# Машинное обучение в гидрологии

Вводная лекция

Семенова Наталья Кирилловна
snkone132@mail.ru

# Из чего состоит курс?

- Теоретическая часть – 3 лекции:
  - Постановка задач машинного обучения
  - Методы классификации
  - Линейные модели, регрессия
  - Композиции алгоритмов

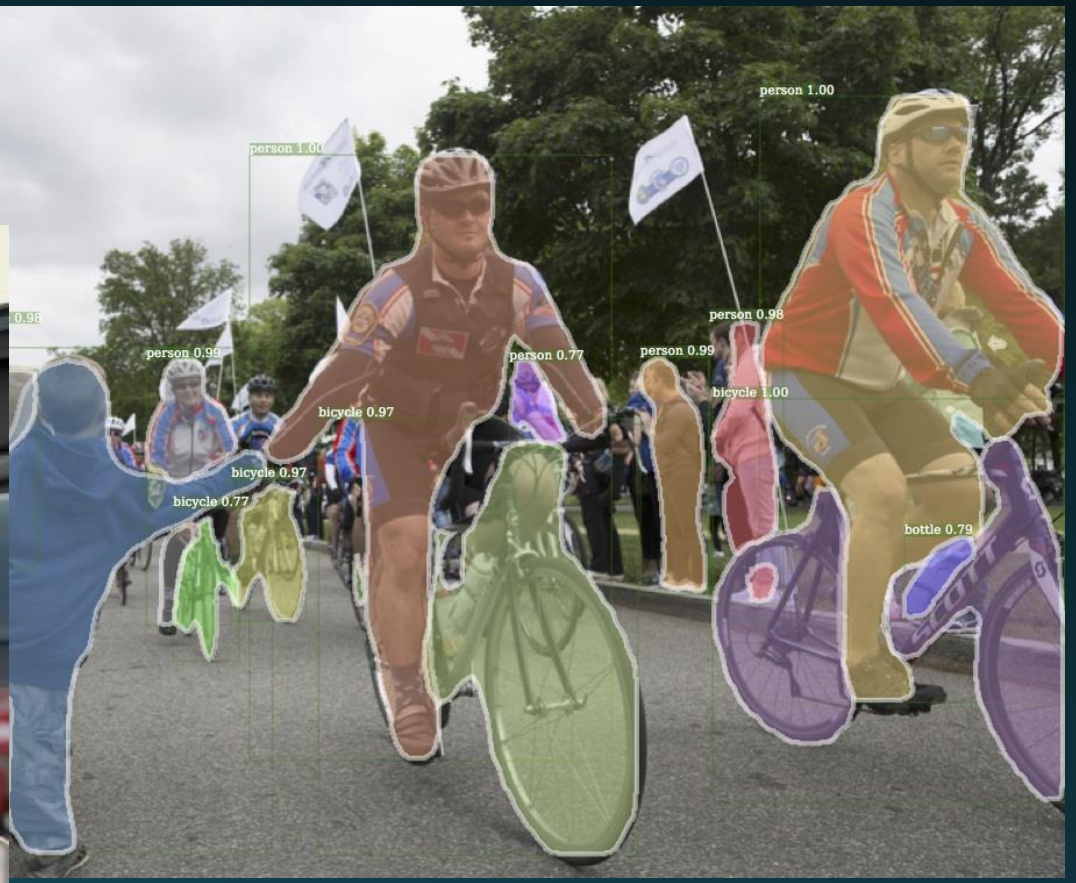
- Практическая часть – 3 семинара:
  - Первичный и визуальный анализ данных с Python
  - Построение моделей классификации и регрессии
  - Практические домашние задания

# Содержание лекции

1. Современное машинное обучение
2. Основные понятия и обозначения
   - Задание данных и постановка задачи
   - Модели и методы обучения
   - Обучение и переобучение
3. Примеры прикладных задач
   - Задача классификации
   - Задача регрессии

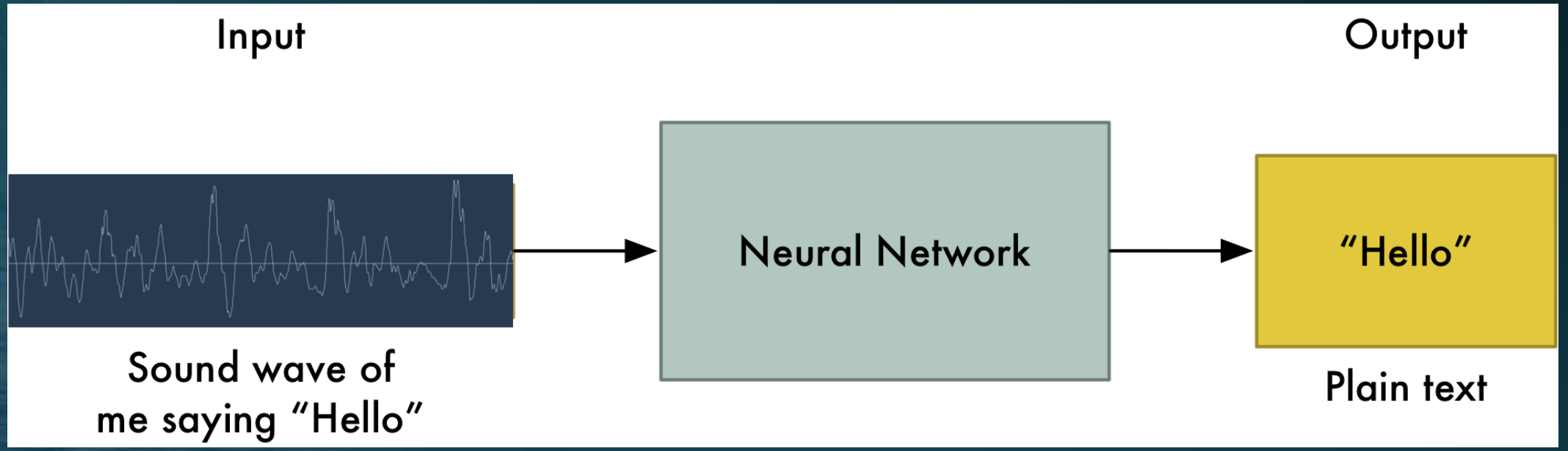
# Какие задачи сейчас решаются с помощью машинного обучения?

