

Введение в машинное обучение

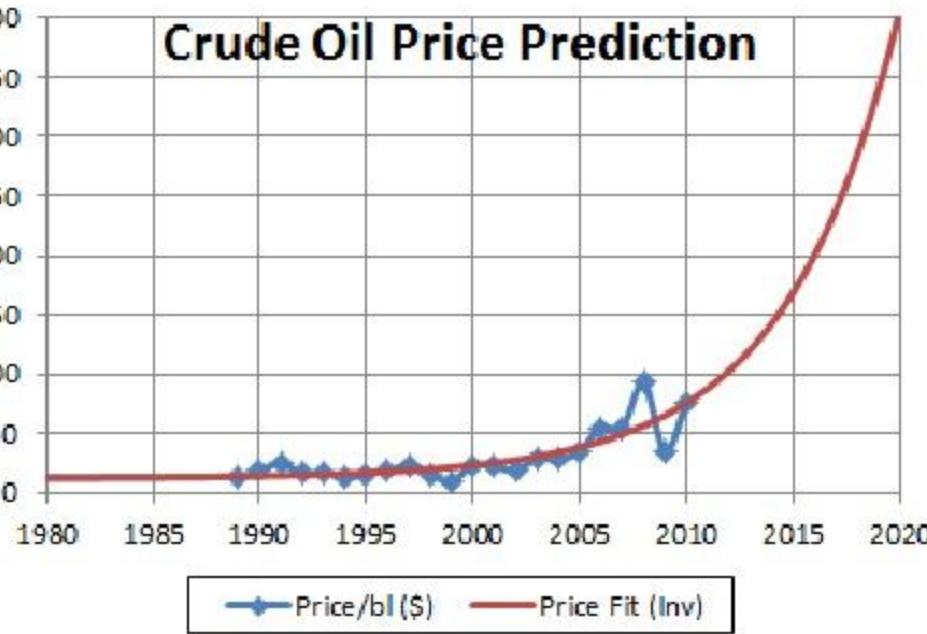
Татьяна Гайнцева
МФТИ, ШАД, DLSchool

@atmyre

Problems statement



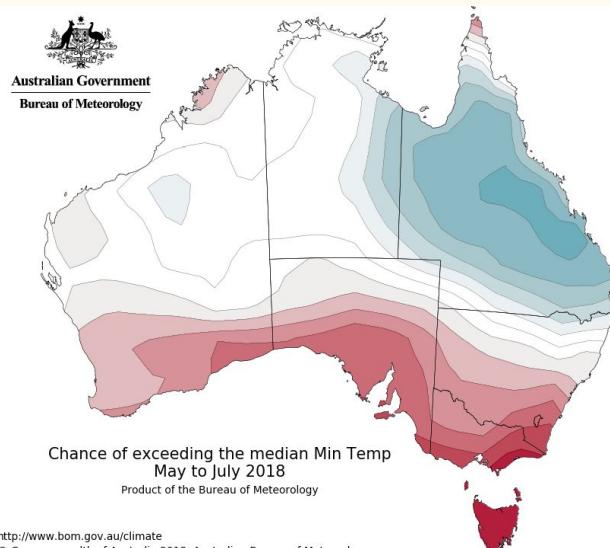
Crude Oil Price Prediction



Good Credit Score



Australian Government
Bureau of Meteorology



Issued: 12/04/2018
Model Run: 08/04/2018
Base Period: 1981-2010

Специально для Вас

Аффинаж, rsac, номер скрыт и другие

▶ СЛУШАТЬ ВСЕ



Приглядитесь к этим предложениям



3 295 ₽

-50 %

6 590 ₽

Кеды VANS



7 030 ₽

-35 %

10 800 ₽

Внешняя звуковая



1 875 000 ₽

Виниловый
проигрыватель Spira...



4 400 ₽

-30 %

6 290 ₽

Кеды VANS



11 790 ₽

Лонгборд GoldCoast
Standard

где найти

где найти

где найти работу

где найти девушку

где найти друзей

где найти парня

где найти мужа

где найти деньги

где найти ответы на огэ 2018

где найти ответы на егэ 2018

где найти алису

Страница



Свидание III в Москве
Событие

УДИВИТЕЛЬНОЕ ПРЕДЛОЖЕНИЕ



Линзы ACUVUE® в

"Очкарик"!

ochkarik.ru

Удивительное
предложение на
контактные линзы
ACUVUE OASYS® 1-Day!

Есть противопоказания.
Требуется консультация
специалиста.

Блог Разработчикам

5491_219500906

Сначала интересные



беру! БЕТА



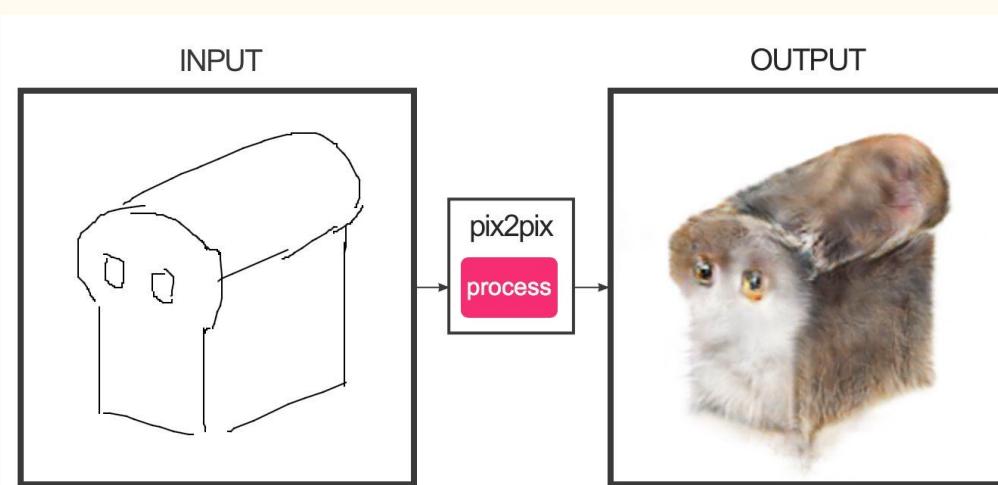
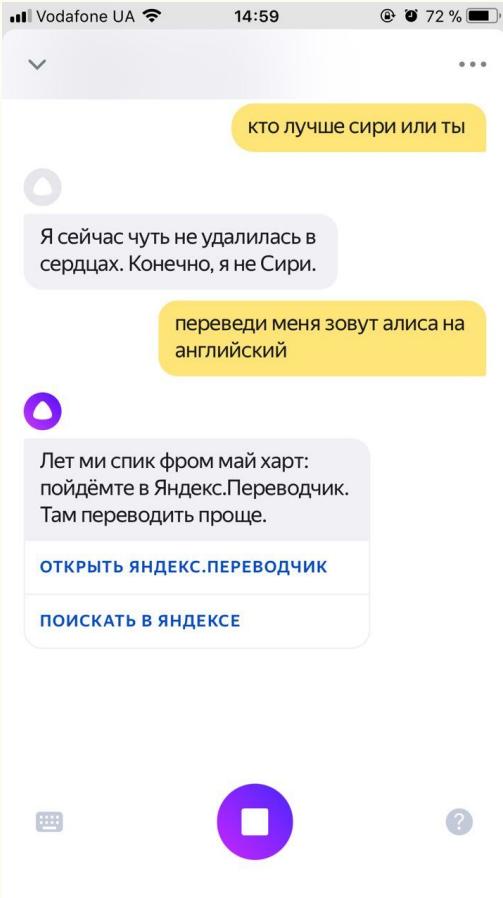




Figure 8: Interpolations between z, c pairs.

