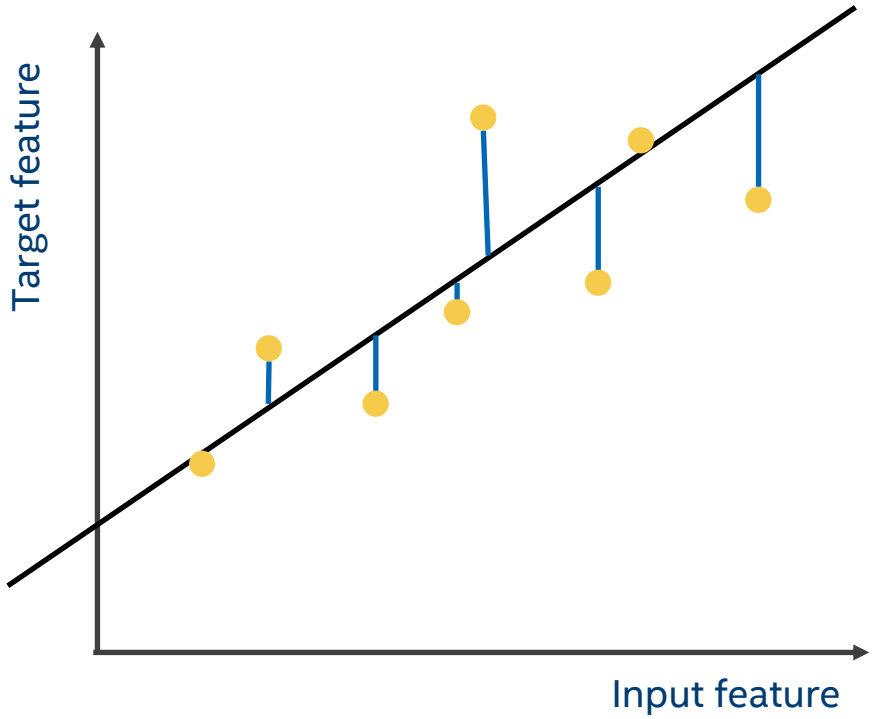
Empirical quality / loss

- As long as only a sample of all viable data is available,
  we can try to approximate model quality (or loss) using data at hand.

- Hence, we have *empirical* quality $\mathcal{Q}(h, \mathcal{D})$ or *empirical* loss $\mathcal{L}(h, \mathcal{D})$ estimates and an optimization problem:

$$h^* = \max_{h \in \mathcal{H}} \mathcal{Q}(h, \mathcal{D}) = \min_{h \in \mathcal{H}} \mathcal{L}(h, \mathcal{D}) \,.$$

Linear regression



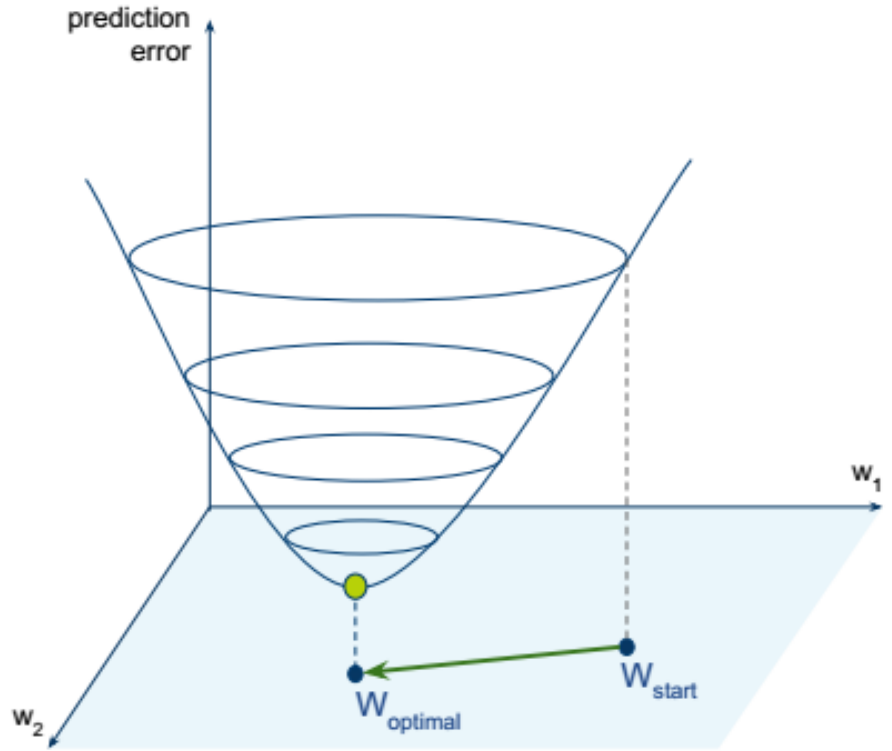
$$\mathcal{H} = \{\beta_0 + \beta_1 x\}$$

$$\mathcal{L}(h, \mathcal{D}) = \frac{1}{N}\sum_{1}^{N}\left(y^{(i)} - h\left(x^{(i)}\right)\right)^2 =$$

$$= \frac{1}{N}\sum_{1}^{N}\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)^2$$