## Распознавание образов

# Какие задачи сейчас решаются с помощью машинного обучения?

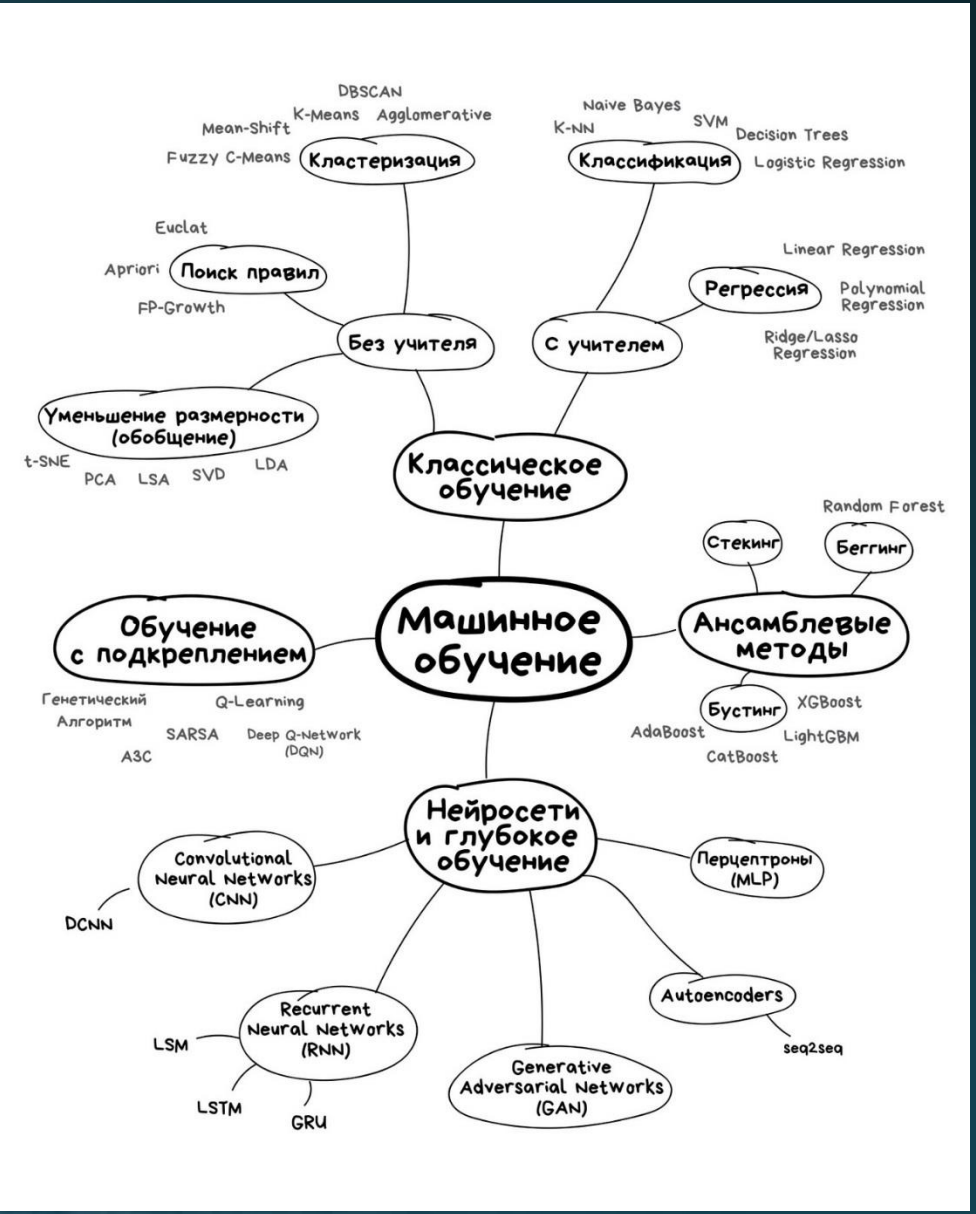
Распознавание речи

# Какие задачи сейчас решаются с помощью машинного обучения?

Медицинская диагностика, разработка лекарств

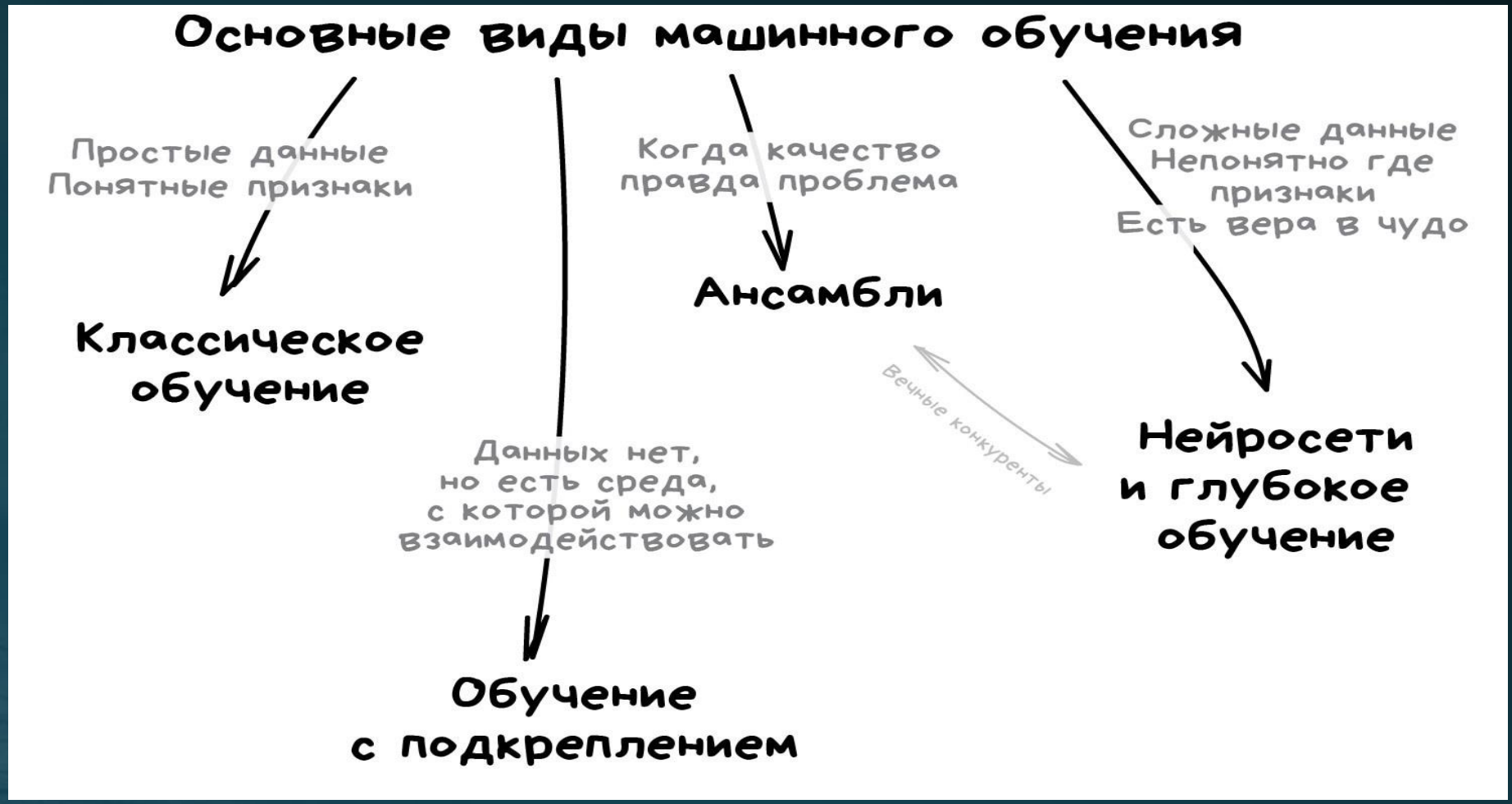
# Карта мира машинного обучения





- Искусственный интеллект – название всей области (как то биология, химия)
- Машинное обучение – только раздел искусственного интелекта
- Нейронные сети – раздел машинного обучения (не единственный!)
- Глубокое обучение – архитектура нейросетей, один из подходов к их построению и обучению

# В современном машинном обучении можно выделить 4 основных направления



# Классическое машинное обучение


Первые классические алгоритмы пришли в 50х

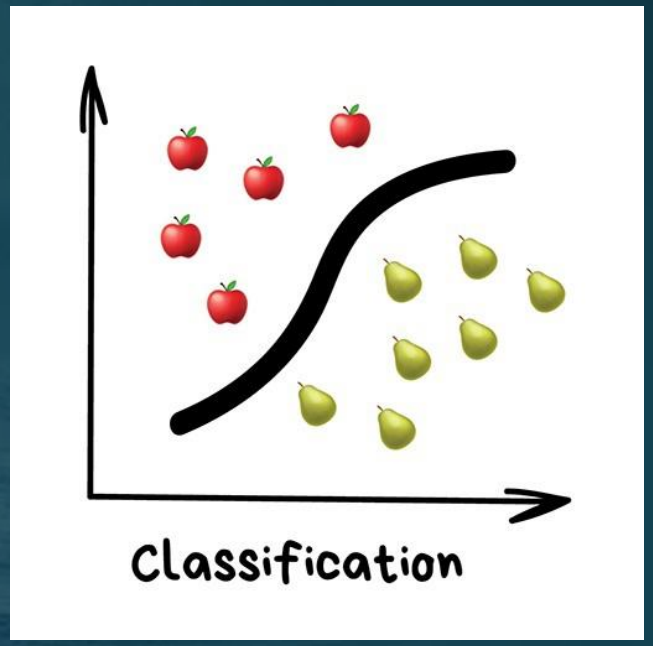
- С учителем: данные размечены заранее (Supervised Learning)
- Без: машине приходится искать закономерности в данных самостоятельно (Unsupervised Learning)

С учителем машина обучается быстрее и точнее. Поэтому в боевых задачах методы обучения с учителем используются чаще.

# Обучение с учителем

## Классификация

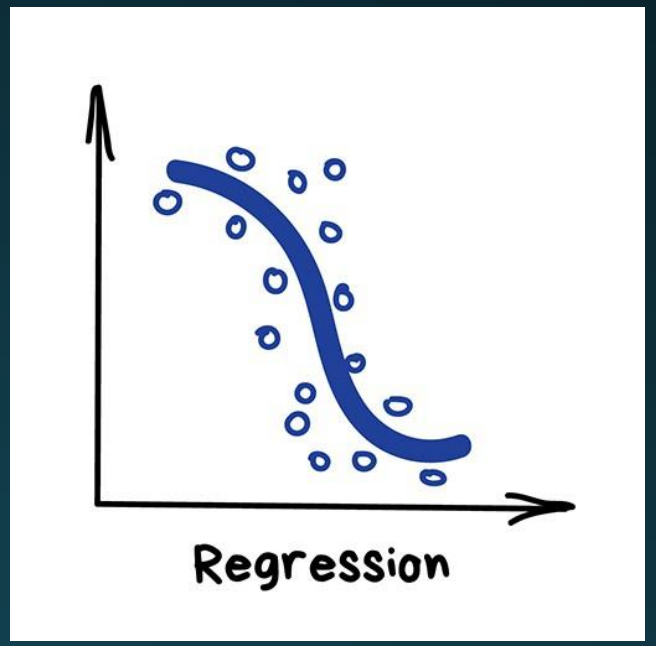
Разделение объектов по заранее известному признаку. Носки по цветам, музыку по жанрам




## Регрессия

Та же классификация, только вместо категории предсказывается число.

Нарисовать линию по точкам



# Обучение с учителем

## Классификация

Где применяется:

- Спам-фильтры
- Определение языка
- Поиск похожих документов
- Анализ тональности
- Распознавание рукописных букв и цифр

## Регрессия

Где применяется:

- Прогноз стоимости ценных бумаг
- Банковская сфера
- Медицинские диагнозы
- Анализ спроса, объема продаж

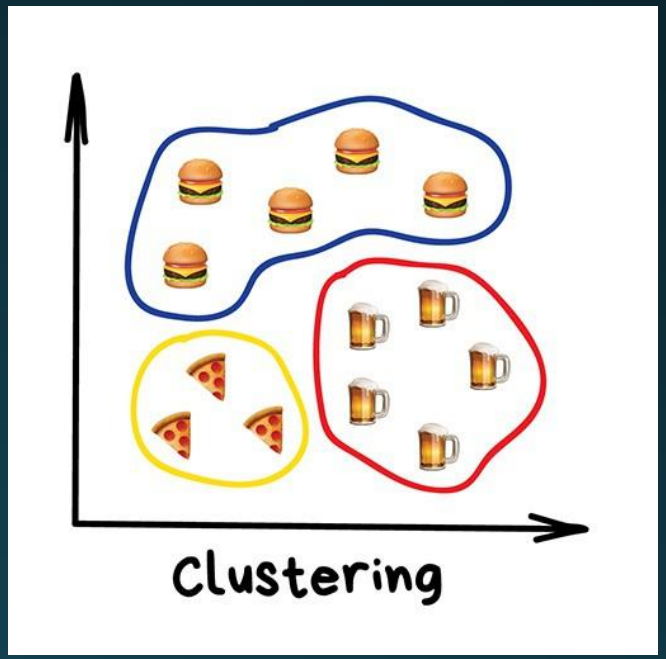
# Обучение без учителя

- Размеченные данные – дорогая редкость.
- Чаще всего обучение без учителя используют для анализа данных, а не как основной алгоритм.
- Используется для кластеризации и поиска зависимостей

**Кластеризация**

Разделение объектов по неизвестному признаку.