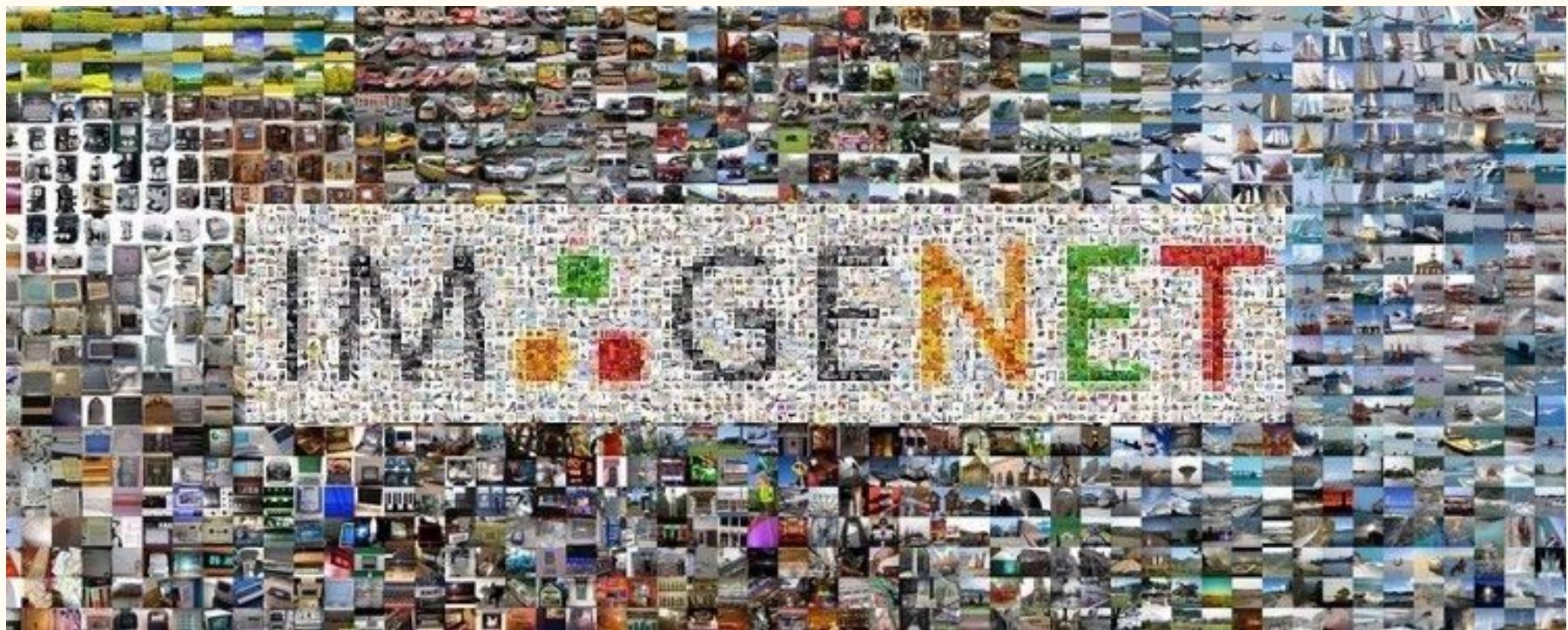
<https://arxiv.org/pdf/1809.11096.pdf>

<https://deepmind.com/blog/>

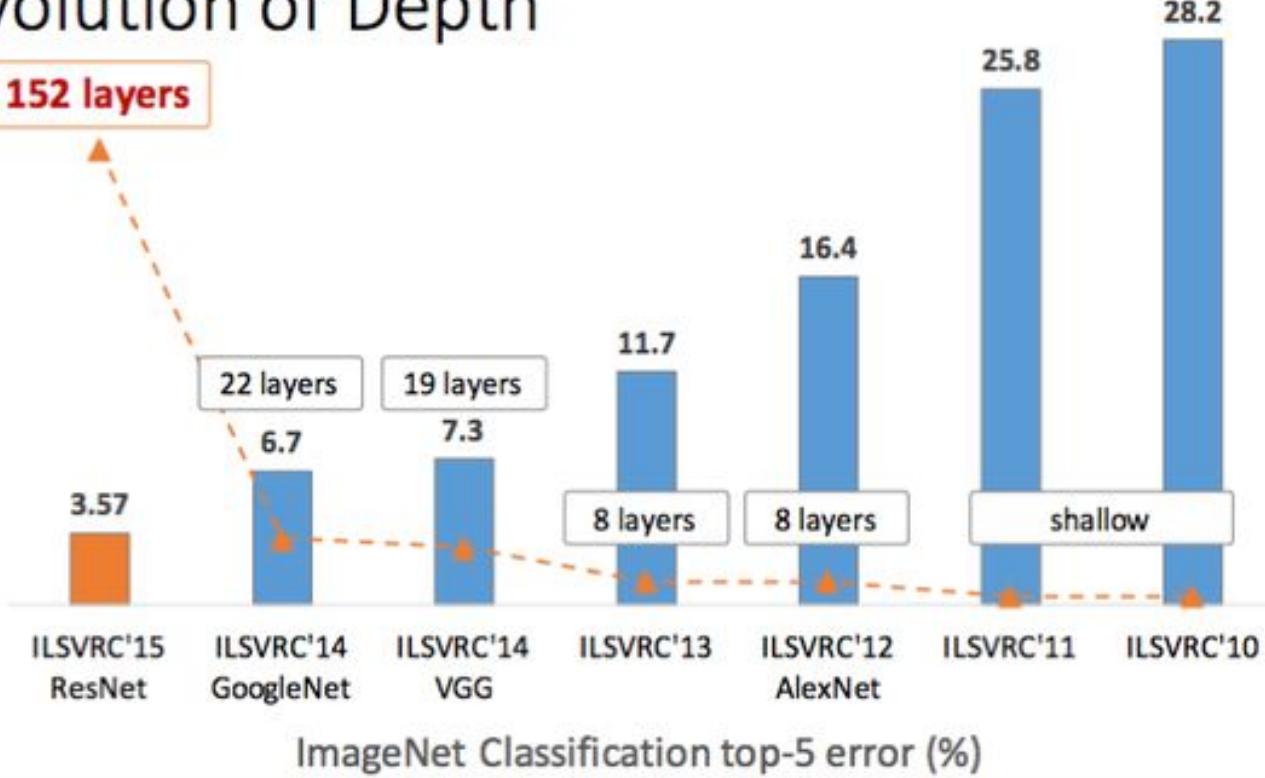
Timeline

<https://mipt.ru/science/labs/laboratoriya-neyronnykh-sistem-i-glubokogo-obucheniya/>

- 1997 IBM Deep Blue обыграл чемпиона мира по шахматам
- 2005 Беспилотный автомобиль: DARPA Grand Challenge
- 2006 Google Translate – статистический машинный перевод
- 2011 40 лет DARPA CALO привели к созданию Apple Siri
- 2011 IBM Watson победил в ТВ-игре «Jeopardy!»
- 2011–2015 ImageNet 25% → 3.5% ошибок против 5% у людей
- 2012 Google X Lab: распознавание видеокадров с котами
- 2014 Facebook DeepFace распознаёт лица с точностью 97%
- 2015 Фонд OpenAI в \$1 млрд. Илона Маска и Сэма Альтмана
- 2016 DeepMind, OpenAI: динамическое обучение играм Atari
- 2016 Google DeepMind обыграл чемпиона мира по игре го
- 2017 OpenAI обыграл чемпиона мира по компьютерной игре Dota 2

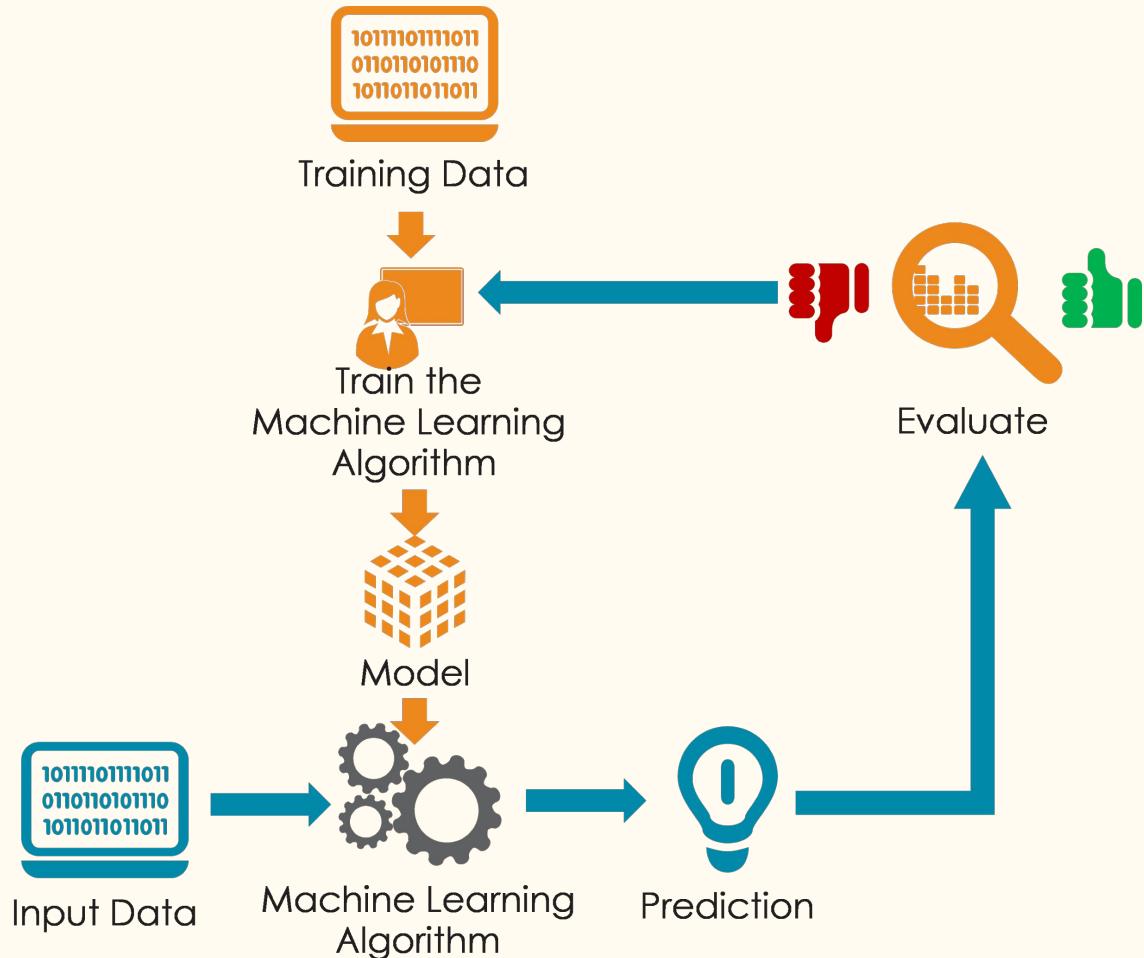


Revolution of Depth



ML Language





X – множество **объектов**

Y – множество **допустимых ответов**

y^* – целевая функция, $y^*: X \rightarrow Y$, $y_i = y^*(x_i)$ известны только на **конечном** подмножестве объектов x_1, \dots, x_m из X

Пары (x_i, y_i) – прецеденты

Совокупность пар таких пар при i из $1, \dots, m$ – **обучающая выборка** (X_{train})

a – **решающая функция** (алгоритм), которая любому объекту из X ставит в соответствие допустимый ответ из Y и приближает целевую функцию y^*

X_{test} – **выборка прецедентов** для тестирования построенного алгоритма a

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 310128
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450
6	0	3	Moran, Mr. James	male		0	0	330877
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909

Titanic dataset (<https://www.kaggle.com/c/titanic>)

Признак (feature) f объекта x — это результат измерения некоторой характеристики объекта. Формально признаком называется отображение $f : X \rightarrow D_f$, где D_f — множество допустимых значений признака. В частности, любой алгоритм $a : X \rightarrow Y$ также можно рассматривать как признак

Пусть дан набор признаков $f_1(x), \dots, f_n(x)$.

Признаковое описание объекта x — вектор (одномерный массив) (f_1, \dots, f_n) . Совокупность признаковых описаний всех объектов выборки длины m , записанную в виде таблицы размера mp , называют матрицей объектов–признаков.

X

y*

features

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 310128
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450
6	0	3	Moran, Mr. James	male		0	0	330877
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909

X_train=Object_ids+Features+y*

Y={0,1}

Titanic dataset (<https://www.kaggle.com/c/titanic>)

X

features

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47	1	0	363272	7
894	2	Myles, Mr. Thomas Francis	male	62	0	0	240276	9.6875
895	3	Wirz, Mr. Albert	male	27	0	0	315154	8.6625
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22	1	1	3101298	12.2875
897	3	Svensson, Mr. Johan Cervin	male	14	0	0	7538	9.225
898	3	Connolly, Miss. Kate	female	30	0	0	330972	7.6292
899	2	Caldwell, Mr. Albert Francis	male	26	1	1	248738	29
900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18	0	0	2657	7.2292
901	3	Davies, Mr. John Samuel	male	21	2	0	A/4 48871	24.15

X_test=Object_ids+Features

Кредитный скоринг

CATEGORY	RANGE
Excellent (28% of people)	750 - 850
Good (10% of people)	700 - 749
Fair (16% of people)	650 - 699
Poor (32% of people)	550 - 649
Very Poor (14% of people)	350 - 549

Feature importance



1. Пол
2. Доход семьи
3. Опыт работы
4. Кредитная история
5. Возраст
6. Уровень образования
7. Профессия
8. Место проживания
9. Недвижимость в собственности

ML task

По выборке X_{train} построить решающую функцию (*decisionfunction*)
 $a : X \rightarrow Y$, которая приближает целевую функцию y^* , причём не
только на объектах **обучающей выборки, но и на всём
множестве X .**

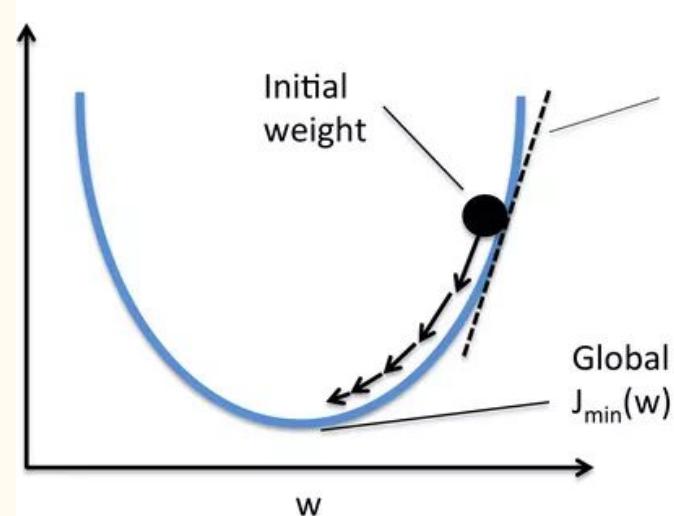
Решающая функция a должна быть вычислимой.

