

Лекция 1.

Принятие решений на основе данных.  
Основы машинного обучения

---

Юрий Яровиков

# Содержание лекции

- Как машинное обучение меняет мир
- Постановка задачи машинного обучения
- Примеры
- Метрики и функции потерь
- Фреймворк машинного обучения
- Методология CRISP-DM

# Как машинное обучение меняет мир

*“Nations with the strongest presence in AI R&D will establish leading positions in the automatization of the future.”*

— отчёт Белого дома США, октябрь 2016

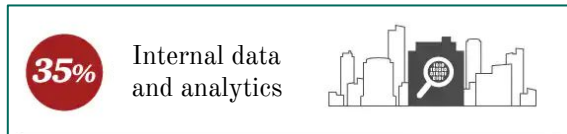
- цифровая и распределённая экономика
- автоматизация и сокращение издержек
- автономный транспорт и роботизация
- автоматизация банковских услуг
- персонализированная медицина
- и многое другое

# Data-driven decision making

*“Highly data-driven organizations are three times more likely to report significant improvement in decision making. But only one in three executives say their organization is highly data-driven.”*

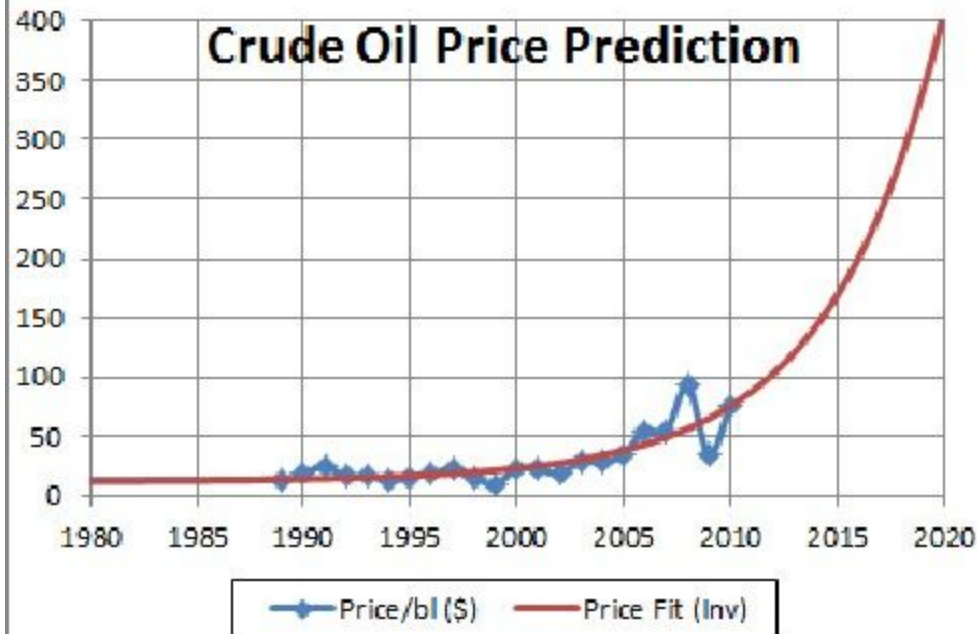
— Dan DiFilippo, Chief Analyst at PwC

What will you rely on most when making your next strategic decision?  
Global base: 2,106 senior executives



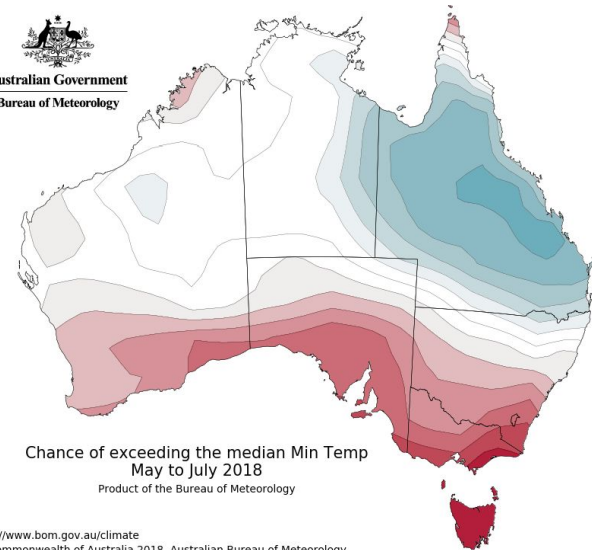
ИСТОЧНИК: исследование PwC

## Crude Oil Price Prediction



Good Credit Score

Australian Government  
Bureau of Meteorology



<http://www.bom.gov.au/climate>  
© Commonwealth of Australia 2018, Australian Bureau of Meteorology

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

## Специально для Вас

Аффинаж, gsac, номер скрыт и другие

▶ СЛУШАТЬ ВСЕ



## Приглядитесь к этим предложениям



3 295 Р -50 %  
6 590 Р  
Кеды VANS



7 030 Р -35 %  
10 800 Р  
Внешняя звуковая



1 875 000 Р  
Виниловый  
проигрыватель Spira...