$$h^* = \min_{h \in \mathcal{H}} \mathcal{L}(h, \mathcal{D})$$

Gradient descent

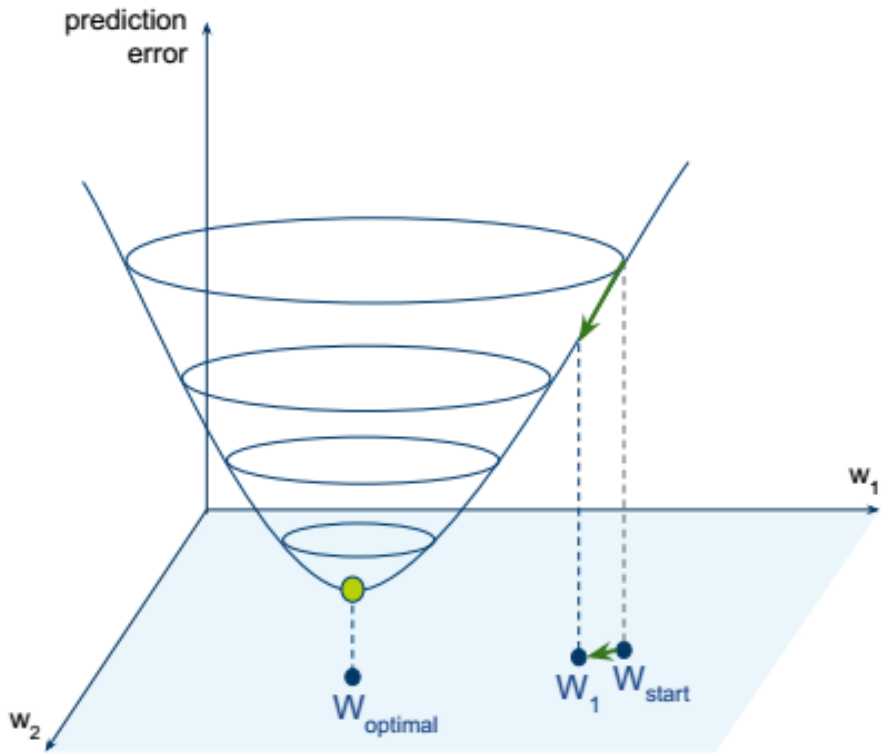


Let $\mathbf{W}$ be a model parameter space.

Get some (random?) initial position $\mathbf{W_0}$.

Image credits: Anna Petrovicheva, Introduction to Deep Learning.

Gradient descent



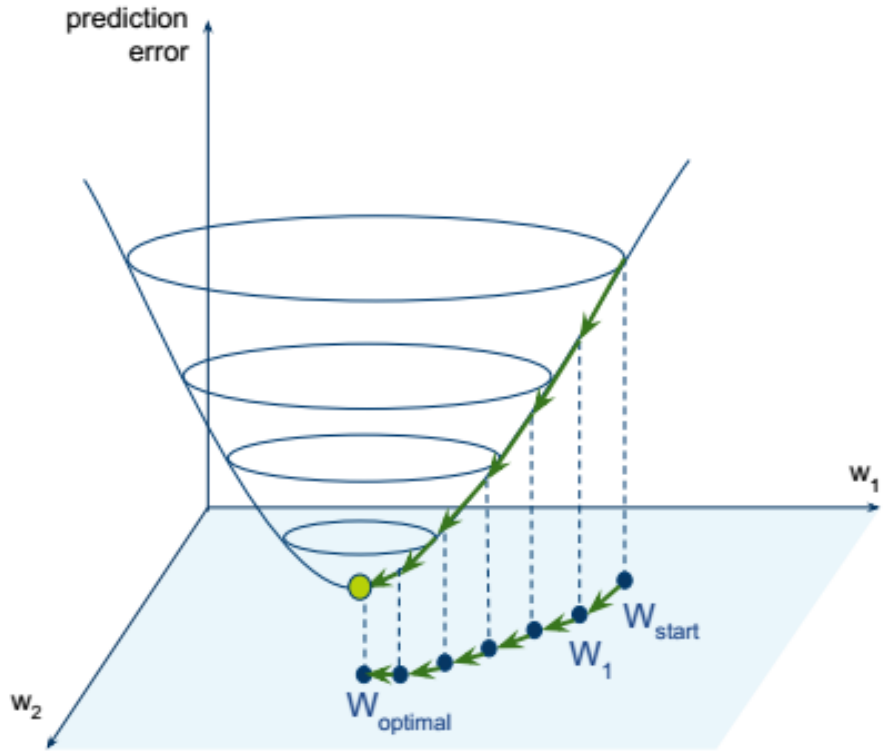
Compute the gradient (vector of partial derivatives) of $\mathcal{L}$ in this point. It will point to the direction of the functions' most rapid local ascent. Opposite direction – most rapid local descent.

Take a small step in this direction.

$$W_1 = W_0 - \alpha \nabla \mathcal{L}(W; \mathcal{D})$$

Image credits: Anna Petrovicheva, Introduction to Deep Learning.

Gradient descent



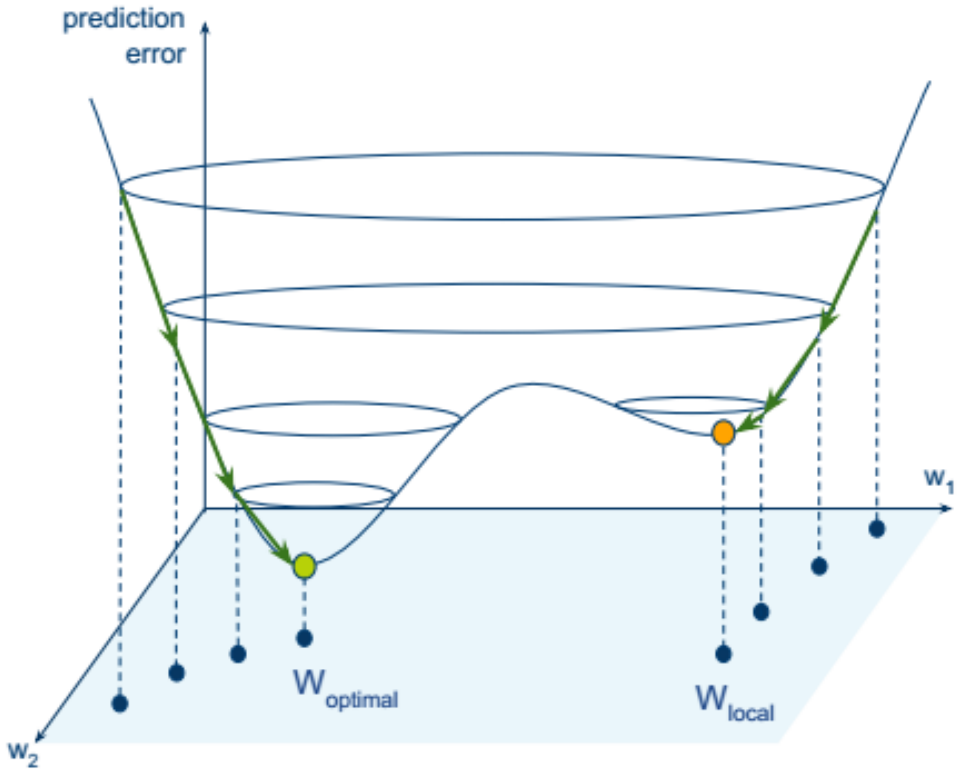
Repeat this procedure until convergence.

$$W_{i+1} = W_i - \alpha \nabla \mathcal{L}(W; \mathcal{D}).$$

α – parameter of the learning algorithm (hyperparameter), and is referred to as learning rate.