Машина сама определяет, как будет лучше



# Обучение без учителя

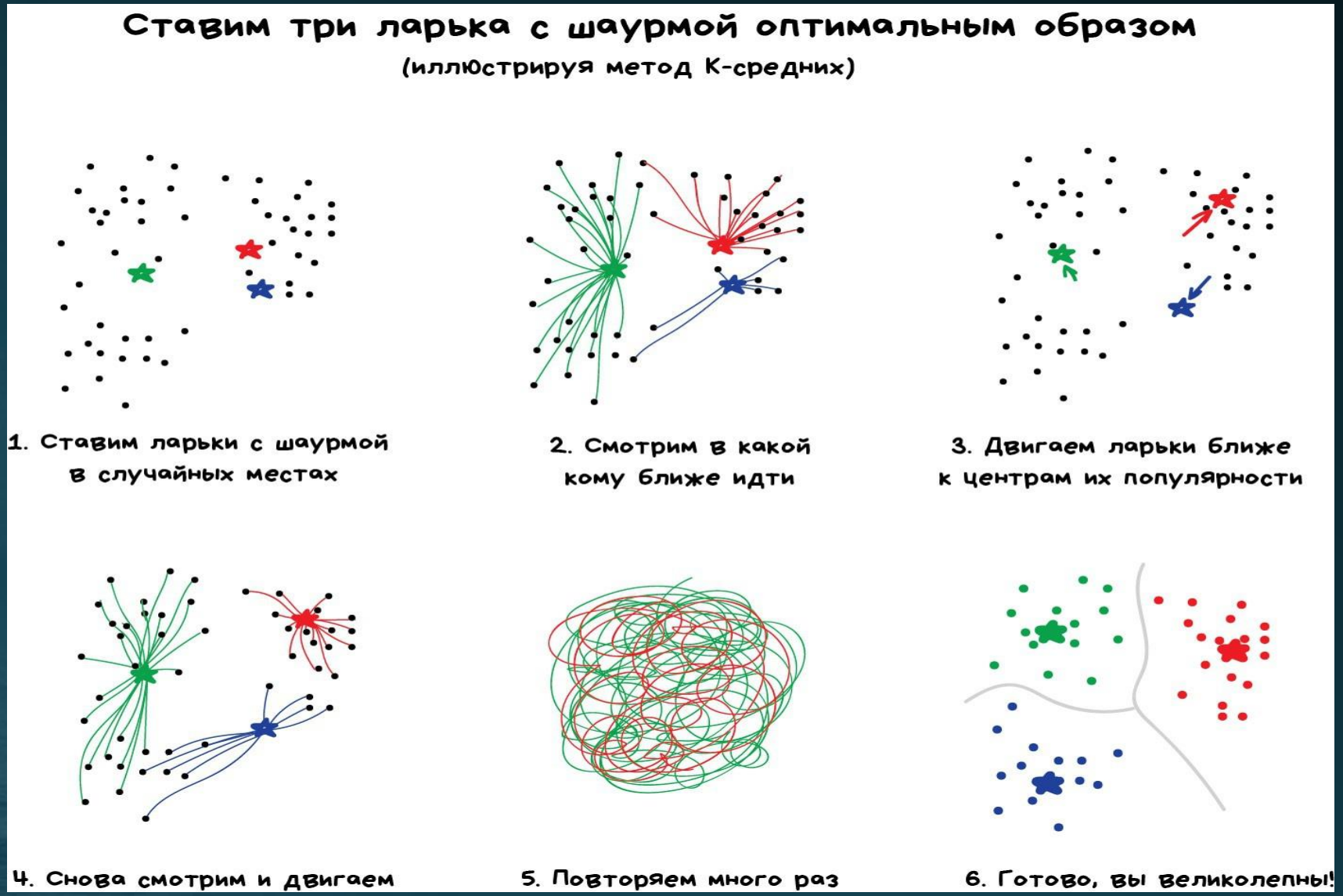
- Размеченные данные – дорогая редкость.
- Чаще всего обучение без учителя используют для анализа данных, а не как основной алгоритм.
- Используется для кластеризации и поиска зависимостей

**Кластеризация**

Где применяется:

- Сегментация рынка
- Объединение близких точек на карте
- Сжатие изображений
- Детекторы аномального поведения

# Кластеризация



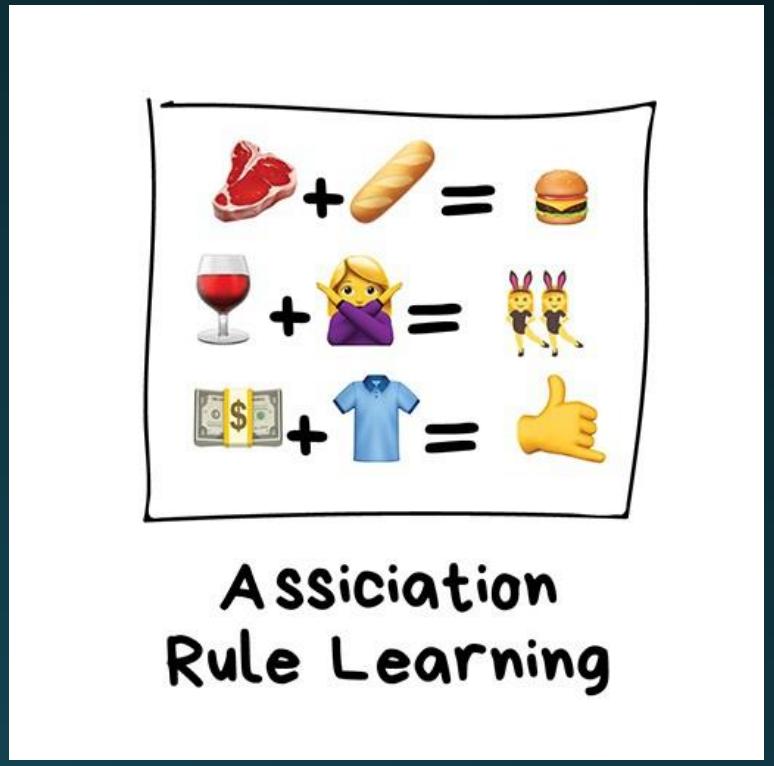
# Обучение без учителя

## Уменьшение размерности

Сведение признаков в абстракции более высокого уровня


## Поиск правил

Поиск закономерностей в потоке «заказов»



# Уменьшение размерности

Где используется :

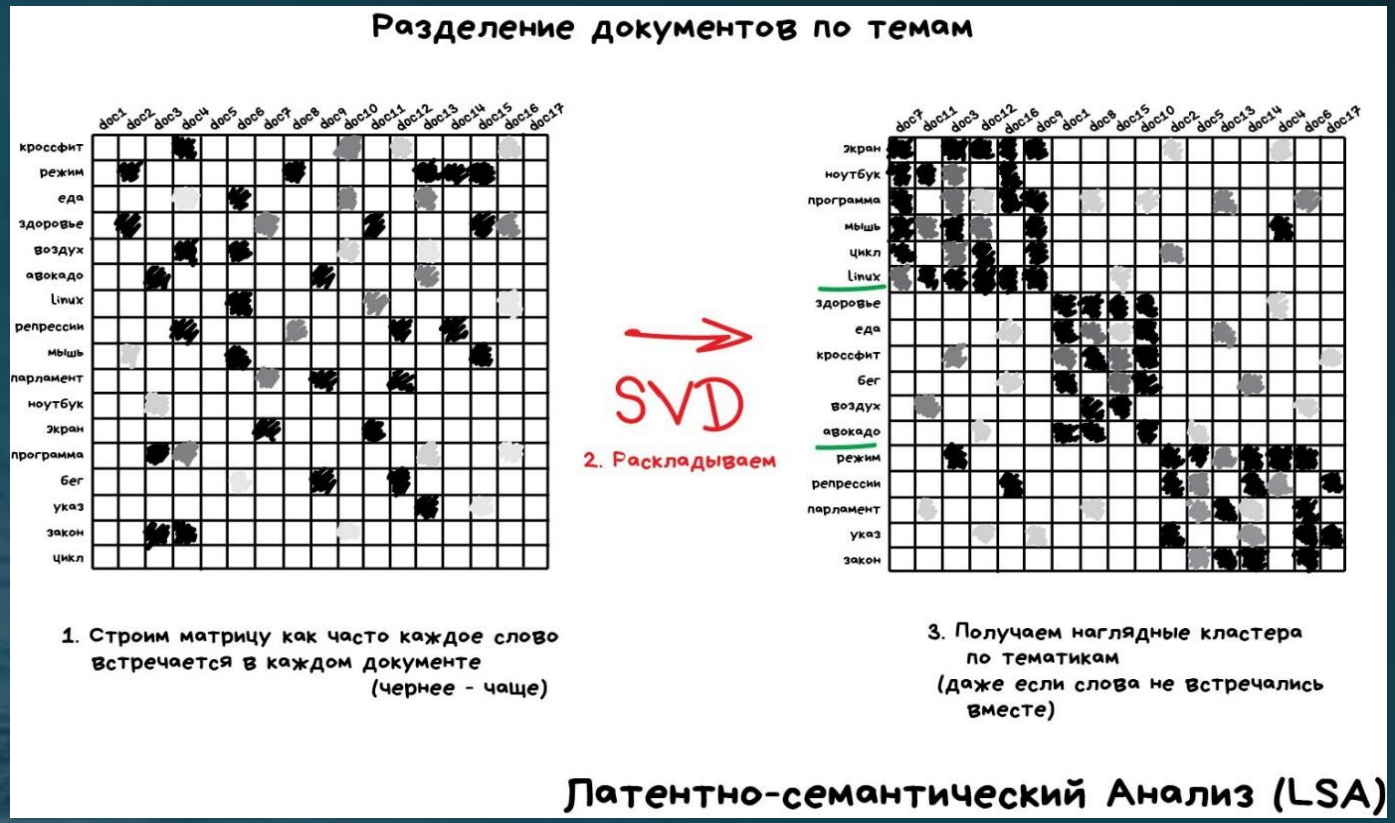
- Рекомендательные системы
- Определение тематики и поиска схожих документов
- Анализ фейковых изображений
- Риск-менеджмент

Объединяем несколько признаков в одну «абстракцию». Да, теряем информацию о конкретных объектах. Но абстракция полезнее деталей. Плюс обучение происходит быстрее.

# Уменьшение размерности

Определение тематик текстов  - латентно семантический анализ (LSA алгоритм).

Идея заключается в том, что частота появления слов зависит от тематики текста. В научных словах больше технических слов, в текстах женских журналов – косметики.

