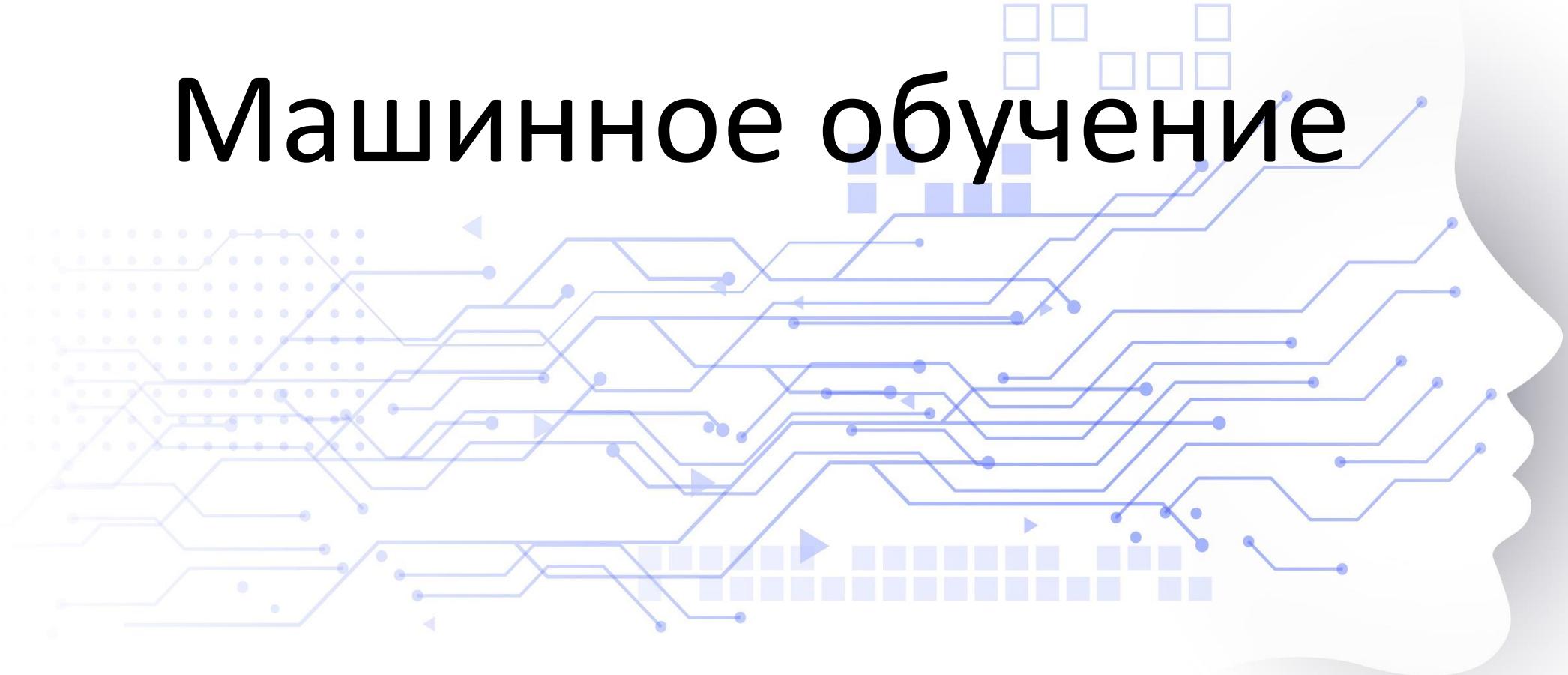


Машинное обучение





Резаиан Наим

E-mail: rezaian-n@rudn.ru

Telegram: [@NaeimRezaeian](https://t.me/NaeimRezaeian)

1. Заведующий лабораторией искусственного интеллекта
2. Руководитель направления разработок Центра развития цифровых технологий в образовательных процессах
3. Старший преподаватель факультета искусственного интеллекта

Вы должны знать

✓ Разработка алгоритмов

Временная сложность алгоритма и вычислительная сложность

✓ Линейная алгебра

Матрицы и операции над матрицами, Векторы, Система линейных алгебраических уравнений

Обратная матрица, Собственный вектор, Невырожденная матрица, Сингулярное разложение

✓ Анализ функций многих переменных

Производная функции, Интеграл, Касательное пространство

✓ Теория вероятностей

Случайная величина, Математическое ожидание , Дисперсия случайной величины, ...

✓ Программирование

Python

- Домашнее задание – **50 баллов**
- Промежуточная аттестация – **20 баллов**
- Итоговая аттестация – **20 баллов**
- Активность на занятиях – **10 баллов**

Мы ежедневно взаимодействуем с ИИ

(и не всегда знаем об этом)

текущее положение дел: миром правит слабый ИИ



КИНОПОИСК



Яндекс **Go Такси**



Яндекс **Музыка**

СБЕР БАНК

маруся

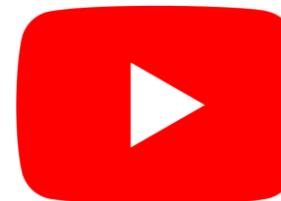
NETFLIX



иви



Яндекс **Директ**



История развития искусственного интеллекта

❖ Идея — 1943 год

❖ Активное развитие — с 2022 года

1943

Первая математическая модель искусственной сети

1950

Алан Тьюринг публикует статью в журнале *Mind*, где предлагает тест на «интеллектуальность машины»

1956

Проведён Дартмутский семинар, заложивший теоретические основы искусственного интеллекта

1980-е

Разработаны первые свёрточные нейронные сети

1991

Опубликована первая версия языка программирования Python

1997

Deep Blue побеждает Гарри Каспарова в шахматном турнире

2005

Джеффри Хинтон и Йошуа Бенджи начинают обучать первые глубокие нейронные сети

2011

IBM Watson побеждает в *Jeopardy!*

2012

Свёрточная нейросеть AlexNet выигрывает конкурс по распознаванию изображений

История развития искусственного интеллекта

❖ Идея — 1943 год

❖ Активное развитие — с 2022 года

2013

Создана исследовательская лаборатория искусственного интеллекта Facebook*, которая позже превратится в Meta* AI, создающую ИИ для метавселенной

2015

Илон Маск и Сэм Альтман основывают НКО Open AI

2015

DeepMind создаёт AlphaGo — нейросеть, способную победить человека в игре го

2016

AlphaGo побеждает Ли Седоля

2017

Выход знакового препринта, заложившего архитектуру «Attention is all you need»

2018

DeepMind создаёт AlphaFold

2020

DeepMind выпускает AlphaFold 2

2022

Запуск ChatGPT

2022

Опубликована «белковая» языковая модель ESM-2

2023

Разработана генеративная модель для белкового дизайна RFdiffusion

Экспертная система

ЕСЛИ

(Температура тела > 37,5)

И

(Головная боль отсутствует)

И

...

И

(Дискомфорта в глазах нет)

ТО

ПРОСТУДА



История машинного обучения



Tom Mitchell (Tom Mitchell, 1998)

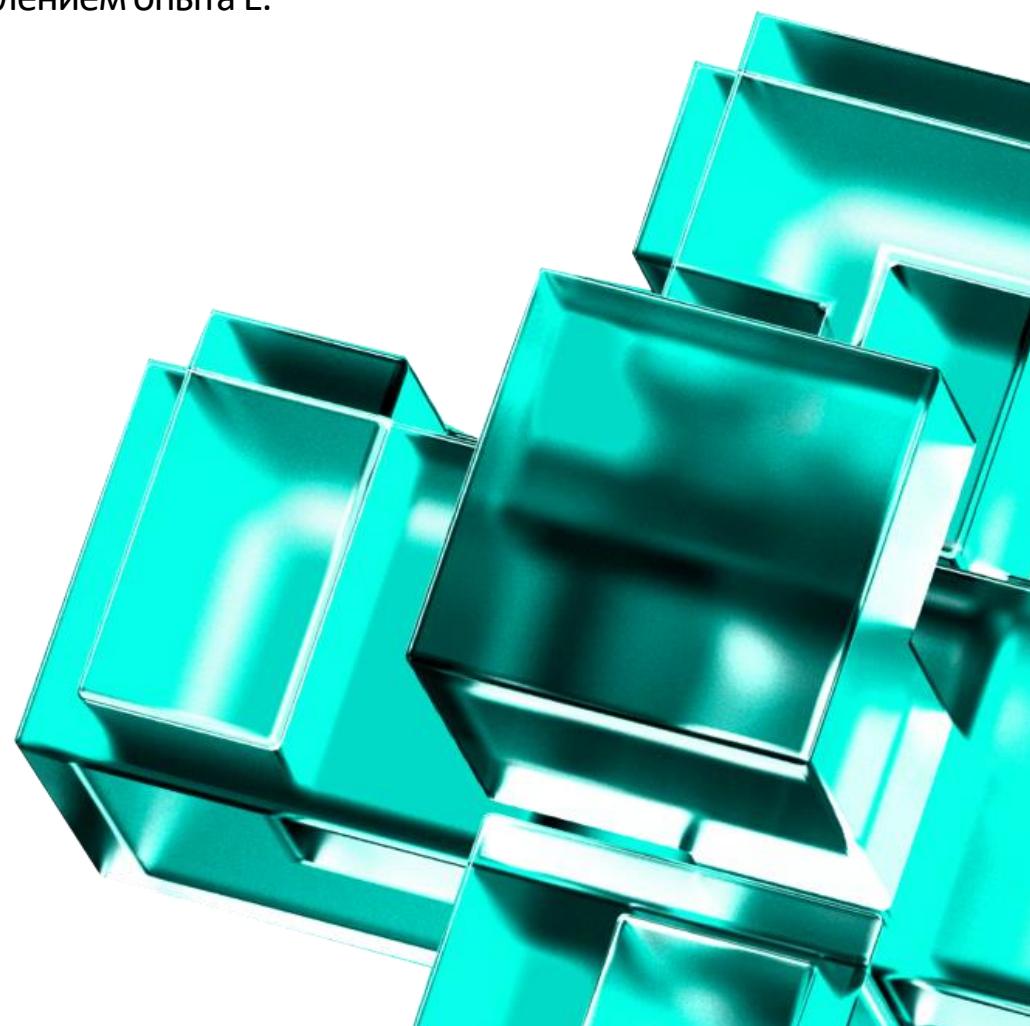
Компьютерная программа обучается с использованием опыта E относительно задачи T и меры успеха P, если качество ее работы в задаче T, измеряемое мерой успеха P, улучшается с накоплением опыта E.

Шашки

Задача Т: Игра в шашки

Опыт Е: Опыт включает в себя историю игры

Мера успеха Р: Мерой успеха



Что такое машинное обучение?

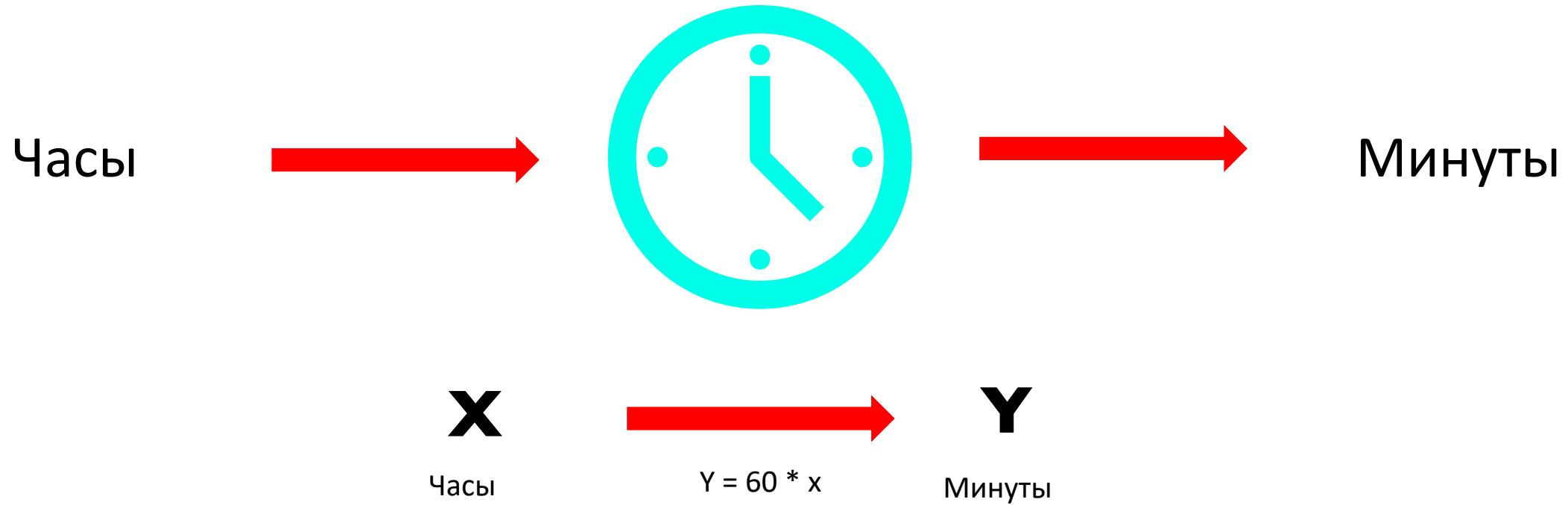
Искусственный интеллект

Машинное обучение
сотни других методов обучения

Нейросети

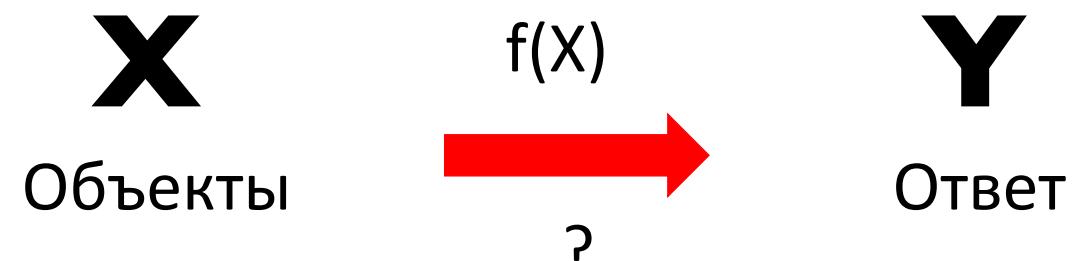
Глубокое обучение

Машинное обучение

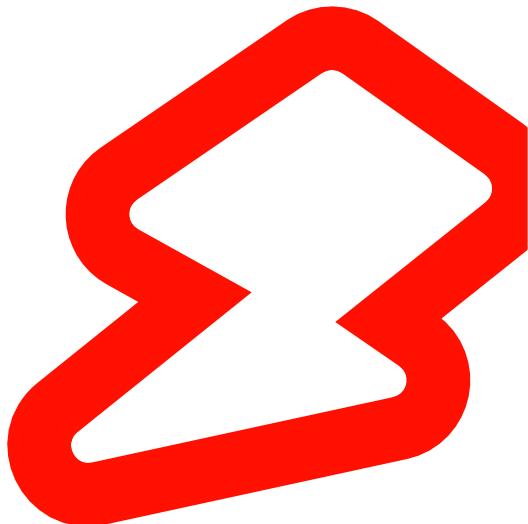


Машинное обучение

Больше сложных зависимостей



$$Y \approx f(X)$$



Основные виды методов машинного обучения



Обучение с учителем (Supervised Learning)

Модель обучается на основе размеченных данных, где для каждого примера входных данных имеется соответствующая метка.

Целью этого обучения является разработка модели, которая способна предсказывать целевую переменную для новых, ранее не виденных данных.



Обучение без учителя (Unsupervised Learning)

Модель обучается на неразмеченных данных, то есть данных, для которых нет предварительно заданных меток классов или целевых переменных.

Вместо этого, задачей в обучении без учителя является выявление скрытых структур, паттернов, группировок или зависимостей в данных.



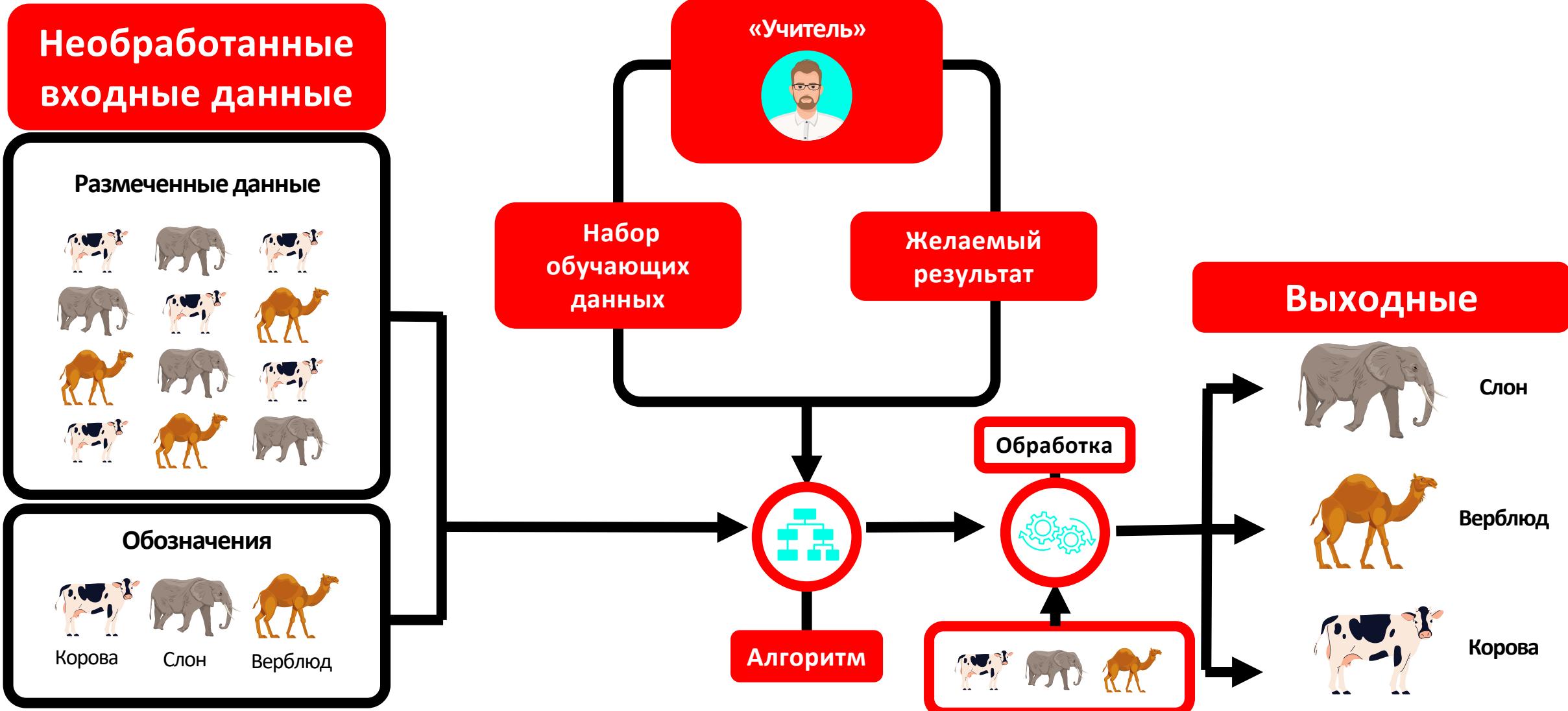
Обучение с подкреплением (Reinforcement Learning):

Это вид машинного обучения, в котором агент (искусственный интеллект) взаимодействует с окружающей средой и принимает решения с целью максимизации награды или определенного критерия успеха. В RL нет предварительных пар входных данных и целевых переменных, как в обучении с учителем.

Вместо этого агент учится, испытывая различные действия в среде и анализируя их последствия.

Обучение с учителем

Supervised Learning



Обучение с учителем

Supervised Learning



Набор данных:

Обучающий набор, в котором для каждой записи предоставлен **правильный ответ**

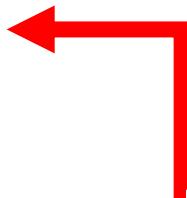


Цель:

Поиск функции f , которая наилучшим образом приближает зависимость между x и y

$$f : X \rightarrow Y$$

Входные
данные



$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$



Целевая переменная



Обучение с учителем (Supervised Learning)



Классификация (Classification):

Модель обучается присваивать входным данным одну из заранее определенных меток классов.

Примеры включают задачи, такие как определение категории электронного письма (спам или не спам) или распознавание изображений (классификация на категории).



Регрессия (Regression):

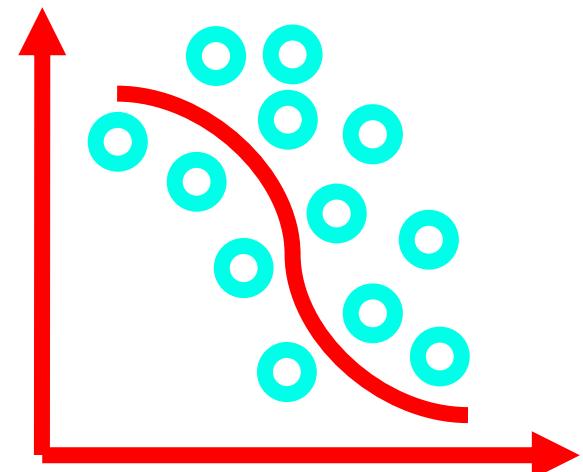
Модель предсказывает числовое значение на основе входных данных.

Примеры включают задачи, такие как прогнозирование цен на недвижимость или предсказание временных рядов, таких как температура и курс валюты.



С учителем:

- Классификация
- Регрессия
 - Какого цвета конфета?
 - Сколько будет стоить квартира?
 - Какая будет цена на нефть в 2025 году?



Идентификация спама



Вход: письмо



Выход: спам / не спам



Dear Sir. First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...



To be removed from future mailings, simply reply to this message and put "remove" in the subject. in the subject. 99 million email addresses for only \$99



Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.



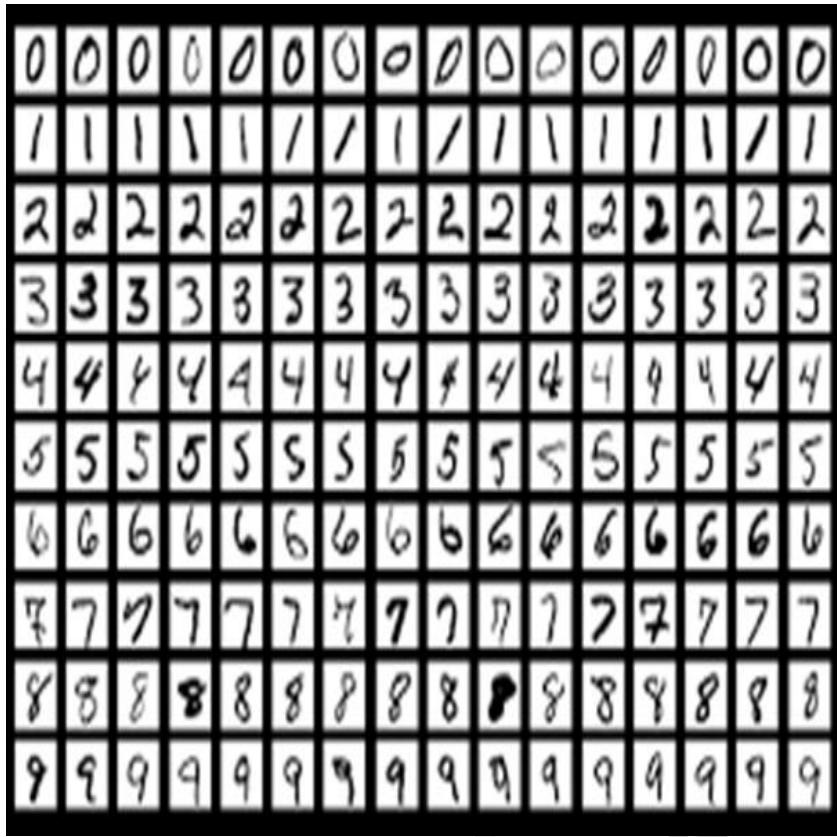
Распознавание рукописных номеров



Вход: Изображение рукописного номера

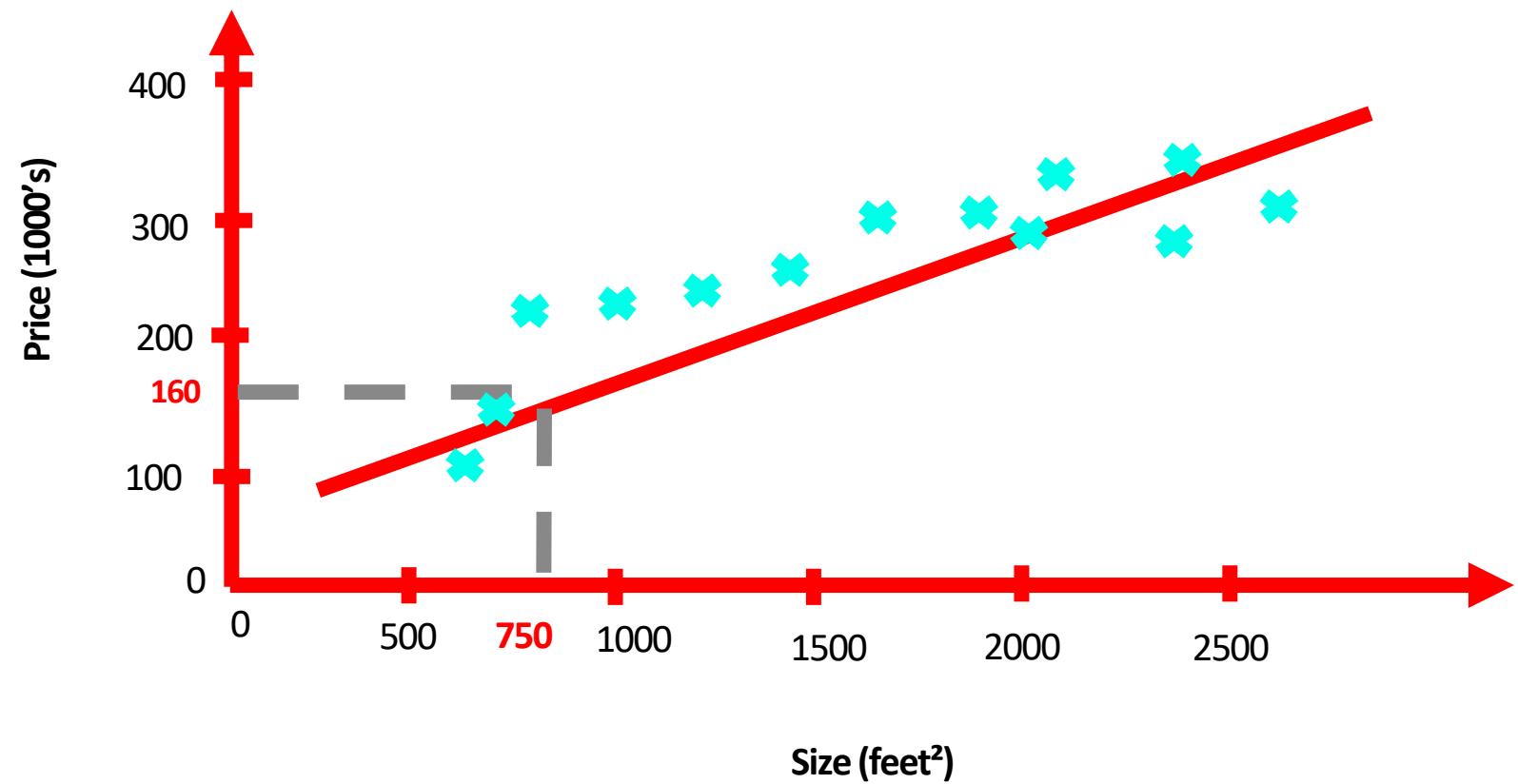


Выход: Число



Стоимость жилого недвижимого имущества

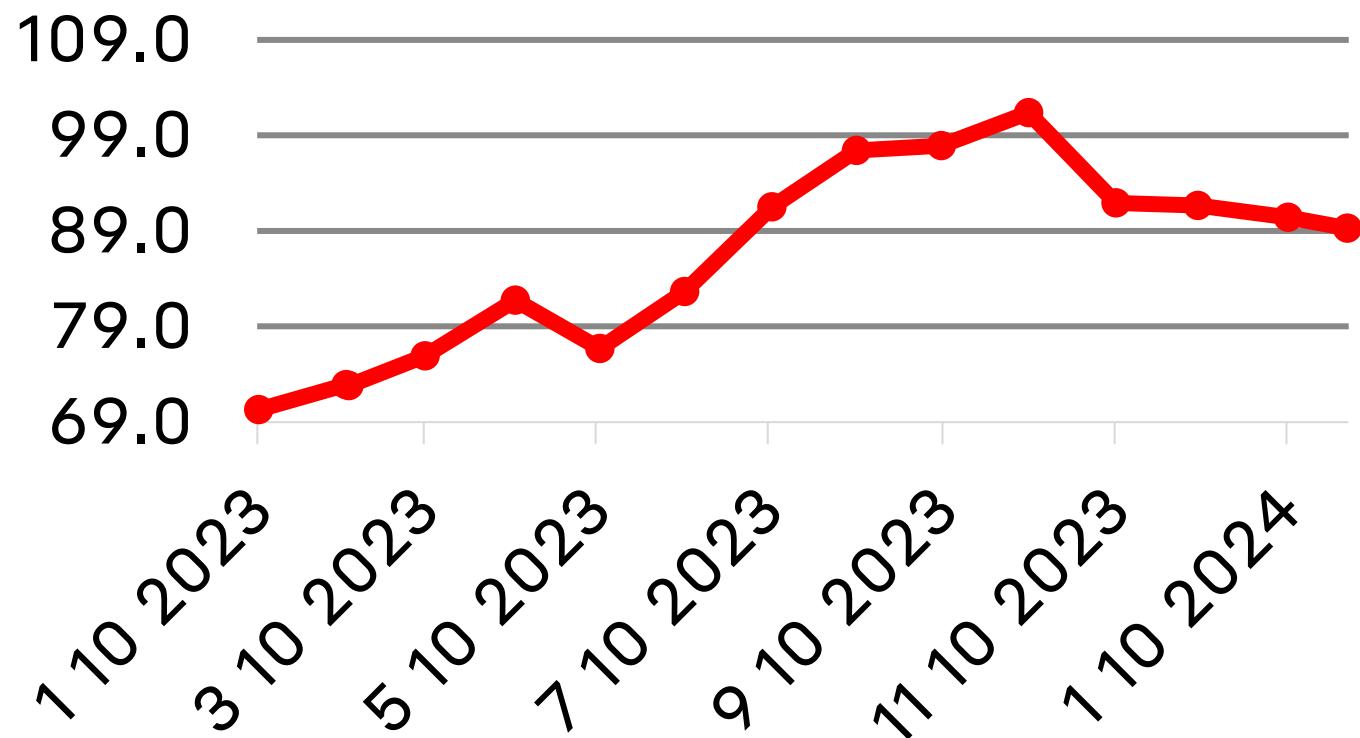
- ➡ Вход: площадь квадрата
- ➡ Выход: стоимость квартиры