4 400 Р -30 %  
6 290 Р  
Кеды VANS



11 790 Р  
Лонгборд GoldCoast  
Standard



Сначала интересные



# Google

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

# беру! BETA

кто лучше сири или ты

Я сейчас чуть не удалилась в сердцах. Конечно, я не Сири.

переведи меня зовут алиса на английский

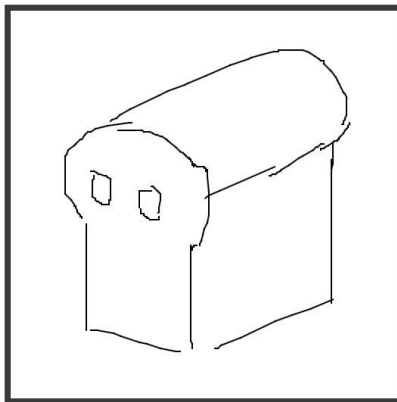
Лет ми спик фром май харт:  
пойдёмте в Яндекс.Переводчик.  
Там переводить проще.

[ОТКРЫТЬ ЯНДЕКС.ПЕРЕВОДЧИК](#)

[ПОИСКАТЬ В ЯНДЕКСЕ](#)



INPUT



pix2pix

process

OUTPUT



# Пример: оцените стоимость ноутбука

	Кол-во ядер	RAM (Гб)	Объем жесткого диска (ГБ)	Диагональ/ разрешение	Работа от аккумулятора	Цена (руб.)
1 	2	4	500 (HDD)	15"/1920x1080 пикс.	до 5 часов	31 490
2 	4	8	256 (SSD)	14"/1920x1080 пикс.	до 12 часов	60 990
3 	4	16	1000 (HDD)	17"/1920x1080 пикс.	до 3 часов	65 990
4 	8	16	1000 (HDD) + 256 (SSD)	17"/1920x1080 пикс.	до 11 часов	109 990
5 	4	16	1000 (HDD)+ 128 (SSD)	17"/1920x1080 пикс.	до 6 часов	?



# Пример: оцените стоимость ноутбука

	Кол-во ядер	RAM (Гб)	Объем жесткого диска (ГБ)	Диагональ/ разрешение	Работа от аккумулятора	Цена (руб.)
1 	2	4	500 (HDD)	15"/1920x1080 пикс.	до 5 часов	31 490
2 	4	8	256 (SSD)	14"/1920x1080 пикс.	до 12 часов	60 990
3 	4	16	1000 (HDD)	17"/1920x1080 пикс.	до 3 часов	65 990
4 	8	16	1000 (HDD) + 256 (SSD)	17"/1920x1080 пикс.	до 11 часов	109 990
5 	4	16	1000 (HDD)+ 128 (SSD)	17"/1920x1080 пикс.	до 6 часов	86990

# Постановка задачи машинного обучения

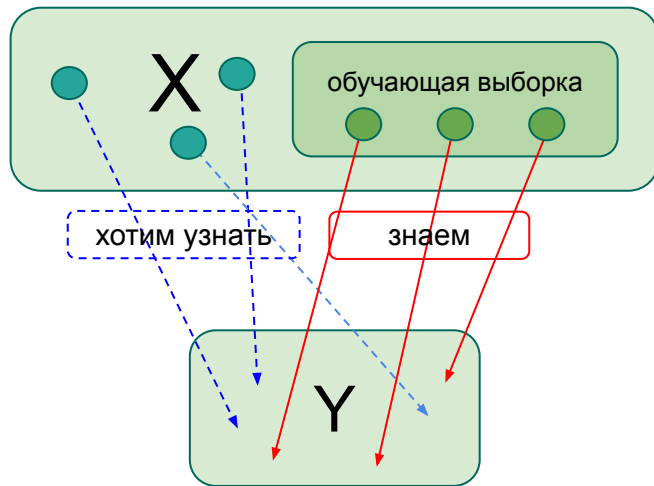
$X$  — множество *объектов*

$Y$  — множество *ответов* (например, два класса или произвольные числа)