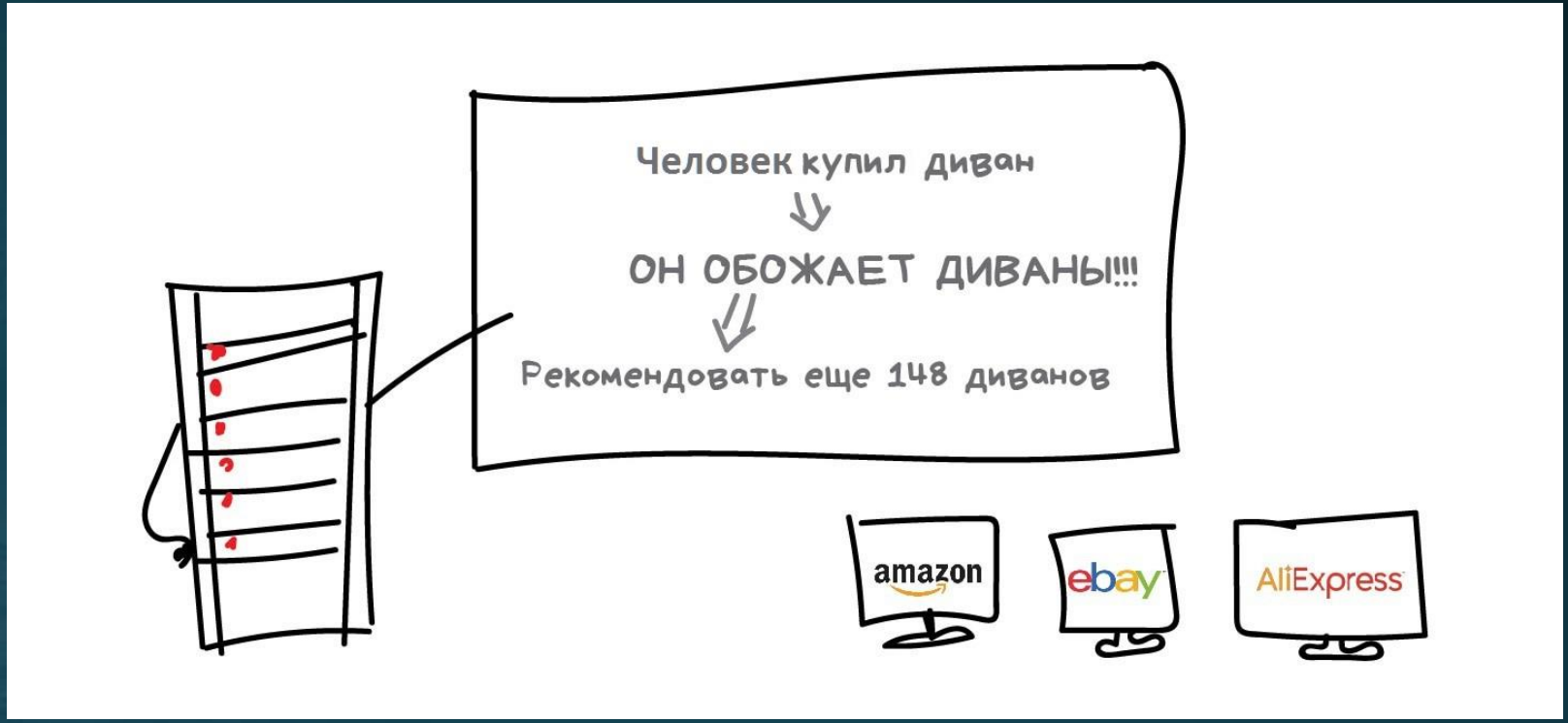



Необходимо объединять слова и документы в один признак, не теряя скрытые связи (в разных документах разные слова могут обозначать одно понятие)

Сингулярное разложение (SVD) – метод, выявляющий полезные тематические кластеры из слов, которые встречаются вместе.

# Поиск правил

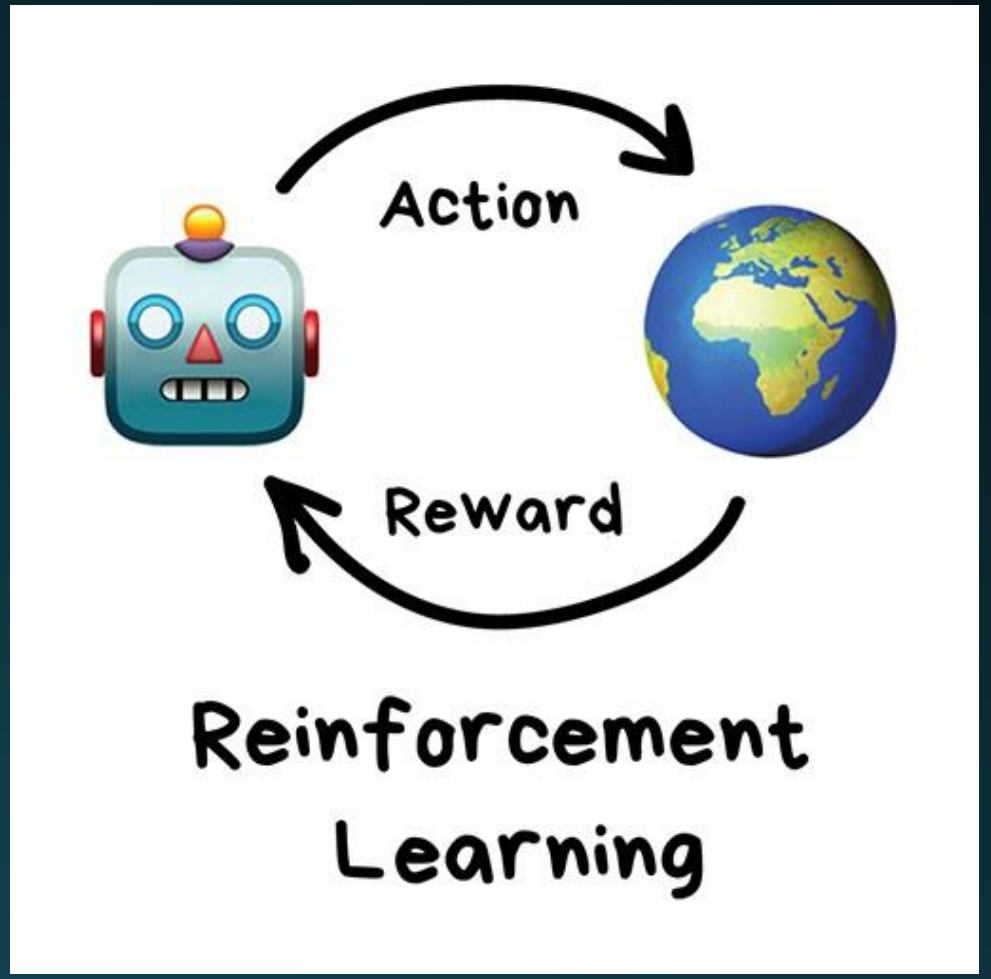
- Прогноз акций и распродаж, анализ покупок, расстановка товара
- В реальности классические способы заключаются в переборе пар всех купленных товаров, плохо обобщают закономерности и плохо воспроизводят их на новых данных



# Обучение с подкреплением

«Брось робота в лабиринт и пусть ищет выход»

- Данных нет. Есть среда и взаимодействие со средой.
- Главная задача – не анализ данных, а выживание в среде.
- Рассчитать все ходы невозможно. Цель – минимизировать ошибки.
- Грубо говоря: штрафуем за ошибки и награждаем за правильные поступки.



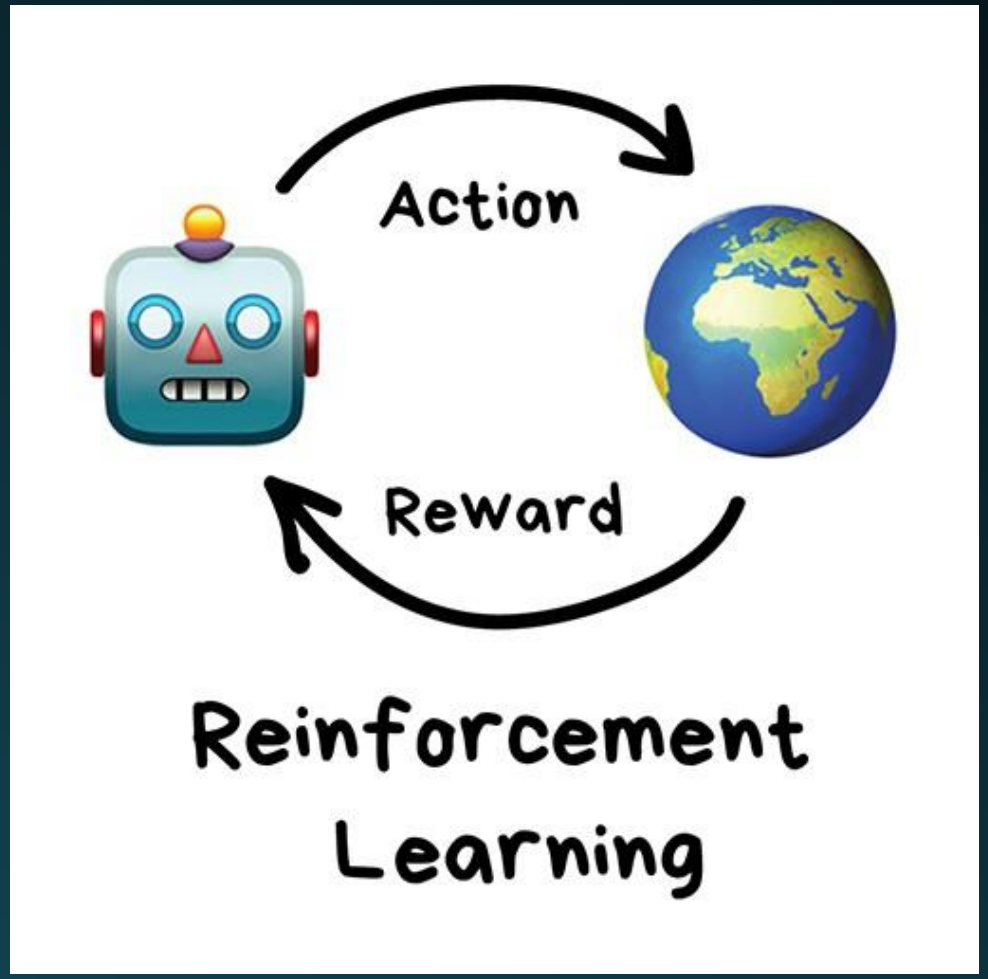
# Обучение с подкреплением

Где используют:

- Самоуправляемые автомобили
- Компьютерные игры
- Автоматическая торговля

Из очень известного:

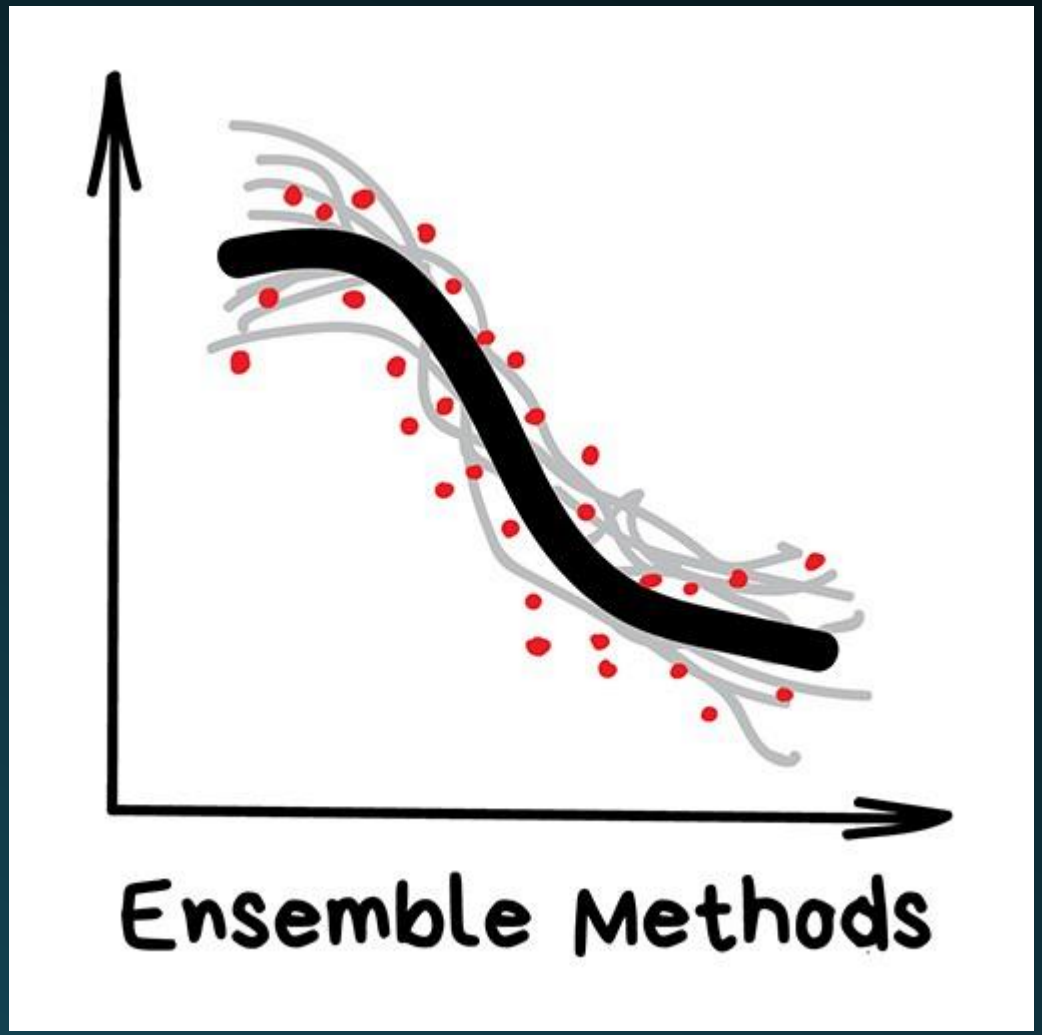
Alpha Zero – машина, которая в течение 24 часов достигла сверхчеловеческого уровня игры в шахматы, сёги и го, победив чемпионов мира среди программ!



# Ансамбли

«Куча глупых деревьев учится распознавать ошибки друг друга»

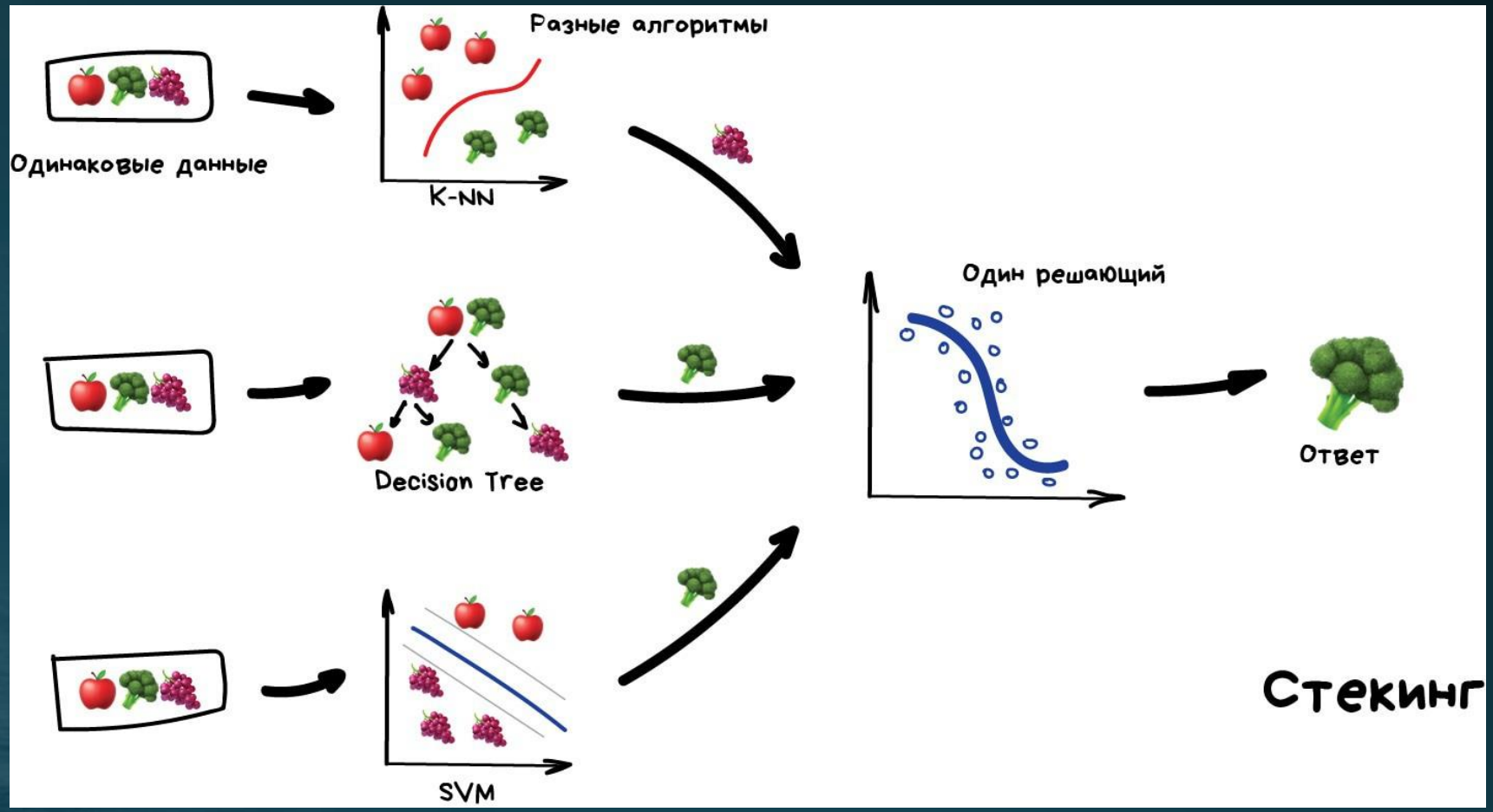
- Идея проста: взять несколько простых, может не самых эффективных алгоритмов, но обучить их исправлять ошибки друг друга.
- Качество такой системы может получиться выше, чем каждого метода по отдельности.
- Где используется:
  - Любые классические алгоритмы
  - Поисковые системы
  - Компьютерное зрение



# Три способа задания ансамблей

## Стекинг:

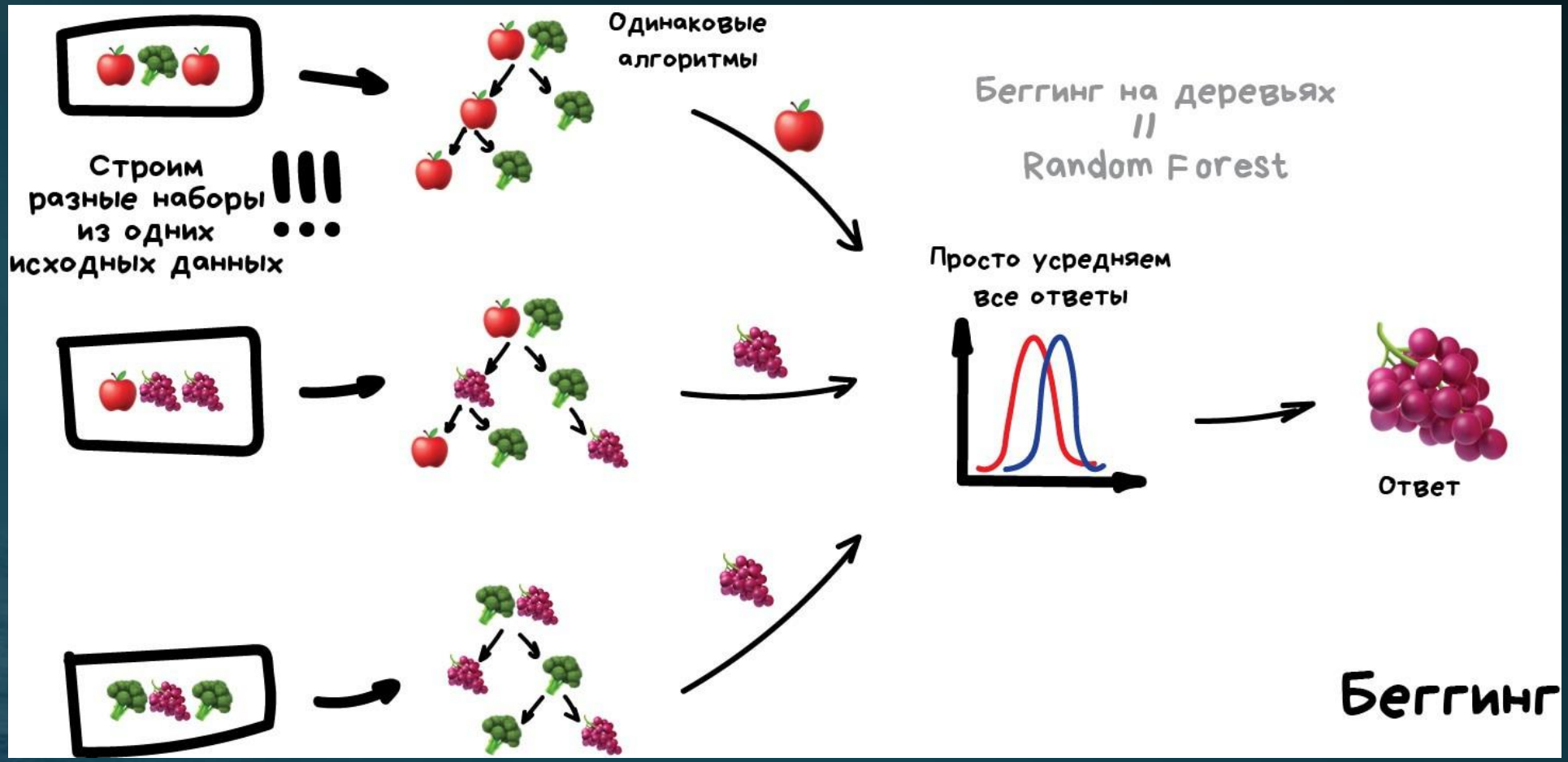
одинаковые данные → разные алгоритмы → результаты подаются на вход решающему