Image credits: Anna Petrovicheva, Introduction to Deep Learning.

Gradient descent



- Pros:
  - Quite generic.
  - Easy to implement.
  - Resource friendly, especially in a stochastic modification.
- Cons:
  - Target function must be differentiable.
  - Can get stuck in a local minima or saddle point. So, no guarantee to reach global minima.
  - Solution depends heavily on initial guess.
  - Depending on the target function landscape may require long time to converge.

Image credits: Anna Petrovicheva, Introduction to Deep Learning.

Model generalization

- How good is empirical quality estimate?
- How good is the model that optimizes empirical quality on some training set?

- It’s easy to show that optimization of empirical quality on training set does not guarantee that model generalizes to the new data.

Model generalization

- Let's build model h in the following way:

$$h(x) = \begin{cases} y_i \;\; if \;\; \exists\, x_i \in \mathcal{X} \;\; and \;\; x_i = x, \\ random \;\; y \;\; from \;\; \mathcal{Y}. \end{cases}$$

  i.e., just remember the whole training dataset.

- $Q(h, \mathcal{D})$ is high, but this model makes random guesses on data points outside of the training datasets, hence $Q(h)$ is low.

Model generalization

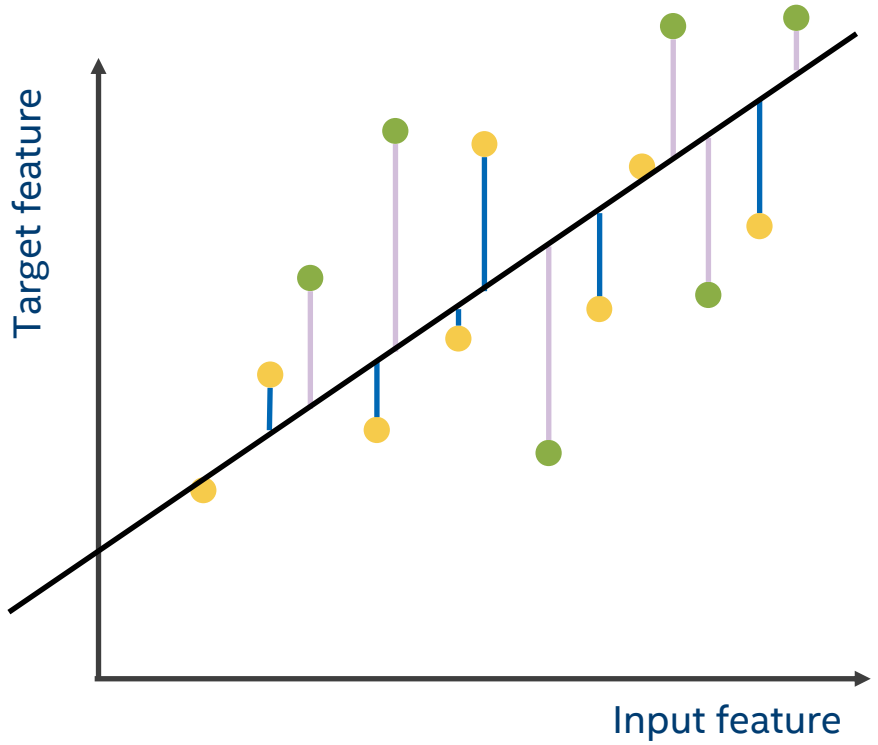
- Conclusions:
  - Model generalization for the data unseen at training time is essential.
  - Measure of model generalization is required.
  - Generalization measure should be incorporated into the optimization objective somehow.

Estimate of model generalization

- Having a dataset for the same task but independent from the training one, which is sampled under some reasonable constraints (i.i.d.), may provide us an unbiased estimate of model quality (not empirical!).

- This dataset is commonly referred to as ***validation*** or ***test*** one.

Linear regression



Overfitting

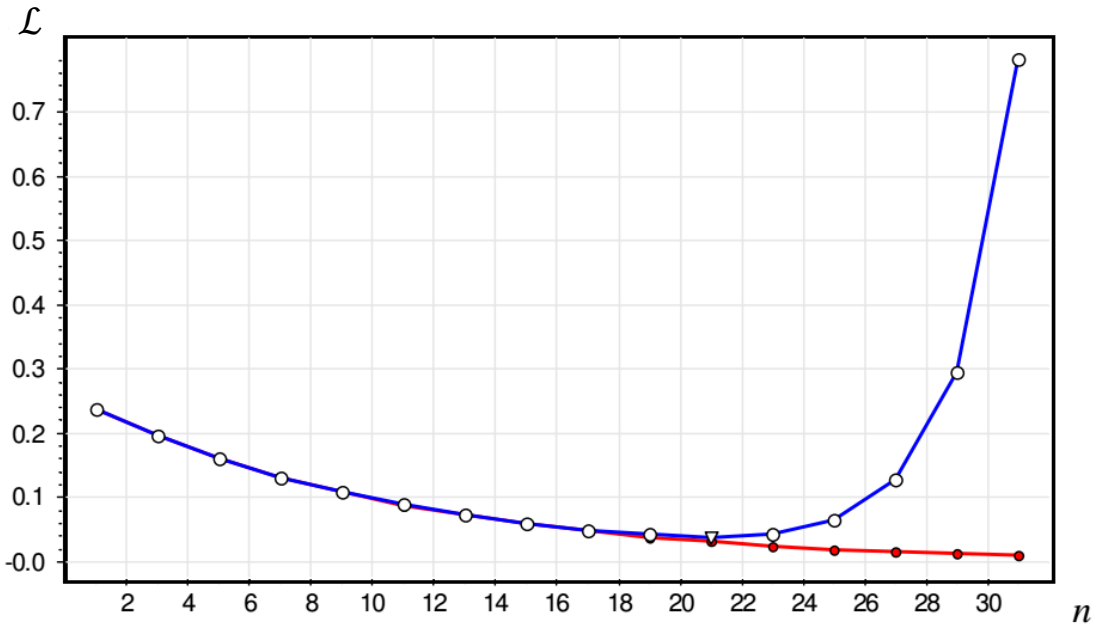
- Situation when $Q(h, \mathcal{D}_{train}) \gg Q(h, \mathcal{D}_{test})$ is called overfitting.
- Overfitting is one of the main concepts in machine learning.
- Overfitting occurs when:
  - We don't have enough data.
  - Our set of viable models $\mathcal{H}$ is too broad, and model h^* is overcomplicated.