Курс валют

⌚ Вход: дата

💸 Выход: стоимость курса в рублях



Задача кредитного скоринга

❖ Набор содержит
13 функций

1	Кредит	Уникальный идентификатор
2	Пол	Пол заявителя Мужчина/женщина
3	Женат	Семейное положение заявителя, значения будут равны «Да»/«Нет»
4	Зависимые	Он показывает, есть ли у заявителя какие-либо иждивенцы или нет
5	Образование	Это покажет нам, получил ли заявитель высшее образование или нет
6	Self_Employed	Это определяет, что заявитель является самозанятым, т.е. «Да»/«Нет»
7	Доход соискателя	Доход кандидата
8	Coapplicantincome	Доход соавтора заявки
9	Количество кредитов	Сумма кредита (в тысячах)
10	Loan_Amount_Term	Условия кредита (в месяцах)
11	Credit_History	Кредитная история погашения физическим лицом своих долгов
12	Property_Area	Площадь собственности, т.е. Сельская/Городская/Полугородская
13	Loan_Status	Статус одобренного кредита: Y – да, N – нет

Обучение без учителя

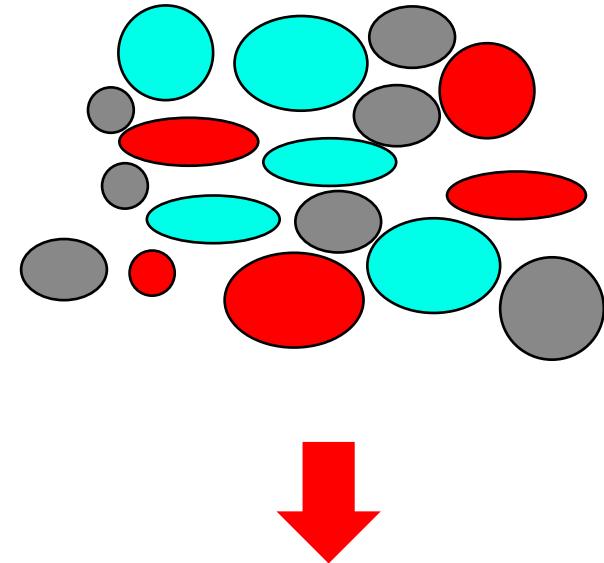
Unsupervised Learning

❖ Модель обучается на неразмеченных данных, то есть данных, для которых нет предварительно заданных меток классов или целевых переменных. Вместо этого, задачей в обучении без учителя является выявление скрытых структур, паттернов, группировок или зависимостей в данных.

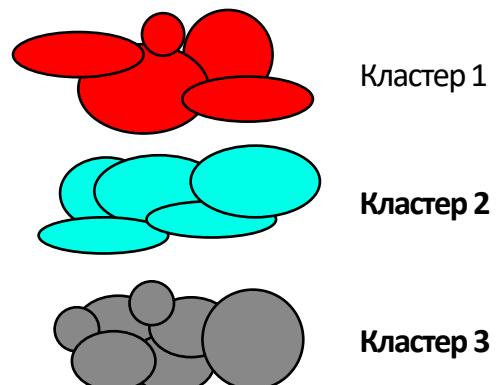
3 Кластеризация

- Машинное обучение – финансы – игры
- Разложить похожие вещи по кучкам

До кластеризации



После кластеризации



Обучение без учителя

Unsupervised Learning



Кластеризация

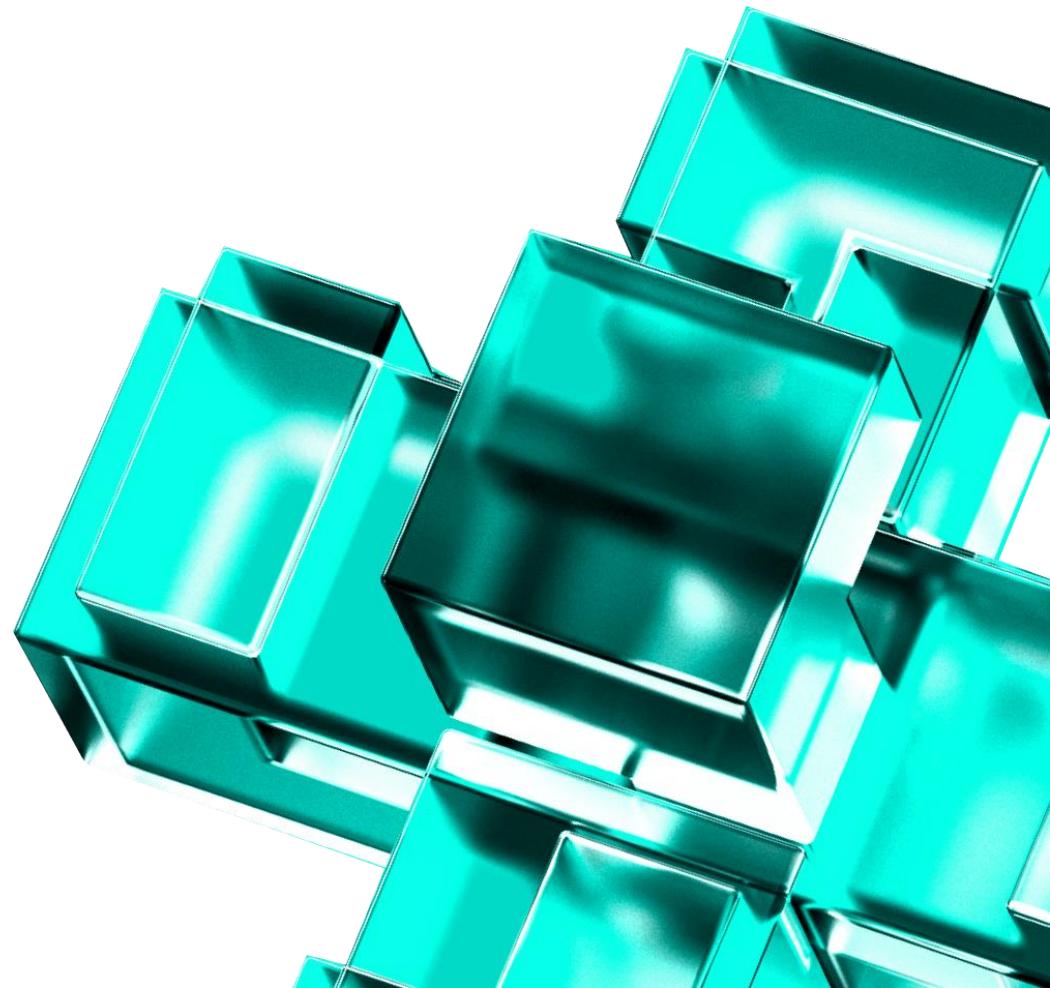


Найти похожих людей



Сфера применения

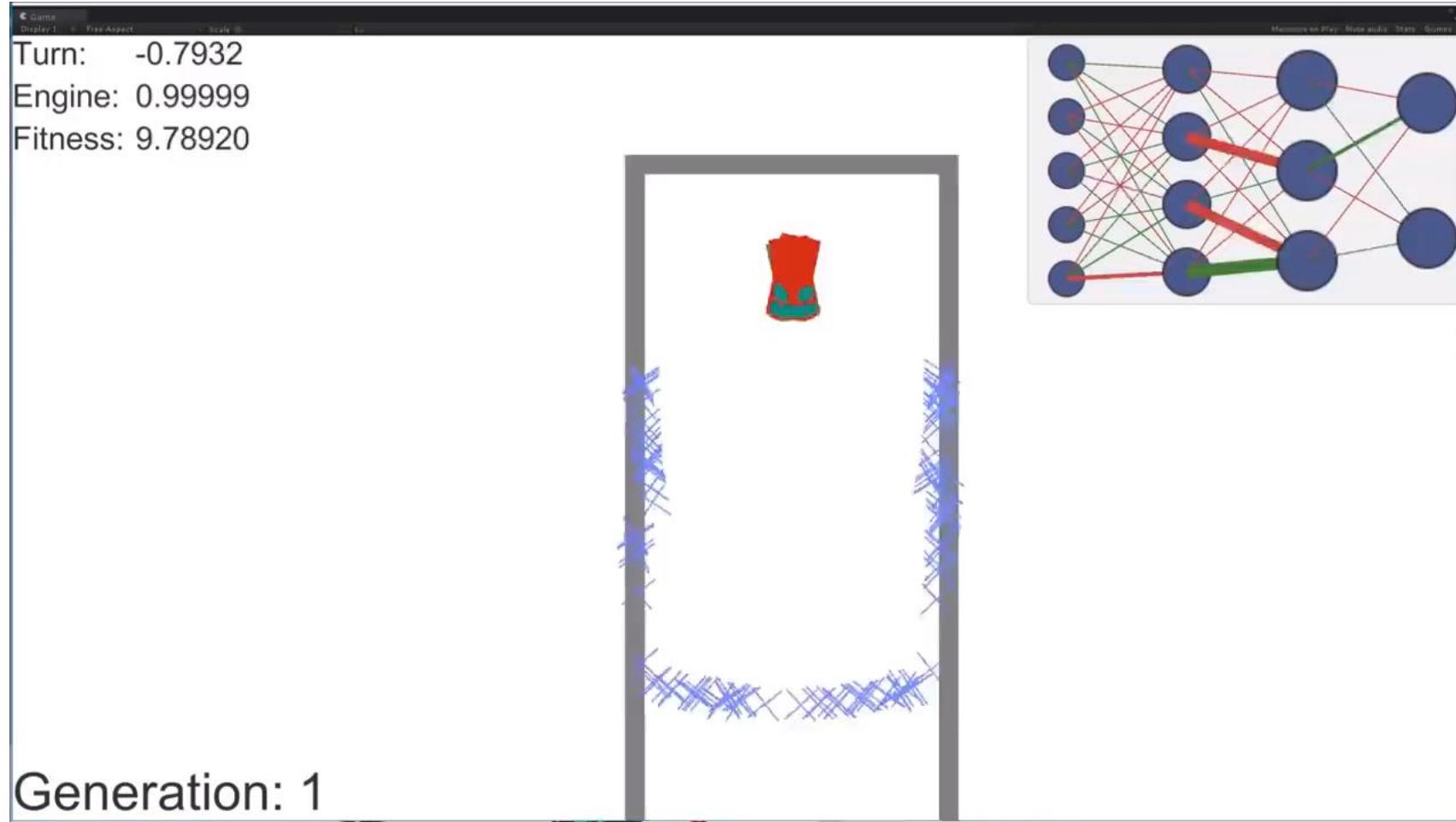
- ⌚ Анализ поведения отдельных клиентских групп
- ⌚ Исследование рынка конкурентов
- ⌚ Изучение статистики выздоровления
- ⌚ Анализ мнений при опросах в разных группах людей
- ⌚ SEO-ключи для формирования тематик страниц сайта
- ⌚ Группировка документов по тематике



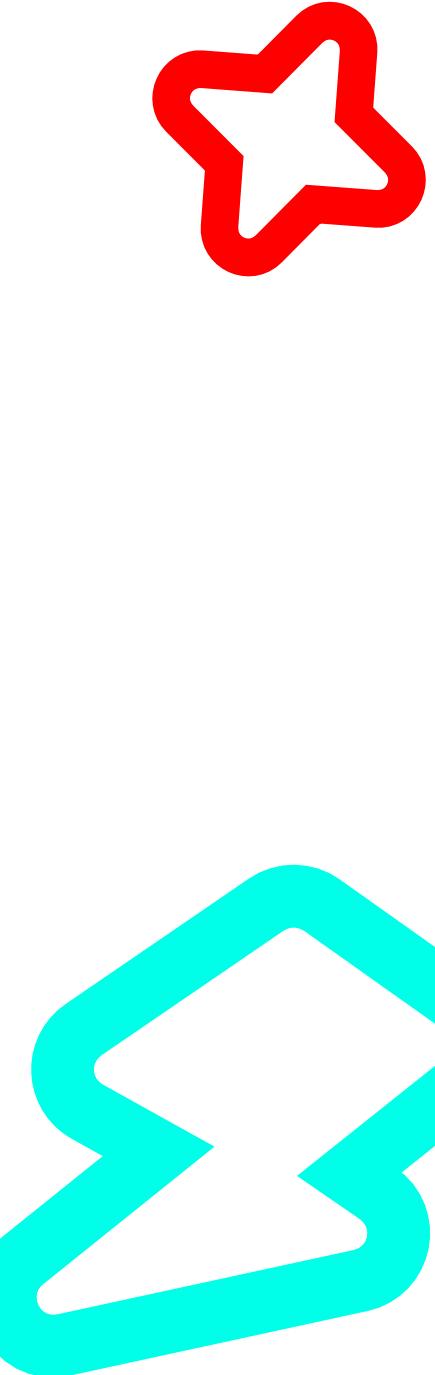
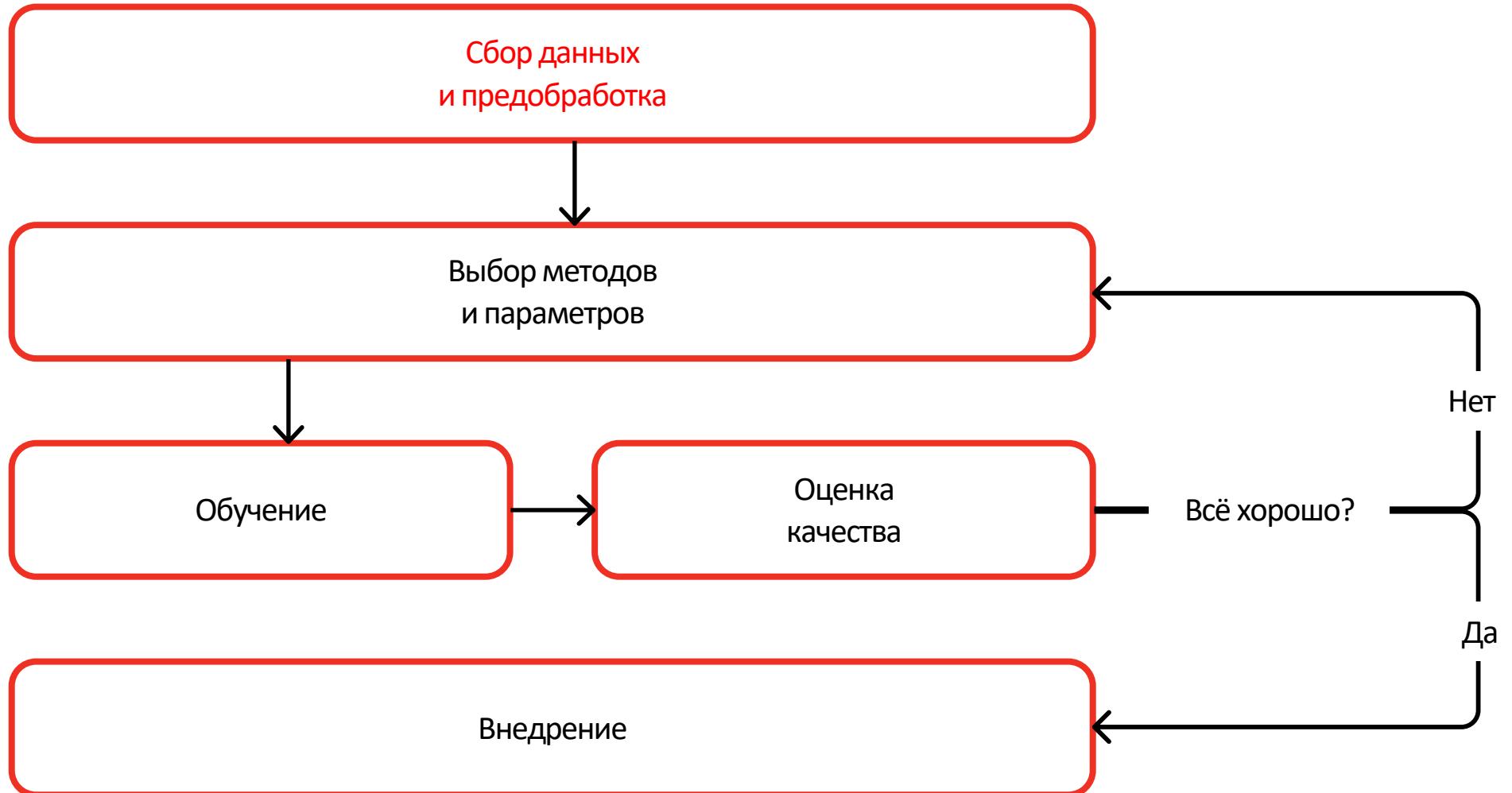
Обучение с подкреплением (Reinforcement Learning)



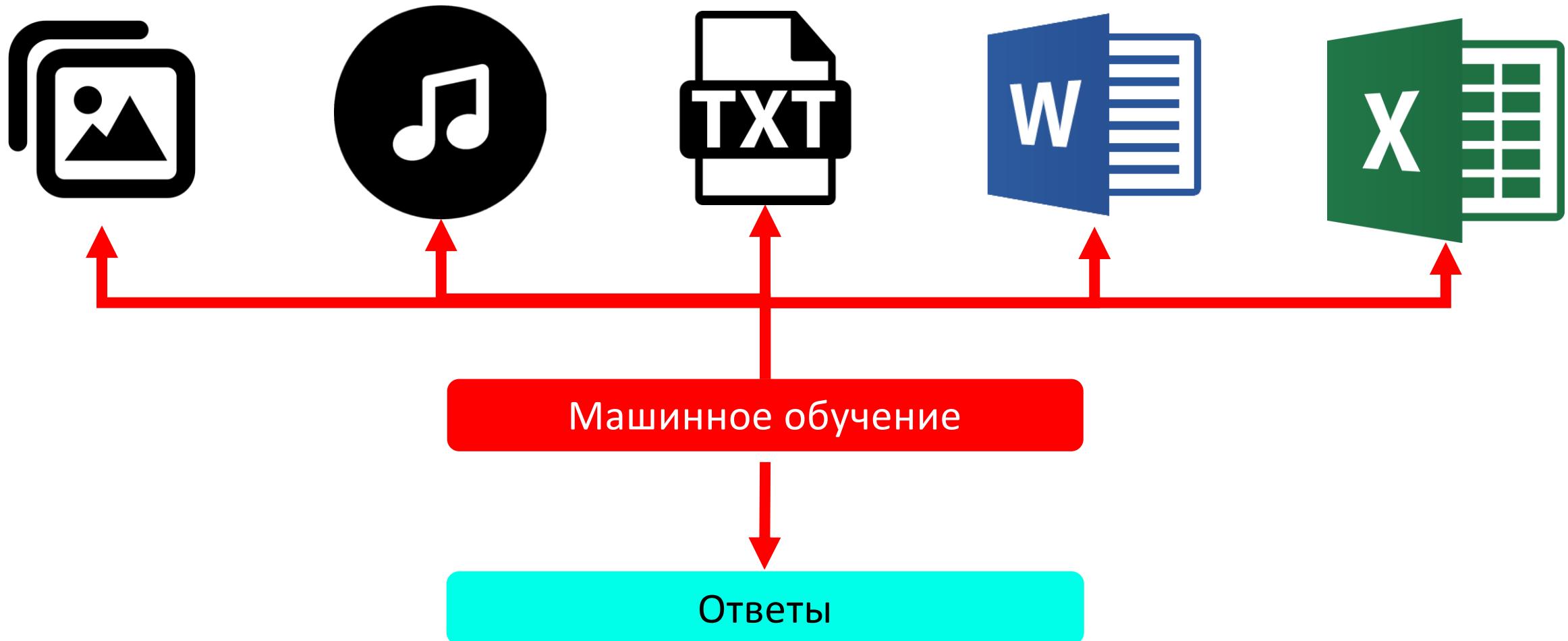
Обучение с подкреплением (Reinforcement Learning)



Ход работы



Типы данных



Сбор и предобработка данных



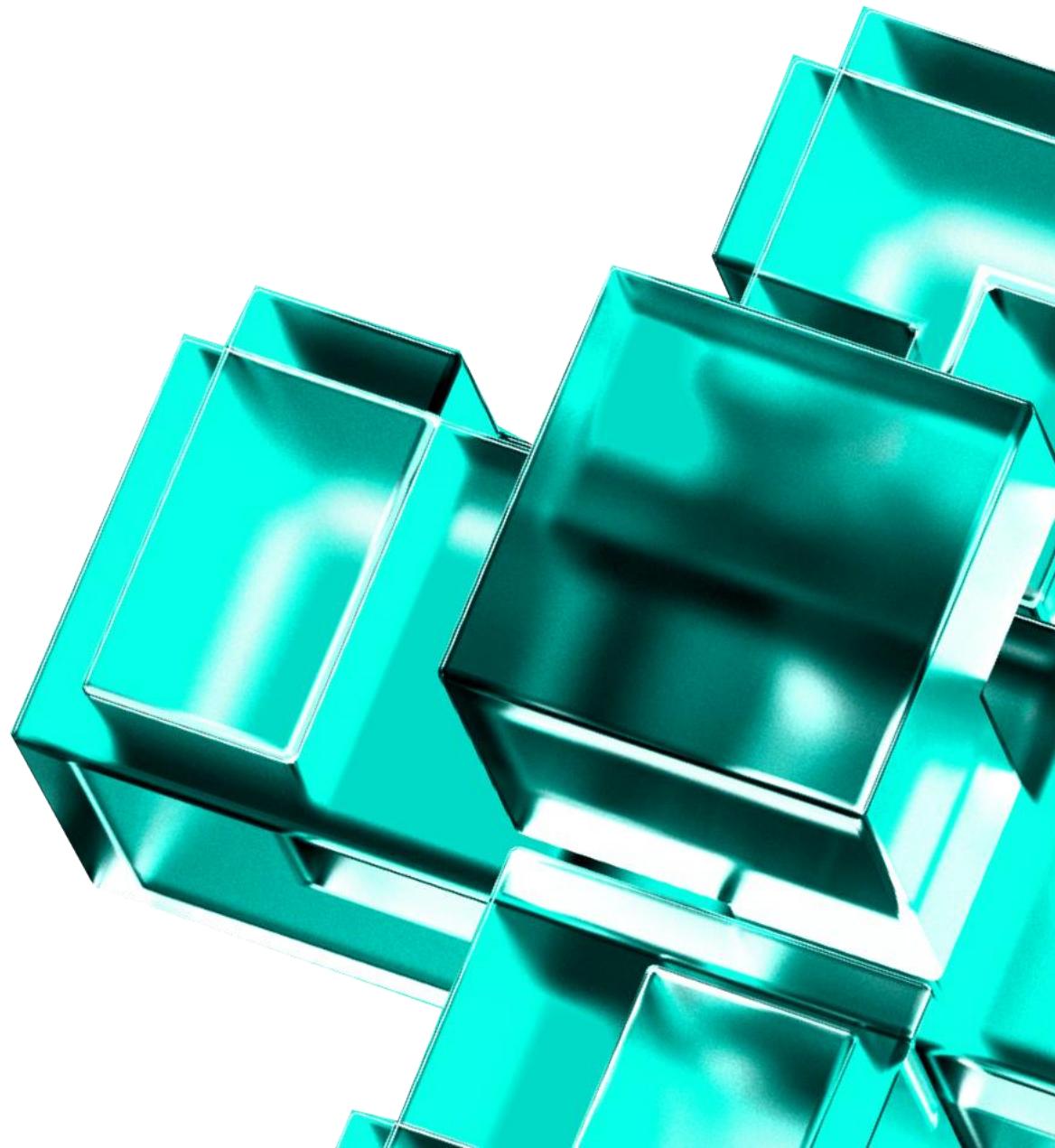
Сбор данных

- MNIST
- Открытые базы данных
- Моделирование событий
- Данные с детекторов
- Каптча