# Три способа задания ансамблей

## Беггинг:

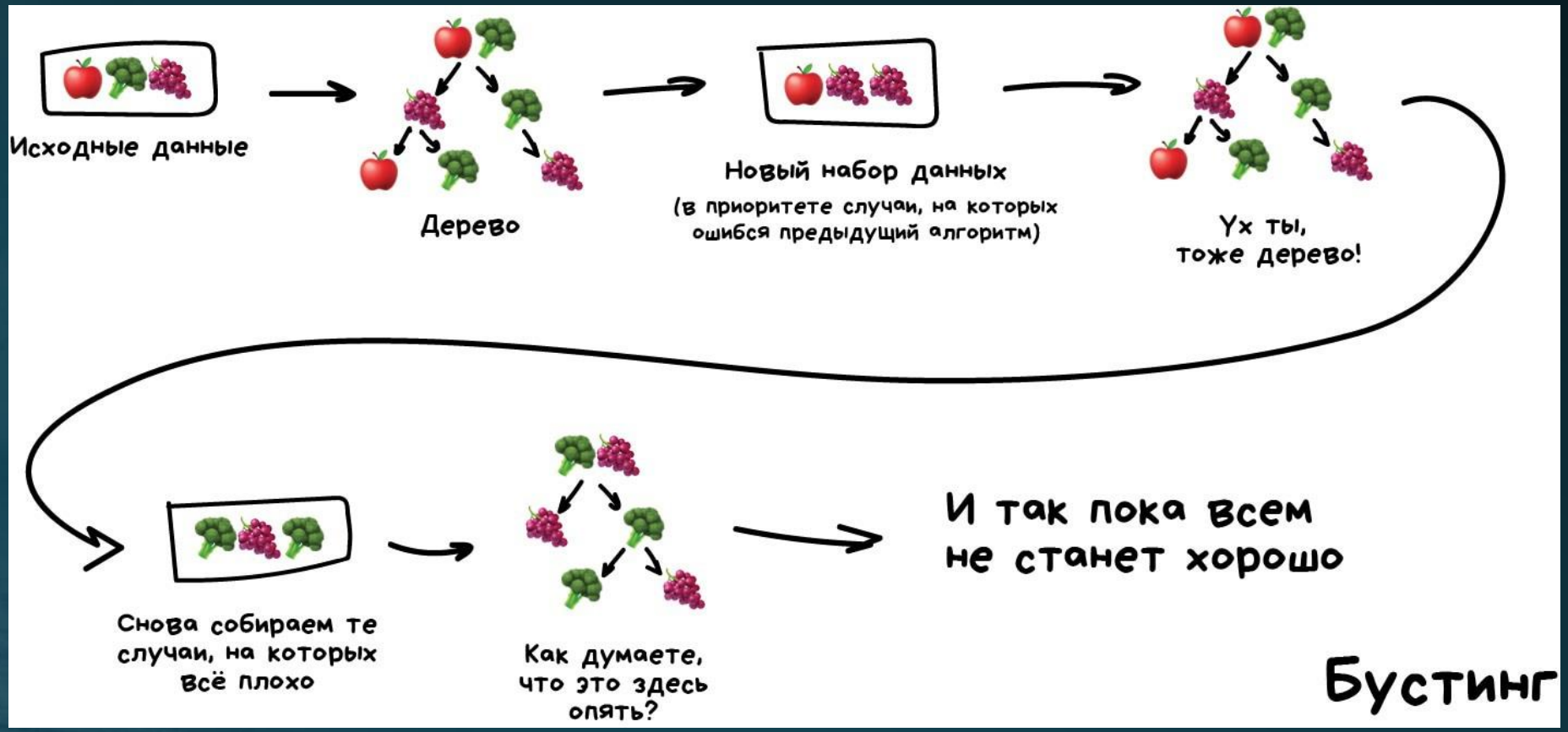
разные случайные данные → одинаковые алгоритмы → усредняем ответы



# Три способа задания ансамблей

## Бустинг:

Последовательное обучение, уделяем внимание ошибкам на прошлом шаге.
Аналог беггинга, но данные выбираются не случайно




Результатам позавидуют все!

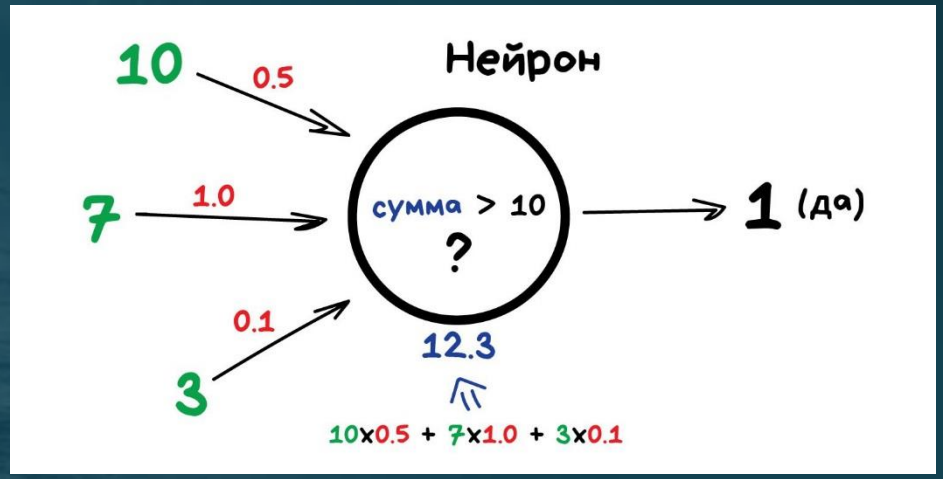
# Три способа задания ансамблей

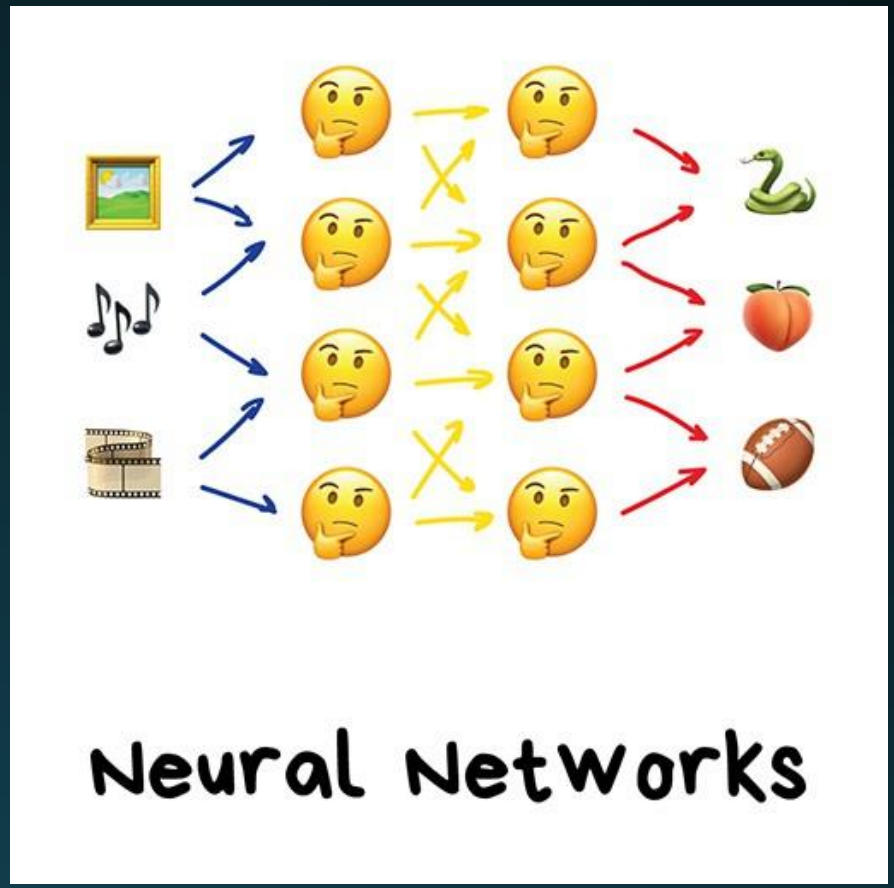
## Бустинг:

Последовательное обучение, уделяем внимание ошибкам на прошлом шаге.
Аналог беггинга, но данные выбираются не случайно

# Нейросети и глубокое обучение

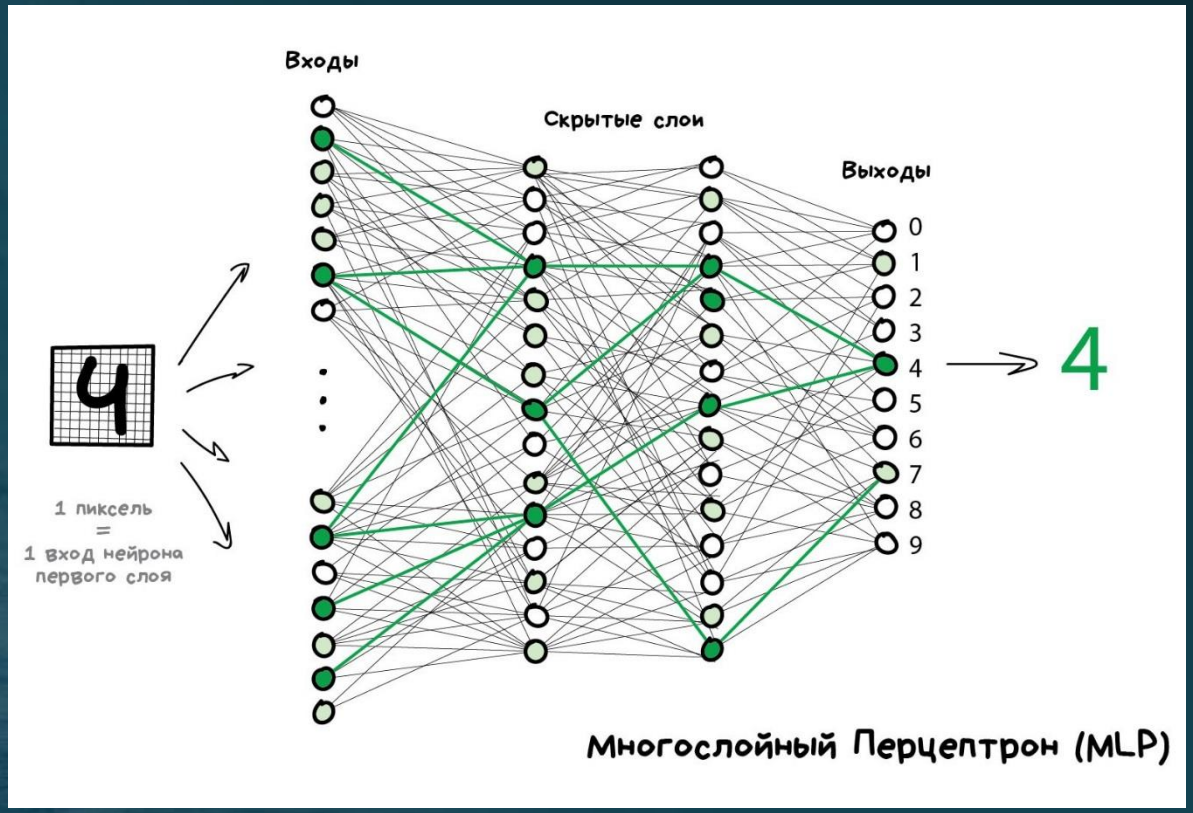
- Нейросеть – набор нейронов и связей между ними.
- Нейрон – функция с кучей входов и одним выходом.
- Связь – канал по которому нейроны передают друг другу информацию. Все каналы имеют свой вес. Нейрон сам особо не разбирается, что к нему приходит – это регулируют веса