Как строится функция а?

Обучающая выборка — выборка, по которой производится настройка (оптимизация параметров) модели зависимости.

Тестовая выборка — выборка, по которой оценивается качество построенной модели.

Функционал качества (обучение с учителем) — определяется как средняя ошибка ответов, выданных алгоритмом, по всем объектам выборки.



Мы хотим найти в пространстве параметров такой набор параметров, чтобы ошибка алгоритма с этими параметрами была минимальна (функционал качества был максимален)

Метрики качества

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$logloss = -\frac{1}{l} \cdot \sum_{i=1}^l (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

$$PRE = \frac{TP}{TP + FP}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

$$F_1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC}$$

Алгоритмы

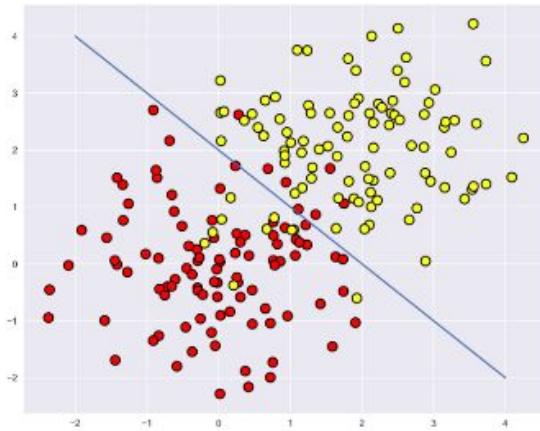






Классификация

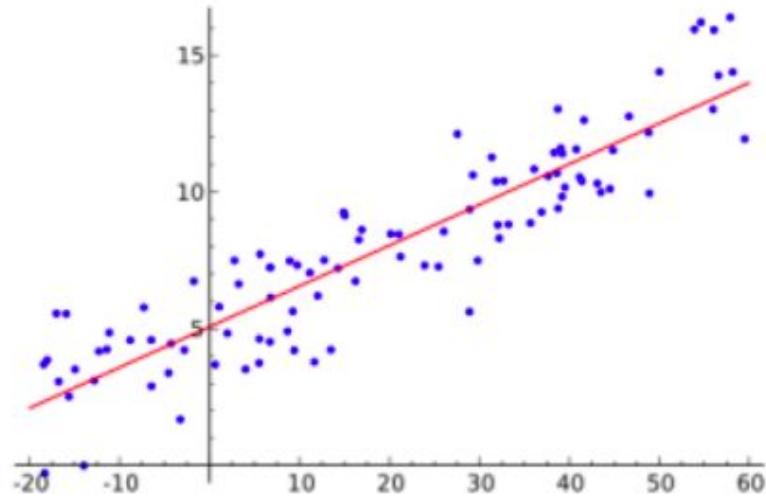
*Множество допустимых ответов конечно. Их называют метками классов (*class label*). Класс — это множество всех объектов с данным значением метки.*





Регрессия

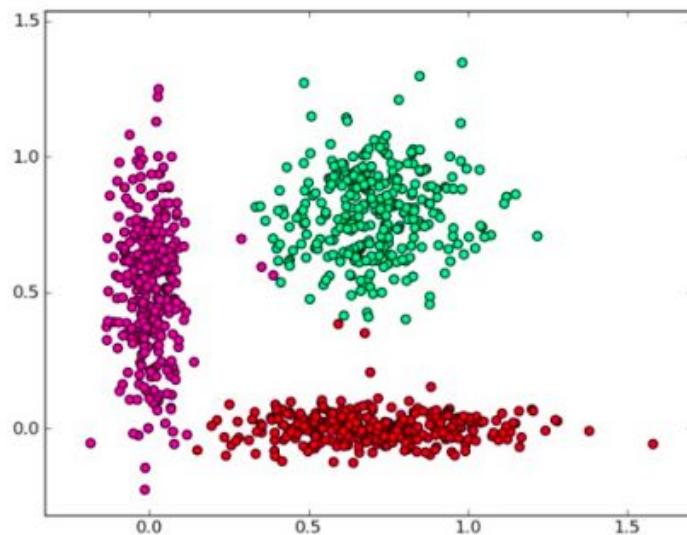
Отличается тем, что допустимым ответом является действительное число или числовой вектор.



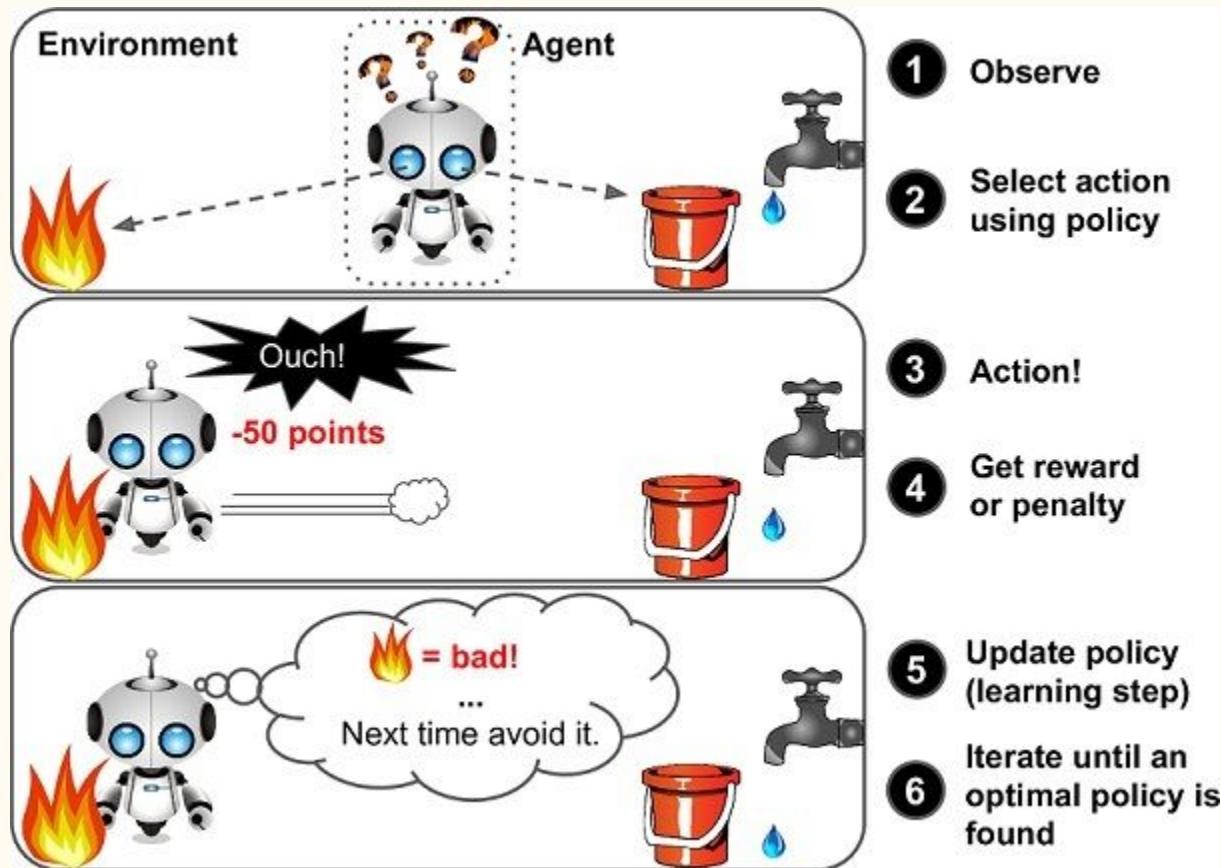


Кластеризация

Заключается в том, чтобы сгруппировать объекты в кластеры, используя данные о попарном сходстве объектов. Функционалы качества могут определяться по-разному, например, как отношение средних межкластерных и внутрикластерных расстояний.