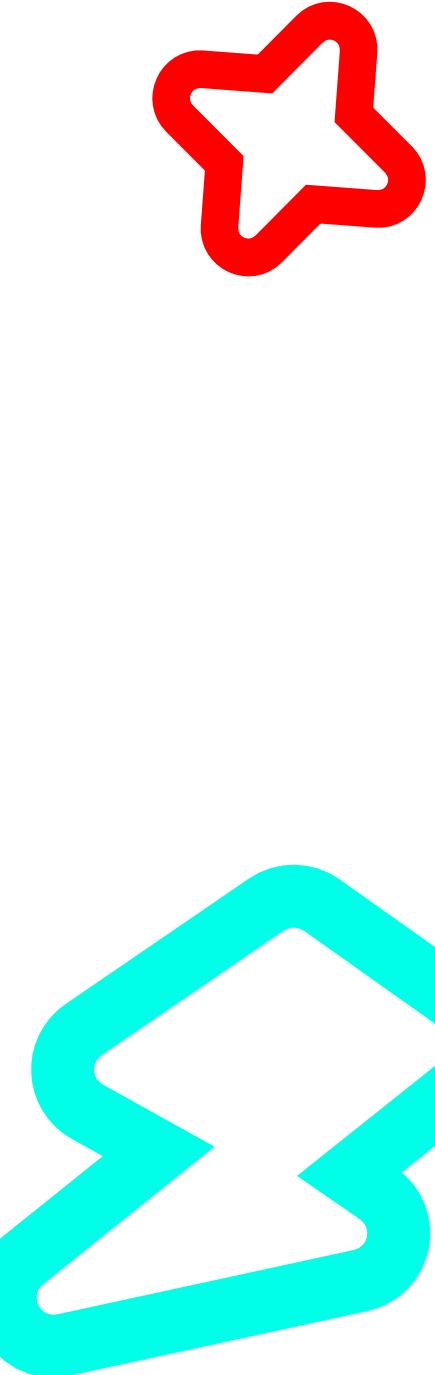
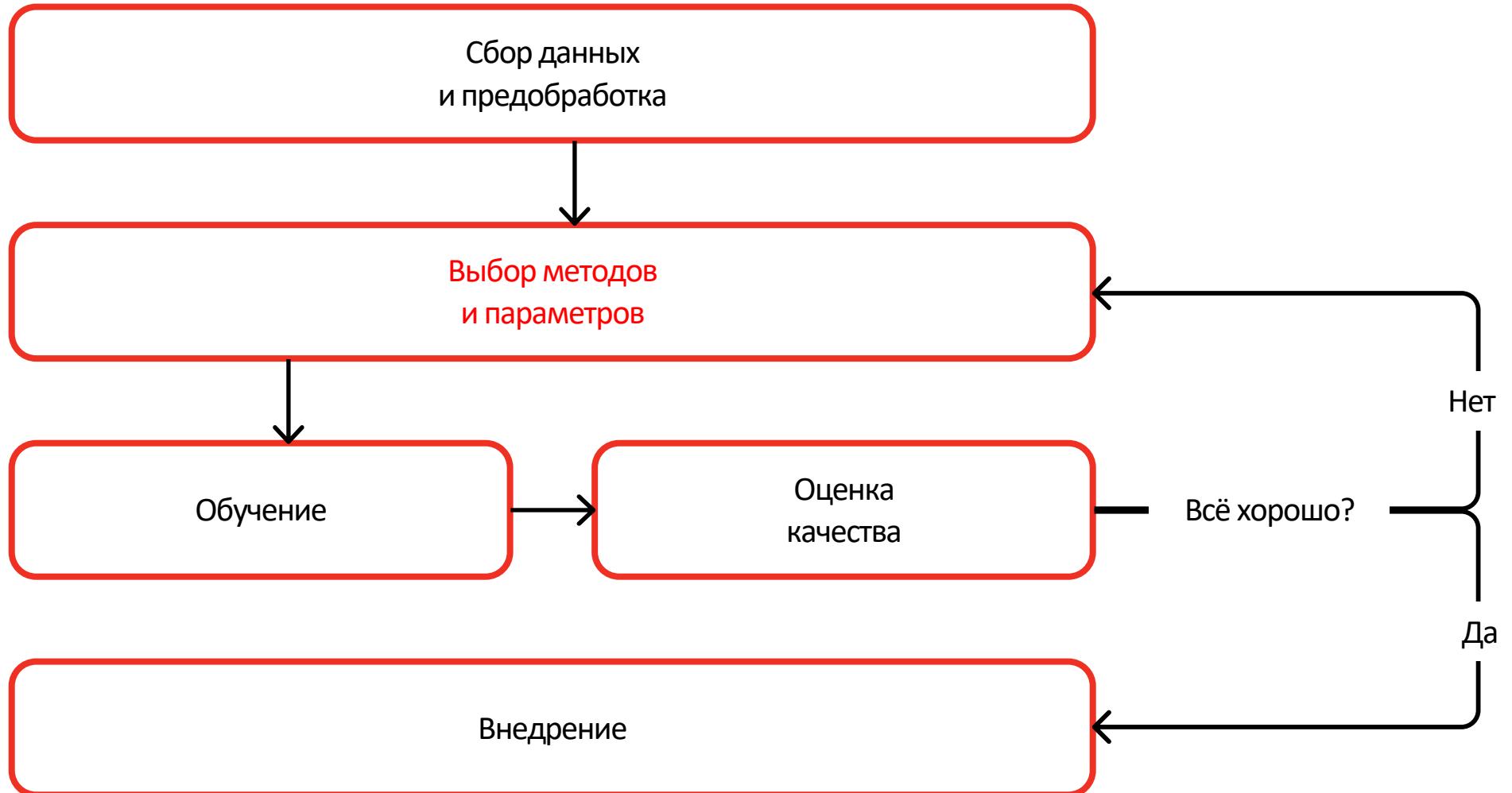


Предобработка

- Исключение выбросов и NaN событий
- Исключение шумовых данных



Ход работы

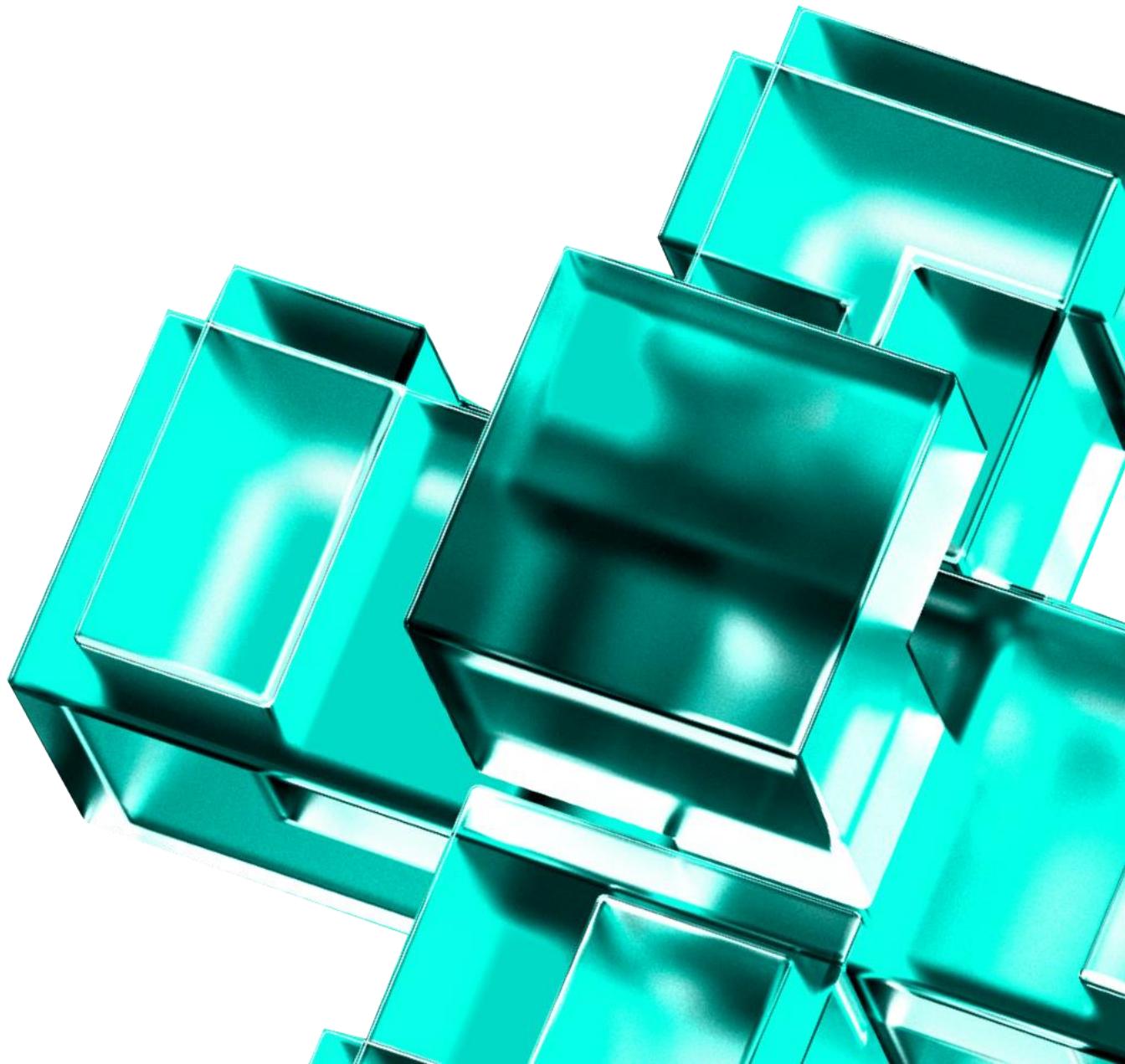


Какой алгоритм выбрать?



«Базовый» список моделей машинного обучения

- Наивный байесовский классификатор
- Дерево решений
- Метод опорных векторов
- Регрессия
- Логистическая регрессия
- Принцип главных компонент
- Другие

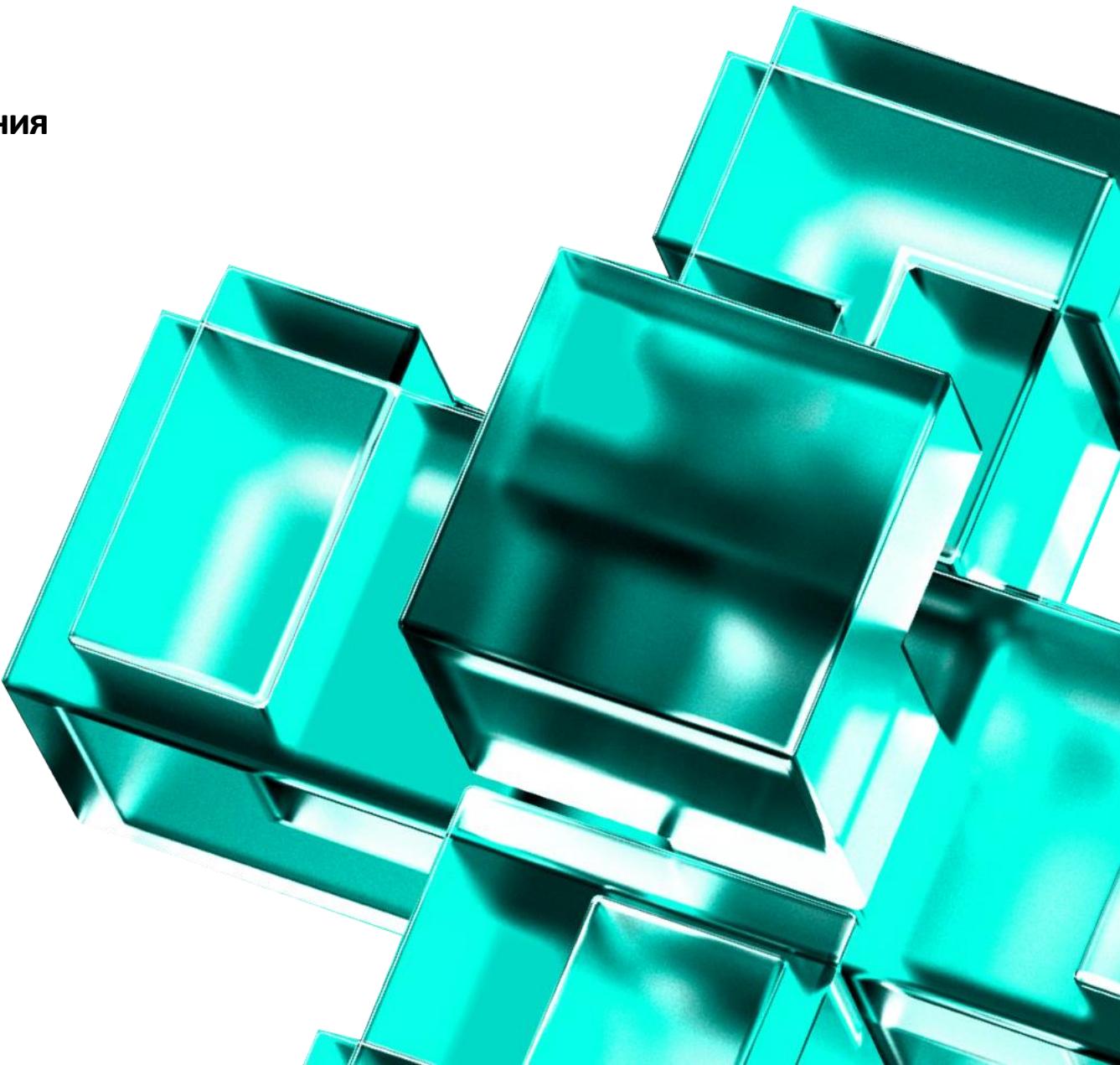


Какой алгоритм выбрать?

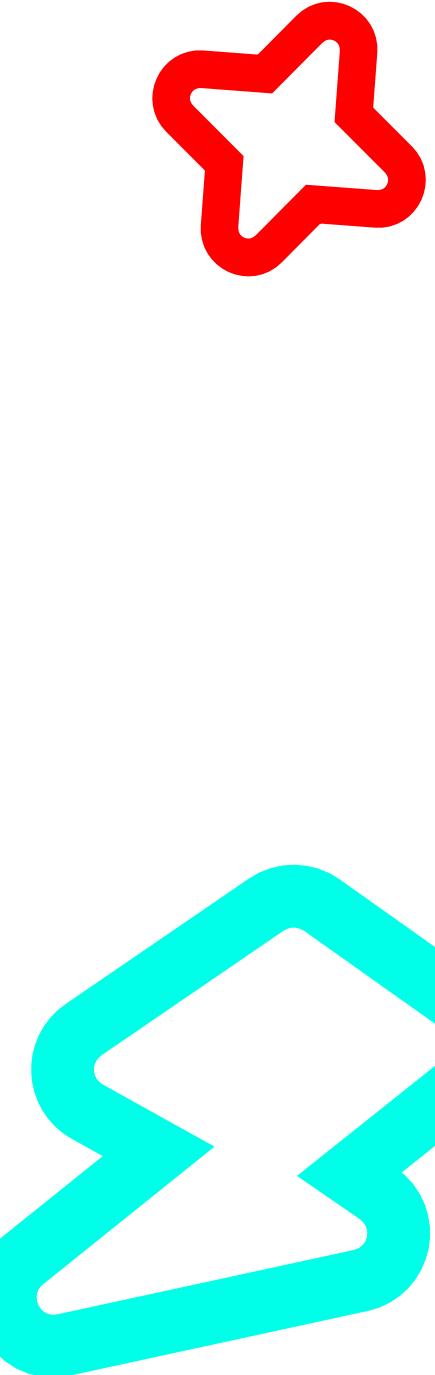
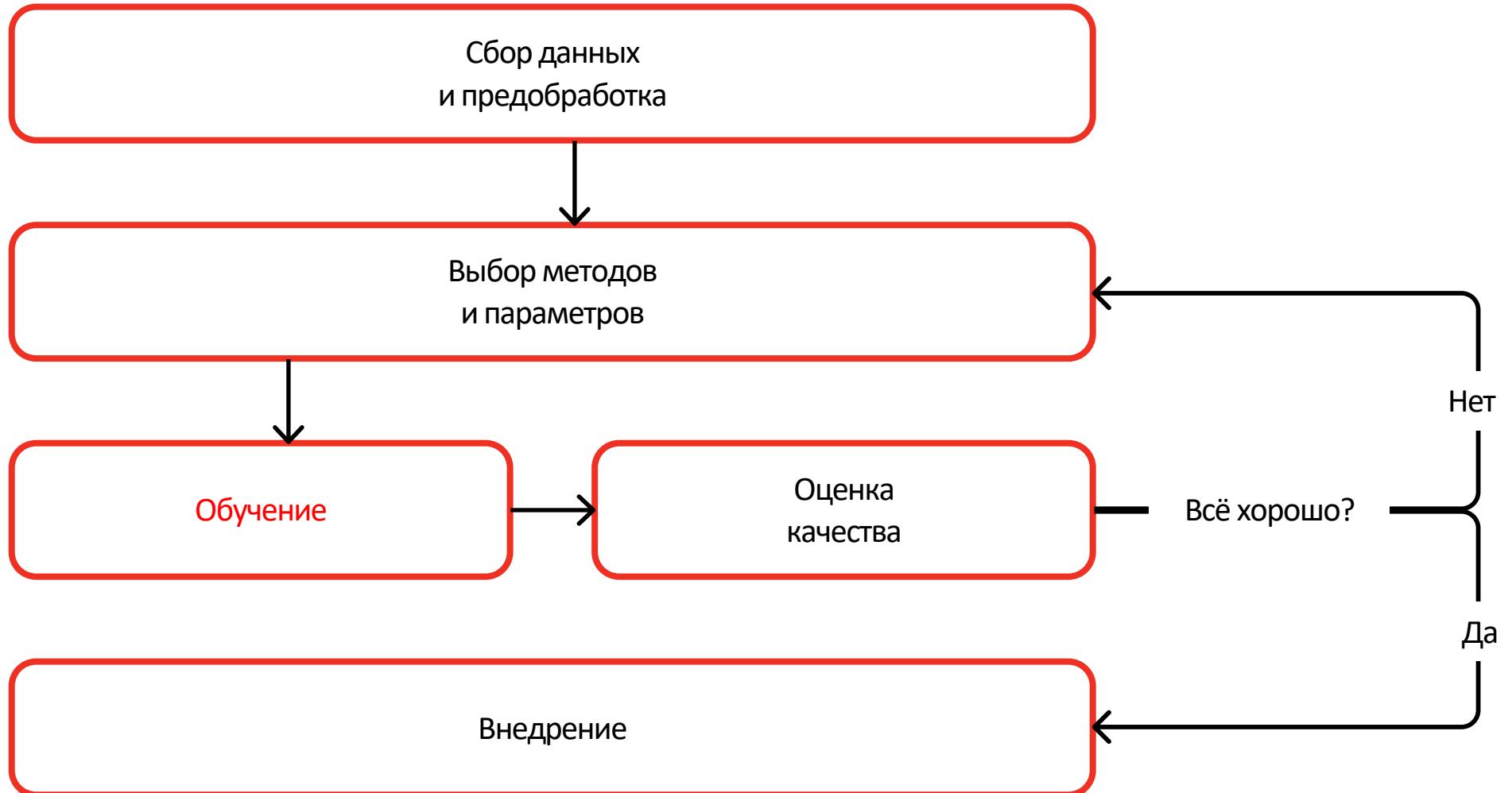


«Расширенный» список моделей машинного обучения

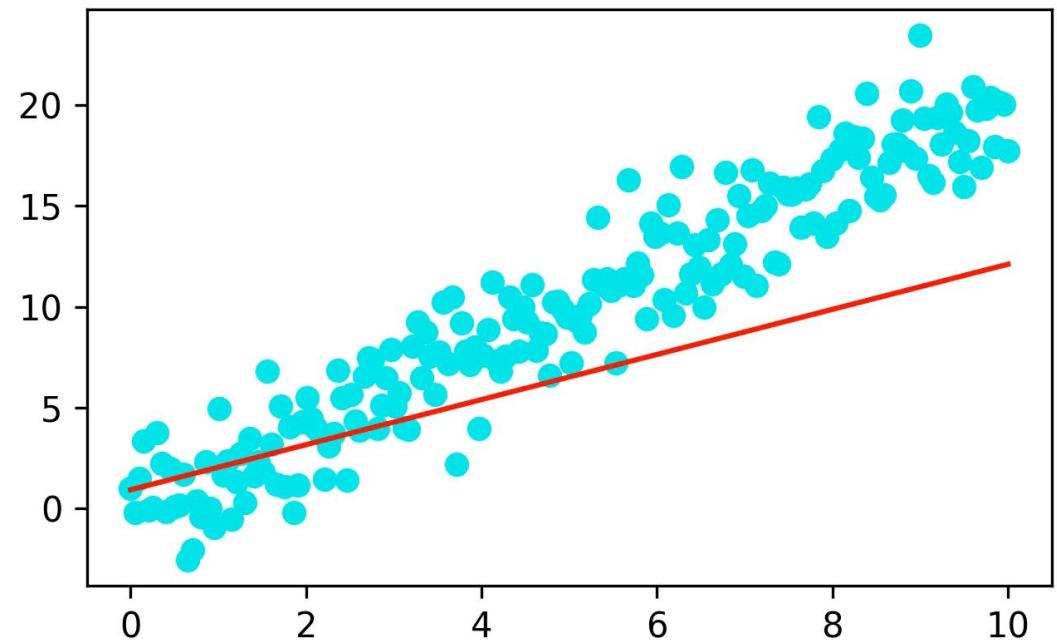
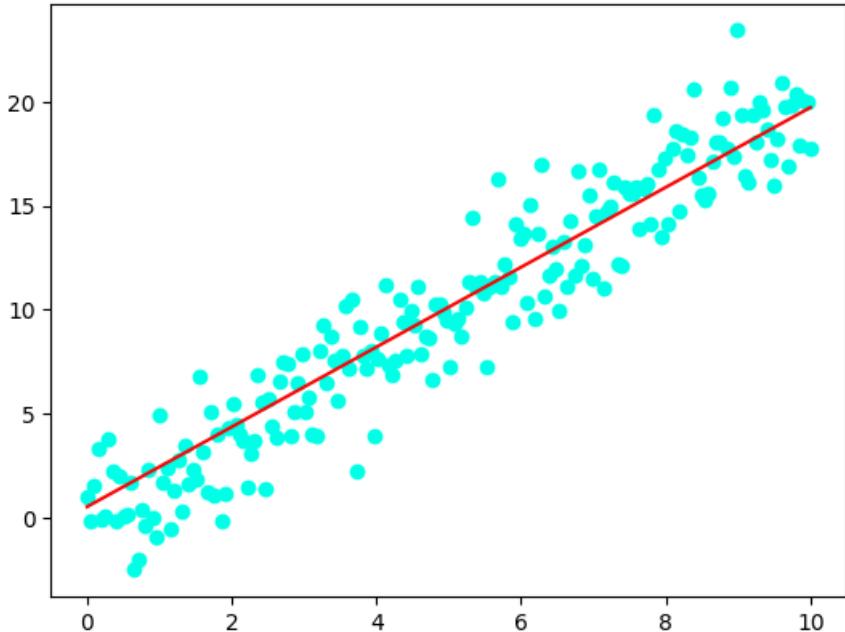
- Нейронная сеть
- Свёрточная нейронная сеть
- К-средних (kNN)
- Случайный лес
- Бустинг над деревьями решений
- Ансамбль моделей
- Другие



Ход работы



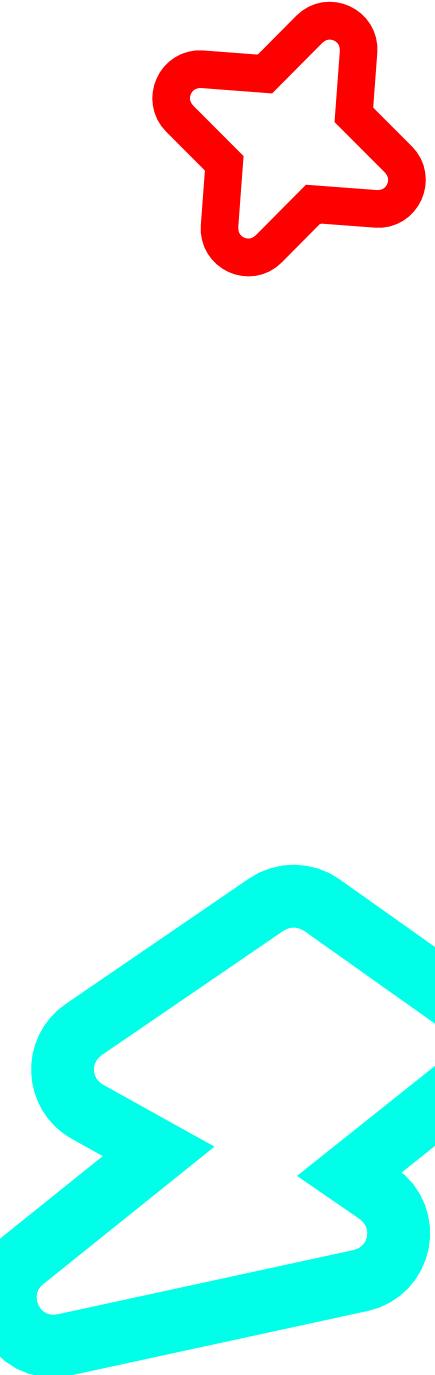
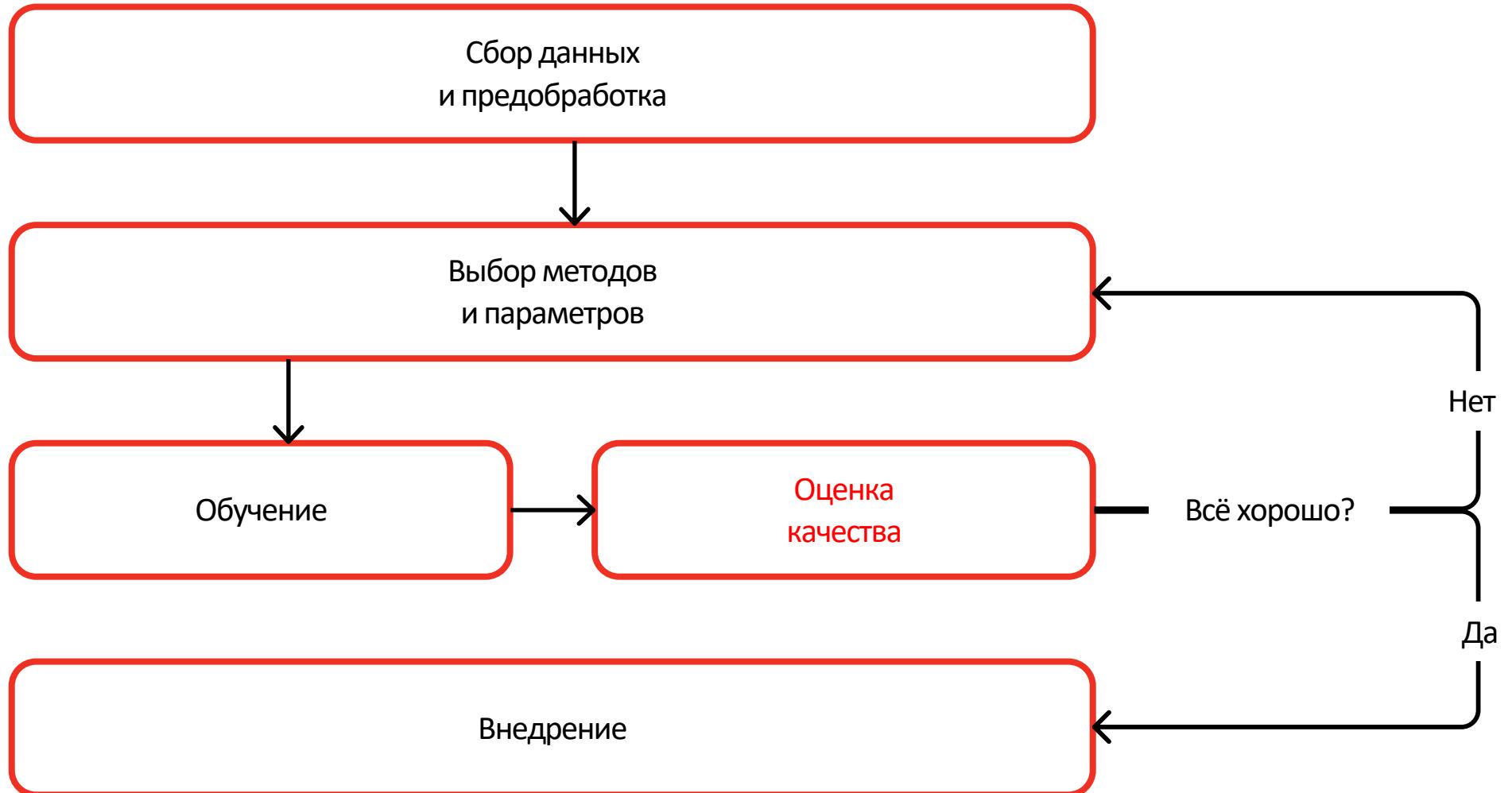
Обучение



Обучение

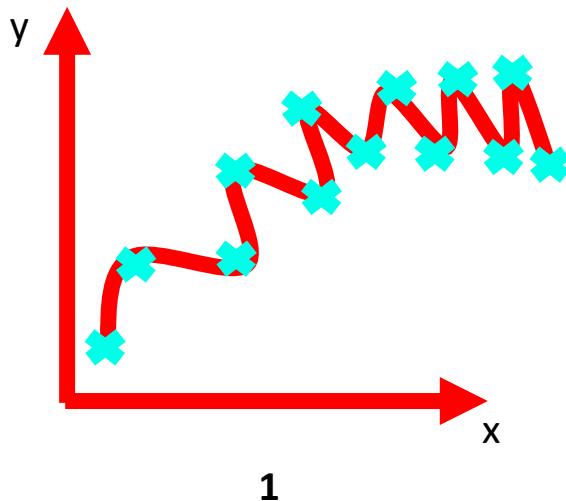


Ход работы

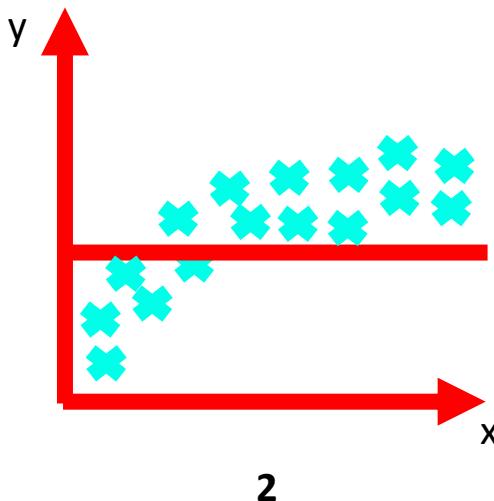


Интерактивное задание

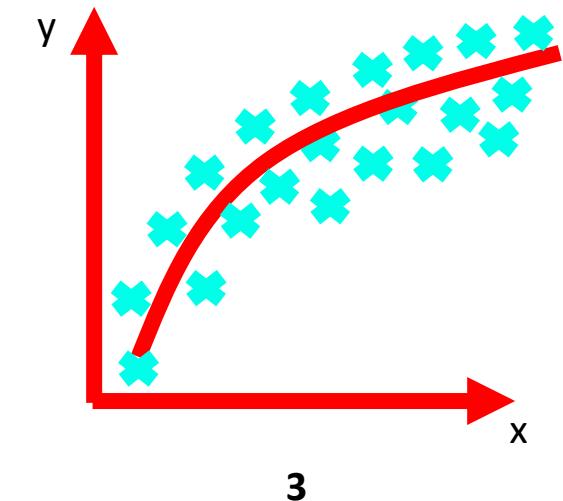
Какой график или модель вы считаете самыми лучшими?