Overfitting

$$y(x) = \frac{1}{1+25x^2}$$

$$\mathcal{H} = \{a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n\}$$

$$\mathcal{L} = \frac{1}{\#D}\sum_{(x,y)\in D}(h(x) - y)^2$$



n = 38

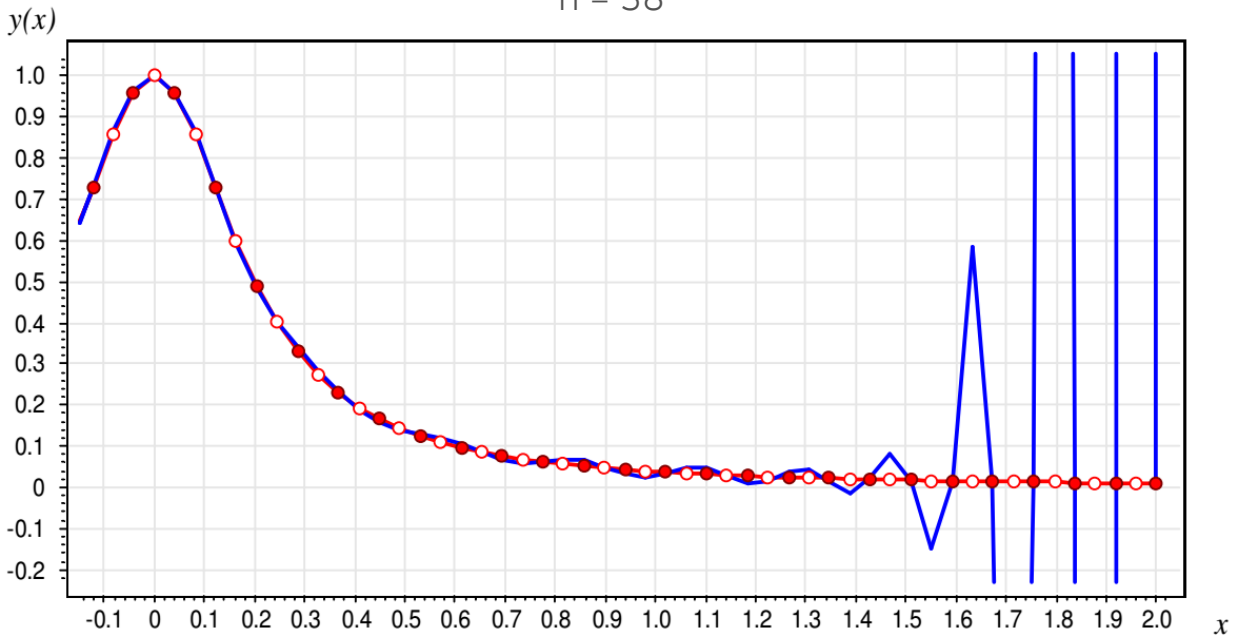


Image credits: http://www.machinelearning.ru/wiki/images/f/fc/Voron-ML-Intro-slides.pdf