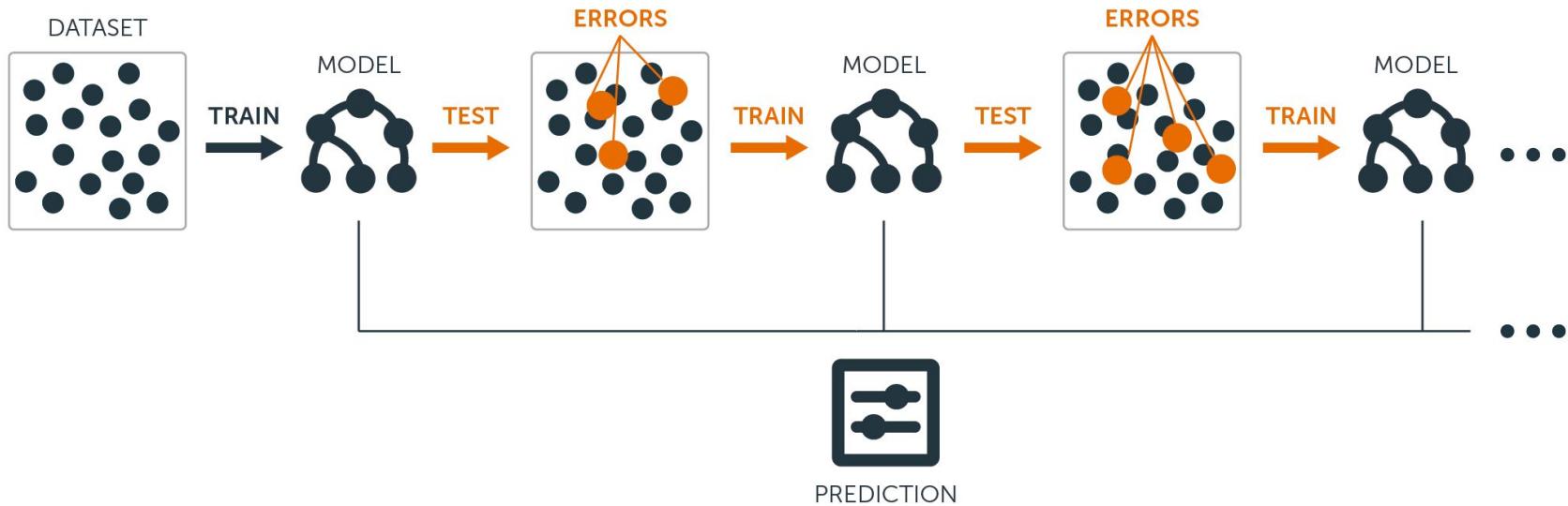


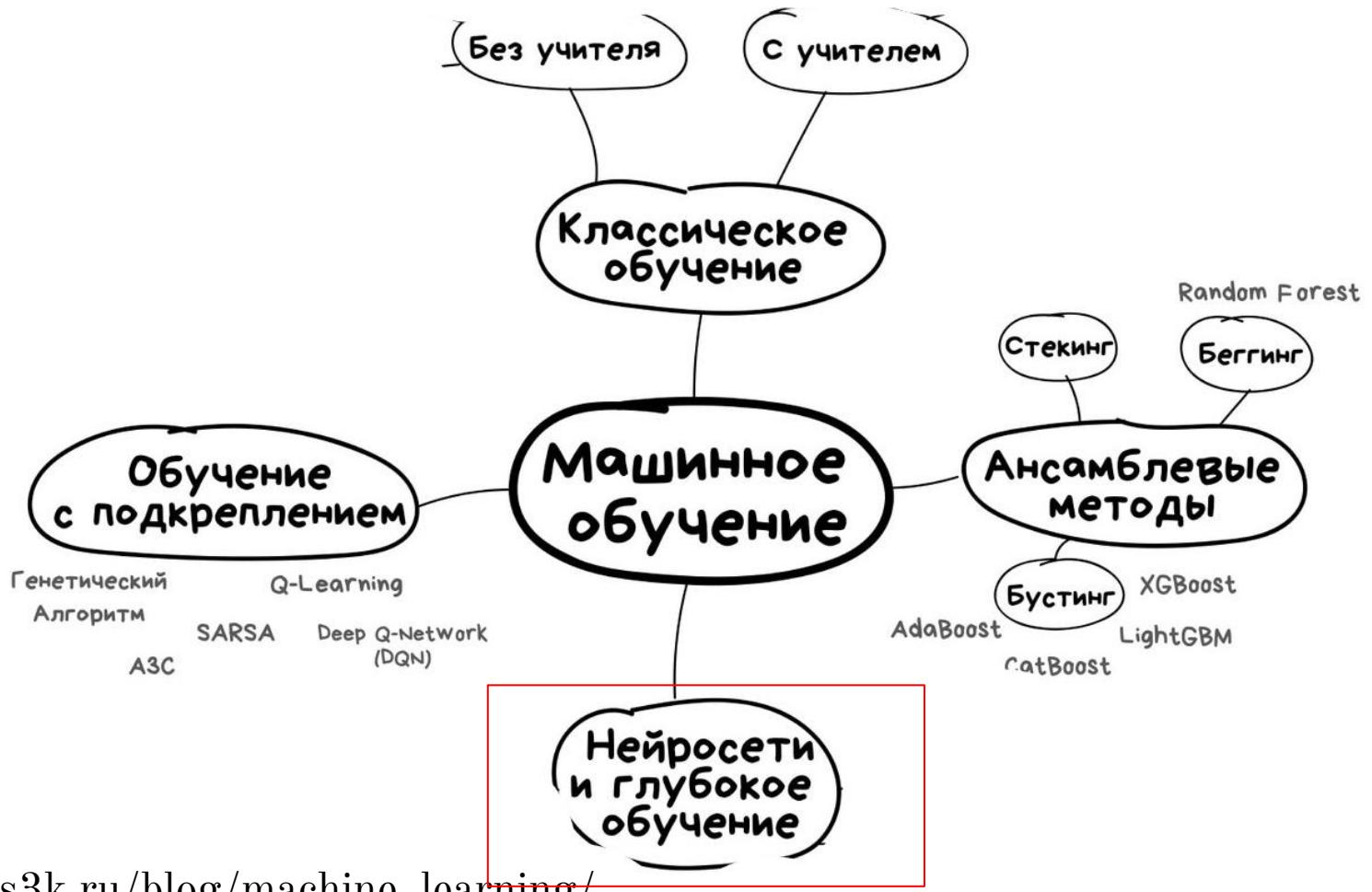




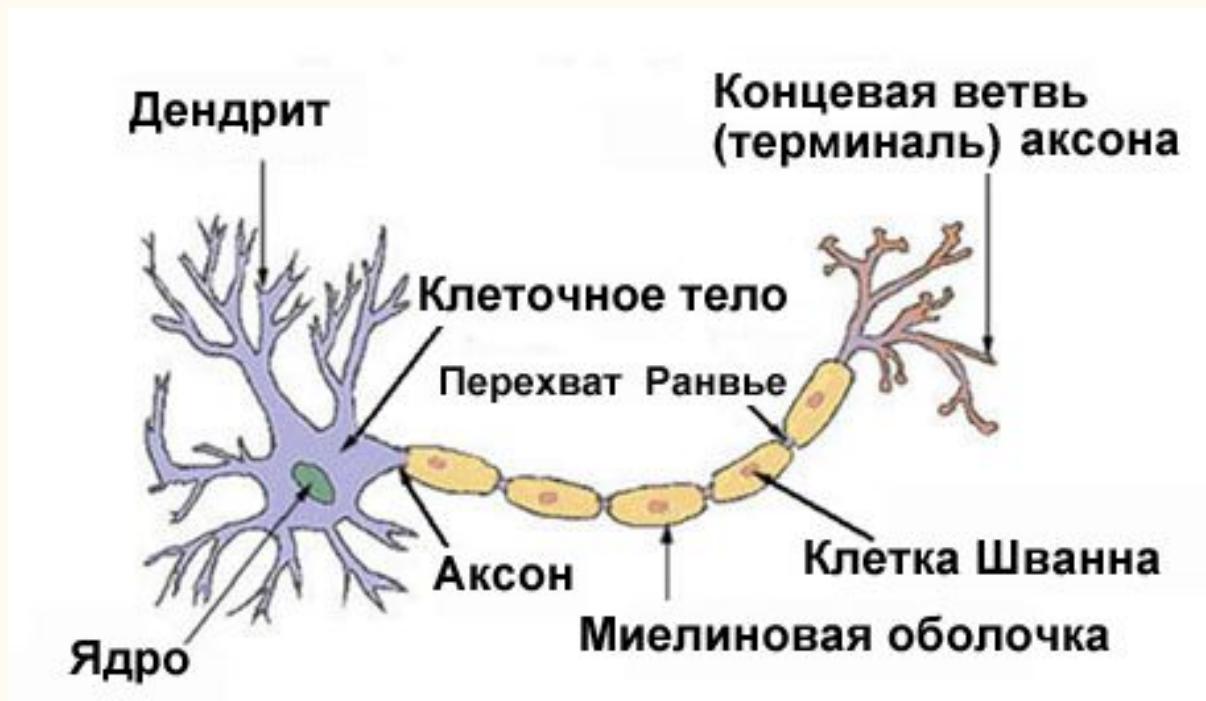




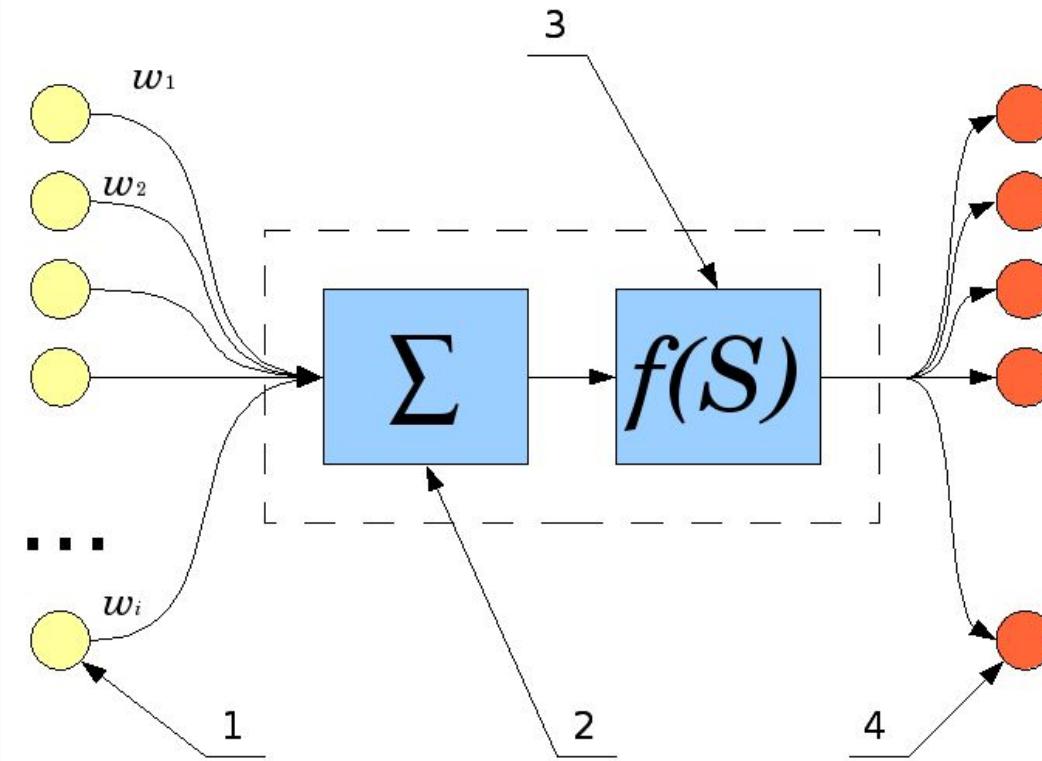


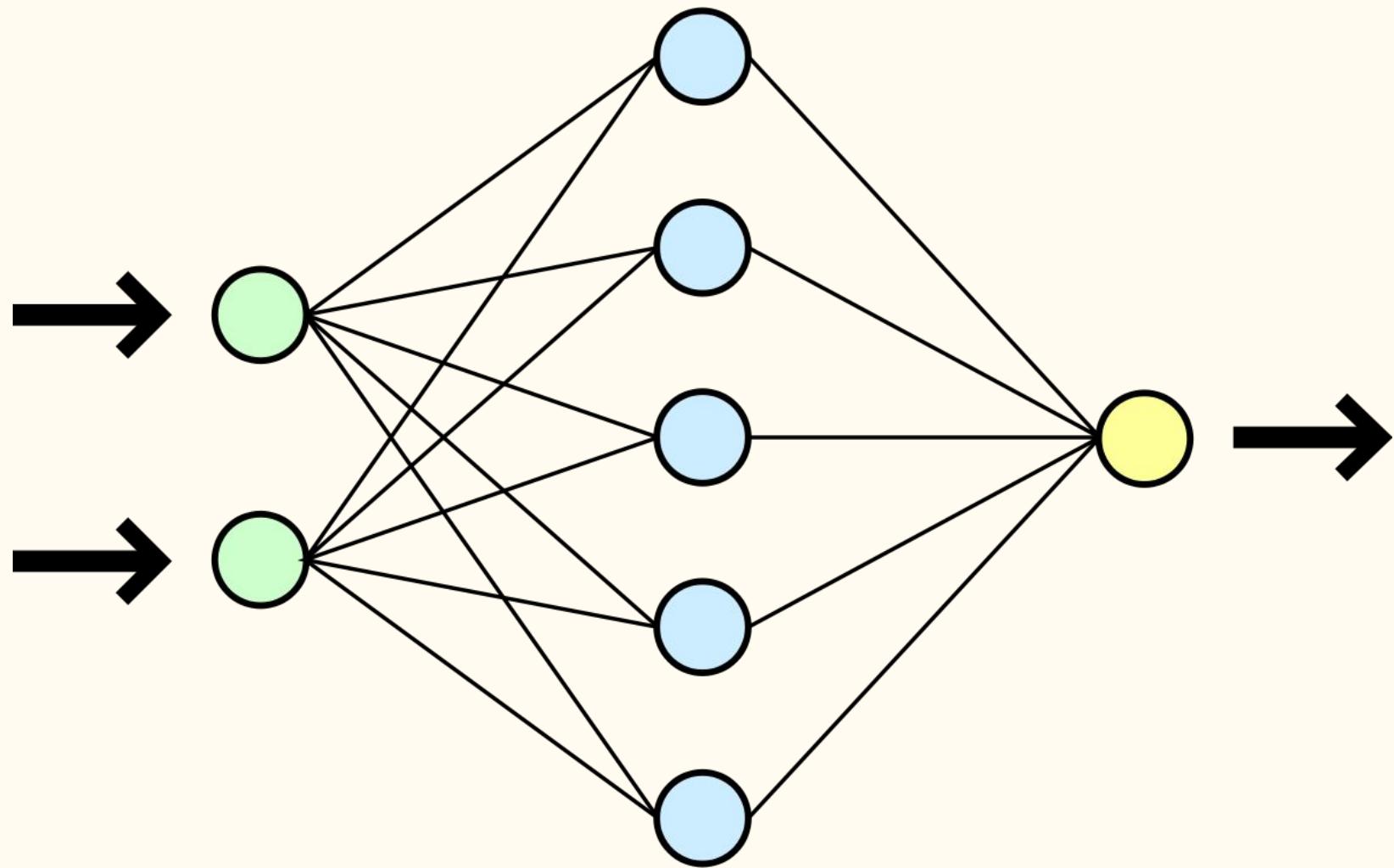


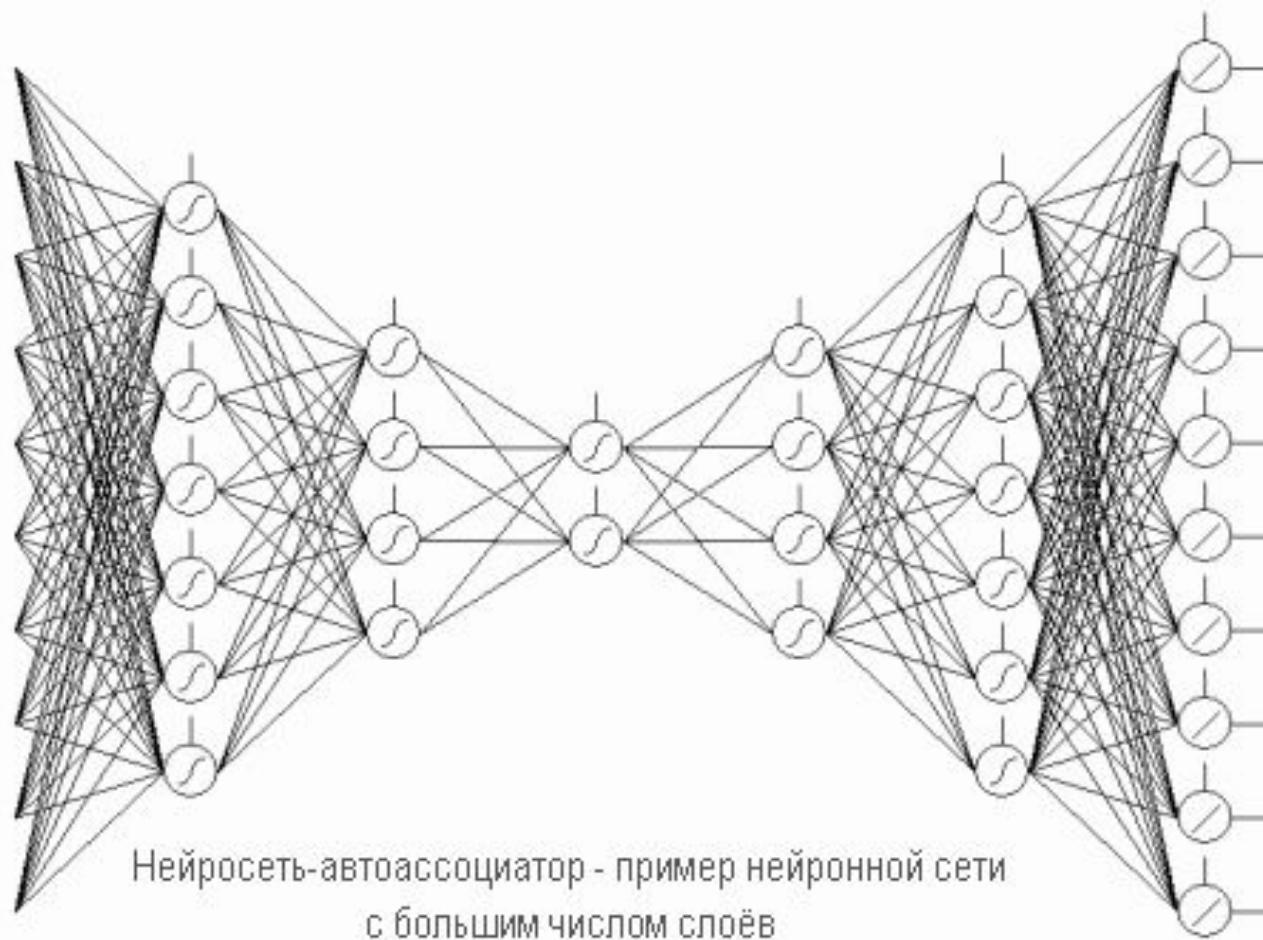
Биологический нейрон человека



Искусственный нейрон





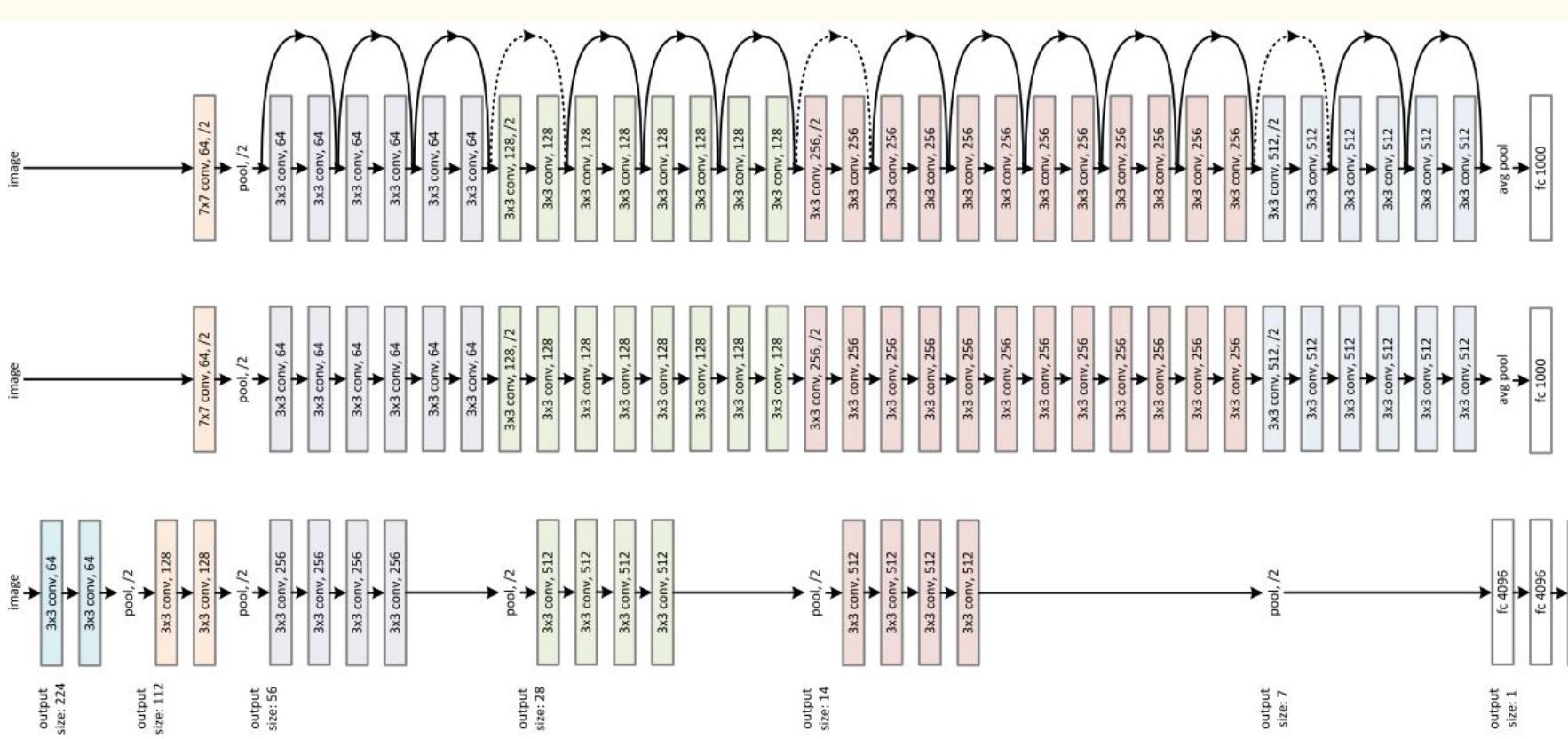


Нейросеть-автоассоциатор - пример нейронной сети
с большим числом слоёв

VGG-19

34-layer plain

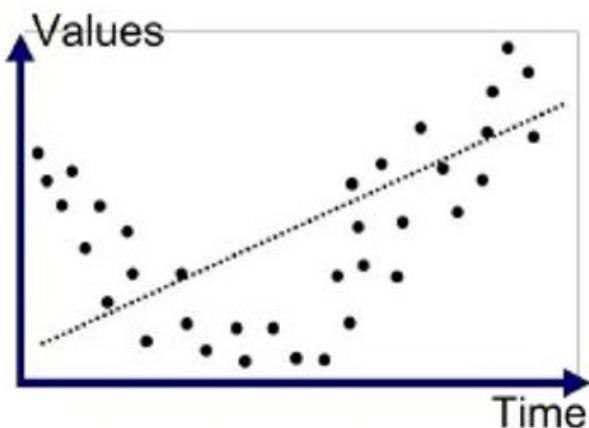
34-layer residual