1

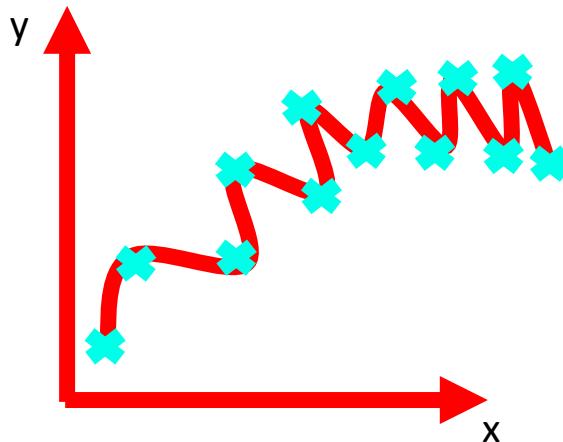


2

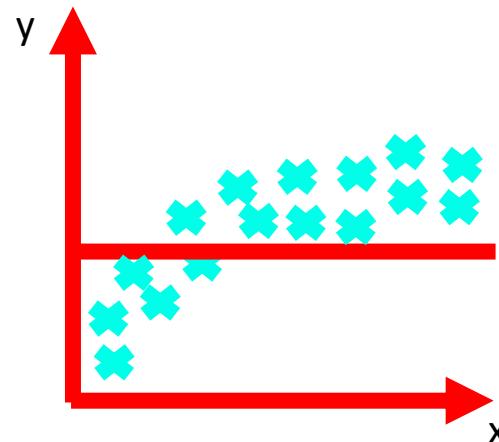


3

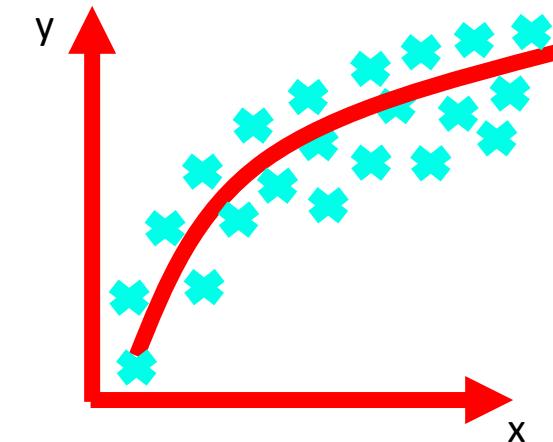
Оценка качества



Overfitting (High Variance)

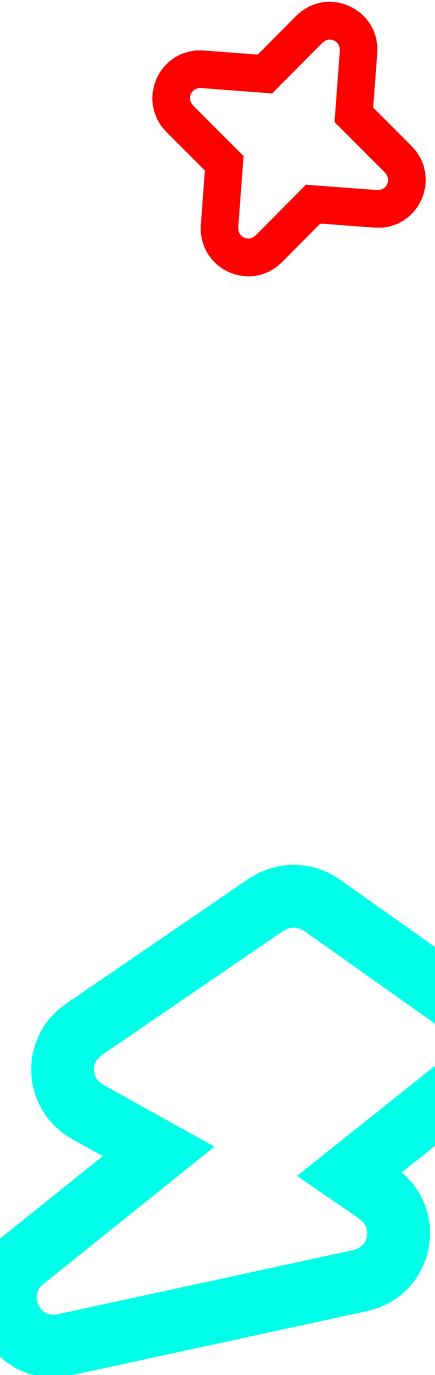
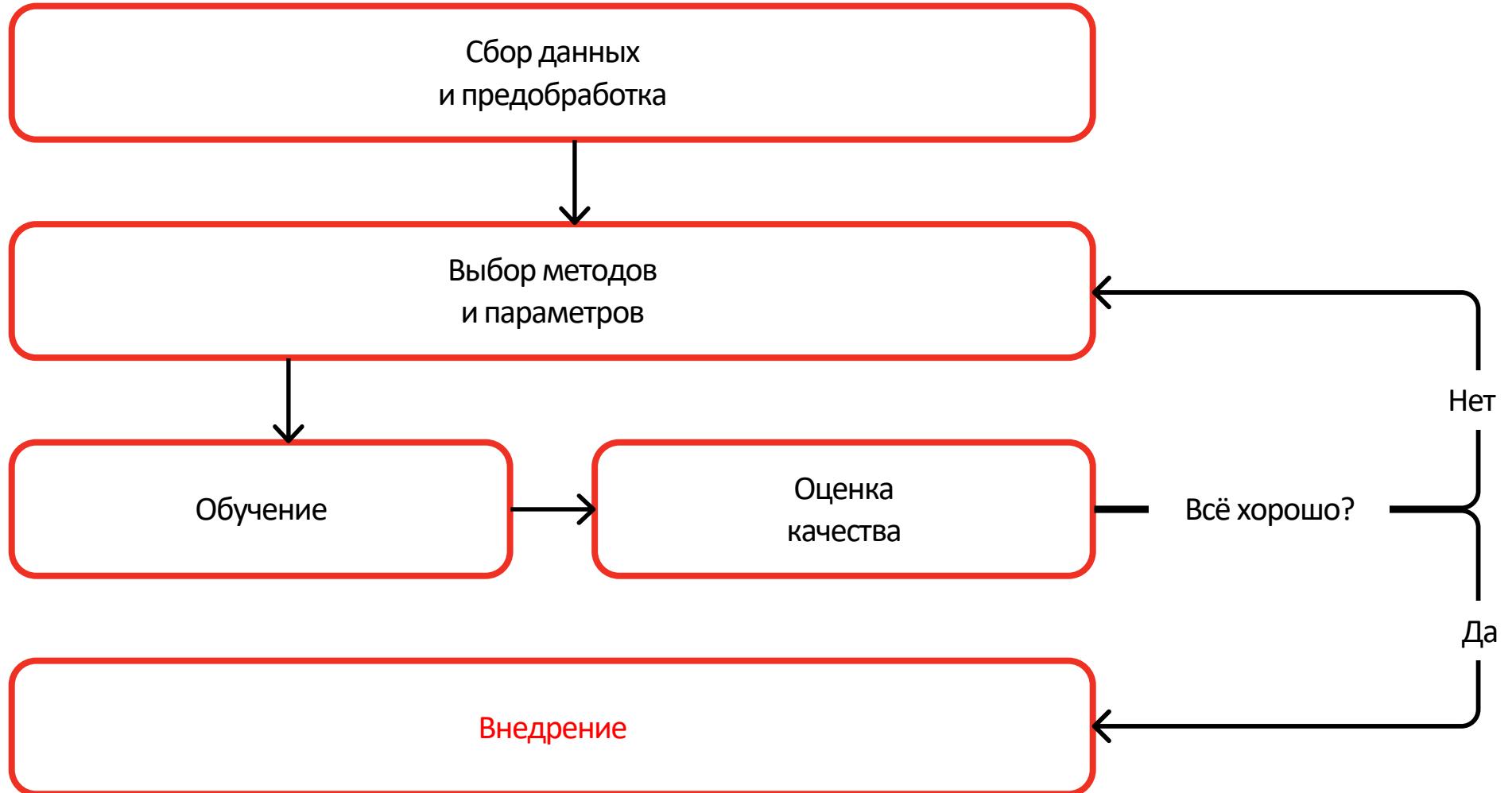


Underfitting (High Bias)



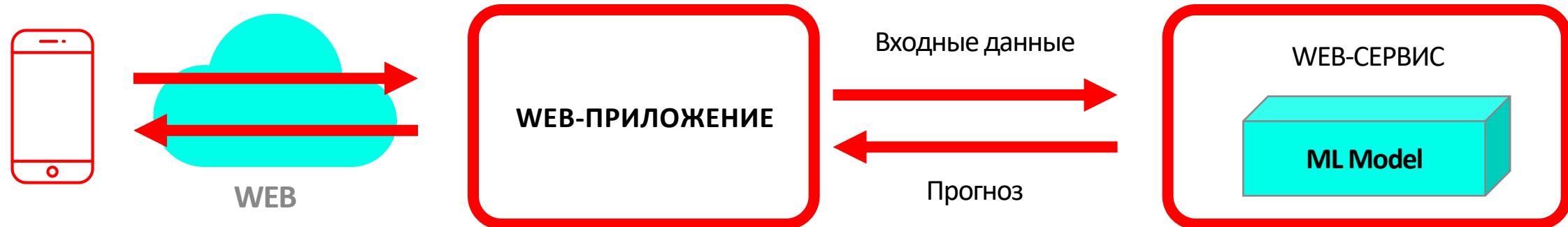
Just Right

Ход работы

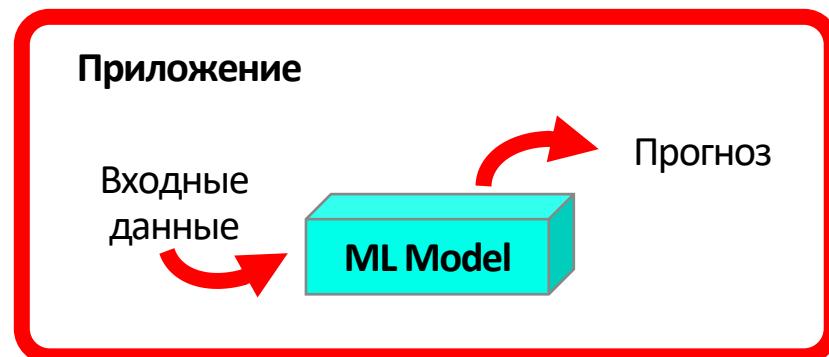


Внедрение

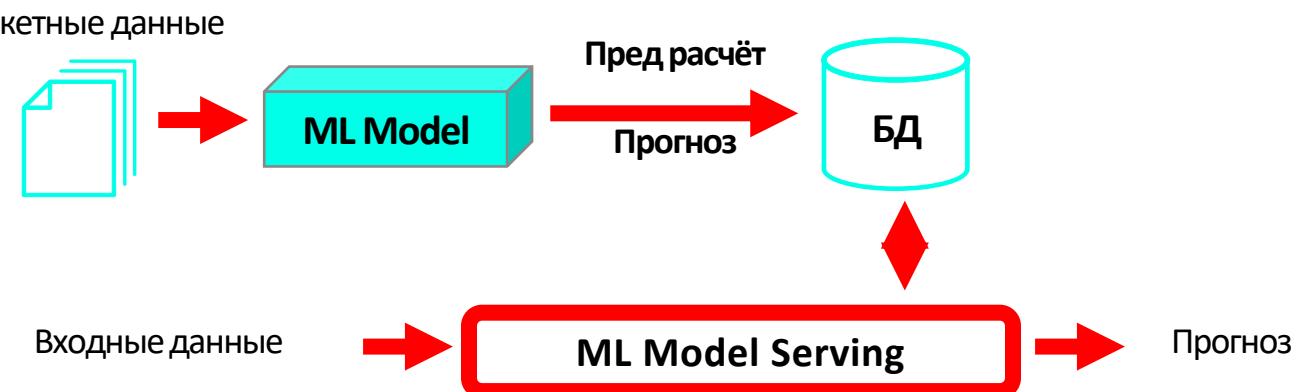
⚡ Модель как услуга



⚡ Модель как зависимость



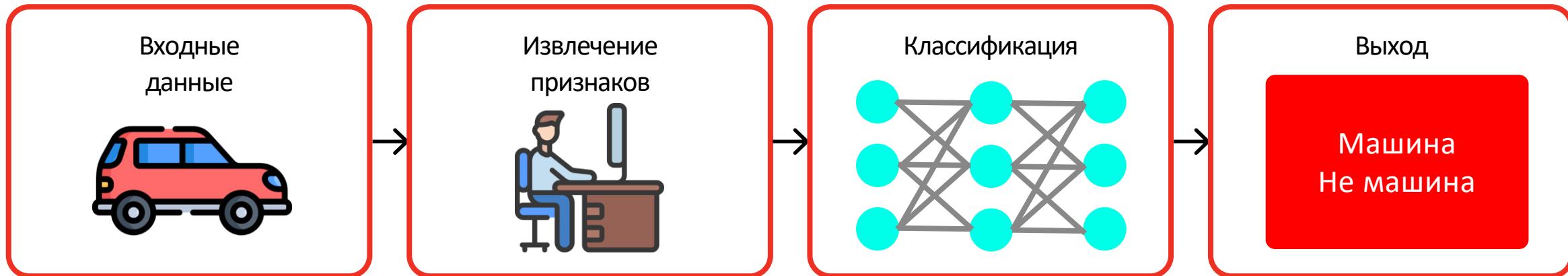
⚡ Предварительный расчёт



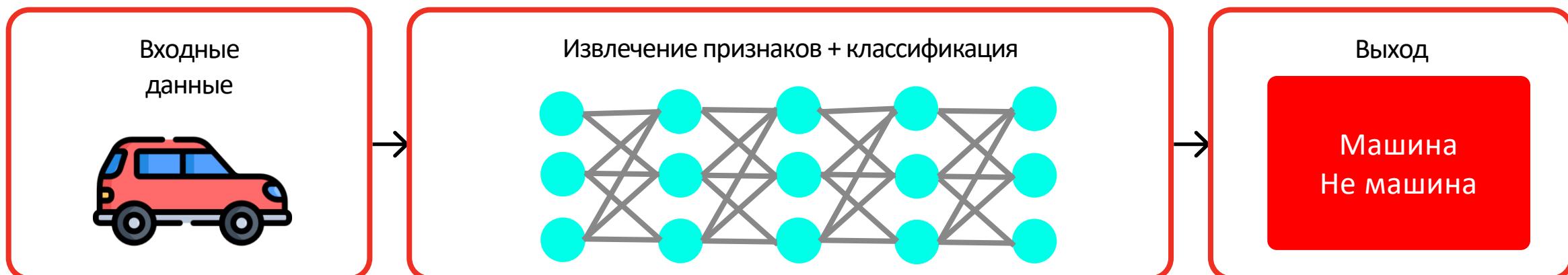
Feature extraction

Извлечение признаков

Машинное обучение



Глубокое обучение

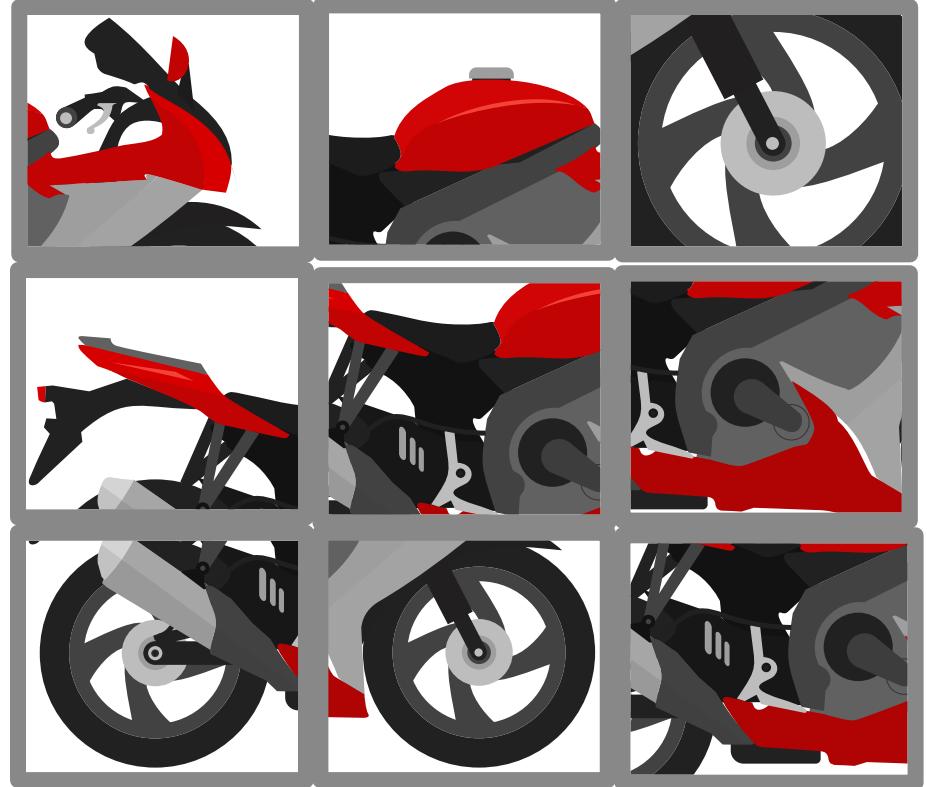


Feature extraction

Извлечение признаков



Алгоритм
извлечени
я
признаков

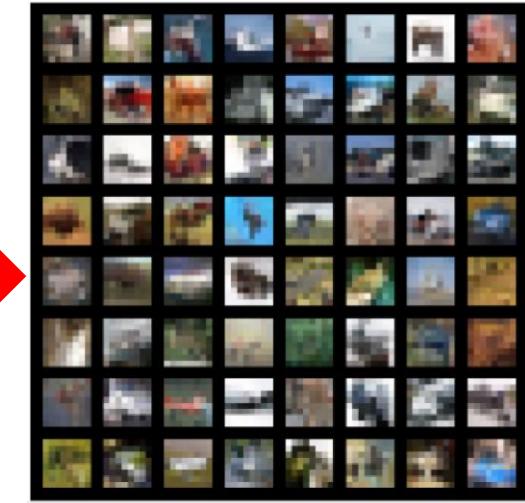
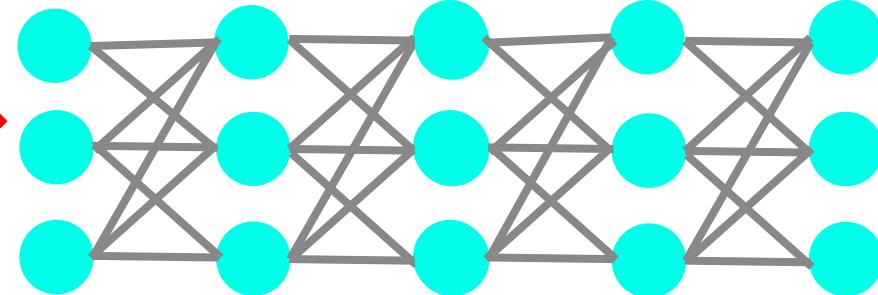


Feature extraction

Извлечение признаков



Глубокое обучение

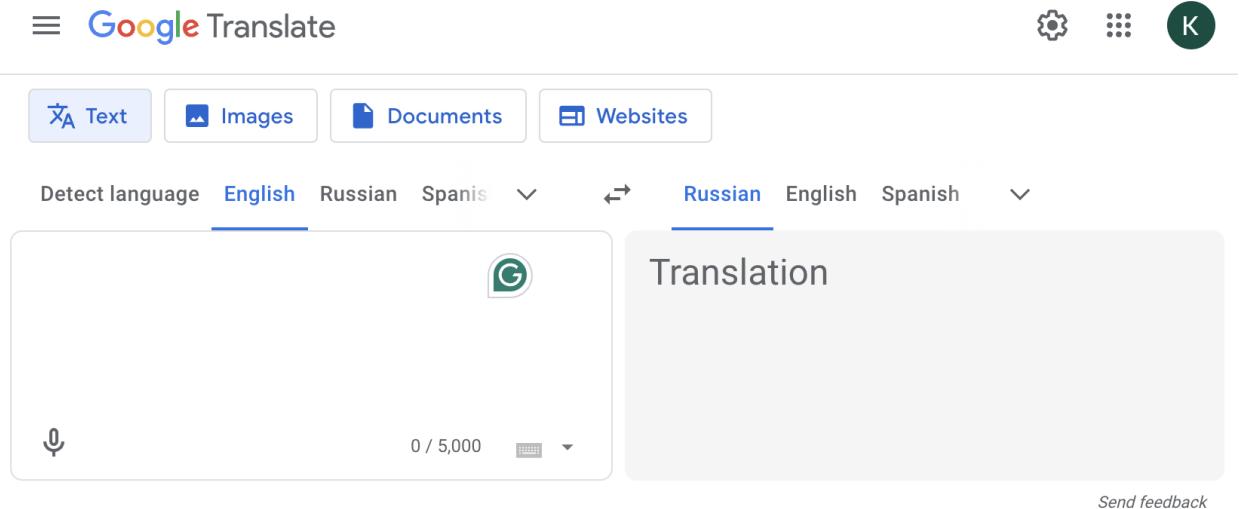
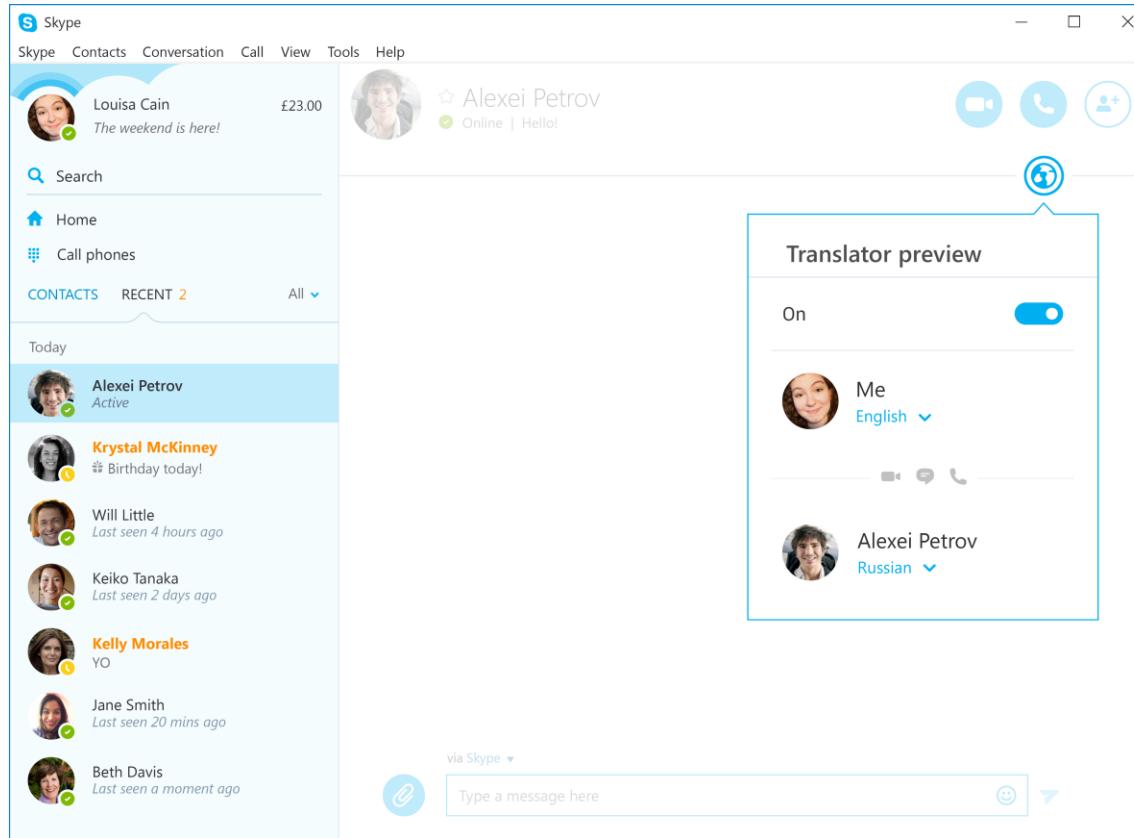