$y: X \rightarrow Y$  — неизвестная закономерность

**Дано:** обучающая выборка,  $\{x_1, x_2, \dots, x_n\}$  — подмножество множества  $X$

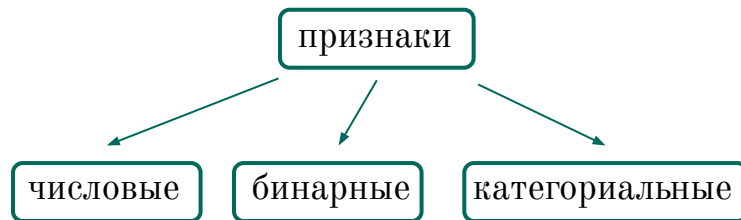
**Цель:** подобрать *алгоритм*  $a: X \rightarrow Y$ , приближающий функцию  $y(x)$  на всём  $X$



# Как задаются объекты. Признаковое описание

Объект  $x$  задаётся *признаковым описанием*

$f_1, f_2, \dots, f_k$  — признаки (features) объекта  $x$



$$\begin{bmatrix} f_1(x_1), & f_2(x_1), & \dots, & f_k(x_1) \\ f_1(x_2), & f_2(x_2), & \dots, & f_k(x_2) \\ & & \dots & \\ f_1(x_n), & f_2(x_n), & \dots, & f_k(x_n) \end{bmatrix}$$

— матрица “объекты-признаки”  
объект, пригодный для применения  
алгоритмов машинного обучения

# Задачи машинного обучения

- Обучение с учителем

- Задача классификации

Предсказываем класс объекта:  $Y = \{\text{class}_1, \text{class}_2, \dots, \text{class}_m\}$

- Задача восстановления регрессии

Предсказываем число:  $Y$  — множество действительных чисел

- Задача ранжирования

Предсказываем, какие объекты наиболее релевантны запросу

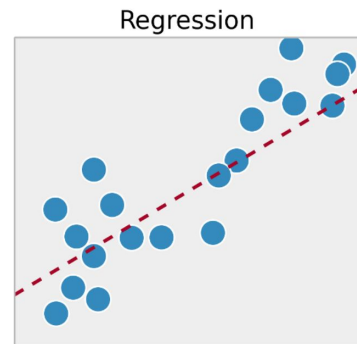
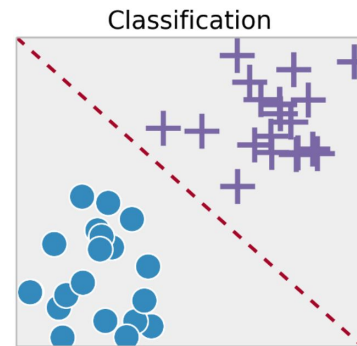
- Обучение без учителя

- Задача кластеризации

Разбиваем объекты на группы похожих

- Задача понижения размерности

“сжимаем” данные, пытаясь потерять как можно меньше информации



# Пример: медицинская диагностика

**Объект** — пациент в определённый момент времени

**Классы** — болен ли человек данной болезнью

**Примеры признаков:**

- бинарные: пол, головная боль, слабость, тошнота
- количественные: возраст, пульс, давление, результаты анализов, геном

**Особенности задачи:**

- пропуски в данных
- нужен интерпретируемый алгоритм
- нужна оценка вероятности исхода

# Пример: открытие нового ресторана

**Объект** — место для открытия нового ресторана

**Цель предсказания** — ожидаемая прибыль через год

**Примеры признаков:**

- количественные: демография, цены на недвижимость в округе, удалённость существующих ресторанов
- категориальные: характеристики здания, окружающей местности, наличие

**Особенности задачи:**

- мало объектов, много признаков
- разнотипные признаки
- разнородные объекты, есть выбросы

# Пример: Avito Context Ad Clicking Prediction

**Объект** — тройка  $\langle \text{пользователь}, \text{запрос}, \text{баннер} \rangle$

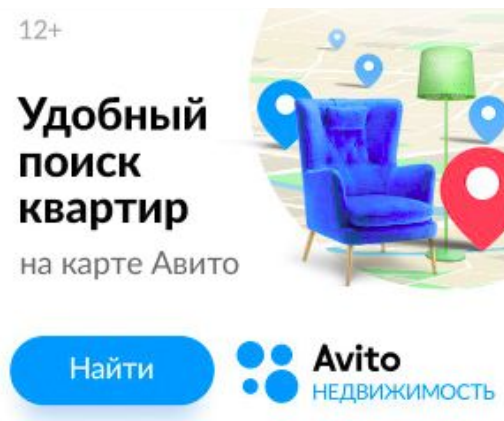
**Цель предсказания** — кликнет ли пользователь по контекстной рекламе