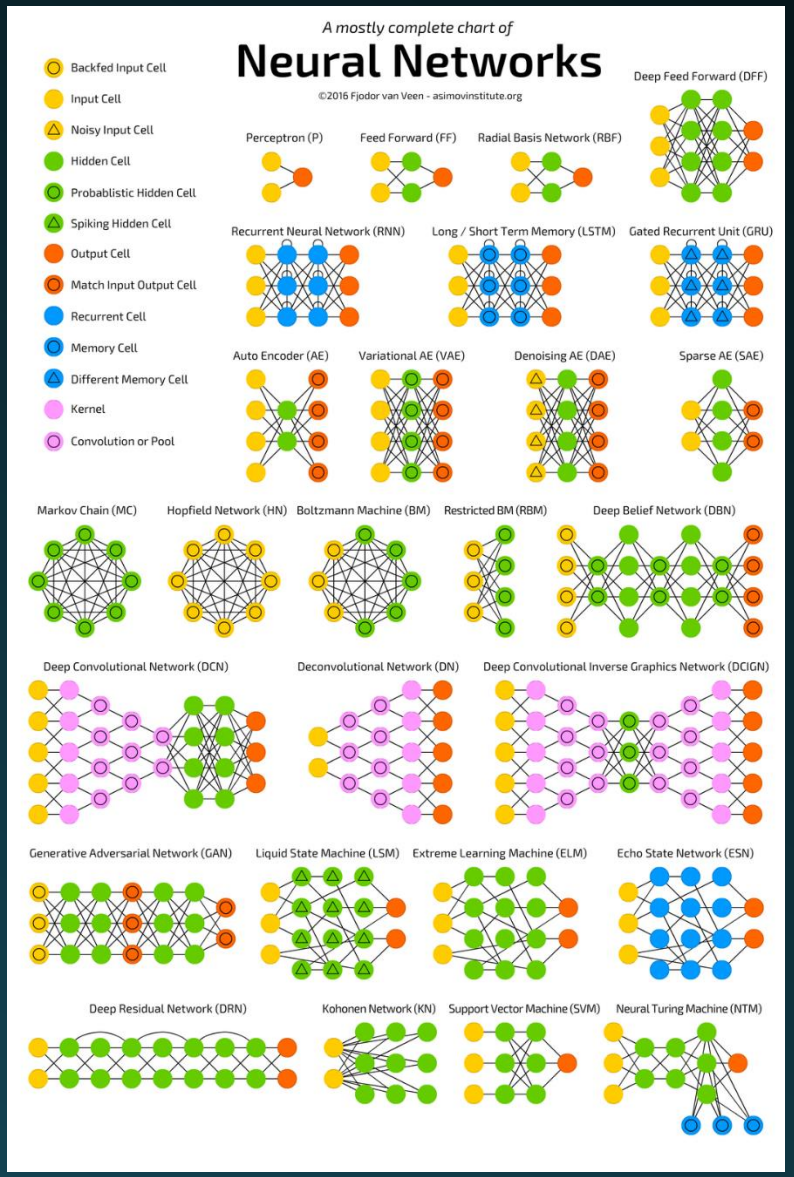


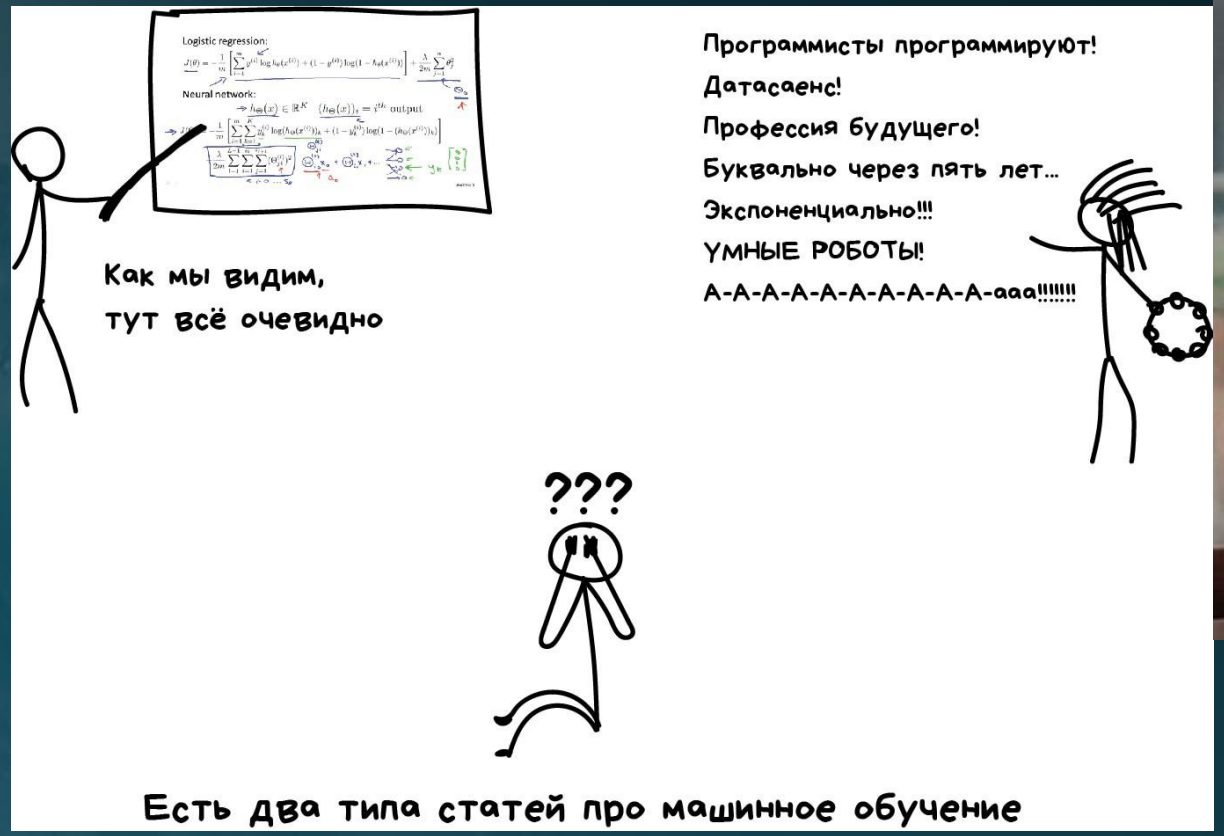



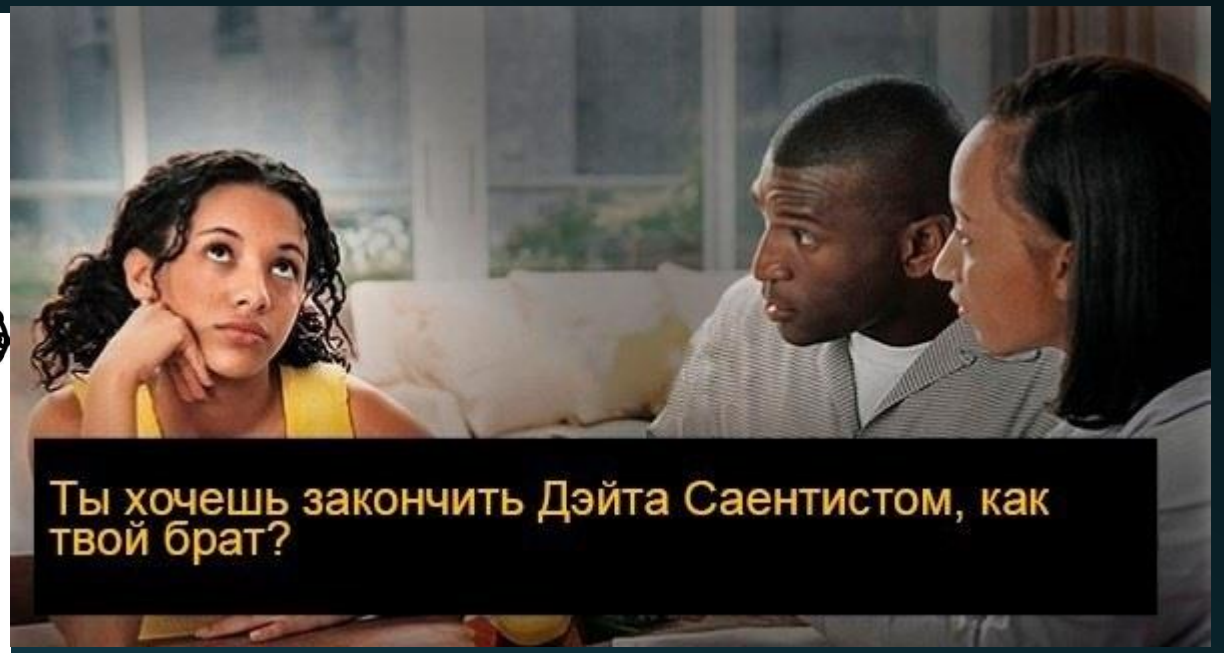
# Нейросети и глубокое обучение

- Нейроны расположены по слоям, дабы не допустить анархии. Внутри слоя нейроны никак не связаны и соединяются с лишь со следующим по порядку слоем.