Overfitting

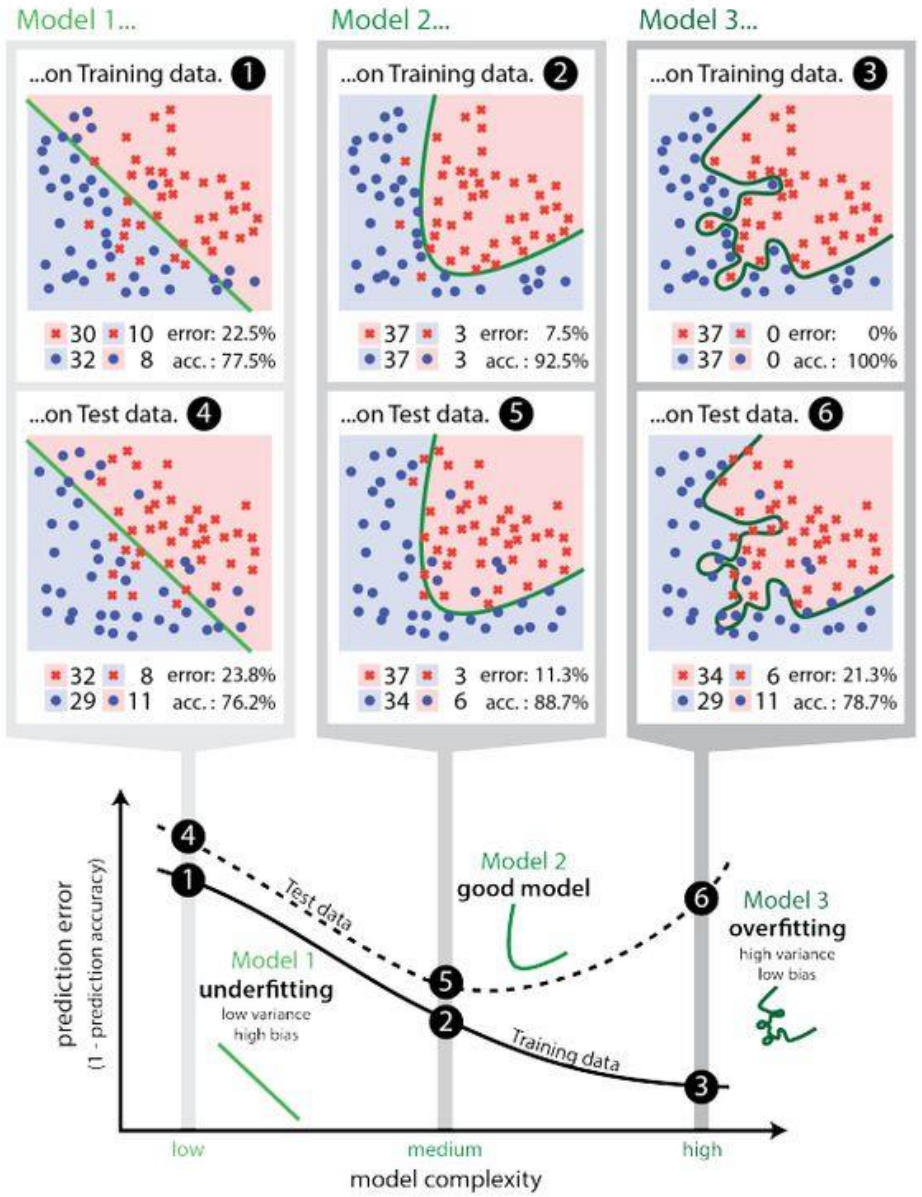


Image credits: https://pin.it/otqalezmvhx4zr

Regularization

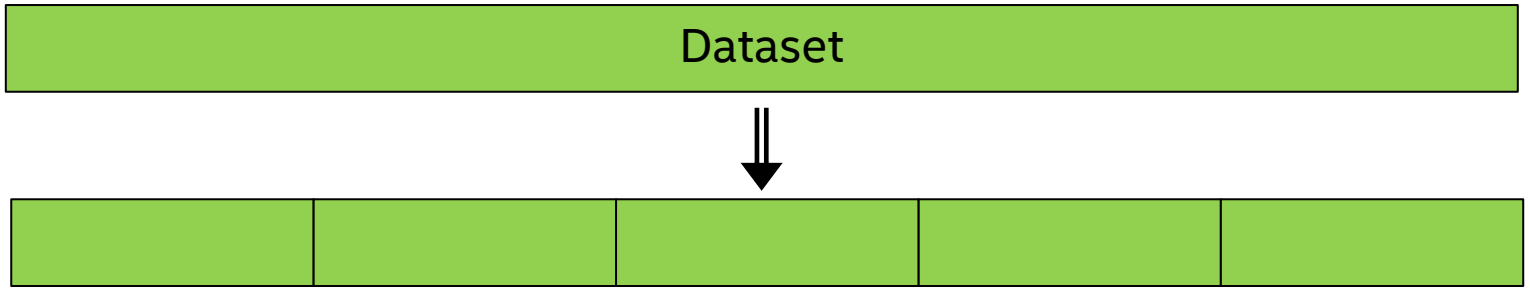
- Regularization is a common name for techniques that are intended to reduce generalization (test) error rather than simply reduce training loss.

- Example:

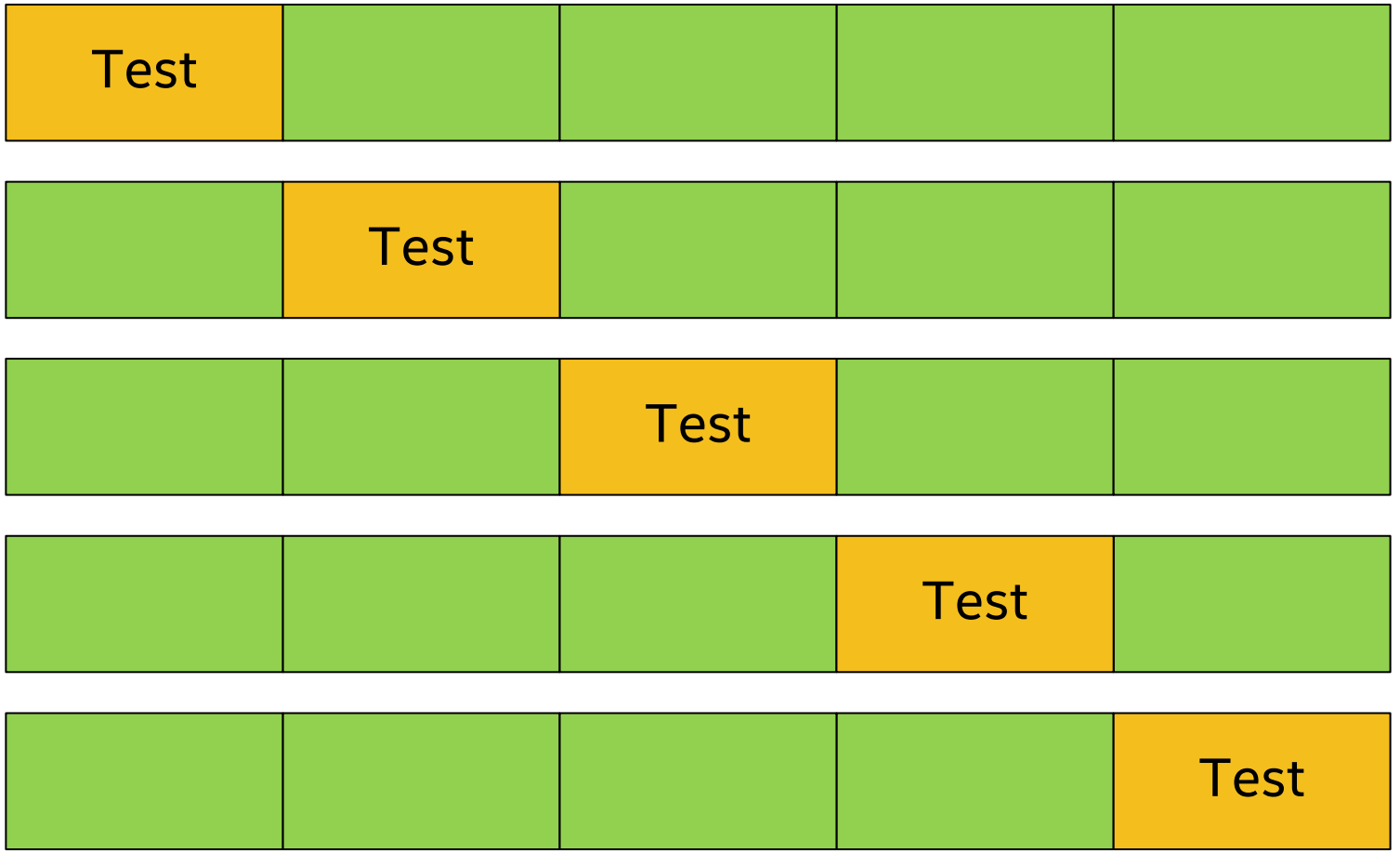
$$h^* = \max_{h \in \mathcal{H}} Q(h, \mathcal{D}) + \lambda \sum_{i=1}^{n} a_i^2 .$$