**Примеры признаков:**

- действия пользователей на сайте
- профиль пользователя
- история показов и кликов другим пользователям

**Особенности задачи:**

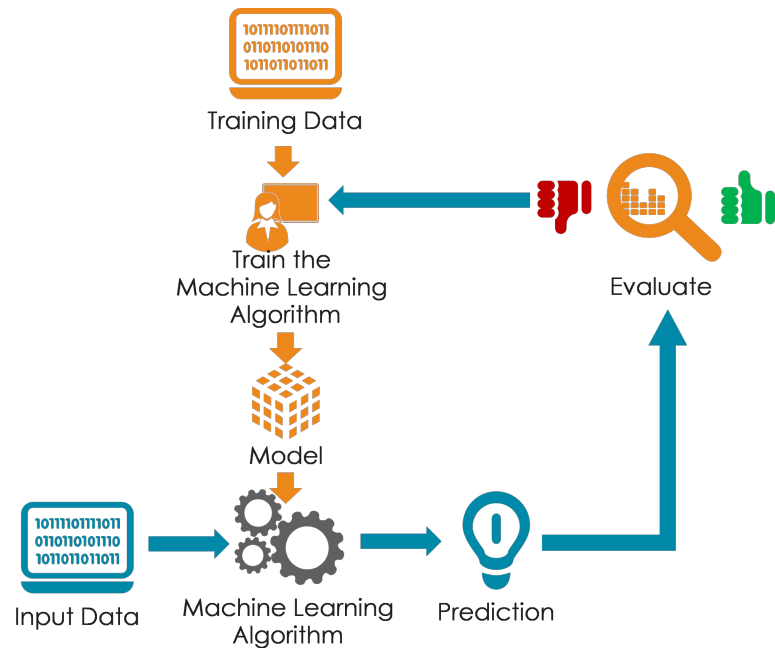
- признаки надо придумывать
- данных много
- главный критерий качества — доход рекламной площадки



# Фреймворк машинного обучения

- Формируем матрицу “объекты-признаки” по размеченным данным
- Разбиваем данные на train и test
- Настраиваем алгоритм  $a : X \rightarrow Y$  так, чтобы  $a$  приближал  $y$  на train
- Тестируем, насколько хорошо  $a$  приближает  $y$  на test

Что значит “хорошо приближает”?



фреймворк машинного обучения: [medium.com](https://medium.com)



# Метрики и функции потерь

Метрика — функционал, оценивающий качество предсказания

- основная оценка качества модели
- легко интерпретируемая
- примеры: accuracy, precision, recall, F1, ROC-AUC, MAE,  $R^2$

Функция потерь — функционал, который оптимизируется в процессе обучения

- легко оптимизируемая
- не обязательно интерпретируемая
- нужна для процесса обучения
- примеры: logloss, MSE

# Метрики в задаче классификации

- accuracy (доля верно угаданных ответов)

Метрика не подходит для случая, когда классы разбалансированы. Алгоритму выгодно относить все объекты наиболее частотного класса

- precision (точность, доля верно идентифицированных объектов первого класса к общему количеству идентифицированных объектов)

Алгоритму не выгодно относить объекты класса 0 к классу 1

- recall (полнота, доля верно идентифицированных объектов первого класса к общему количеству объектов первого класса)

Алгоритму не выгодно оставлять объекты класса 1 в классе 0

- Другие метрики: ROC-AUC, F-мера, logloss

# Confusion matrix

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	<b>TP</b> True Positive	<b>FP</b> False Positive
	negatives	<b>FN</b> False Negative	<b>TN</b> True Negative

# Accuracy

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

- легко интерпретируема
- не подходит для несбалансированных классов

# Пример: кредитный скоринг

- В банк пришло 90 надёжных и 10 ненадёжных заёмщиков
- Выдача кредита 10 ненадёжным заёмщикам принесёт большие убытки
- Алгоритм, который советует выдать кредит каждому заёмщику, будет иметь accuracy 90%!