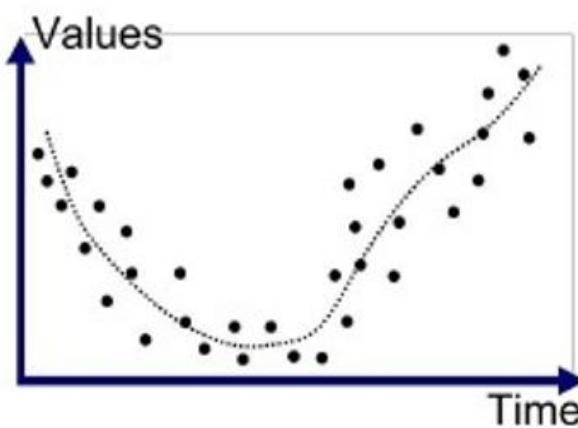
Problems



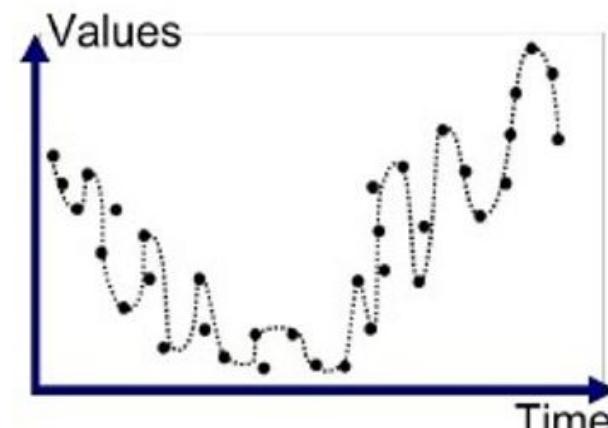
Overfitting/Underfitting



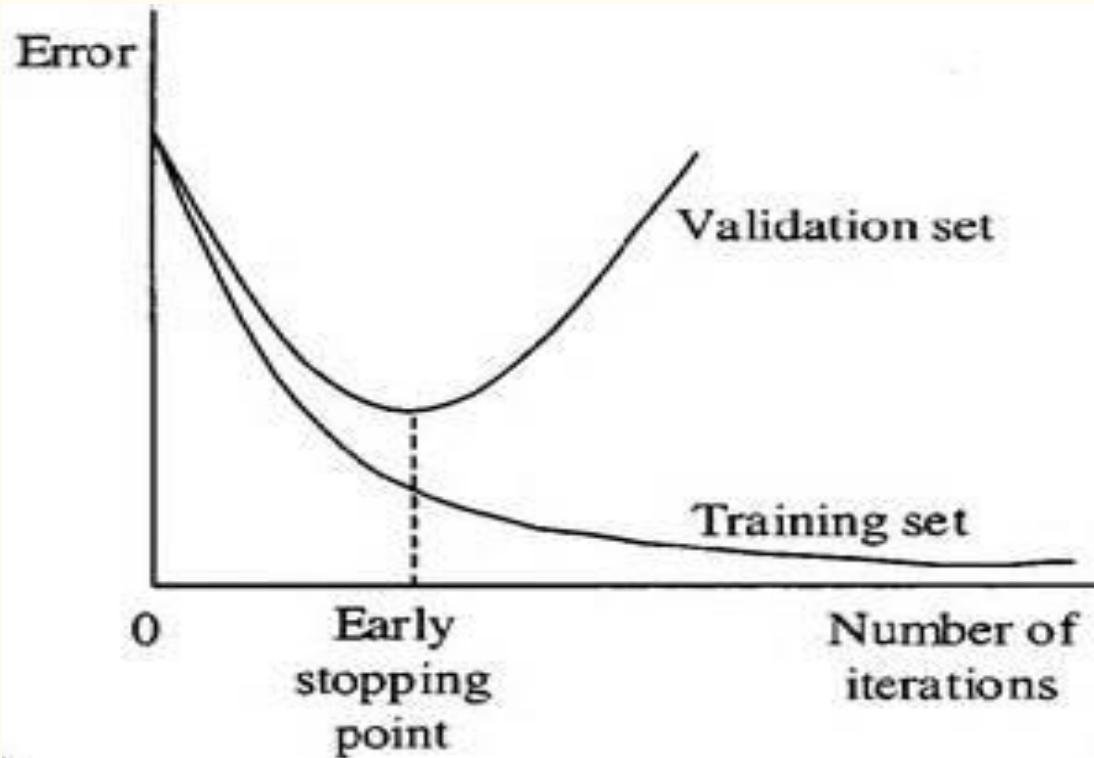
Underfitted



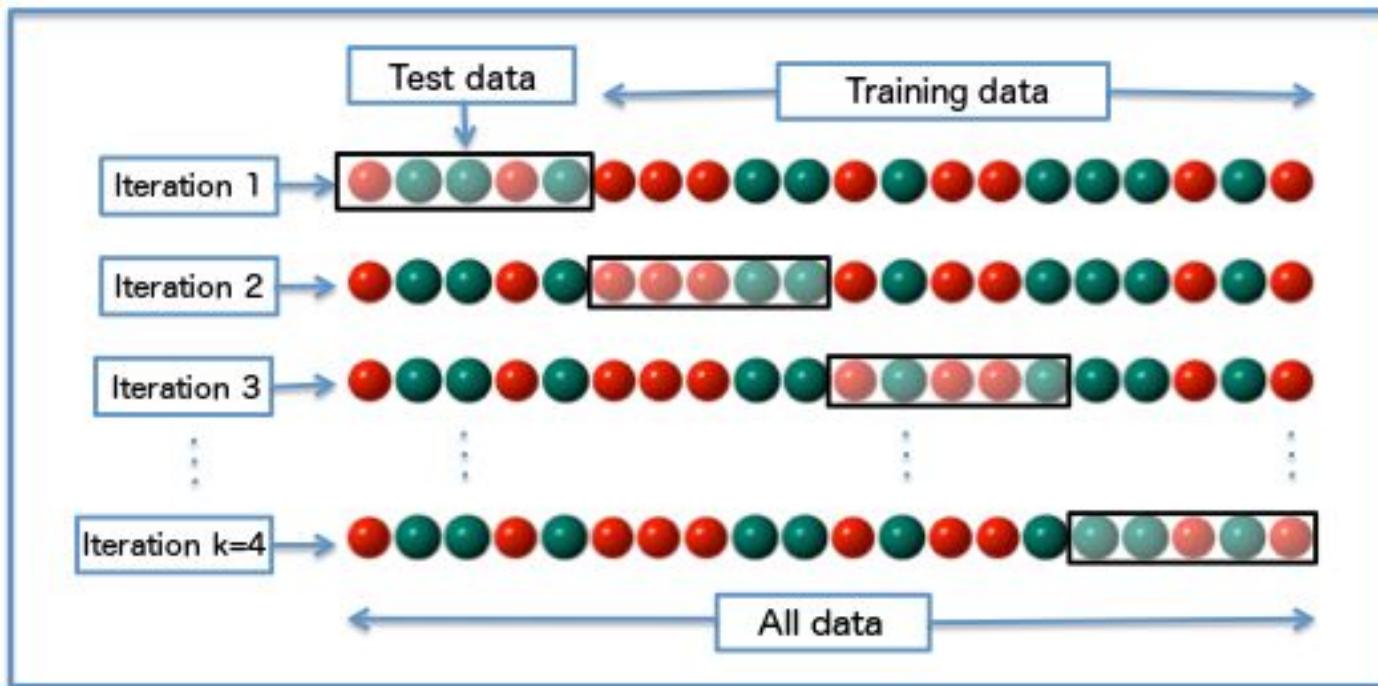
Good Fit/Robust



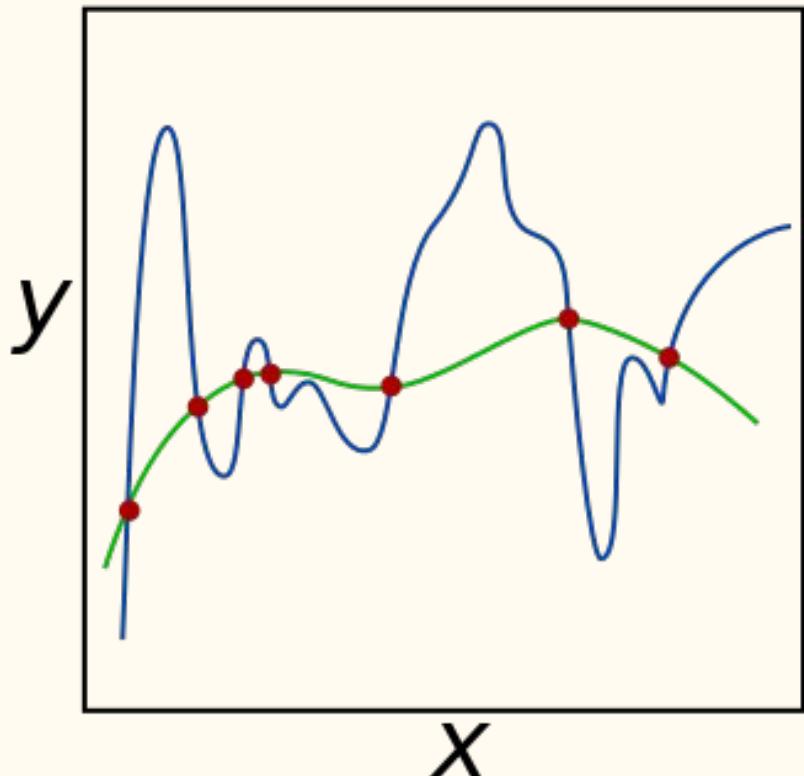
Overfitted



Cross Validation



Regularization



L1 Regularization

$$\text{Cost} = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2 + \lambda \sum_{j=0}^M |W_j|$$

L2 Regularization

$$\text{Cost} = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2 + \lambda \sum_{j=0}^M W_j^2$$

Loss function Regularization Term

Sparse Data

Item	Date	Qty	Supplier
Apples	2011-20-29	60	Figoni
Asparagus	2011-10-30	34	Giusti Farms
Bananas	\N	\N	\N
Cantelope	\N	\N	\N
Grapes	\N	\N	\N
Onions	2011-10-27	66	Pastorino
Oranges	\N	\N	\N
Peaches	\N	\N	\N
Pears	\N	\N	\N
Pineapples	\N	\N	\N
Plums	\N	\N	\N
Strawberries	\N	\N	\N
Yams	2011-11-03	52	Iacopi Farms

Метрические модели



Гипотеза компактности (для классификации):

Близкие объекты, как правило, лежат в одном классе.