Применение глубоких нейронных сетей Google's AlphaGo



Применение глубоких нейронных сетей

Skype Translator, переводчик Google



Применение глубоких нейронных сетей

Беспилотные автомобили и не только



Сингапур



Питтсбург, США



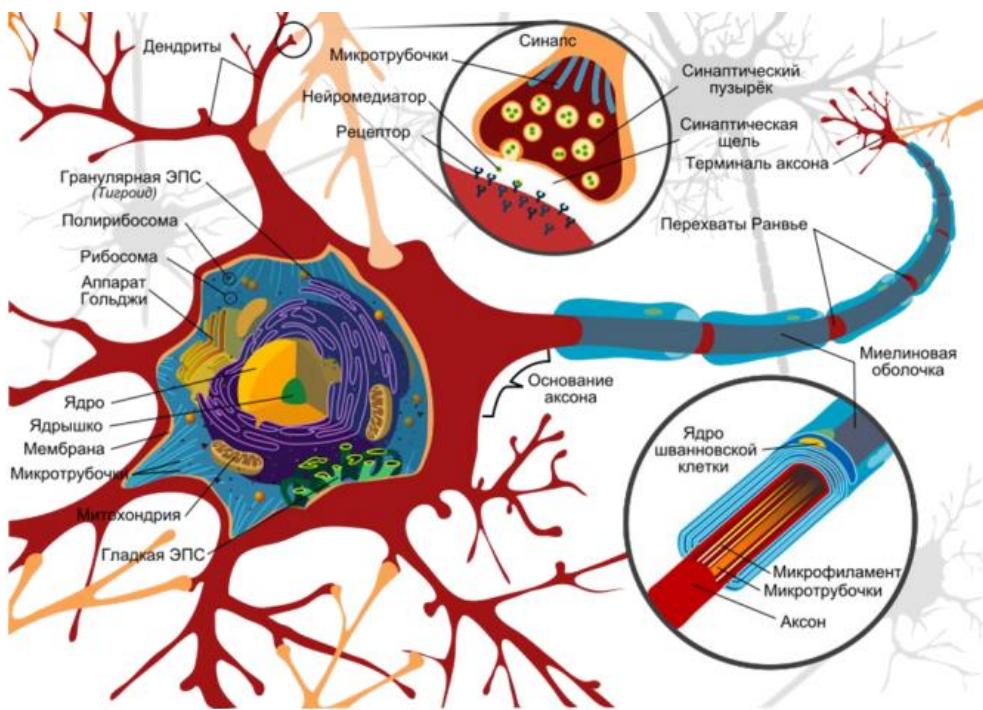
Лондон, Великобритания



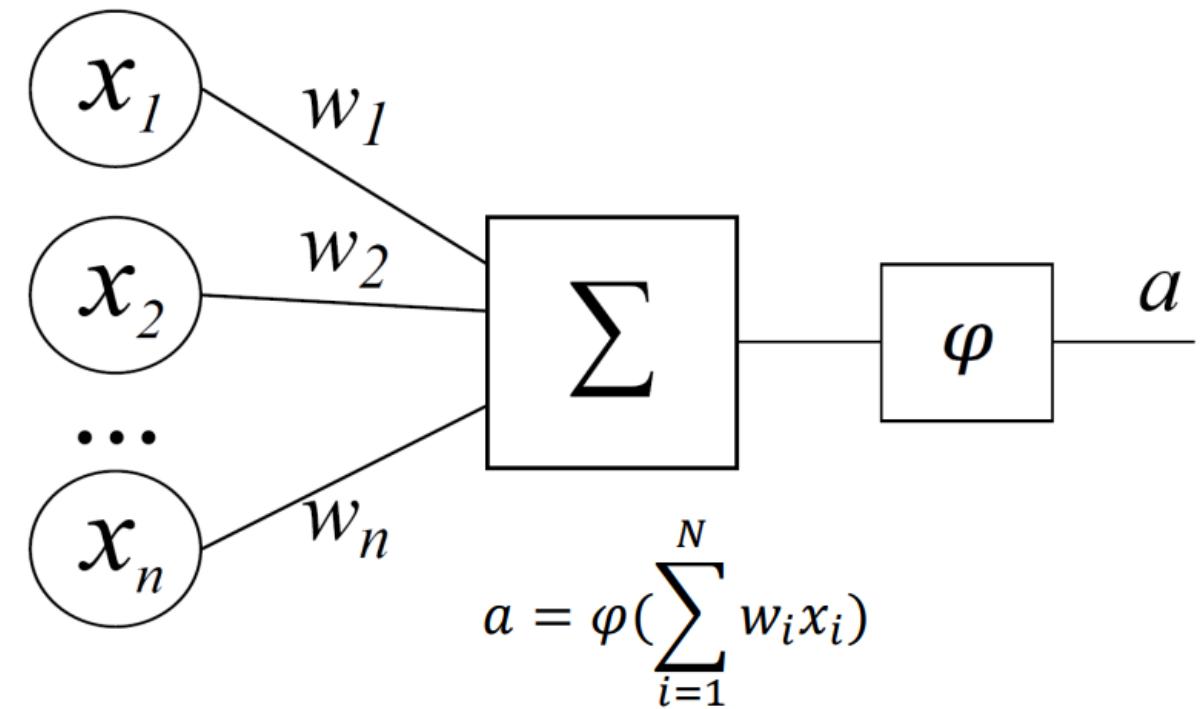
Москва,
Россия

Нейрон головного мозга

Нейрон головного мозга



Искусственный нейрон



Набор данных

X	Площадь квадрата (x_1)	Этаж квартиры (x_2)	Площадь кухни(x_3)	Количество комнат(x_4)	Стоимость квартиры (Y)
L	460	2	15	6	195
	230	7	9	4	130
	315	1	20	3	140
	178	3	25	4	80

- X = Входные данные (длина, ширина, высота, цвет, текст и так далее)

Каждый объект X характеризуется набором признаков x_1, x_2, \dots, x_n

- Y = Целевая переменная (Target)
- L = Количество обучающих данных
- (X, Y) = Обучающий набор данных
- $(x^{(i)}, y^{(i)})$ = i-го примера в вашем обучающем наборе данных

Типы признаков

Вещественные

- Бинарные $\in \{0,1\}$
- Числовые $\in \mathbb{R}$

Число лайков от пользователей



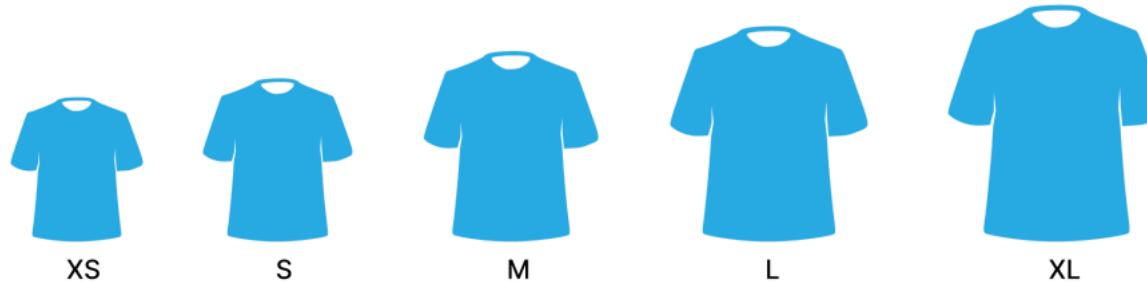
Температура



Типы признаков

Категориальные $\in \{c_1, c_2, \dots, c_n\}$

- Упорядоченные (ординальные) – для каждой пары возможных категорий можем сказать, какая больше, а какая меньше. Например, размер одежды



- Неупорядоченные (номинальные) – категории между собой несравнимы.



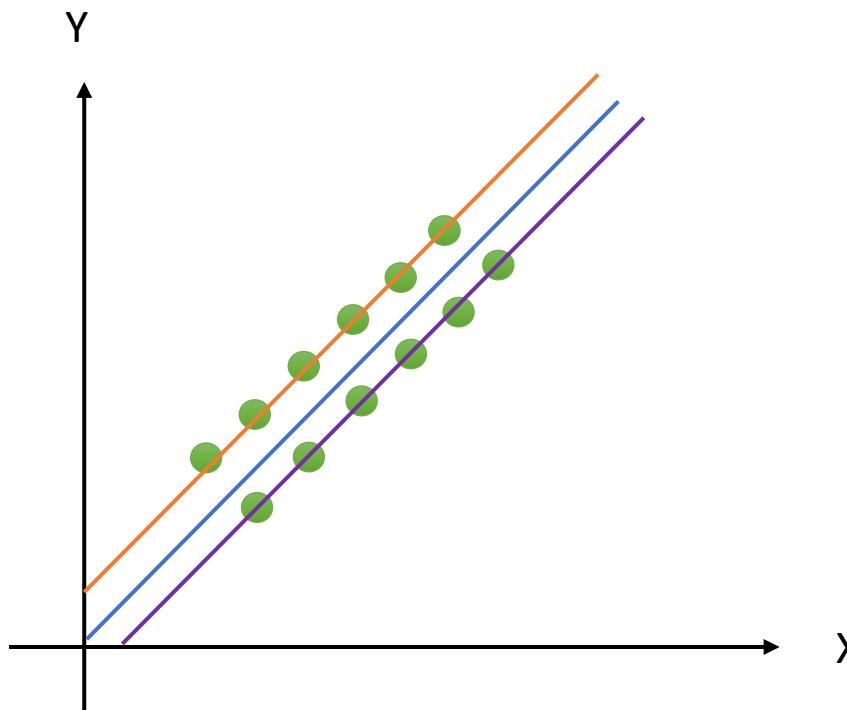
Регрессия (Regression)

$a = \mathcal{A}$ Семейство моделей

$a : \mathbb{X} \rightarrow \mathbb{Y}$

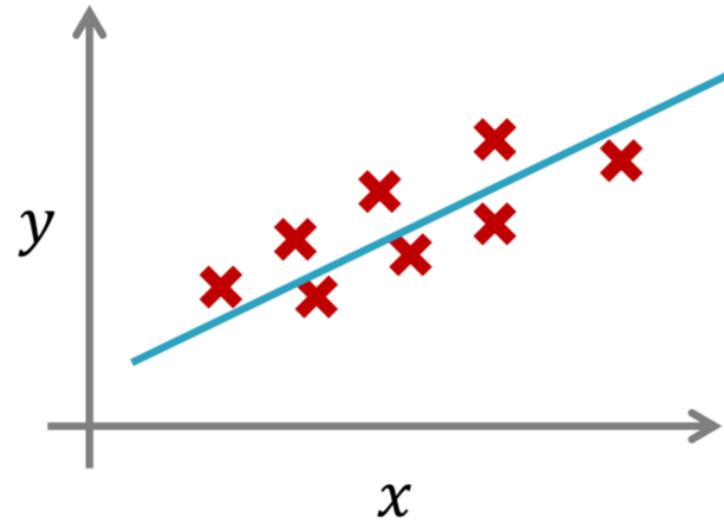
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

$$\mathcal{A} = \{a(x) = \theta_0 + \theta_1 x_1 + \dots + \theta_d x_d \mid \theta_0, \theta_1, \dots, \theta_d \in \mathbb{R}\}$$

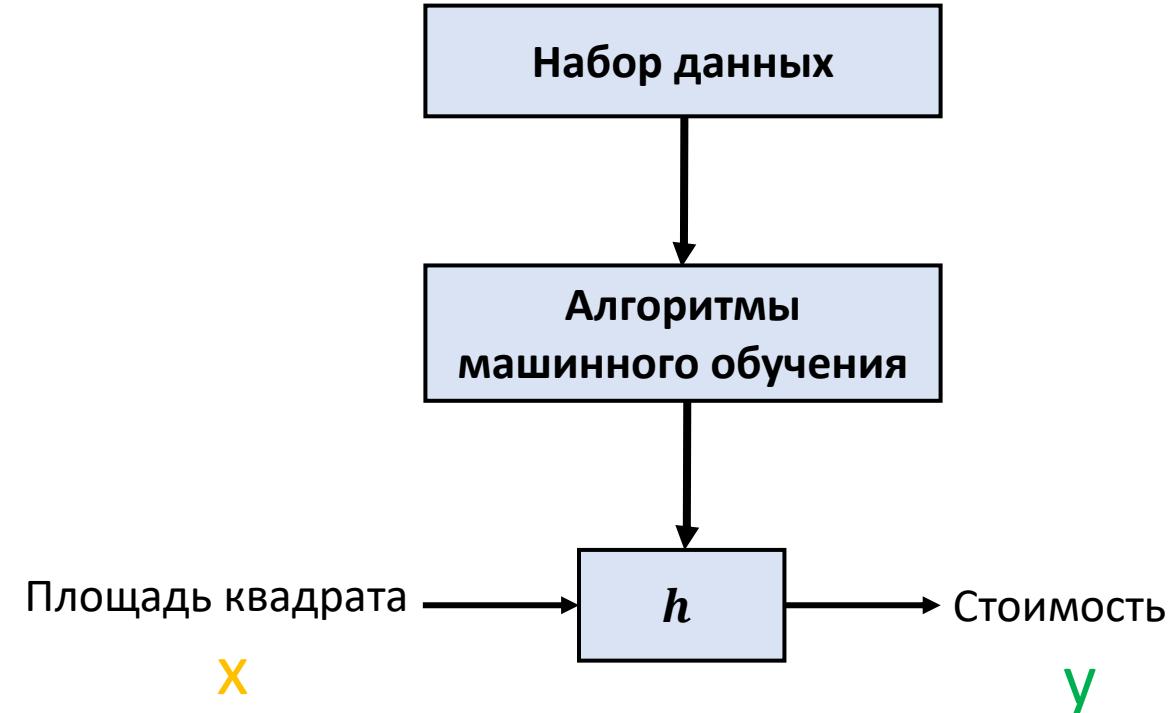


Регрессия (Regression)

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



Линейная регрессия с одной переменной



$$h : x \rightarrow y \in \mathbb{R}$$

Функция потерь (Lost function)

Набор данных:

Площадь квадрата (X)	Стоимость квартиры (Y)
460	195
230	130
315	140
178	80
....

Функция гипотезы:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

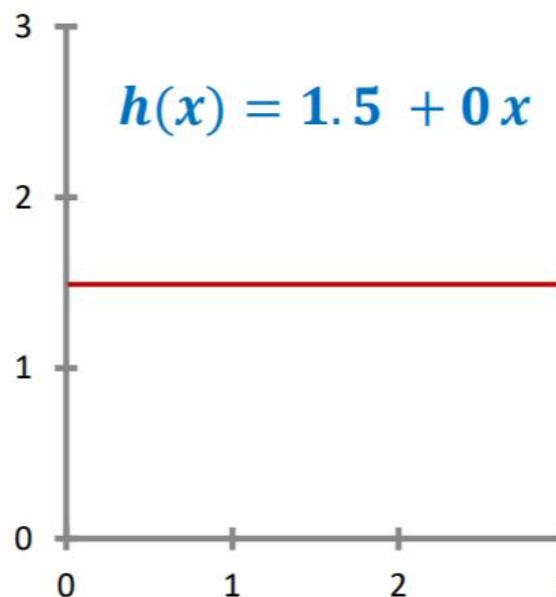
Параметры модели:

$$(\theta_0, \theta_1)$$

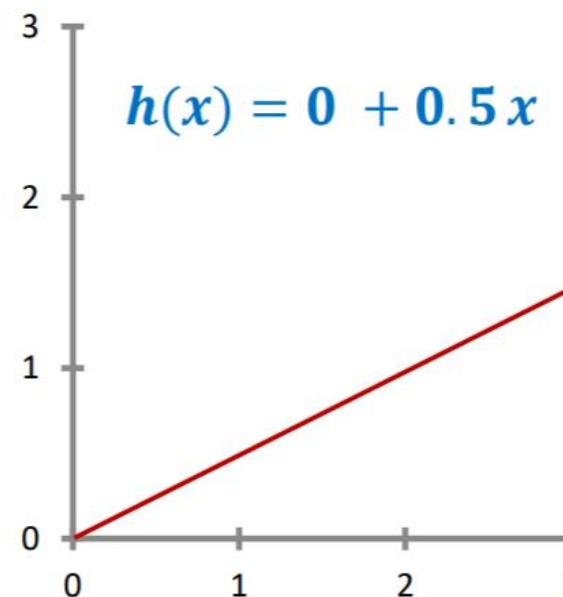
Цель обучения в машинном обучении заключается в нахождении оптимальных значений параметров θ , которые наилучшим образом соответствуют данным и позволяют функции гипотезы точно предсказывать или оценивать целевую переменную y на новых данных.

Функция потерь (Lost function)

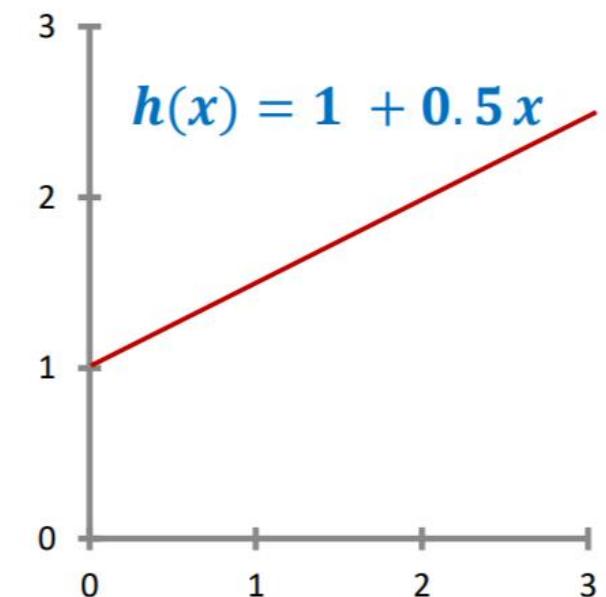
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



$$\begin{aligned}\theta_0 &= 1.5 \\ \theta_1 &= 0\end{aligned}$$



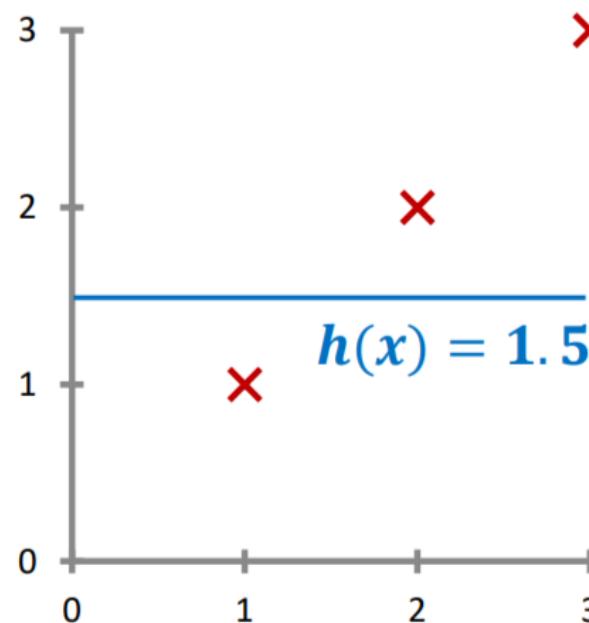
$$\begin{aligned}\theta_0 &= 0 \\ \theta_1 &= 0.5\end{aligned}$$



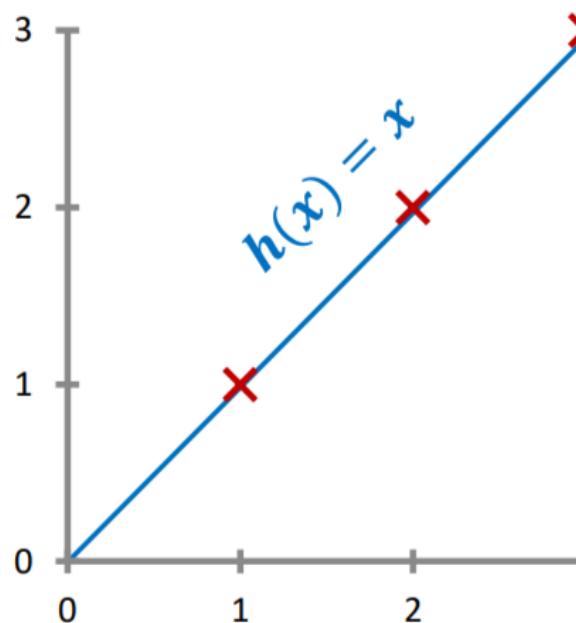
$$\begin{aligned}\theta_0 &= 1 \\ \theta_1 &= 0.5\end{aligned}$$

Функция потерь (Lost function)

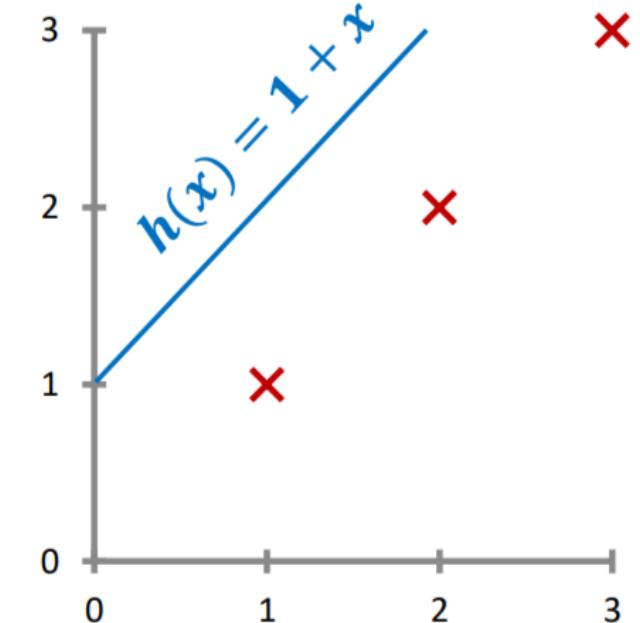
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



$$\begin{aligned}\theta_0 &= 1.5 \\ \theta_1 &= 0\end{aligned}$$



$$\begin{aligned}\theta_0 &= 0 \\ \theta_1 &= 1\end{aligned}$$



$$\begin{aligned}\theta_0 &= 1 \\ \theta_1 &= 1\end{aligned}$$

Функция потерь (Lost function)

Функция потерь: Среднеквадратичная ошибка ([Mean Squared Error](#))

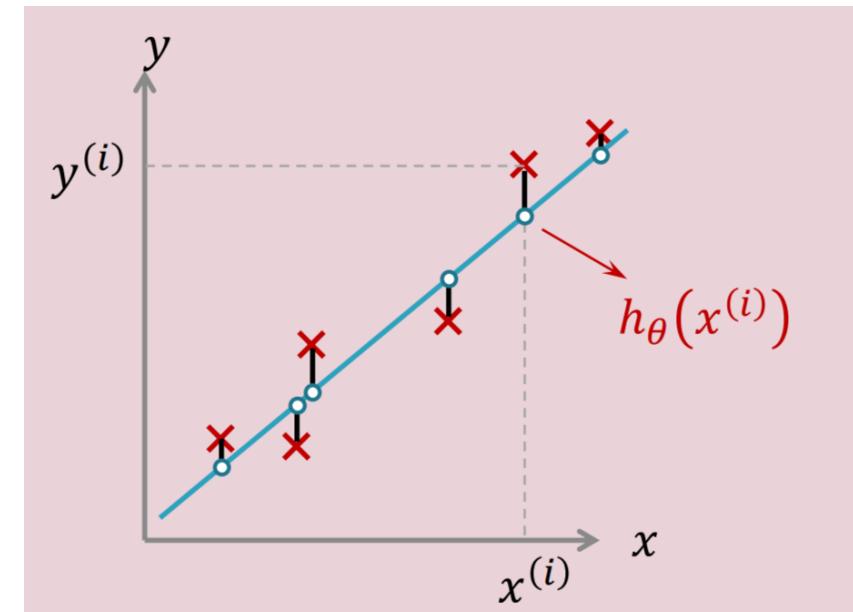
$$L : h_{\theta}(x^{(i)}), y^{(i)}$$

$$L : (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Функционал ошибки

$$J(\theta_0, \theta_1) = \frac{1}{l} \sum_{i=1}^l (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Цель: $\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$



$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Выбирая параметры θ так, чтобы функция гипотезы $h_{\theta}(x)$ приближалась к значениям y на обучающей выборке

Функция потерь (Lost function)

Функция потерь: Среднеквадратичная ошибка ([Mean Squared Error](#))

Площадь квадрата (x_1)	Этаж квартиры (x_2)	Площадь кухни(x_3)	Количество комнат(x_4)	Стоимость квартиры (Y)	$h_{\theta}(x)$	$(h_{\theta}(x^{(i)}) - y^{(i)})^2$
460	2	15	6	195	210	$(210 - 195)^2 = 225$
230	7	9	4	130	120	$(120 - 130)^2 = 100$
315	1	20	3	150	130	$(130 - 150)^2 = 400$
178	3	25	4	80	88	$(88 - 80)^2 = 64$
150	3	15	3	75	80	$(75 - 80)^2 = 25$

Функционал ошибки

$$J(\theta) = \frac{1}{l} \sum_{i=1}^l (h_{\theta}(x^{(i)}) - y^{(i)})^2 = \frac{225 + 100 + 400 + 64 + 25}{5} = 162.8$$

Функция потерь (Lost function)

Функция потерь: Корень из среднеквадратичной ошибки ([Root Mean Squared Error - RMSE](#))

$$L : h_{\theta}(x^{(i)}), y^{(i)}$$

$$L : \sqrt{(h_{\theta}(x^{(i)}) - y^{(i)})^2}$$

Функционал ошибки

$$J(\theta) = \frac{1}{l} \sum_{i=1}^l \sqrt{(h_{\theta}(x^{(i)}) - y^{(i)})^2}$$

$$RMSE = \sqrt{MSE}$$

Цель: $\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta)$

Функция потерь (Lost function)

Функция потерь: Корень из среднеквадратичной ошибки (**Root Mean Squared Error - RMSE**)

Площадь квадрата (x_1)	Этаж квартиры (x_2)	Площадь кухни(x_3)	Количество комнат(x_4)	Стоимость квартиры (Y)	$h_{\theta}(x)$	$(h_{\theta}(x^{(i)}) - y^{(i)})^2$
460	2	15	6	195	210	$(210 - 195)^2 = 225$
230	7	9	4	130	120	$(120 - 130)^2 = 100$
315	1	20	3	150	130	$(130 - 150)^2 = 400$
178	3	25	4	80	88	$(88 - 80)^2 = 64$
150	3	15	3	75	80	$(75 - 80)^2 = 25$

Функционал ошибки

$$J(\theta) = \frac{1}{l} \sum_{i=1}^l \sqrt{(h_{\theta}(x^{(i)}) - y^{(i)})^2} = \sqrt{\frac{225 + 100 + 400 + 64 + 25}{5}} = 12.79$$

Функция потерь (Lost function)

Коэффициент детерминации (R^2)

$$L : h_{\theta}(x^{(i)}), y^{(i)}$$

$$R^2 = 1 - \frac{\sum_{i=1}^l (h_{\theta}(x^{(i)}) - y^{(i)})^2}{\sum_{i=1}^l (y_i - \bar{y})^2}$$

Где $\bar{y} = \frac{1}{l} \sum_{i=1}^l y_i$ Среднее значение целевой переменной

R^2 Это нормированная среднеквадратичная ошибка. Если она близка к единице, то модель хорошо объясняет данные, если она близка к нулю, то прогноз не очень хорошо объясняет данные.