# Precision, Recall

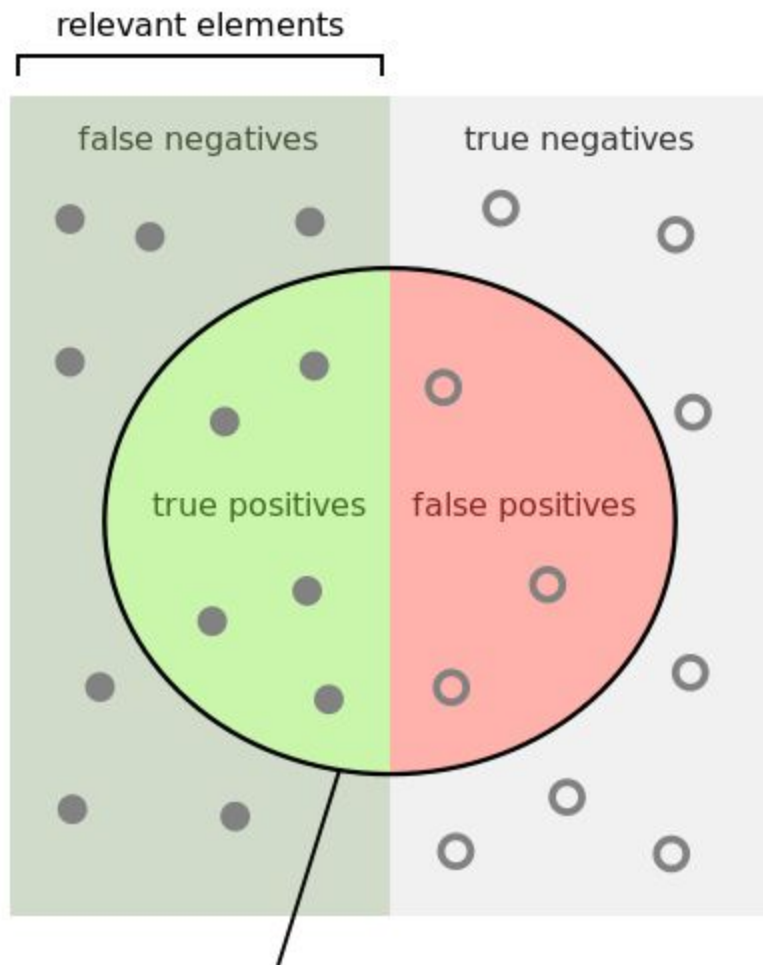
Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики *precision* (точность) и *recall* (полнота)

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

Алгоритму не выгодно относить объекты класса 0 к классу 1

$$recall = \frac{TP}{TP + FN}$$

Алгоритму не выгодно относить объекты класса 1 к классу 1



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

# F-мера, ROC-AUC

- F-мера — среднее гармоническое precision и recall
- ROC-AUC — используется при предсказании вероятностей. Является вероятностью того, что два случайно выбранных объекта разных классов будут отранжированы в нужном порядке
- Обе метрики учитывают возможный дисбаланс классов



# Метрики в задаче регрессии

- MSE (Mean Square Error): 
$$\text{MSE}(y_{\text{true}}, y_{\text{predicted}}) = \overline{(y_{\text{true}} - y_{\text{predicted}})^2}$$

Легко оптимизировать, сложнее интерпретировать
- MAE (Mean Absolute Error): 
$$\text{MAE}(y_{\text{true}}, y_{\text{predicted}}) = \overline{|y_{\text{true}} - y_{\text{predicted}}|}$$

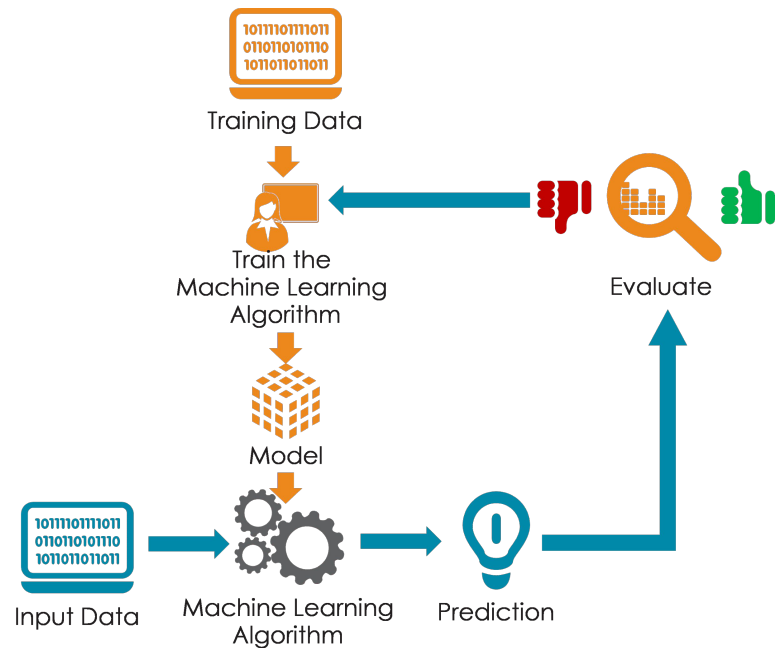
Легко интерпретировать, но сложно оптимизировать
- $R^2$  — коэффициент детерминации: 
$$R^2(y_{\text{true}}, y_{\text{predicted}}) = 1 - \text{MSE} / D y_{\text{true}}$$