Cross-validation

- The more data points are there in test dataset the more reliable model quality estimate is.
- The more data points are there in training dataset the more chances to fit a better model.
- There always is a tradeoff between sizes of train and test datasets.
- One of the popular ways to get better unbiased estimate of model quality is to use cross-validation: split dataset into train/test several times and average quality measures over this folds.



Cross-validation



$$Err_1 \rightarrow$$

$$Err_2 \rightarrow$$

$$Err_3 \rightarrow$$

$$Err_4 \rightarrow$$

$$Err_5 \rightarrow$$

$$Err_{CV} = \frac{1}{k}\sum_{i=1}^{k} Err_i$$

No free lunch

- Is there a single best machine learning algorithm?
  - No. No machine learning algorithm is universally better that any other. All machine learning algorithms have the same error rate on a test dataset if it is averaged over all possible target functions.
- The goal is to understand what kinds of distributions are relevant to real-world tasks and what kinds of machine learning algorithms perform well under these assumptions.

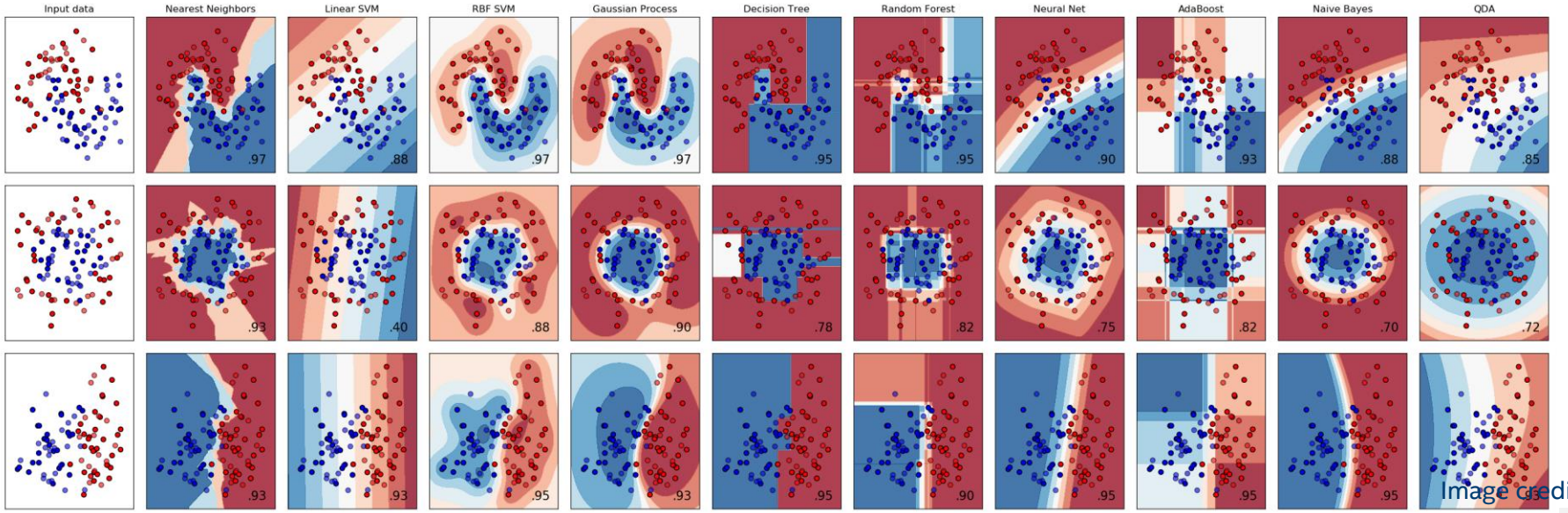


Image credits: http://scikit-learn.org

Beyond supervised learning

- Supervised learning algorithms require labeled data, which is expensive.
- For some tasks these is no single and well-defined target, or it’s not available at all.

Reinforcement learning

- Situations when we know what we want, but don't know how to get it.
- E.g., sequential decision making.
  - It's hard to say what actions should some agent perform to reach the goal.
  - Every action has consequences that affect later actions.
  - It's hard to describe all successful sequences of actions.
- Direct supervision is not available. But allowing the ***agent*** to interact with an ***environment*** (perform ***actions***) we can get a feedback – receive a numerical ***reward*** signal (experience consequences of actions), which can guide learning.
- The goal is to estimate and maximize the long-term cumulative reward by finding the best ***policy*** for agents' behavior.