Функция потерь (Lost function)

Функция потерь: Средняя абсолютная ошибка ([Mean Absolute Error - MAE](#))

$$L : h_{\theta}(x^{(i)}), y^{(i)}$$

$$L : |h_{\theta}(x^{(i)}) - y|$$

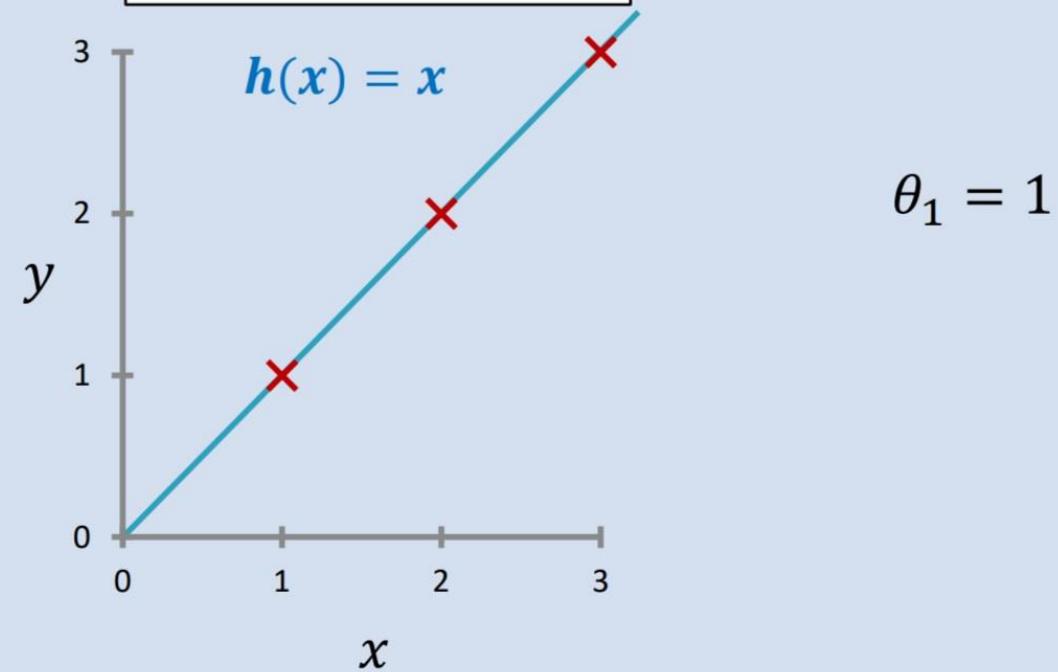
Функционал ошибки

$$J(\theta) = \frac{1}{l} \sum_{i=1}^l |h_{\theta}(x^{(i)}) - y_i|$$

y	h_{θ}	$(h_{\theta}(x^{(i)}) - y^{(i)})^2$	$ h_{\theta}(x^{(i)}) - y $
1	2	1	1
1000	2	996004	998
1	1	0	0
1000	3	994009	897

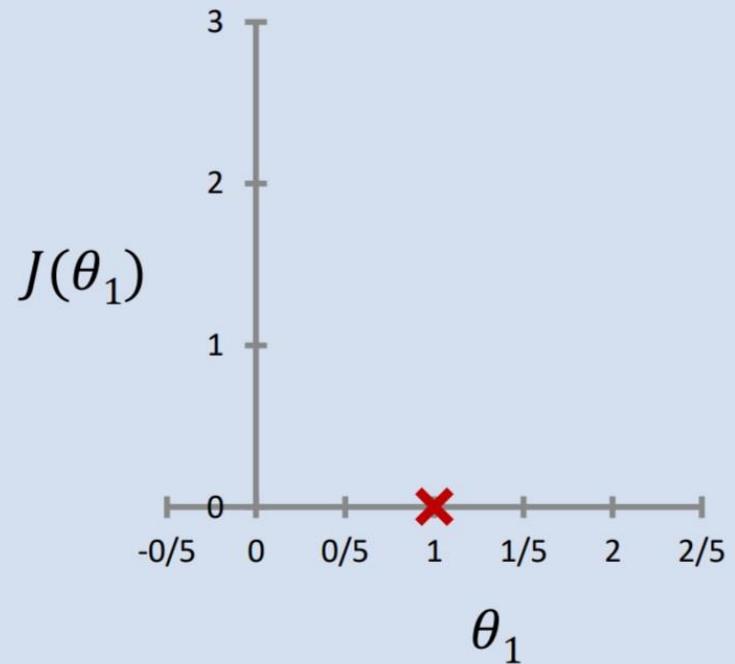
Функция потерь (Lost function)

$$h_{\theta}(x) = \theta_1 x$$



$$\theta_1 = 1$$

$$J(\theta_1)$$

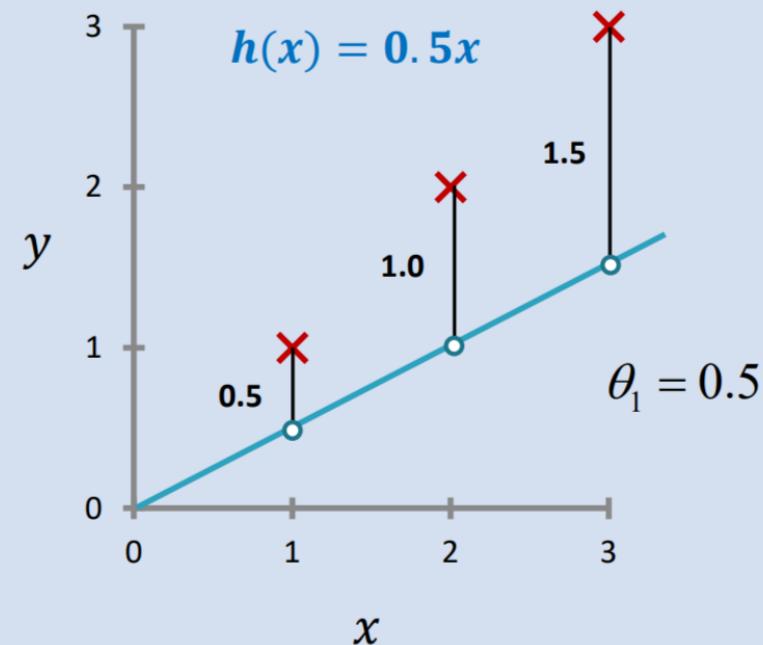


$$\begin{aligned} J(\theta_0, \theta_1) &= \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \\ &= \frac{1}{2} \sum_{i=1}^m (x^{(i)} - y^{(i)})^2 = \frac{1}{2}(0^2 + 0^2 + 0^2) = 0 \end{aligned}$$

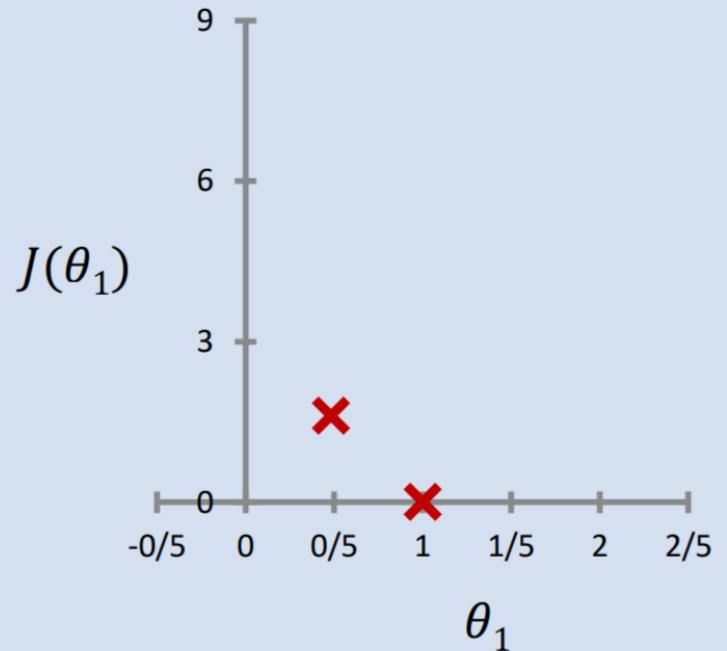
$$J(1) = 0$$

Функция потерь (Lost function)

$$h_{\theta}(x) = \theta_1 x$$



$$J(\theta_1)$$

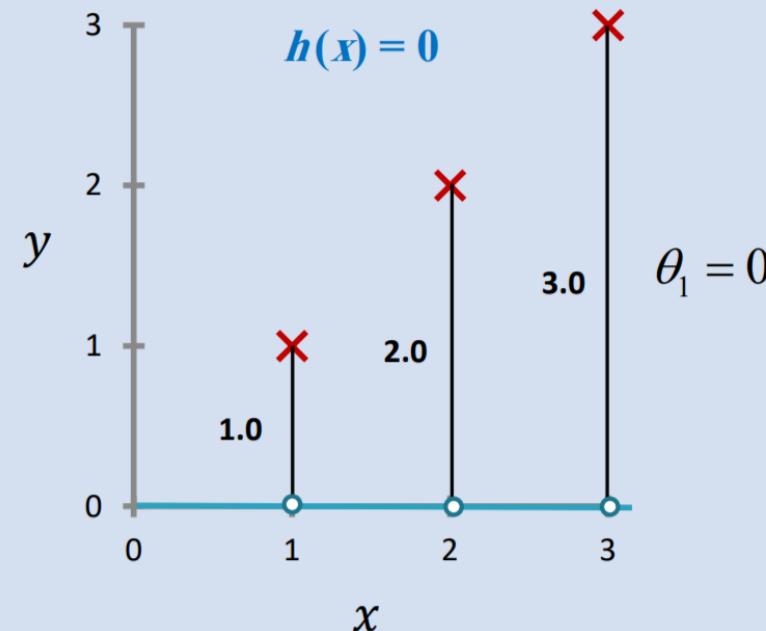


$$J(0.5) = \frac{1}{2}(0.5^2 + 1.0^2 + 1.5^2) = \frac{1}{2}(3.5) = 1.75$$

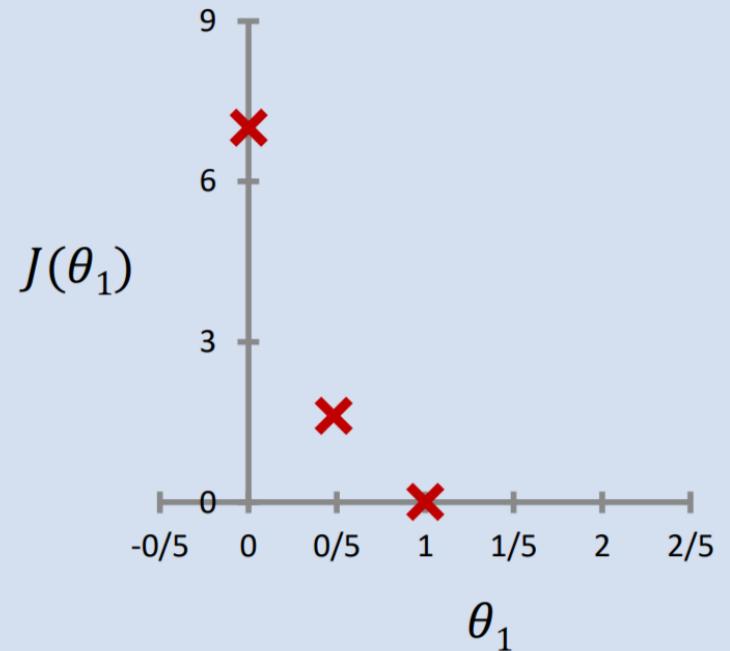
$$J(0.5) = 1.75$$

Функция потерь (Lost function)

$$h_{\theta}(x) = \theta_1 x$$



$$J(\theta_1)$$

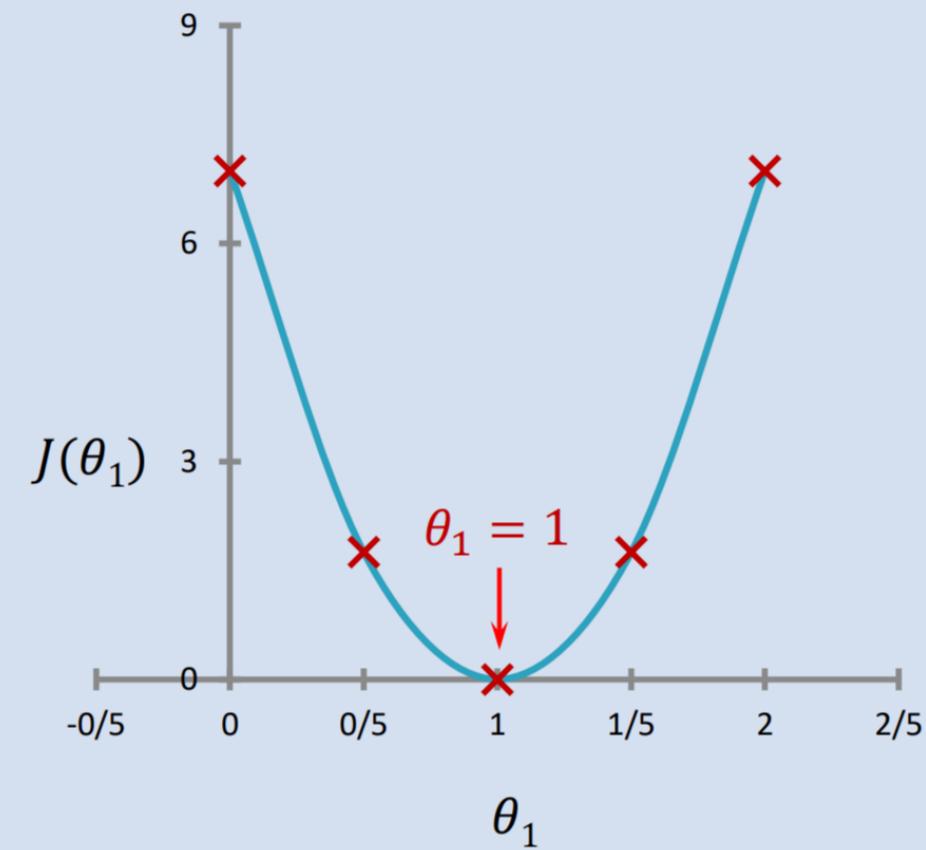


$$J(0) = \frac{1}{2}(1.0^2 + 2.0^2 + 3.0^2) = \frac{1}{2}(14) = 7.0$$

$$J(0) = 7.0$$

Функция потерь (Lost function)

minimize $J(\theta_1)$



Регрессия (Regression)

Функция гипотезы:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Параметры модели:

$$(\theta_0, \theta_1)$$

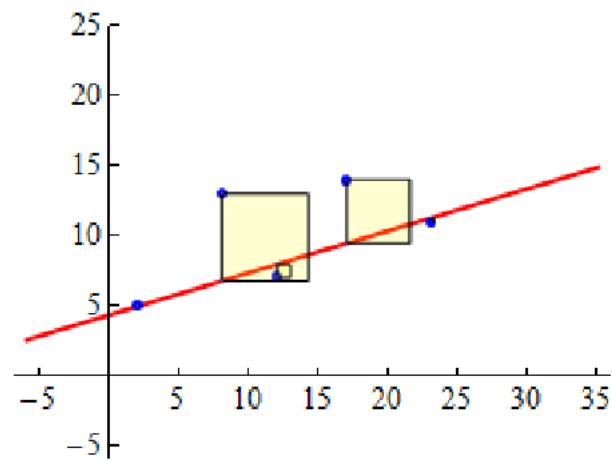
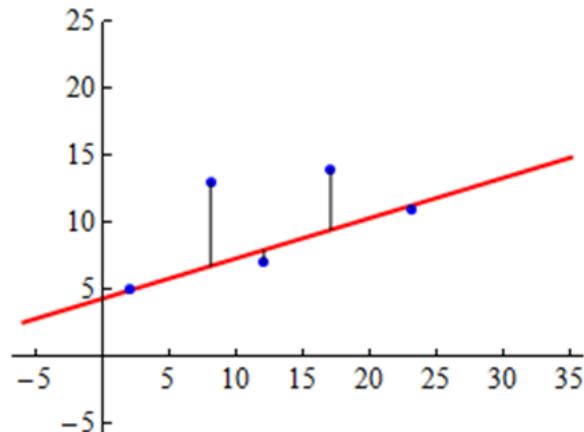
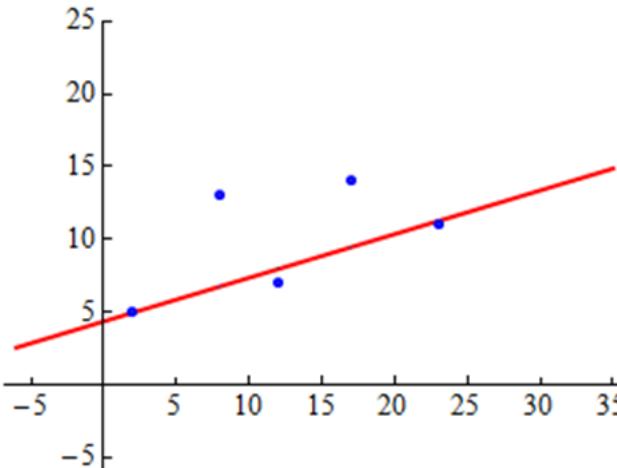
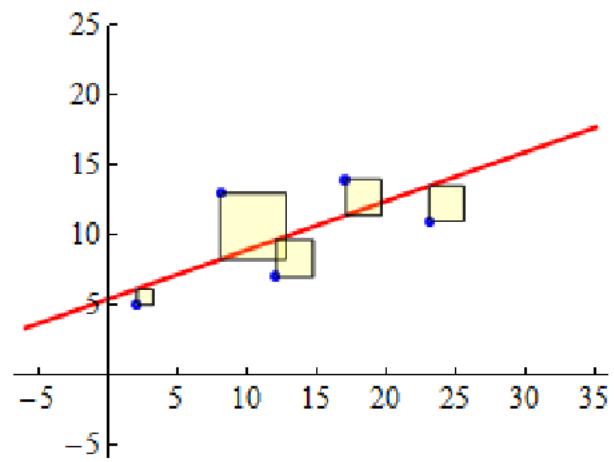
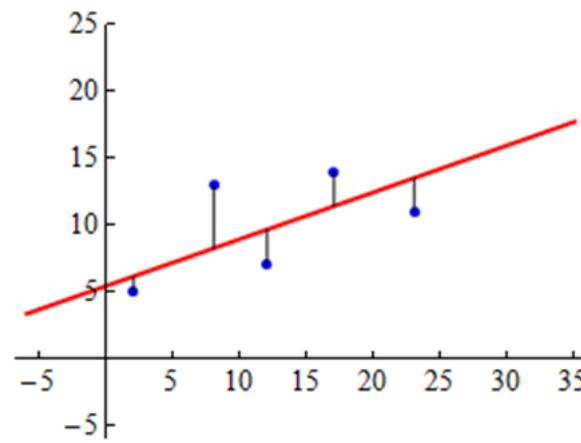
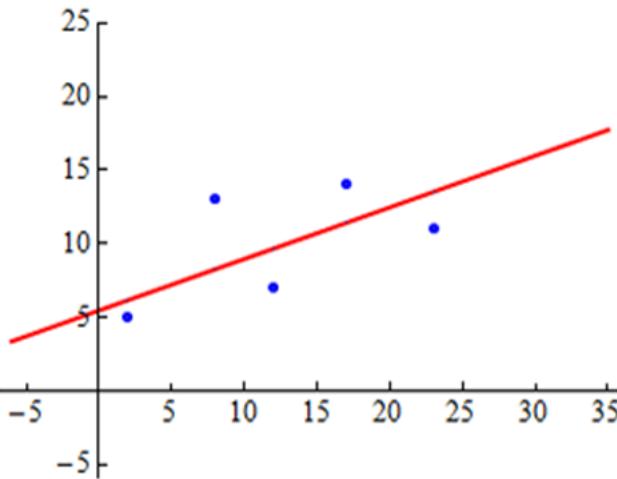
Функция потерь:

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^n (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Цель:

$$\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$$

Метод наименьших квадратов (Least squares)



Метод наименьших квадратов (Least squares)

Функция гипотезы:

$$h_{\theta}(x_i) = \theta_0 + \theta_1 x_i$$

Функция потерь:

$$e_i = y_i - h_{\theta}(x_i) = y_i - \theta_0 + \theta_1 x_i$$

Отклонения значений функции $h_{\theta}(x_i)$ от фактических значений y_i

$$e_1^2 + e_2^2 + \dots + e_n^2 = \sum_{i=1}^n e_i^2$$

Идея метода наименьших квадратов найти параметры θ_0 и θ_1 таким образом, чтобы сумма квадратов была наименьшей

$$\sum_{i=1}^n e_i^2 \rightarrow \min_{\theta_0, \theta_1}$$

Метод наименьших квадратов (Least squares)

Функционал ошибки

$$J = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \theta_0 + \theta_1 x_i)^2 \rightarrow \min_{\theta_0, \theta_1}$$

$$\begin{cases} \frac{\partial J}{\partial \theta_0} = 0 \\ \frac{\partial J}{\partial \theta_1} = 0 \end{cases}$$



$$\begin{cases} \sum_{i=1}^n 2(y_i - \theta_0 + \theta_1 x_i) \cdot (1) = 0 \\ \sum_{i=1}^n 2(y_i - \theta_0 + \theta_1 x_i) \cdot (x_i) = 0 \end{cases}$$

Метод наименьших квадратов (Least squares)

$$\theta_0 = \frac{\sum_{i=1}^n y_i - \theta_1 \sum_{i=1}^n x_i}{n}$$

$$\begin{cases} 2 \left(\theta_1 \sum_{i=1}^n x_i + \theta_0 n - \sum_{i=1}^n y_i \right) = 0 \\ 2 \left(\theta_1 \sum_{i=1}^n x_i^2 + \theta_0 \sum_{i=1}^n x_i - \sum_{i=1}^n x_i y_i \right) = 0 \end{cases}$$



$$\theta_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

Нормальные уравнения (Normal equation)

$$a(x) = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n = \theta_0 + \sum_{i=1}^n \theta_i x_i$$

$$a(x) = \theta_0 + \langle \theta, x \rangle$$

$$a(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n, \quad x_0 = 1$$

$$a(x) = \langle \theta, x \rangle$$

Функцию потерь можно представить в виде:

$$\|x\theta - y\|^2$$

$\|\cdot\|$ Стандартная евклидова норма в пространстве \mathbb{R}^d

$x\theta$ Скалярное произведение $\langle \theta, x \rangle$

Цель:

$$\|x\theta - y\|^2 \rightarrow \min_{\theta}$$

Нормальные уравнения (Normal equation)

Цель:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\langle \theta, x_i \rangle - y_i)^2 \rightarrow \min_{\theta}$$

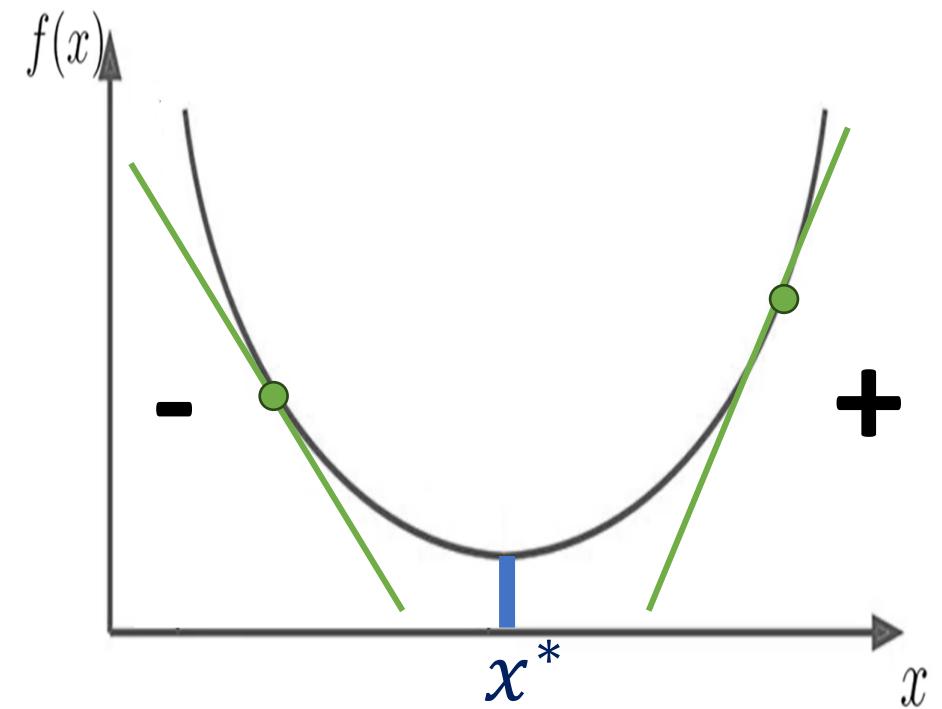
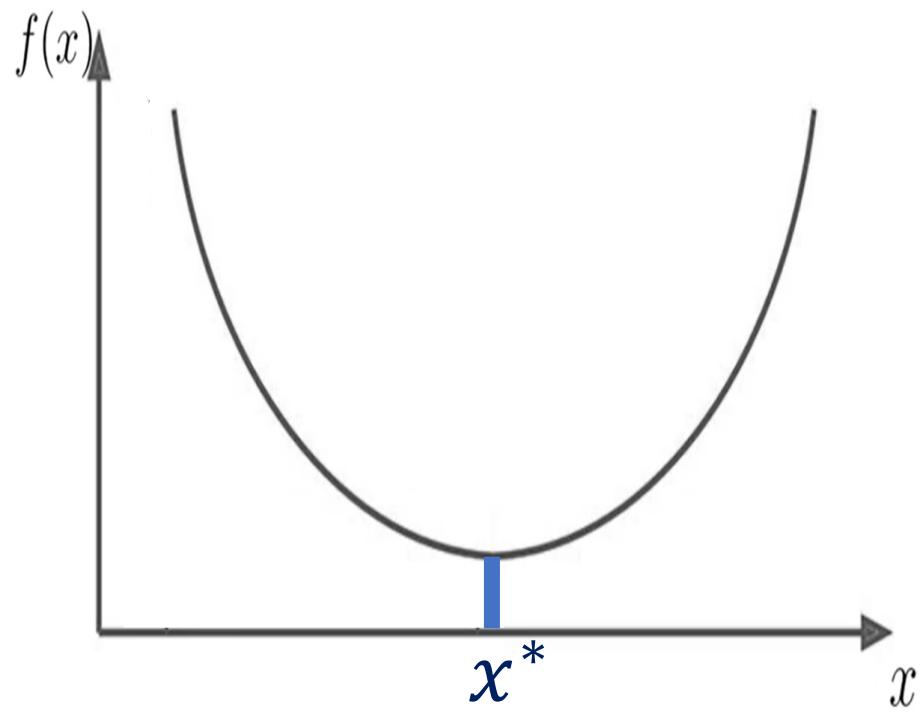
$$X = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

$$\nabla MSE = 0$$



$$\theta = (X^T X)^{-1} X^T y$$

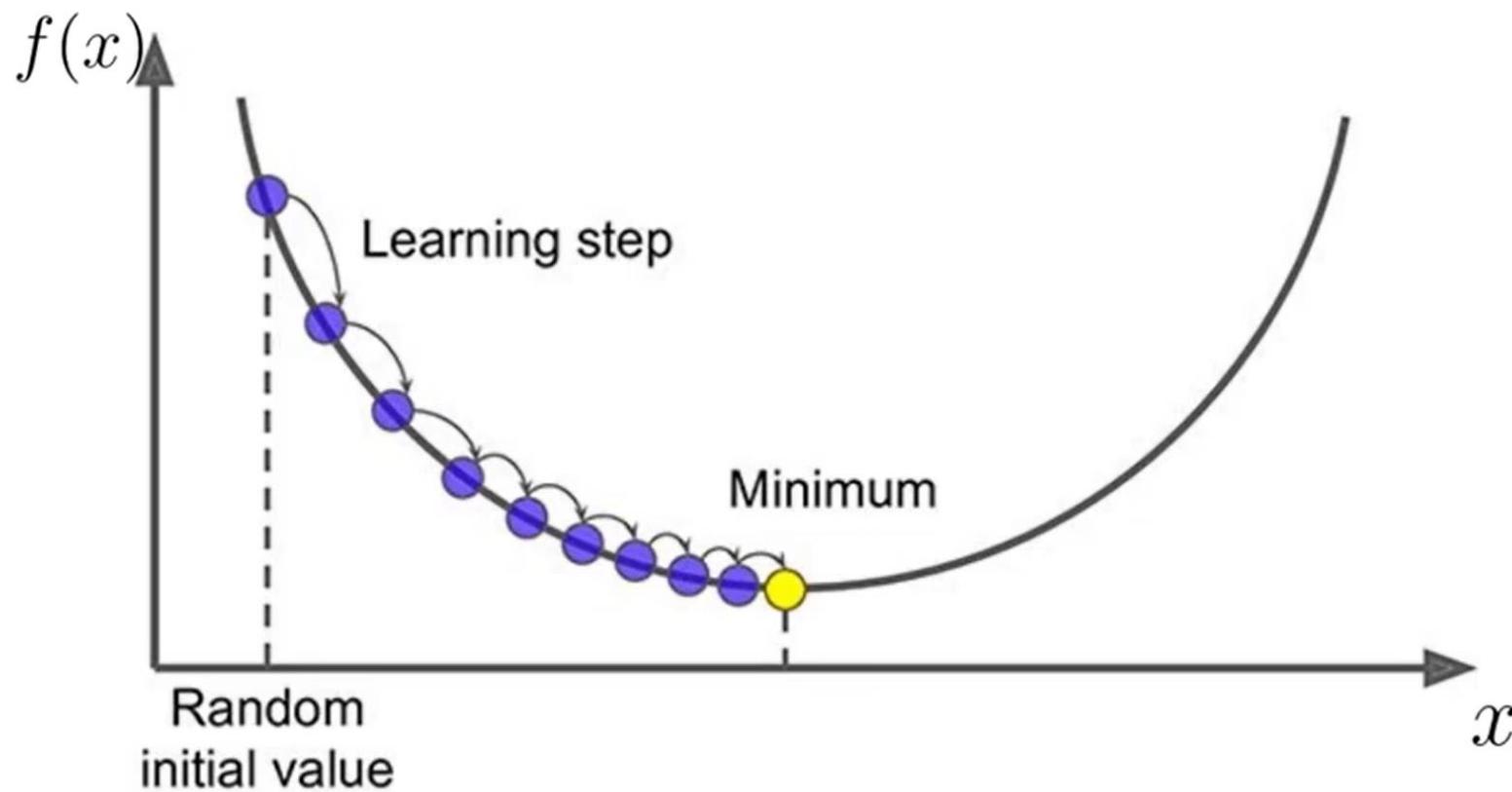
Эвристика градиентного спуска



$$x_{n+1} = x_n - \frac{\partial f(x)}{dx} , n = 0, 1, 2, 3, \dots$$

Градиентный спуск (Gradient descent)