У хороших моделей близок к 1, у не очень хороших близок к 0. Если  $R^2 < 0$ , то модель “очень плохая”.
- Несимметричные функции потерь  
За “недостачу” штрафуют сильнее, чем за “избыток”

# Фреймворк машинного обучения

- Формируем матрицу “объекты-признаки” по размеченным данным
- Разбиваем данные на train и test
- Настраиваем алгоритм  $a : X \rightarrow Y$  так, чтобы  $a$  приближал  $y$  на train
- Тестируем, насколько хорошо  $a$  приближает  $y$  на test

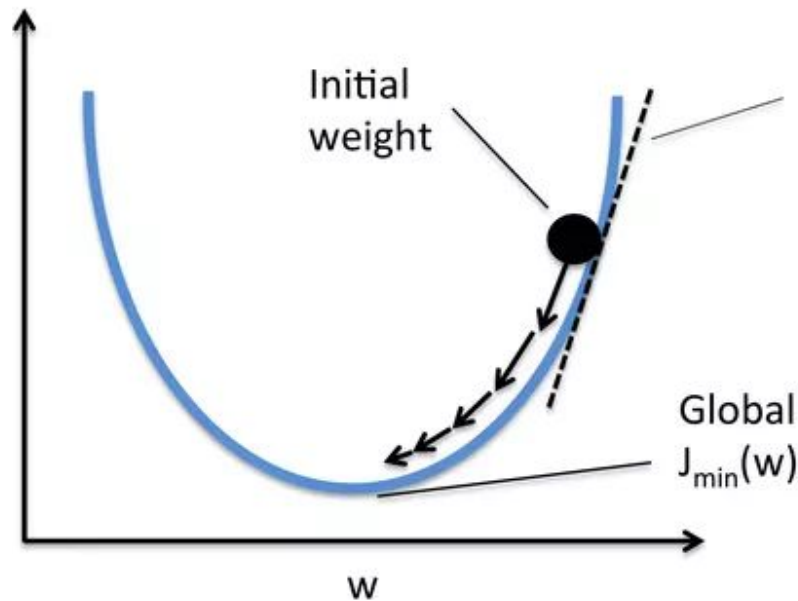
Как настраивать алгоритм?



фреймворк машинного обучения: [medium.com](https://medium.com)

# Оптимизация для настройки алгоритмов

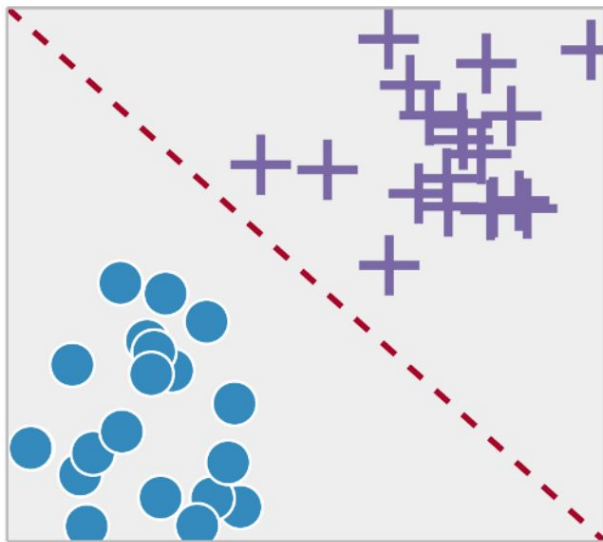
- Ищем алгоритм  $a$  в семействе алгоритмов, параметризуемых вектором весов  $w$
- Находим вектор  $w$ , минимизирующий заданную функцию потерь
- Поиск осуществляется градиентным спуском или любым другим алгоритмом оптимизации