Гипотеза непрерывности (для регрессии):

Близким объектам соответствуют близкие ответы.

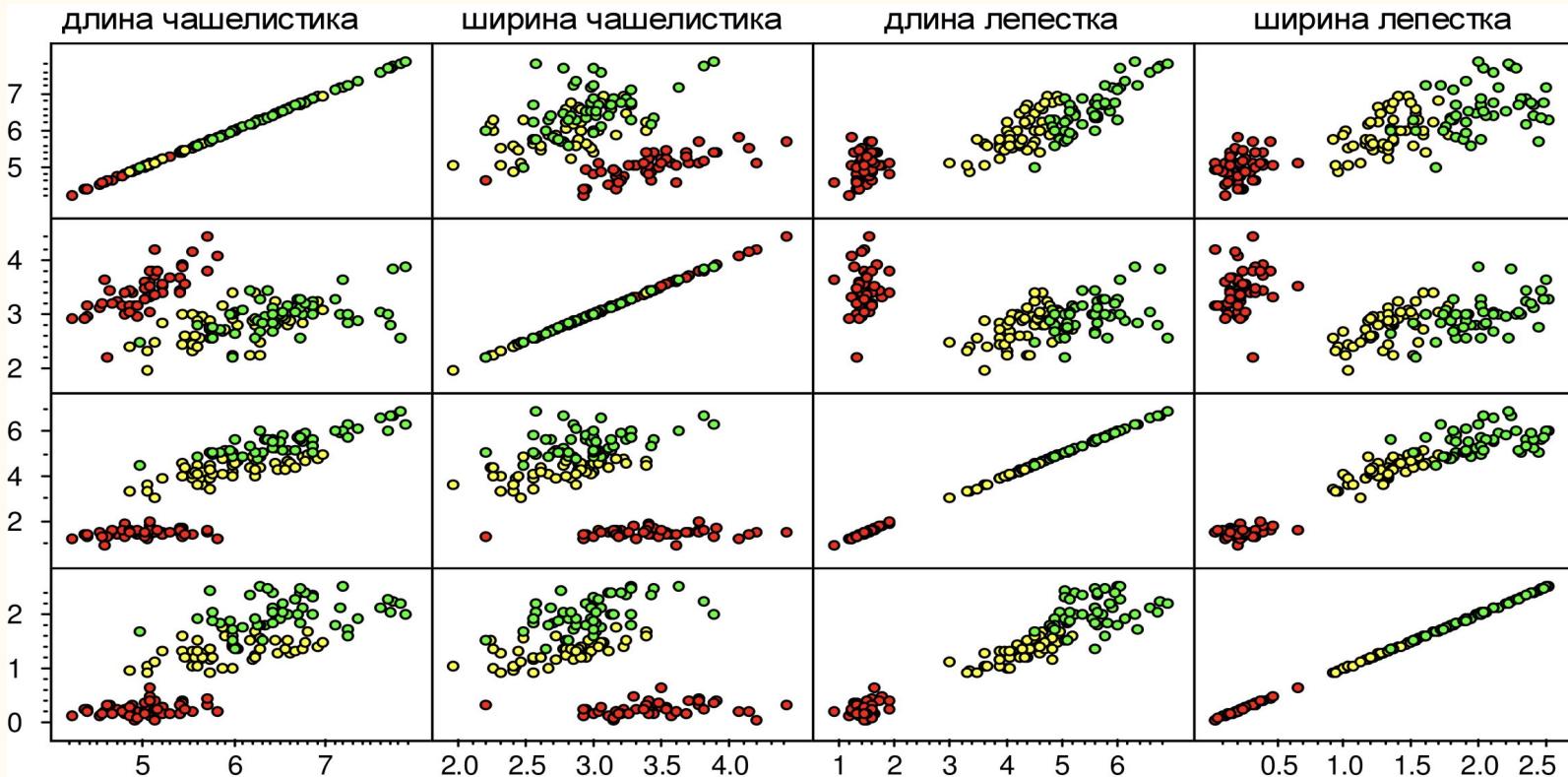
Формализация понятия «близости»:

Задана функция расстояния $\rho: X \times X \rightarrow [0, \infty)$.

Пример. Евклидово расстояние и его обобщение:

$$\rho(x, x_i) = \left(\sum_{j=1}^n |x^j - x_i^j|^2 \right)^{1/2} \quad \rho(x, x_i) = \left(\sum_{j=1}^n w_j |x^j - x_i^j|^p \right)^{1/p}$$

Пример: классификация цветков ириса ($|x|=4$ признака, $|y|=3$ класса)



Ирисы Фишера — это набор данных для задачи классификации, на примере которого Рональд Фишер в 1936 году продемонстрировал работу разработанного им метода дискриминантного анализа.



Ирис щетинистый
(*Iris setosa*)



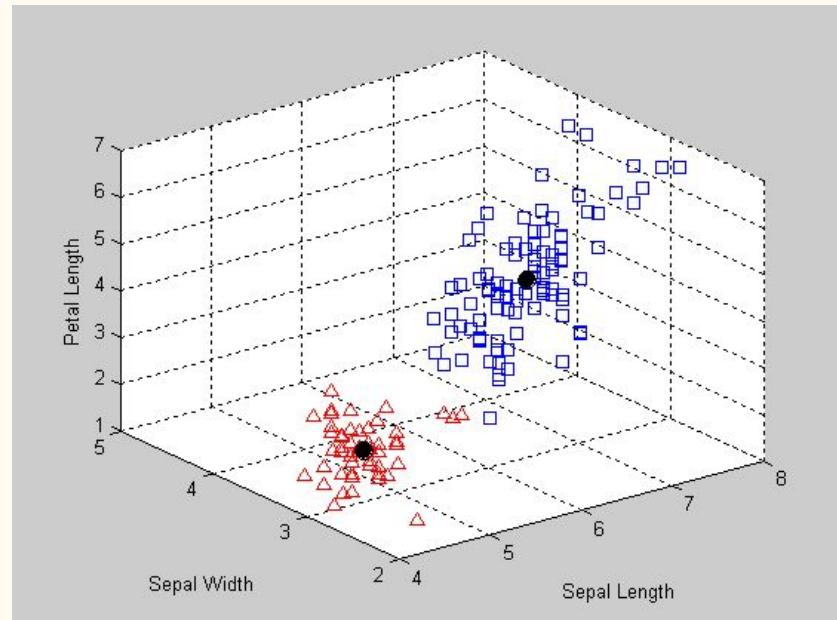
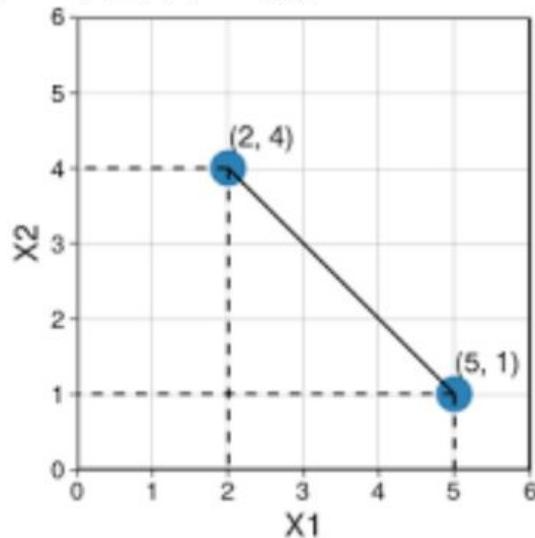
Ирис разноцветный
(*Iris versicolor*)



Ирис виргинский
(*Iris virginica*)

Пусть $\rho(x, y)$ – функция расстояния
Евклидово расстояние:

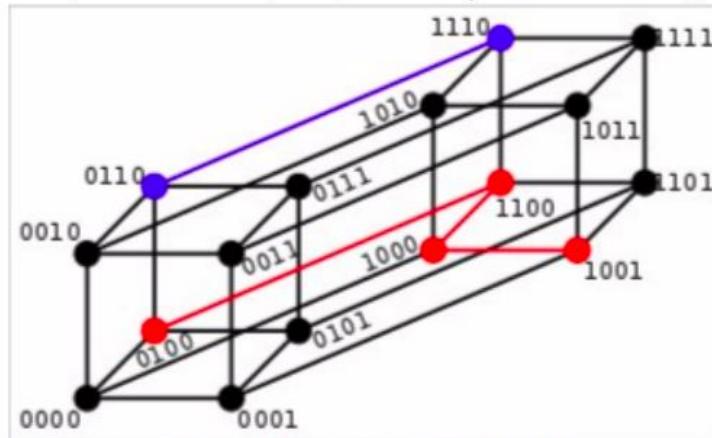
$$\rho(x, y) = \sqrt{\sum (x_i^2 - y_i^2)}$$



Пусть $\rho(x, y)$ – функция расстояния

Расстояние Хэмминга:

число позиций, в которых соответствующие символы двух слов одинаковой длины различны. Расстояние Хэмминга применяется для строк одинаковой длины любых q -ичных алфавитов и служит метрикой различия (функцией, определяющей расстояние в метрическом пространстве) объектов одинаковой размерности.



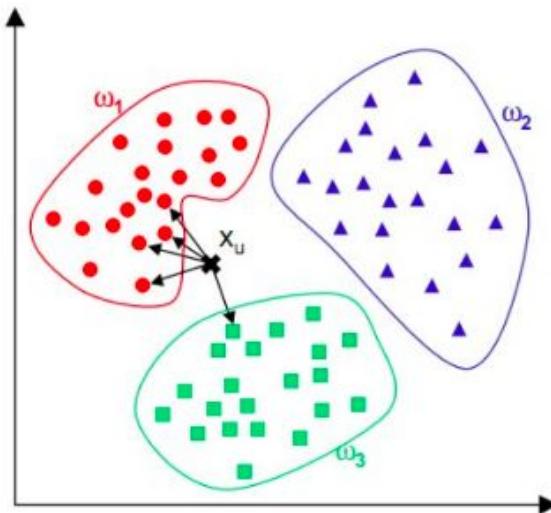
примеры расстояний в двоичном тессеракте:

расстояние 1 0110 → 1110, расстояние 3 0100 → 1001



Пример метрического классификатора:

Метод ближайших соседей — метрический классификатор, основанный на оценивании расстояний между объектами. Классифицируемый объект относится к тому классу, которому принадлежат ближайшие к нему объекты обучающей выборки.