Reinforcement learning

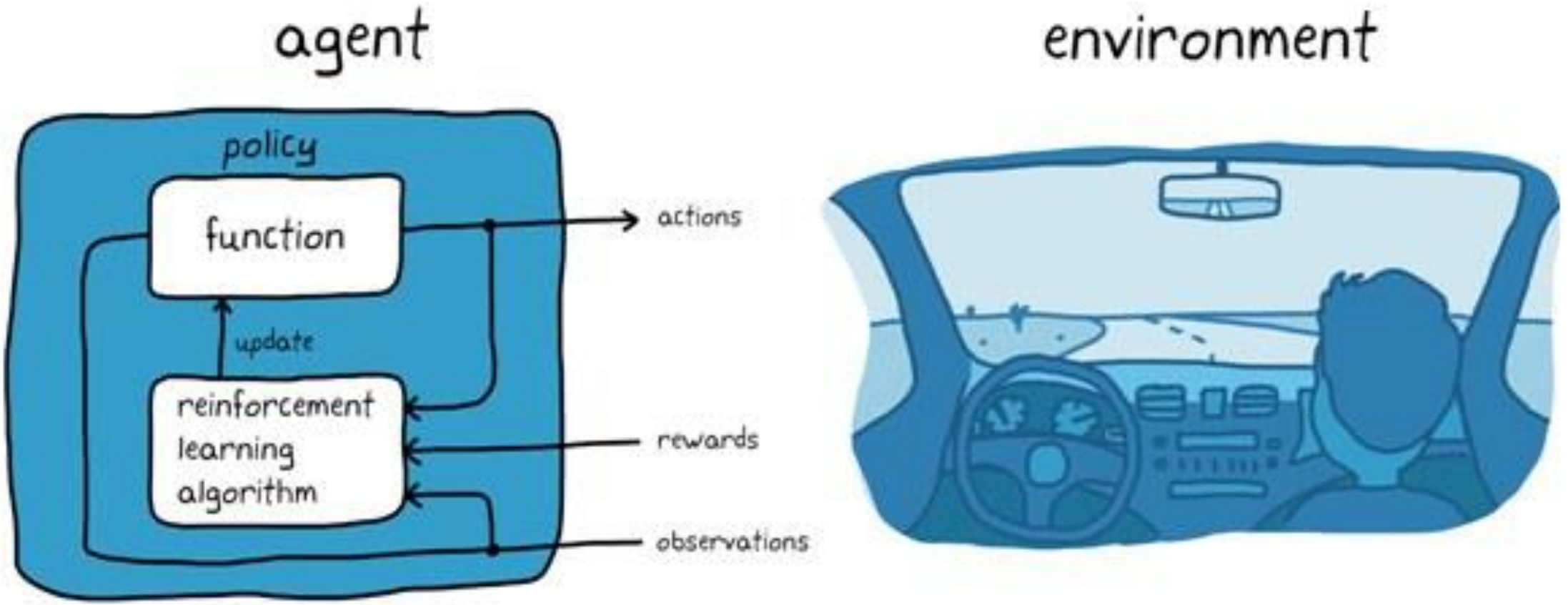


Image credits: https://www.kdnuggets.com/2019/10/mathworks-reinforcement-learning.html

Reinforcement learning

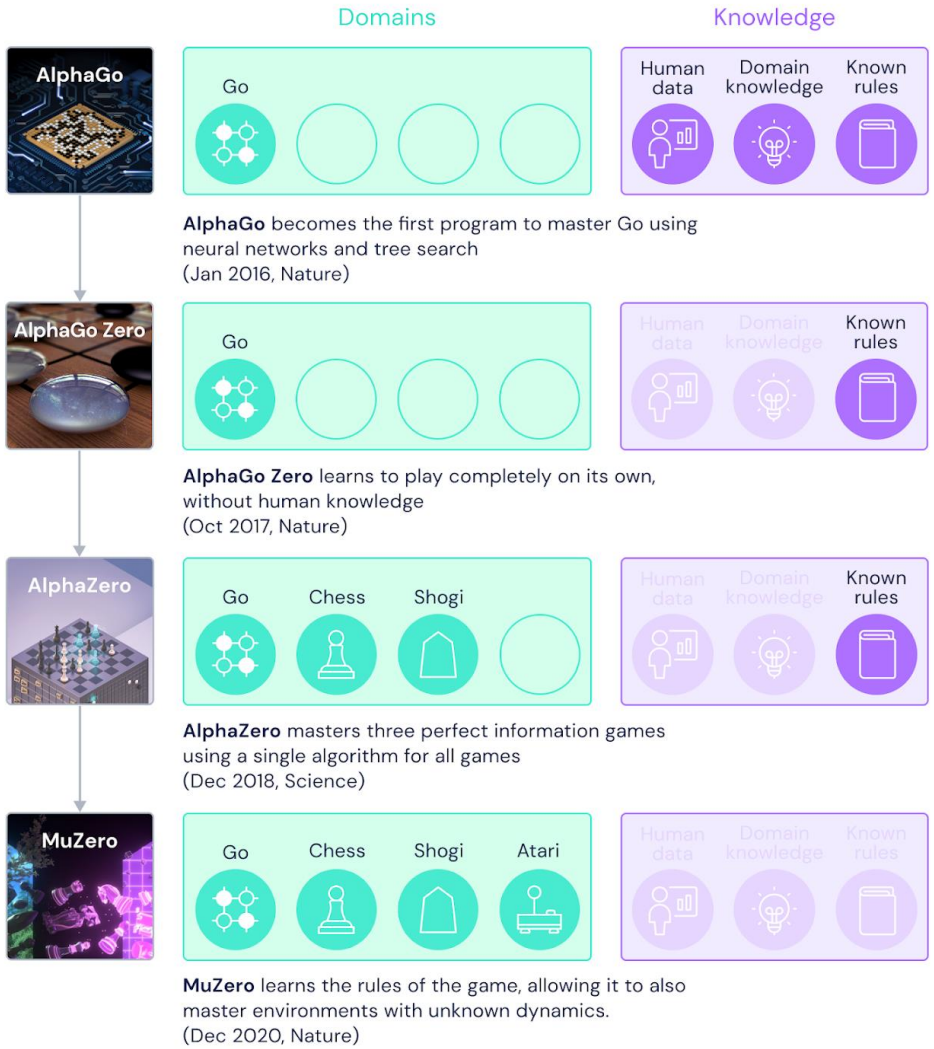
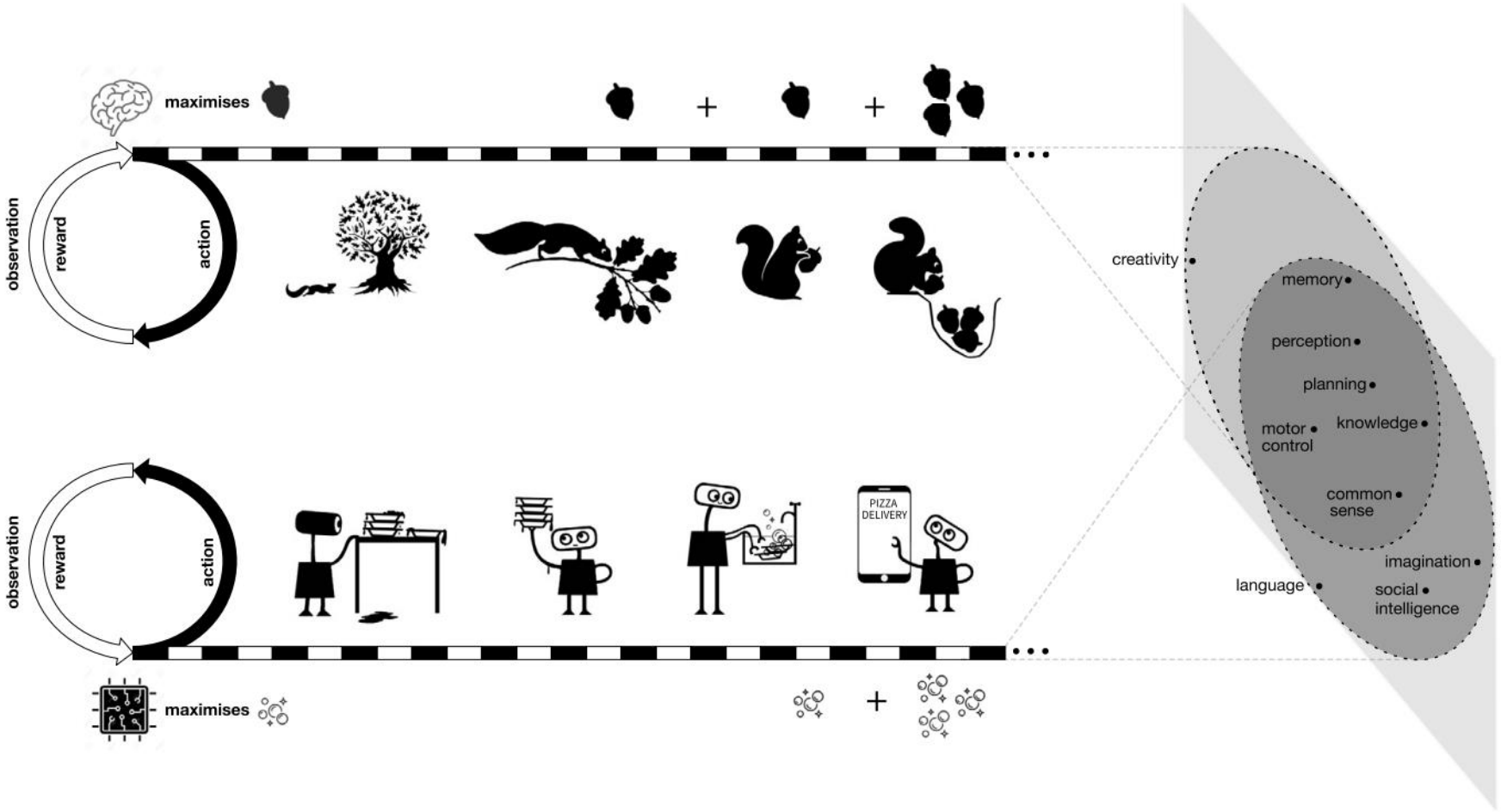