# Линейные алгоритмы

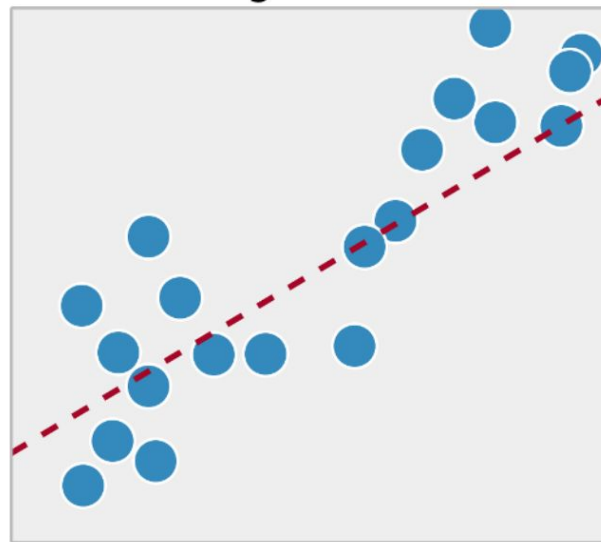
$$y(x) = \text{sign}(\langle w, x \rangle + b)$$

Classification



$$y(x) = \langle w, x \rangle + b$$

Regression



# Пример: линейная регрессия

- Ищем алгоритм  $a$  в классе линейных алгоритмов:

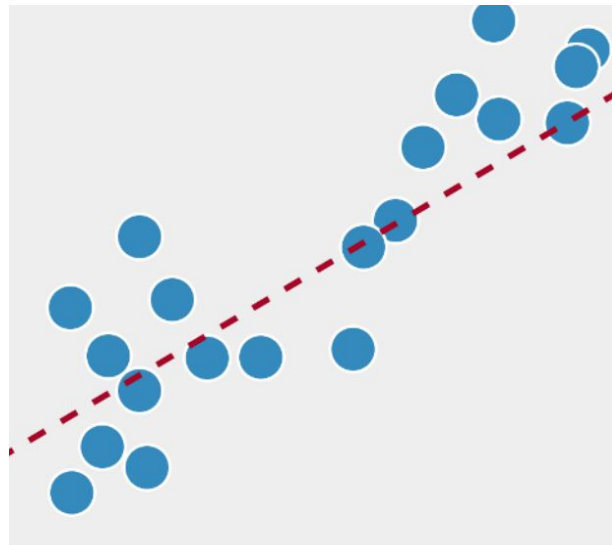
$$y_{\text{predicted}} = a(x) = \langle w, x \rangle - w_0$$

- Настраиваем веса  $w, w_0$  так, чтобы минимизировать MSE:

$$(y_{\text{predicted}} - y_{\text{true}})^2 \rightarrow \min$$

- Итоговая задача оптимизации:

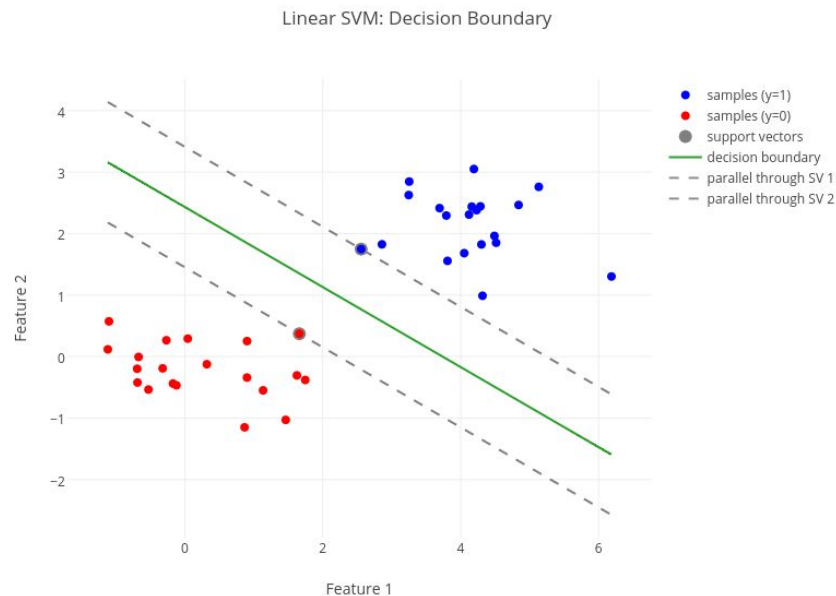
$$(\langle w, x \rangle - w_0 - y_{\text{true}})^2 \rightarrow \min$$



# Пример: SVM — Support Vector Machine

- Ищем полосу максимальной длины, разделяющую объекты обучающей выборки
- Задача эквивалентна задаче оптимизации  $\|w\| \rightarrow \max$  при условии, что точки разделяются прямой  $y = wx - w_0$  на два класса
- Эквивалентная задача оптимизации:

$$c \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}, \quad \text{где} \quad M_i(w, w_0) = y_i (\langle w, x_i \rangle - w_0)$$