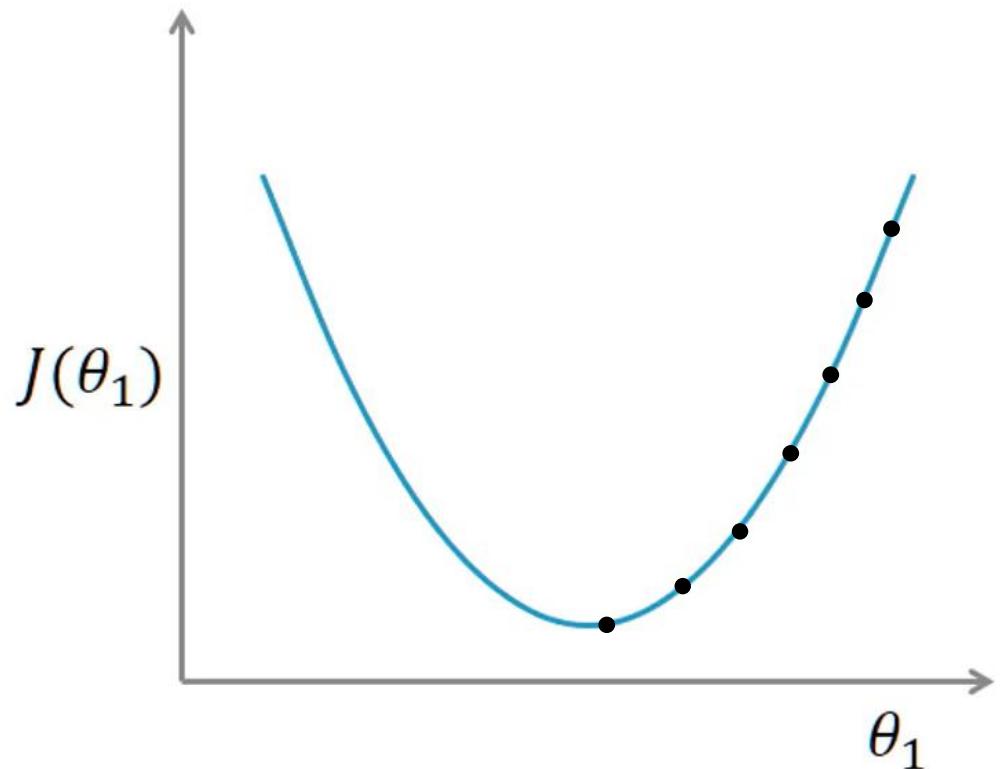
$$x_{n+1} = x_n - \alpha \frac{\partial f(x)}{\partial x}, n = 0, 1, 2, 3, \dots$$



α - шаг сходимости
Learning Rate

Градиентный спуск (Gradient descent)

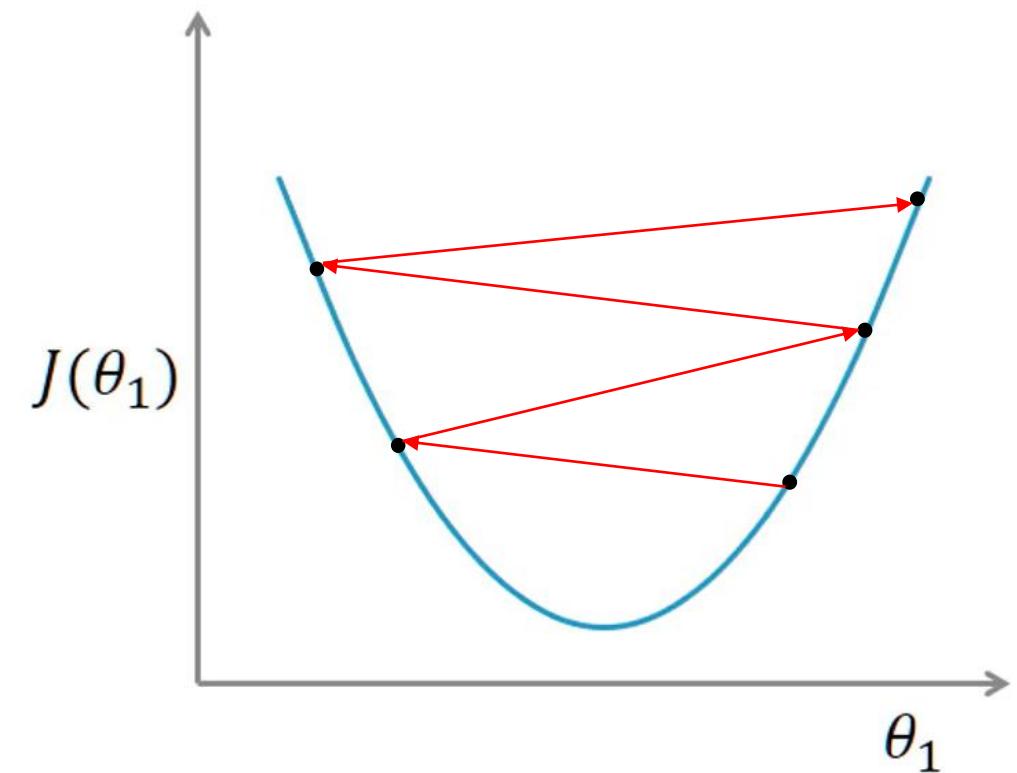
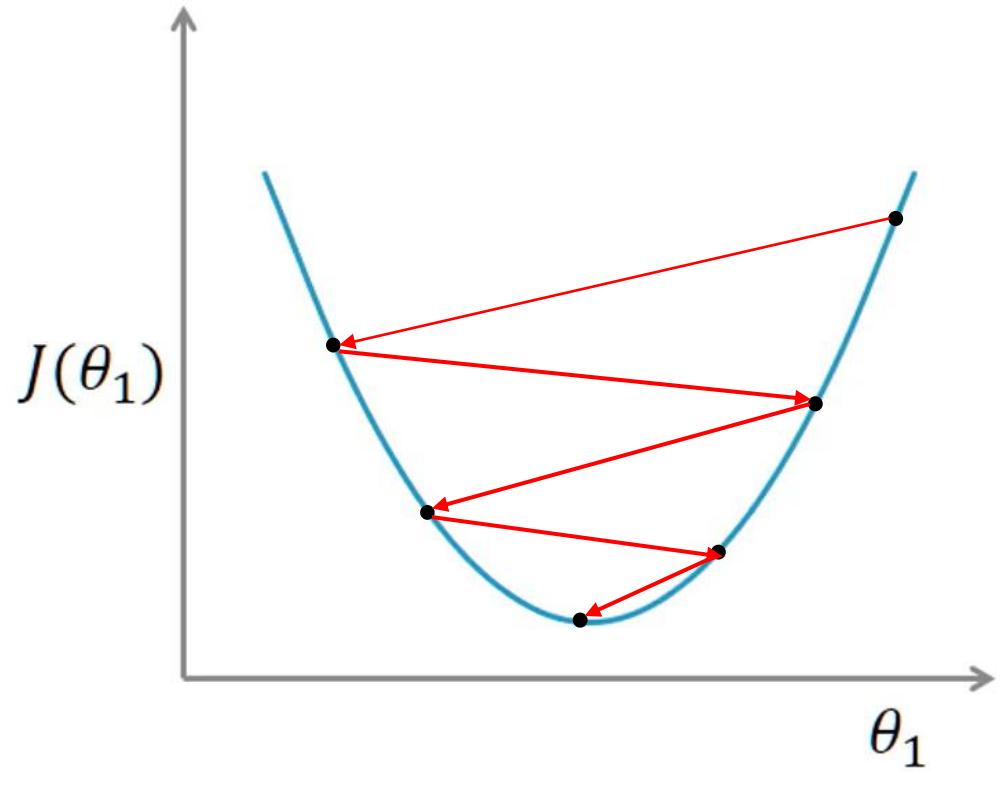
Если размер шага (learning rate) слишком мал, градиентный спуск сходится слишком медленно.



$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

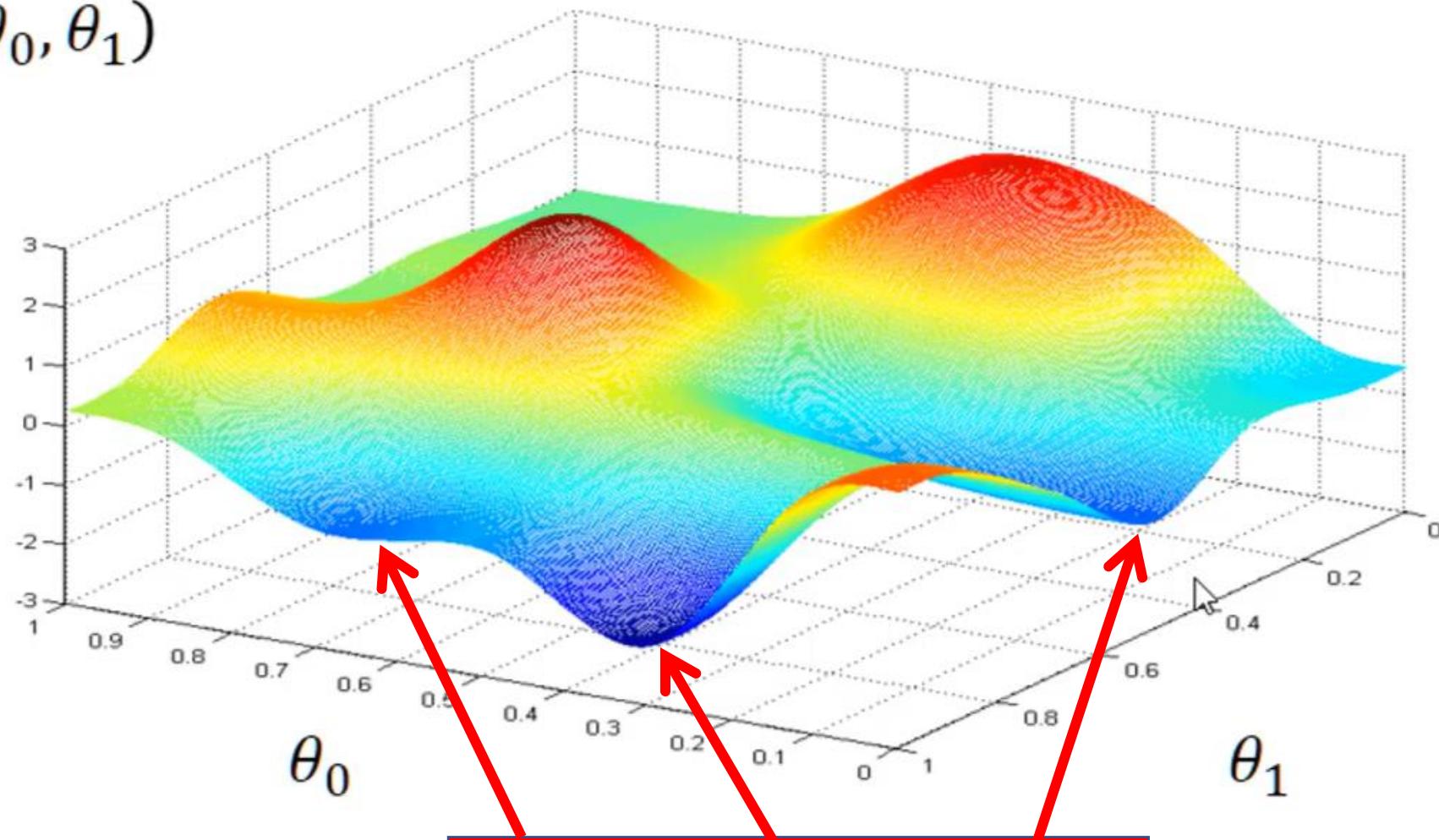
Градиентный спуск (Gradient descent)

Если размер шага (learning rate) слишком большой, градиентный спуск сходится слишком медленно или вообще не сходится.



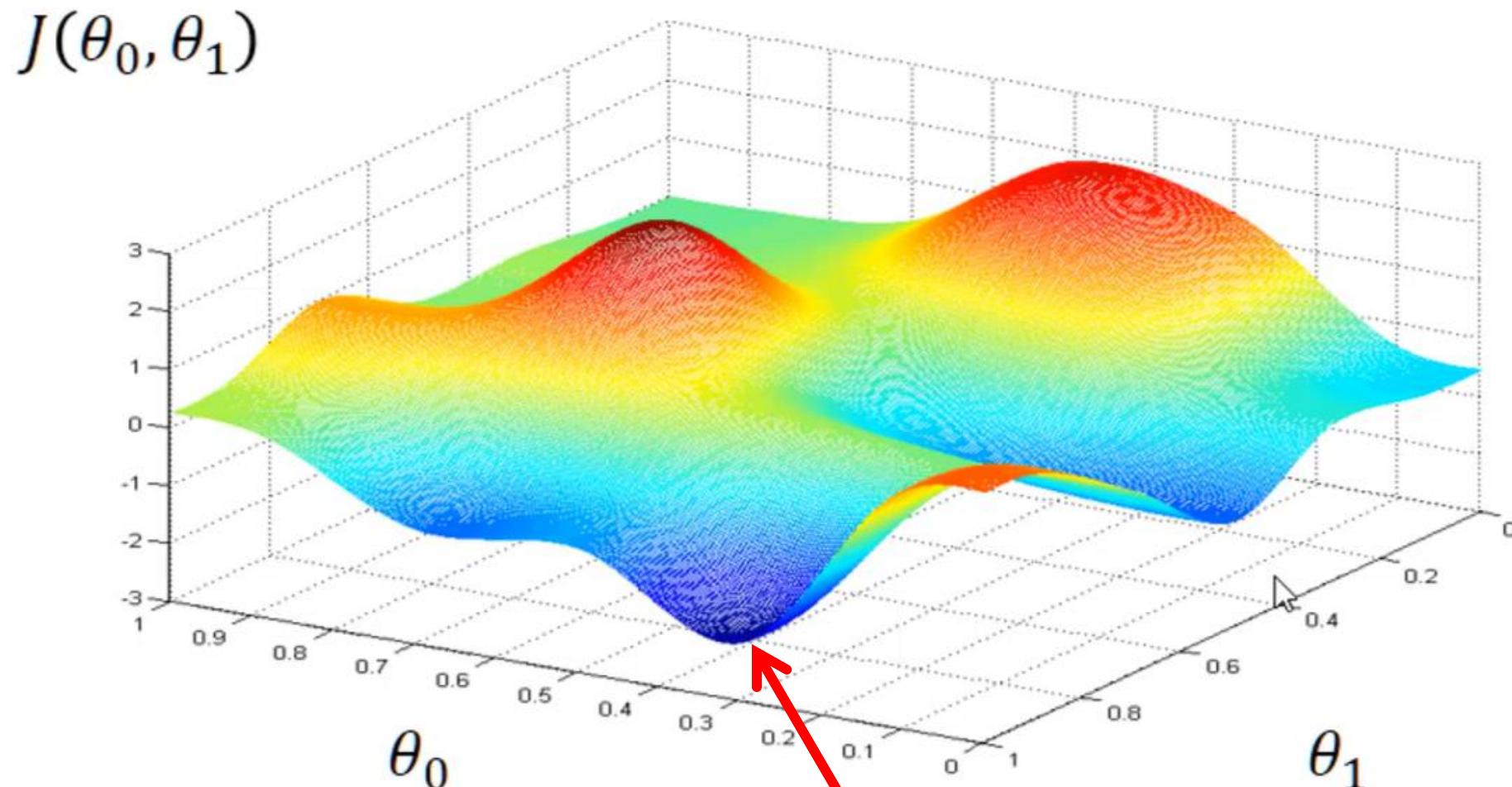
Градиентный спуск (Gradient descent)

$$J(\theta_0, \theta_1)$$



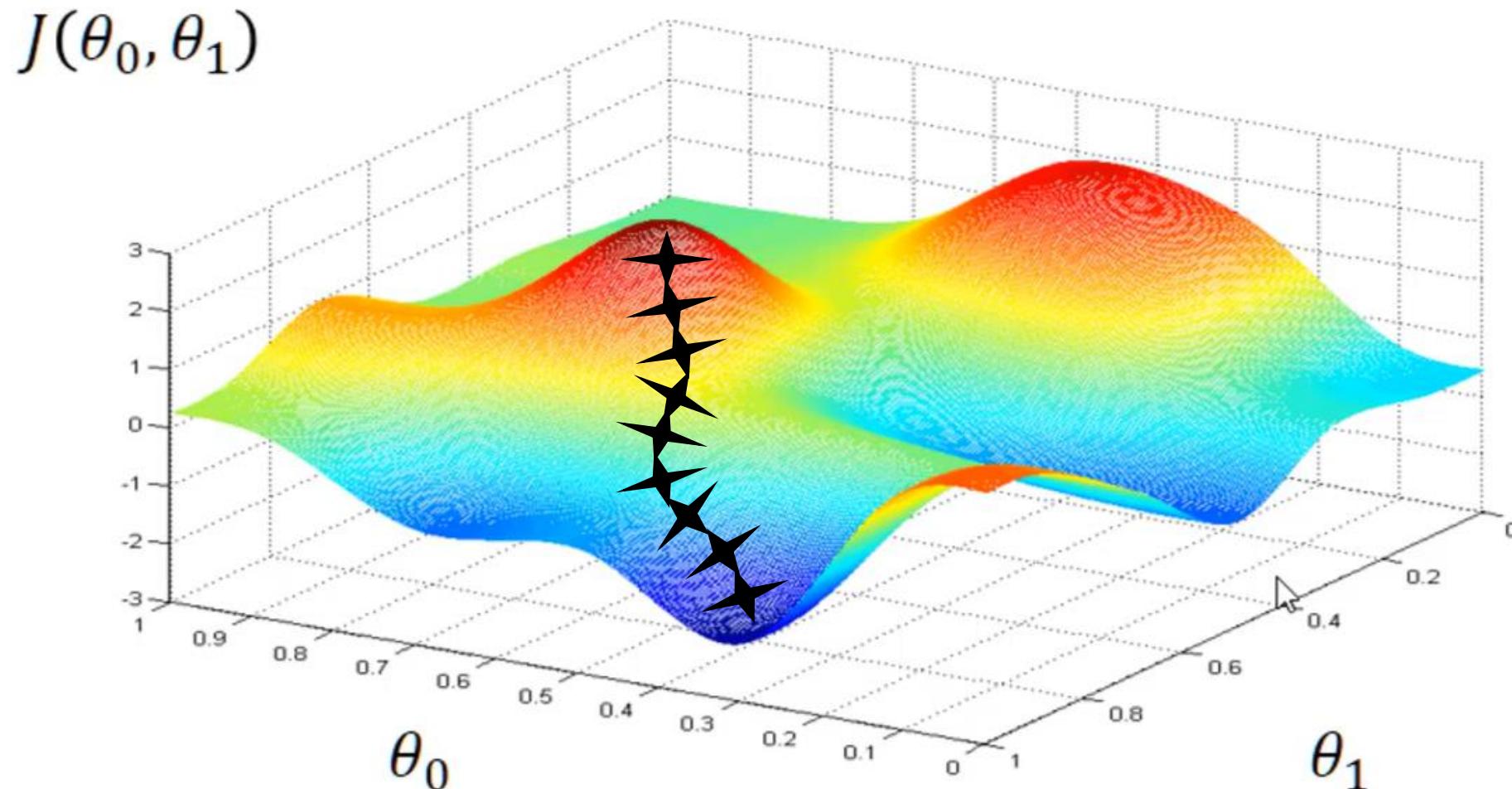
Локальный минимум

Градиентный спуск (Gradient descent)

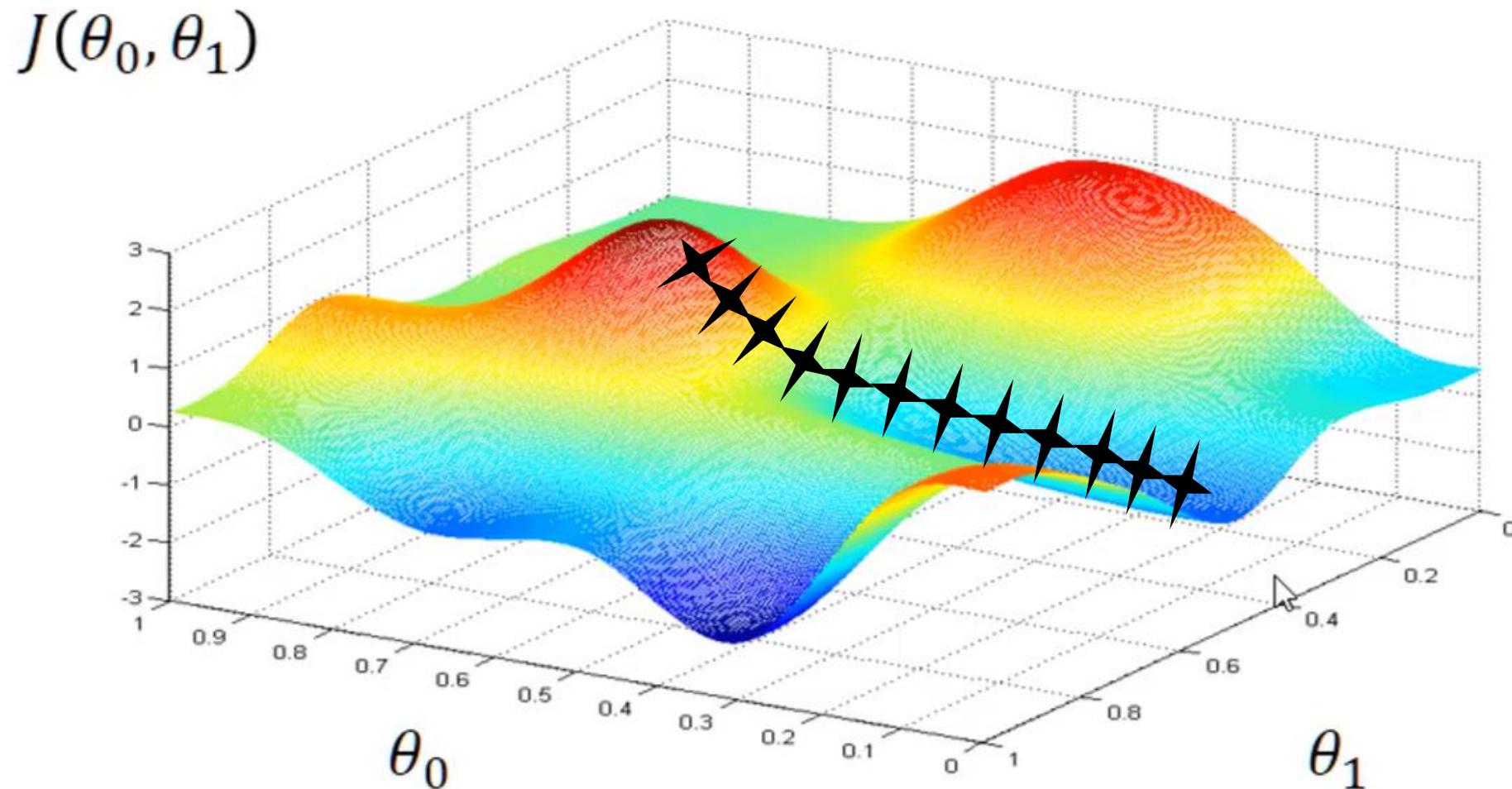


Глобальный минимум

Градиентный спуск (Gradient descent)



Градиентный спуск (Gradient descent)



Формализация

$$f(\theta) \rightarrow \min_{\theta}$$

Чтобы применять метод градиентного спуска, необходимо вычислять градиент функции в точке:

$$\nabla f(\theta_1, \theta_2, \dots, \theta_n) = \left(\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}, \dots, \frac{\partial f}{\partial \theta_n} \right)$$

На каждом шаге будем менять все переменные, от которых зависит функция:

$$\theta_1 = \theta_1 - \alpha \frac{\partial f}{\partial \theta_1}, \dots, \theta_n = \theta_n - \alpha \frac{\partial f}{\partial \theta_n}$$

Повторяем пока изменение не будет достаточно маленьким или пройдет много шагов

Градиентный спуск в машинном обучении

$$\begin{cases} J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^n (\mathbf{h}_\theta(x^{(i)}) - y^{(i)})^2 \\ \mathbf{h}_\theta(x) = \theta_0 + \theta_1 x_i \end{cases} \rightarrow \min_{\theta_0, \theta_1} J$$