Image credits: https://deepmind.com/blog/article/muzero-mastering-go-chess-shogi-and-atari-without-rules

Reinforcement learning



Unsupervised learning

- We just want to find some useful / interesting associations in our data.
- No target feature, no reward.
- Often an ill-defined problem.
- Examples:
  - Clustering: group similar data points together and put dissimilar data points in different groups.
  - Dimensionality reduction: reduce feature space dimension without a loss of some useful dependencies / regularities.

Applications. Marketing and business

- Churn prediction.
  Based on a user behavior predict if he/she wants to quit.
- Profit prediction.
  Given some parameters of the new product and state of the market, predict what will be the revenue from its sales.
- Sentiment analysis.
  Based on text of customer review determine if it is positive or negative.
- Market / customer segmentation.
  Based on customer profile and behavior find similar customer groups.
- Market basket analysis.
  Based on the history of customer purchases determine which products are often bought together.

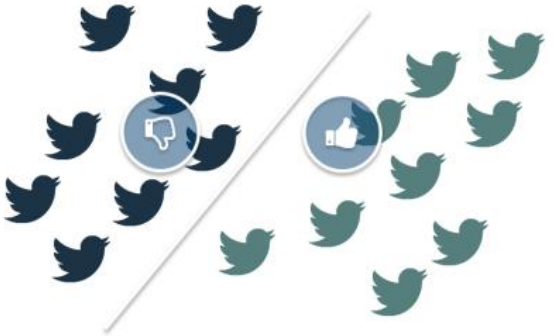


Image credits: https://homelet.co.uk/letting-agents/news/article/Why-landlords-should-target-families-and-retirees; Nervana AI Academy, Deep Learning Student Workshop

Applications. Security

- Fraud / anomaly detection.
  Based on some features of sample (bill, transaction, etc.) determine whether it is a valid sample or not.
- Identification / verification.
  Identify a user/customer based on his behavior.

Applications. Medicine

- Diagnosis.
  Based on patients' test results determine the disease he/she has (if any).
- Treatment planning.
  Based on current patient state develop an optimal treatment plan.
- Personalized medicine.
  Based on medical test results of a particular patient develop an optimal personalized treatment plan.
- Drug discovery.
  Based on known effects of different drugs application and possibly tests of new (generated at algorithms' runtime) drugs, generate novel, more effective drugs.

Applications. Efficient search

- Information retrieval.
  Based on the query and personalized search and click results rank search output.
- Natural language / image understanding.
  Get better understanding what is the semantic meaning of a text/image query and/or documents to do better matching.
- Recommendation systems (collaborative filtering).
  Based on personal user profile, its similarity to other users and their profiles recommend a product that user will like.


intel