# Пример: логистическая регрессия

- Задача классификации, а не регрессии
- Предсказываем вероятности попадания в класс 1 против класса -1
- Минимизируем функцию потерь

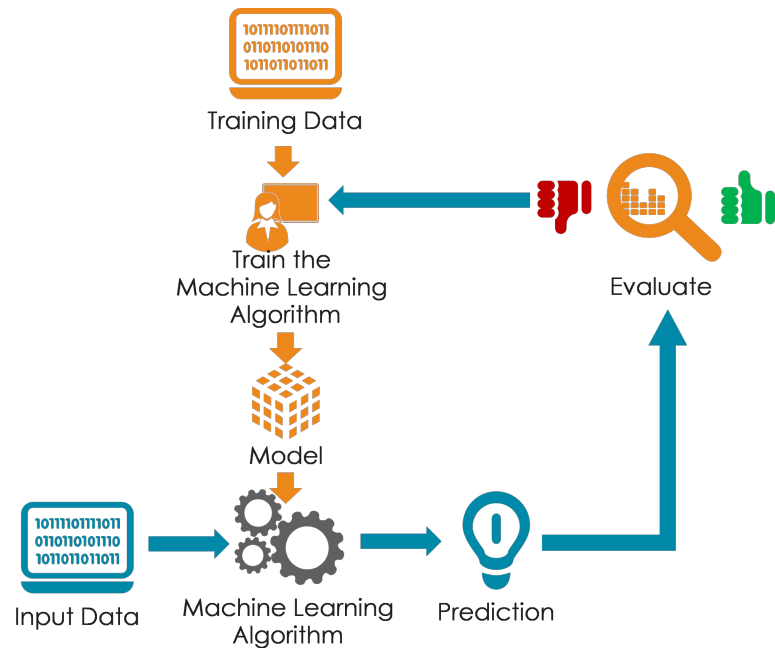
$$L(x, y_{true}) = \sum_{i=1}^n \ln(1 + e^{-y_{true} \langle w, x \rangle}) \rightarrow \min_w$$

- Величина  $\frac{1}{1 + e^{-\langle w, x \rangle}}$  интерпретируется как вероятность попадания объекта  $x$  в класс 1
- Классификация проводится по формуле  $y(x) = \text{sign}(\langle w, x \rangle)$

# Фреймворк машинного обучения

- Формируем матрицу “объекты-признаки” по размеченным данным
- Разбиваем данные на train и test
- Настраиваем алгоритм  $a : X \rightarrow Y$  так, чтобы  $a$  приближал  $y$  на train
- Тестируем, насколько хорошо  $a$  приближает  $y$  на test

Почему train и test необходимо разделять?

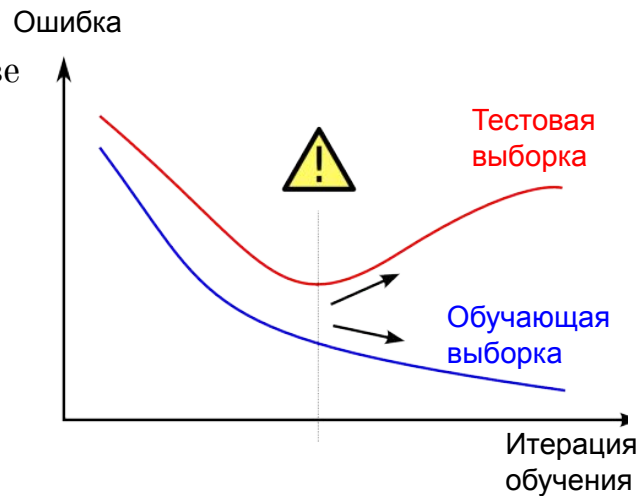


фреймворк машинного обучения: [medium.com](https://medium.com)

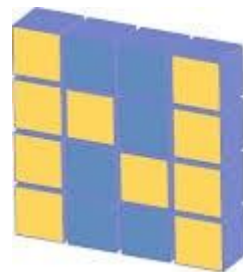


# Переобучение

- Из-за чего возникает переобучение?
  - Переобучение есть всегда, когда выбор делается на основе заведомо неполной информации
  - Слишком сложная/гибкая модель может чрезмерно подстроиться под обучающую выборку и потерять способность находить нижележащие закономерности в новых данных
- Как обнаружить переобучение?
  - Разбивать данные на обучающую и тестовую выборки



# Практика!



NumPy