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^n \mathbf{u}^2 \rightarrow \begin{cases} \frac{\partial J}{\partial \theta_0} = ? \\ \frac{\partial J}{\partial \theta_1} = ? \end{cases}$$

$$\frac{\partial J}{\partial u} = \frac{1}{n} \sum_{i=1}^n 2\mathbf{u} = \frac{2}{n} \sum_{i=1}^n u = \frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J}{\partial \theta_0} = \frac{\partial J}{\partial u} \cdot \frac{\partial u}{\partial h_\theta} \cdot \frac{\partial h_\theta}{\partial \theta_0} = \frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)}) \cdot 1.1$$

$$\frac{\partial u}{\partial h_\theta} = \frac{\partial h_\theta(x^{(i)}) - y^{(i)}}{\partial h_\theta} = 1$$

$$\frac{\partial h_\theta}{\partial \theta_0} = \frac{\partial \theta_0 + \theta_1 x_i}{\partial \theta_0} = 1$$

$$\frac{\partial h_\theta}{\partial \theta_1} = \frac{\partial \theta_0 + \theta_1 x_i}{\partial \theta_1} = x_i$$



$$\frac{\partial J}{\partial \theta_1} = \frac{\partial J}{\partial u} \cdot \frac{\partial u}{\partial h_\theta} \cdot \frac{\partial h_\theta}{\partial \theta_1} = \frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)}) \cdot 1. x_i$$

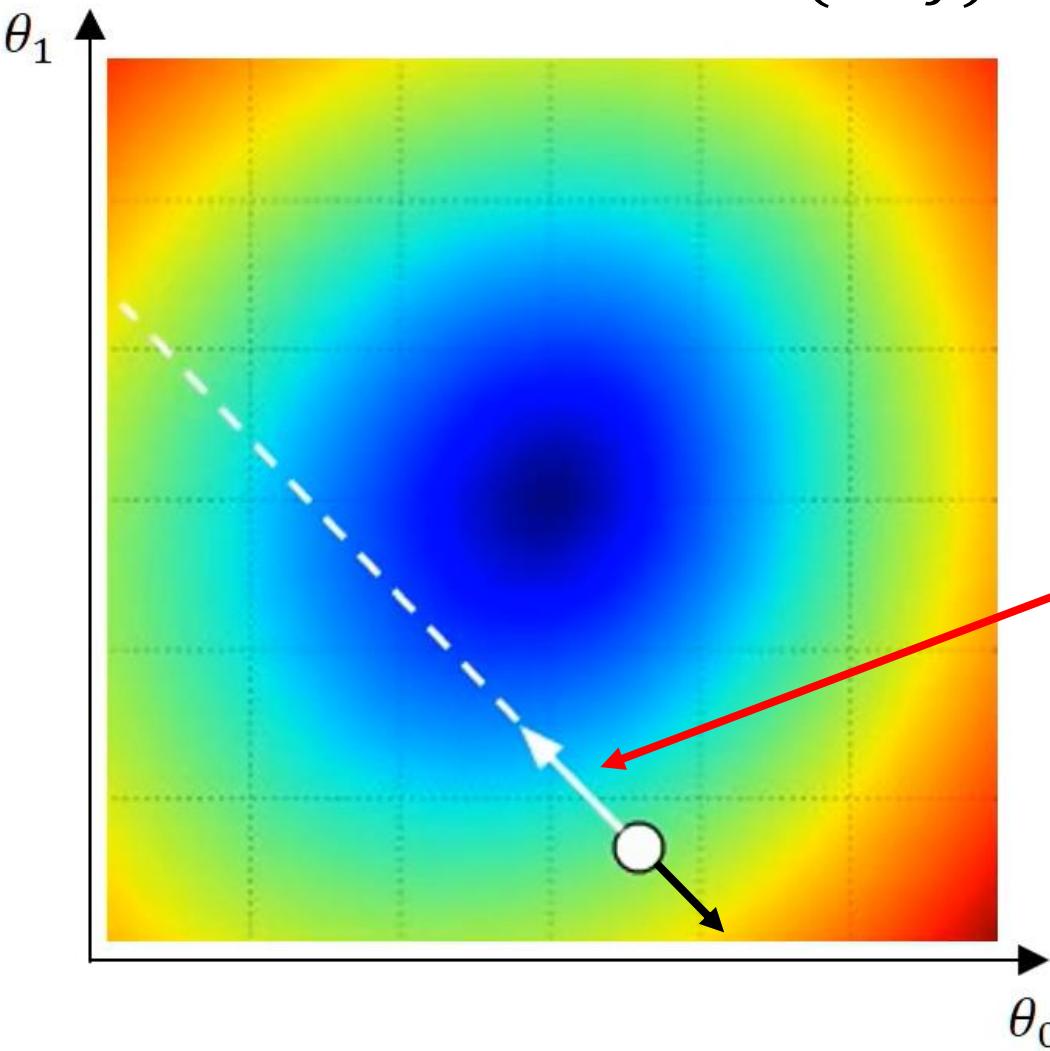
Градиентный спуск (Gradient descent)

$$J(\theta_0, \theta_1) = J(\theta)$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

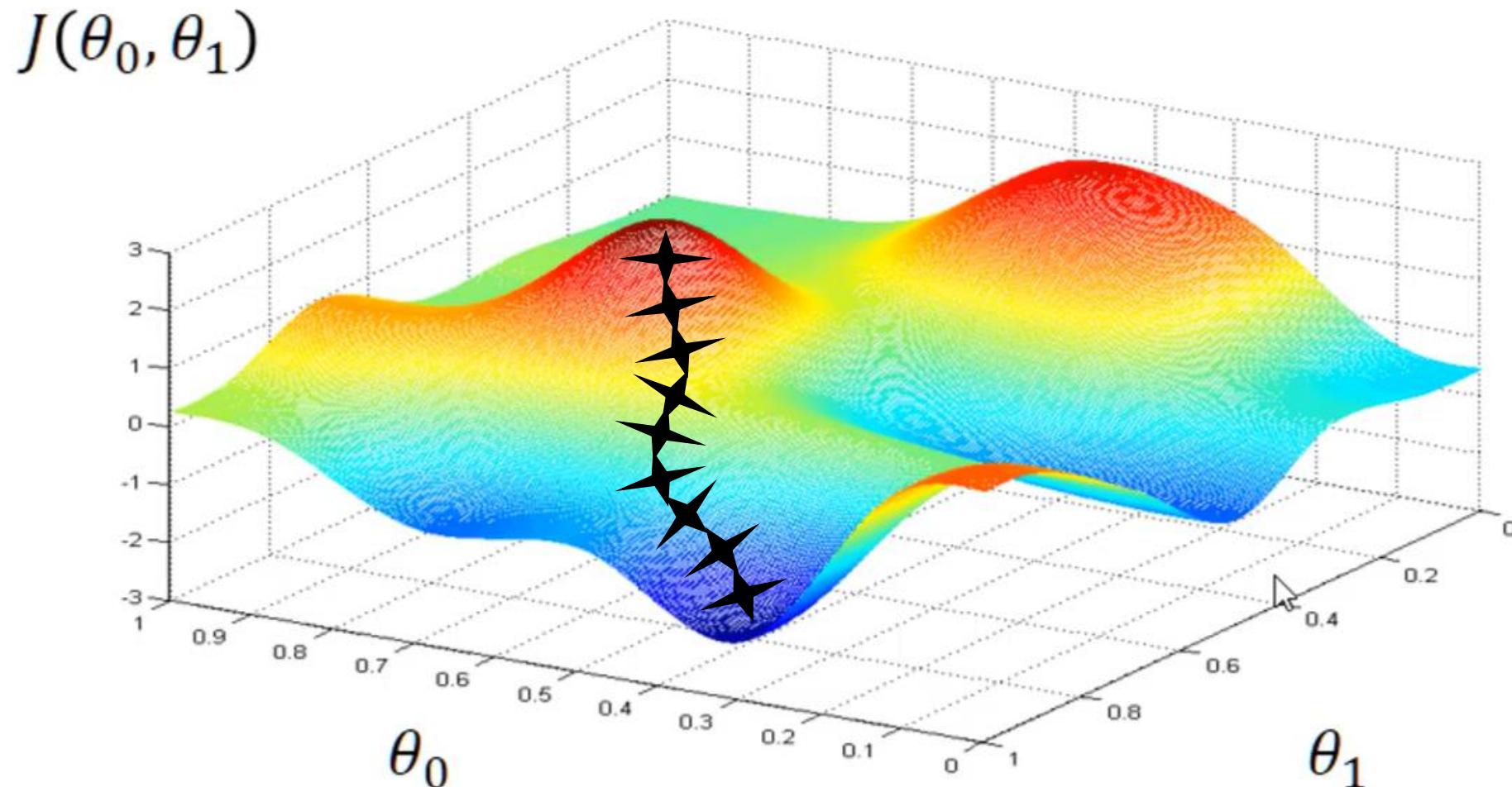
$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \frac{\partial J}{\partial \theta_1} \end{bmatrix}$$

$$\theta^{(new)} = \theta^{(old)} + \alpha(-\nabla J)$$



$$-\nabla J = \begin{bmatrix} -\frac{\partial J}{\partial \theta_0} \\ -\frac{\partial J}{\partial \theta_1} \end{bmatrix}$$

Градиентный спуск (Gradient descent)



$$\theta^{(new)} = \theta^{(old)} + \alpha(-\nabla J)$$

$$\begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} + \alpha \begin{bmatrix} -\frac{\partial J}{\partial \theta_0} \\ -\frac{\partial J}{\partial \theta_1} \end{bmatrix}$$

Градиентный спуск в машинном обучении

Функция гипотезы:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Функция потерь:

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^n (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$\nabla \theta_0 := -\alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\nabla \theta_1 := -\alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \theta_0 + \nabla \theta_0$$

$$\theta_1 := \theta_1 + \nabla \theta_1$$

Repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

Градиентный спуск в машинном обучении

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$j = 0 \Rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)})$$

$$j = 1 \Rightarrow \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_i$$

Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \left(\frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)}) \right)$$

$$\theta_1 := \theta_1 - \alpha \left(\frac{2}{n} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_i \right)$$

}

Линейная регрессия с одной переменной

Площадь квадрата (X)	Стоимость квартиры (Y)
460	195
230	130
315	140
178	80
....

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Регрессия (Regression)

Линейная регрессия с одной переменной

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Линейная регрессия с многими переменными

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Линейная регрессия с многими переменными $x_0 = 1$

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$X = \begin{bmatrix} x_0 = 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$



$$h_{\theta}(x) = \langle x, \theta \rangle$$

$$h_{\theta}(x) = x^T \theta$$

Градиентный спуск (Gradient descent)

Функция гипотезы:

$$h_{\theta}(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

Параметры модели:

$$\theta = (\theta_0, \theta_1, \dots, \theta_n)$$

Функция потерь:

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

$$\frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

Градиентный спуск (Gradient descent)

Функция гипотезы:

$$h_{\theta}(x) = x^T \theta = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad h_{\theta}(x) = x^T \theta$$

Параметры модели:

$$\theta = (\theta_0, \theta_1, \dots, \theta_n)$$

Функция потерь:

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{n} \sum_{i=1}^m (x^T \theta^{(i)} - y^{(i)})^2$$

$$J(\theta) = \sum_{i=1}^n (h_{\theta}(x^{(i)}) - y^{(i)})^2 = \sum_{i=1}^n (x^T \theta^{(i)} - y^{(i)})^2$$

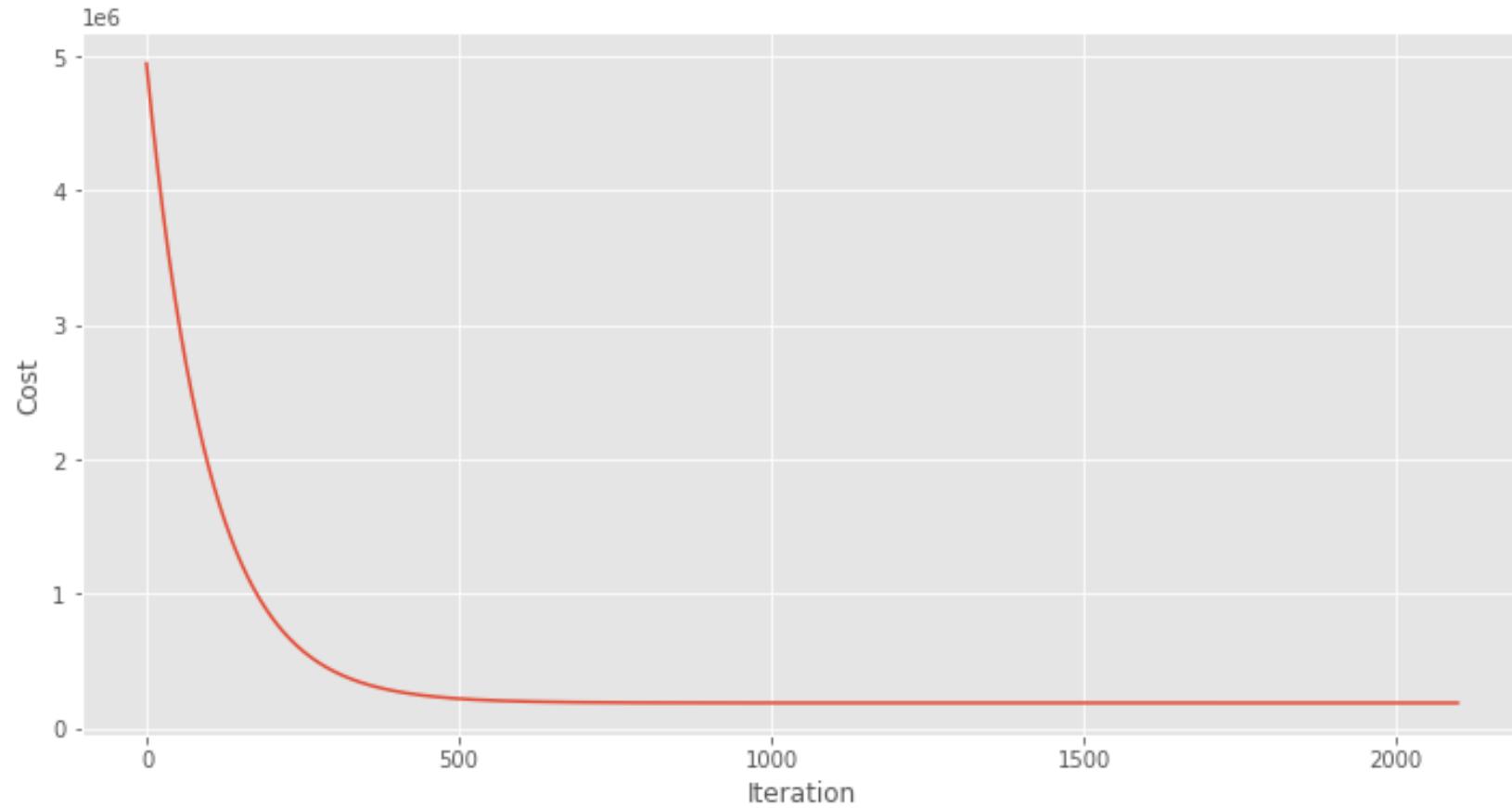
Градиентный спуск (Gradient descent)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (x^T \theta - y^{(i)})^2 \rightarrow \min_{\theta} J$$

$$\frac{\partial J}{\partial \theta_i} = \frac{\partial \sum_{i=1}^n (x_i^T \theta - y^{(i)})^2}{\partial \theta_i} = \frac{1}{n} \sum_{i=1}^n 2((x_i^T \theta - y^{(i)})) \frac{\partial x_i^T \theta}{\partial \theta_i} = \frac{2}{n} \sum_{i=1}^n (x^T \theta - y^{(i)}) \cdot x_i$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial j}{\partial \theta_1}, \dots, \theta_n := \theta_n - \alpha \frac{\partial j}{\partial \theta_n}$$

Cost Function



Стохастический градиентный спуск (Stochastic gradient descent)

Функция потерь

$$L : (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$J(\theta) = \frac{1}{n} \sum_{i=1}^m (h_{\theta}(x) - y^{(i)})^2$$

Градиент:

$$\nabla J(\theta) = \frac{1}{n} \sum_{i=1}^m \nabla L(y_{i_j}, h(x_{i_j}))$$

Может, оценить градиент одним слагаемым?

$$\nabla J(\theta) \cong \nabla L(y_{i_j}, h(x_{i_j}))$$

Градиентный спуск (Gradient descent)

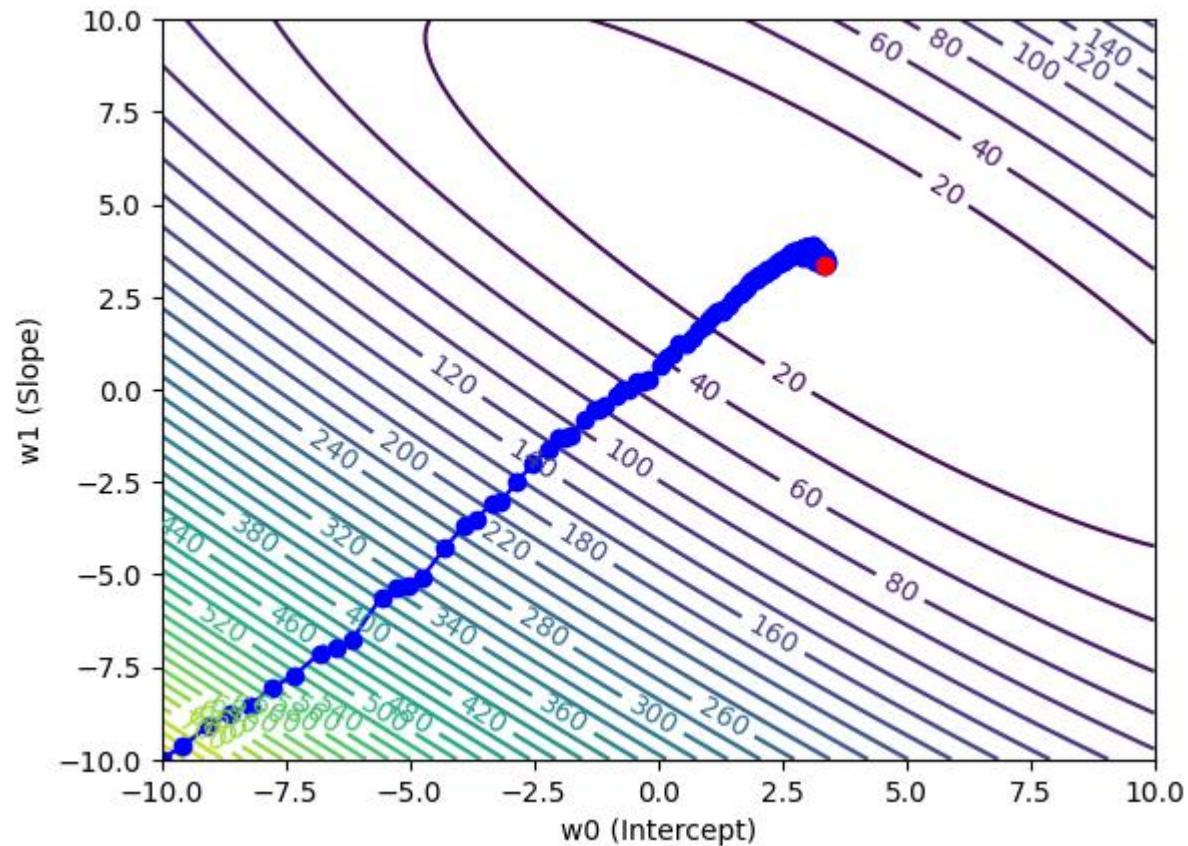
1- Стартуем из случайной точки

θ – Инициализация весов

2- Сдвигаемся по антиградиенту

$$\theta^t = \theta^{t-1} - \alpha \nabla J(\theta^{t-1})$$

3-Повторяем, пока не окажемся в точке минимума



Стохастический градиентный спуск (Stochastic gradient descent)

1- Стартуем из случайной точки

θ – Инициализация весов

2- Повторяем и сдвигается по антиградиенту, каждый раз выбираем случайный объект i_t

$$\theta^t = \theta^{t-1} - \alpha \nabla L(y_{i_j}, h_\theta(x_{i_j}))$$

3- Повторяем, пока не окажемся в точке минимума

Стохастический градиентный спуск (Stochastic gradient descent)

Mini-batch

1- Стартуем из случайной точки

θ – Инициализация весов

2- Повторяем и сдвигается по антиградиенту, каждый раз выбираем m случайных объектов i_1, \dots, i_m

$$\theta^t = \theta^{t-1} - \alpha \frac{1}{m} \sum_{j=1}^m \nabla L(y_{i_j}, h(x_{i_j}))$$

3- Повторяем, пока не окажемся в точке минимума

Сходимость

- Останавливаем процесс, если

$$\|\theta^t - \theta^{t-1}\| < \varepsilon$$

- Другой вариант

$$\|\nabla J(\theta^t)\| < \varepsilon$$

- Или пока не закончилось количество итераций

Градиентный спуск (Gradient descent)

Длина шага

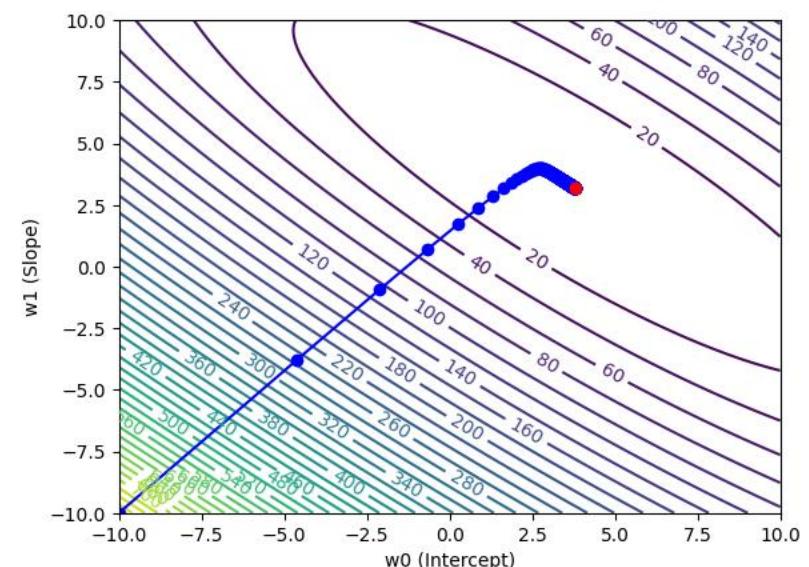
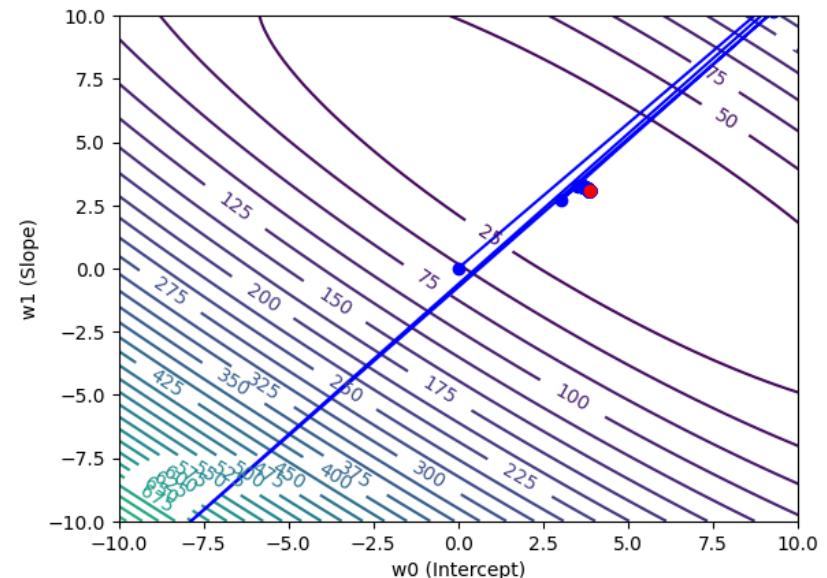
$$\theta^t = \theta^{t-1} - \alpha \nabla J(\theta^{t-1})$$

Длину шага можно менять в зависимости от шага

$$\alpha_t = \frac{1}{t}$$

Еще вариант

$$\alpha_t = \frac{0.1}{t^\beta}$$



Масштабирование данных (Data Scaling)

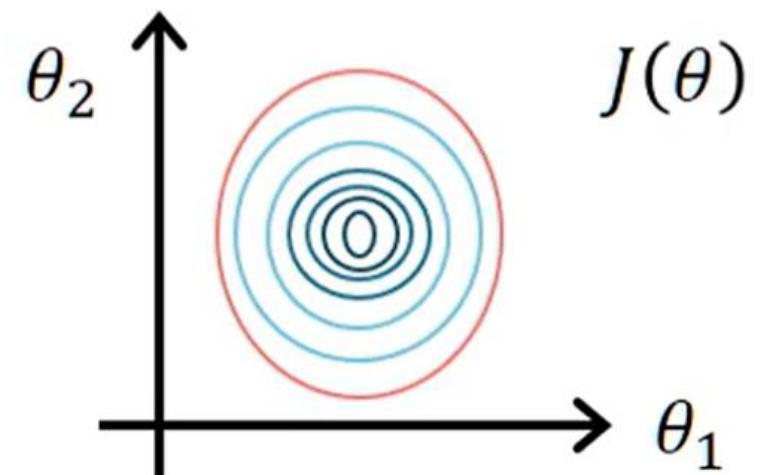
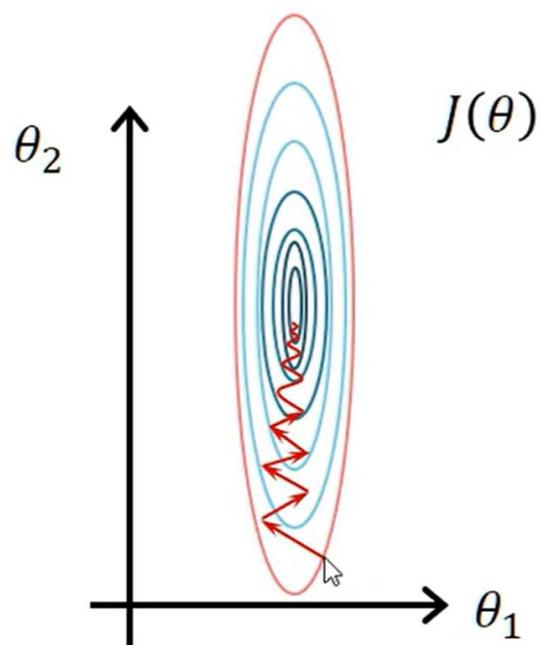
Масштабирование признаков

Процесс изменения данных происходит таким образом, чтобы они имели одинаковый масштаб

Цель: увеличение скорости сходимости в градиентном спуске.

x_1 : Площадь квадрата (0 – 2000)

x_2 : Количество комнат (1 – 10)



Масштабирование признаков

Вычтем из каждого значения признака среднее и поделим на стандартное отклонение

$x_i^{d_j}$ - Признак

μ_j - $\text{mean}(x)$ – Среднее значение

σ_j - $\text{std}(x)$ – Среднеквадратическое отклонение

$$x_i^{d_j} = \frac{x_i^{d_j} - \mu_j}{\sigma_j}$$

- Среднее значение масштабированных данных становится равным 0.
- Стандартное отклонение масштабированных данных становится равным 1.