Для классификации каждого из объектов тестовой выборки необходимо **последовательно выполнить следующие операции**:

- ➊ Вычислить расстояние до каждого из объектов обучающей выборки
- ➋ Отобрать k объектов обучающей выборки, расстояние до которых минимально
- ➌ Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей

Алгоритм можно выразить формулой:

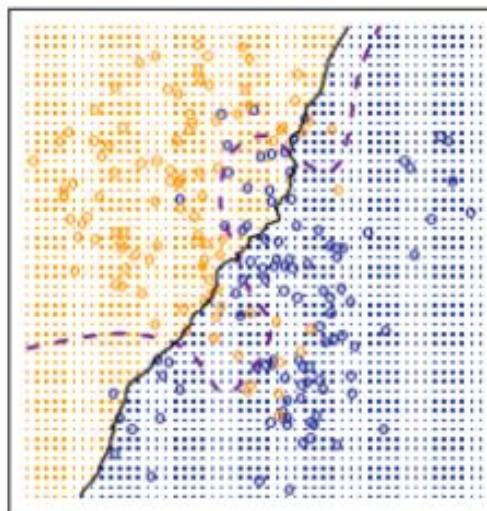
$a(x) = \operatorname{argmax}(\sum_{i=1}^k a_i \cdot [y_{(i)} == y]), y \in Y$ Подбирается с помощью **holdout-выборки** или **кросс-валидации**

Чем больше k, тем проще разделяющая поверхность

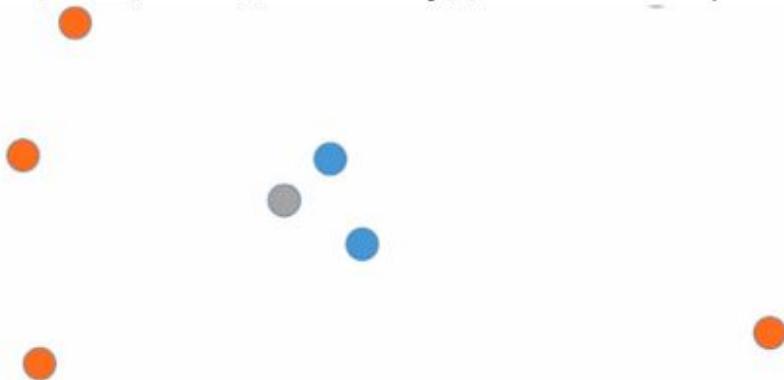
KNN: K=1



KNN: K=100

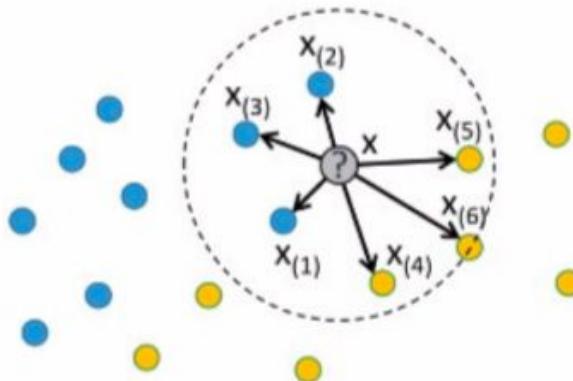


Пусть $k=5$, тогда как будет классифицирован объект?



Решение – учитывать расстояния среди k ближайших соседей: те объекты, которые расположены ближе, должны иметь больший вес.

Пример классификации ($k = 6$):



Tools



Jupyter notebook

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** The URL is 188.93.56.91:1114/notebooks/Style%20Transfer/demo.ipynb. The title is "jupyter demo" and it says "Last Checkpoint: 08/08/2018 (unsaved changes)".
- Toolbar:** Includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a "Connecting to kernel" button.
- Cell Buttons:** Includes standard Jupyter cell controls like Run, Cell, Kernel, and Help.
- Code Block:** The code cell contains the following Python imports:

```
In [15]: from __future__ import print_function
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import torchvision.transforms as transforms
import torchvision.models as models
import copy as copy
import scipy.io.wavfile

import librosa
import librosa.display
import matplotlib.pyplot as plt
import librosa.display
from PIL import Image
import soundfile as sf
```

Git (GitHub)

88 commits 1 branch 0 releases 10 contributors

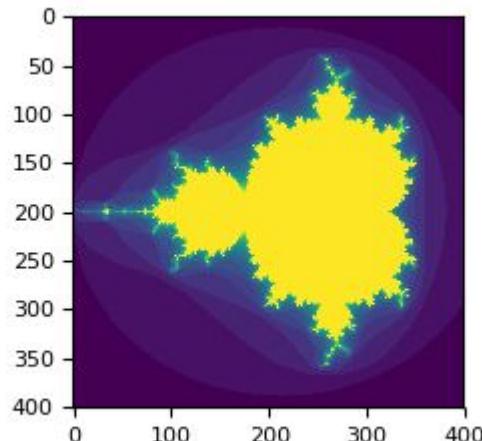
Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

This branch is even with junyanz:master. Pull request Compare

 junyanz	Merge pull request #87 from 1j01/patch-1 ...	Latest commit 5df5a13 on 23 May
data	add cache_dir flag	a year ago
datasets	update README	a year ago
examples	add maps training script to examples/	a year ago
imgs	add a failure case image	a year ago
models	rename identity -> lambda_identity	6 months ago
pretrained_models	Support for keeping the original aspect ratio in one_direction_test_m...	a year ago
util	Update cudnn_convert_custom.lua	5 months ago

NumPy

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> def mandelbrot( h,w, maxit=20 ):
...     """Returns an image of the Mandelbrot fractal of size (h,w)."""
...     y,x = np.ogrid[ -1.4:1.4:h*1j, -2:0.8:w*1j ]
...     c = x+y*1j
...     z = c
...     divtime = maxit + np.zeros(z.shape, dtype=int)
...
...     for i in range(maxit):
...         z = z**2 + c
...         diverge = z*np.conj(z) > 2**2           # who is diverging
...         div_now = diverge & (divtime==maxit)    # who is diverging now
...         divtime[div_now] = i                     # note when
...         z[diverge] = 2                          # avoid diverging too much
...
...     return divtime
>>> plt.imshow(mandelbrot(400,400))
>>> plt.show()
```



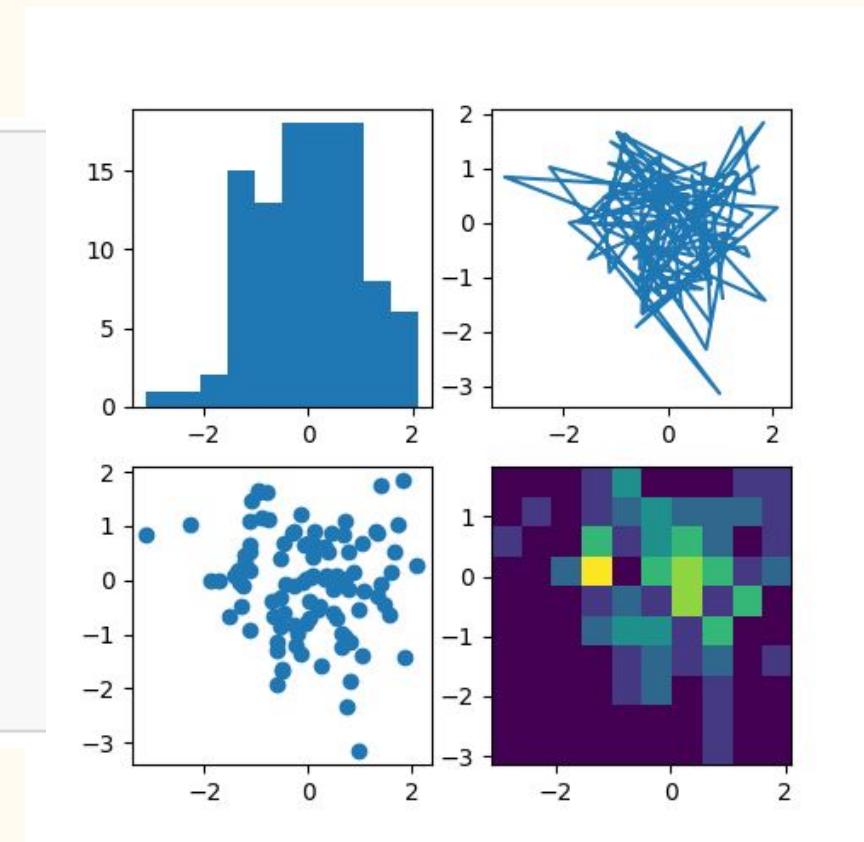
Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(19680801)
data = np.random.randn(2, 100)

fig, axs = plt.subplots(2, 2, figsize=(5, 5))
axs[0, 0].hist(data[0])
axs[1, 0].scatter(data[0], data[1])
axs[0, 1].plot(data[0], data[1])
axs[1, 1].hist2d(data[0], data[1])

plt.show()
```



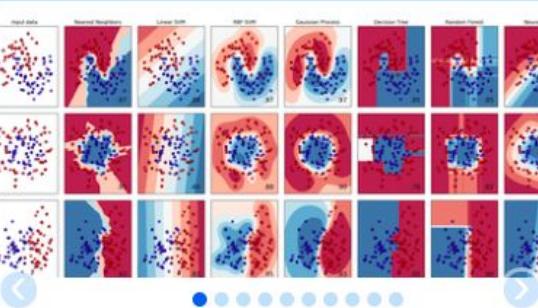
Pandas

```
In [404]: train_data = pd.read_csv("train_med.csv", delimiter=';')
train_data.head()
```

Out[404]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

Scikit-learn



The image shows a 3x9 grid of small plots demonstrating various machine learning models. Each plot displays a 2D dataset with two classes (blue and red) and the decision boundary or regions learned by different models. The models shown include K-Nearest Neighbors, Linear SVM, Logistic Regression, RBF SVM, Gaussian Process, Decision Tree, Random Forest, and Naive Bayes.

scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ...

— Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso,

...

— Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ...

— Examples

Dimensionality reduction

Model selection

Preprocessing

Конец!

Ссылка, которые вам сейчас нужна:

GitHub repo: <https://github.com/Atmyre/Sberbank-ML>