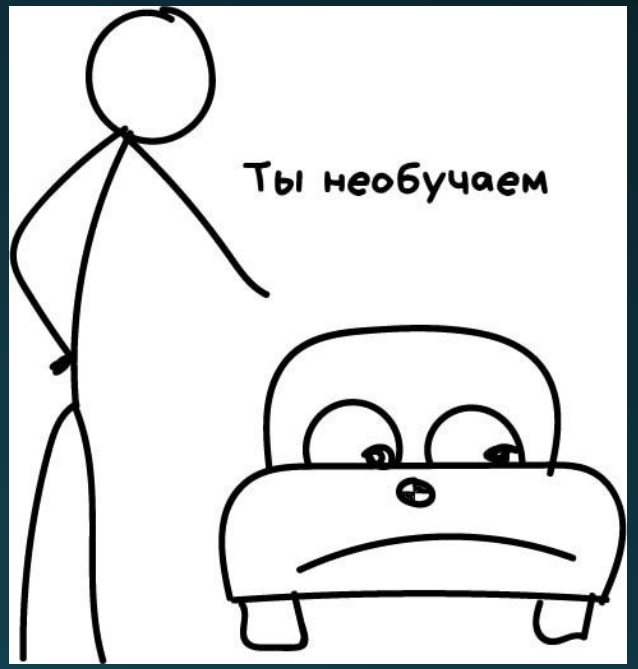





# Ну а теперь серьезно...



# Машинное обучение с учителем

# Постановка задачи

X – множество объектов

Y – множество ответов

y : X → Y – неизвестная зависимость (target function)

## Дано:

$X_{obs} = \{x_1, \dots x_N\} \subset X$ – уже наблюдаемые объекты

$y_i = y(x_i),\ i = \{1, \dots N\}$ – известные ответы

## Найти:

$a: X \to Y$ – алгоритм, решающую функцию (decision function) наилучшим образом приближающую **y** на всем множестве **X**

# Типы задач

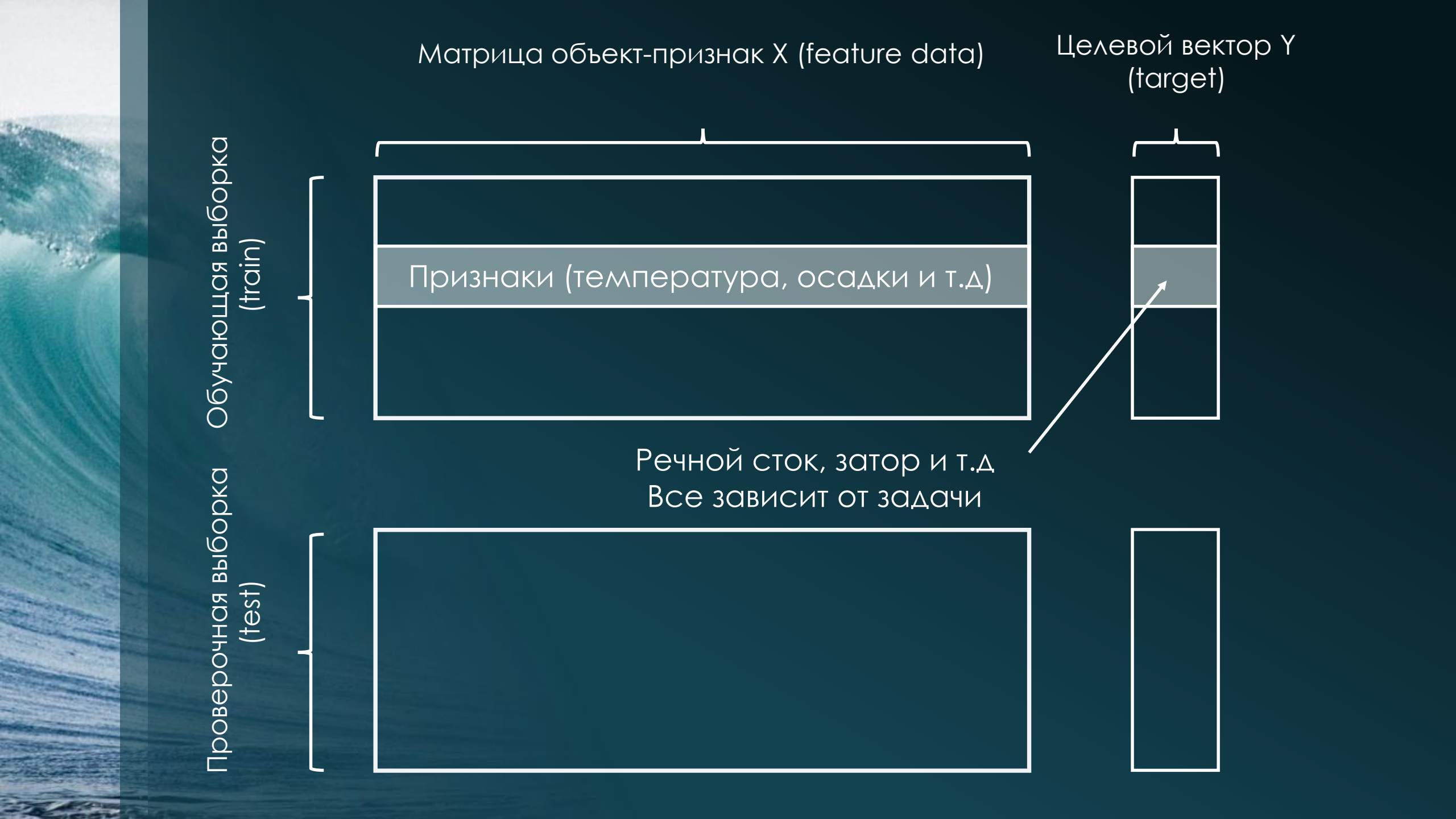
1. Задача классификации
   - Y = {-1, 1} – классификация на 2 класса
   - Y = {1,…M} – классификация на M непересекающихся классов
   - Y = $\{0,1\}^M$ - классификация на M пересекающихся классов
2. Задача регрессии
   - Y = $\mathbb{R}$ или Y = $\mathbb{R}^n$

# Как задаются объекты? Признаки!

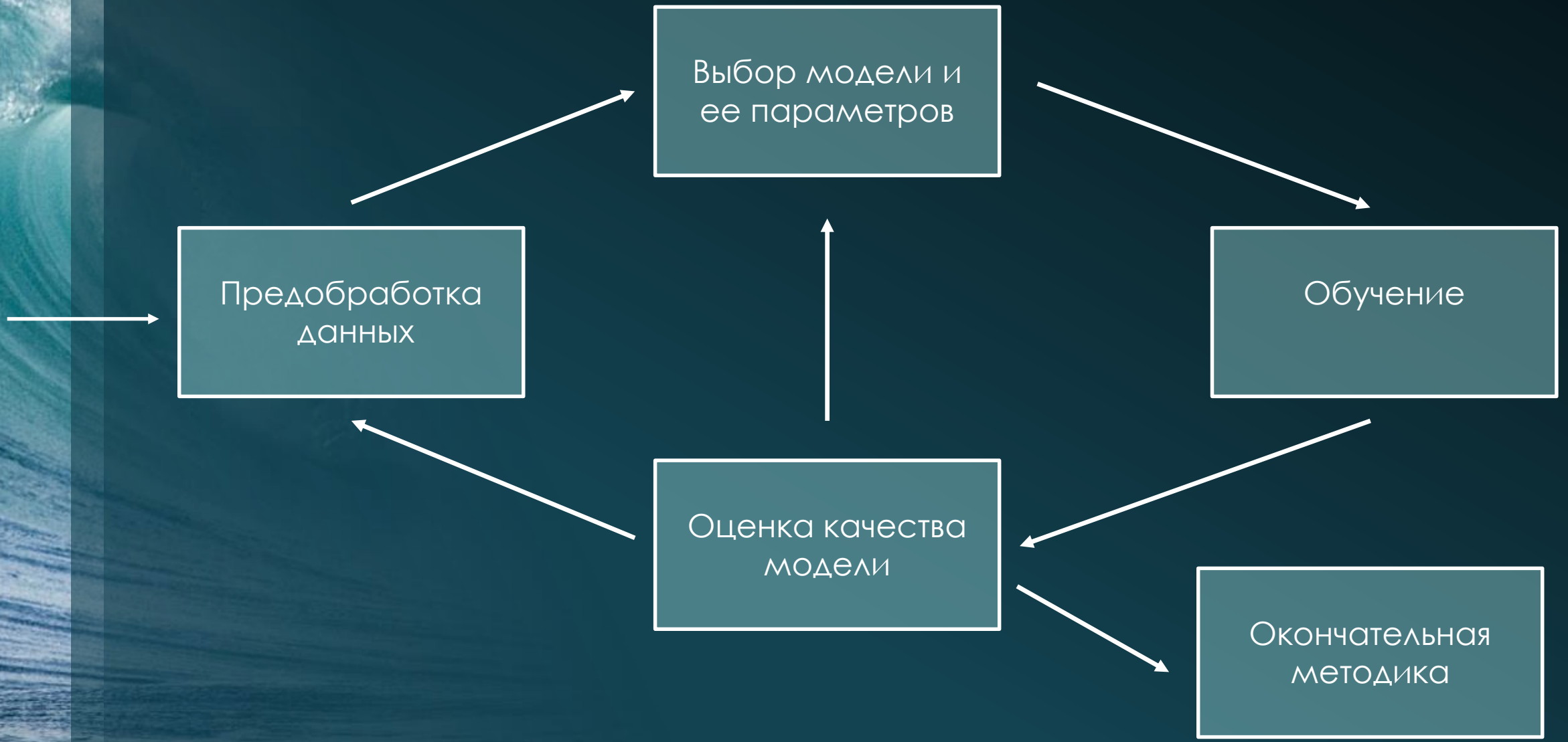
1. Каждому объекту наблюдения X сопоставляется вектор его признаков (характеристик, features)

$$X \to (x_1, x_2, \dots, x_L)$$

2. Типы признаков
   - Бинарные $x_i \in \{0,1\}$
     - 0 – не было затора
     - 1 – был затор
   - Номинальные, категориальные (тип осадков)
   - Порядковые (сила ветра)
   - Количественные $x_i \in \mathbb{R}$ (расход воды, толщина льда, температура)


Матрица объект-признак X (feature data)
Целевой вектор Y (target)
Обучающая выборка (train)
Признаки (температура, осадки и т.д)
Речной сток, затор и т.д
Все зависит от задачи
Проверочная выборка (test)

# Как происходит процесс обучения?



# Как оценивать качество модели?

Вводим функционал качества.

$L(a, x)$ – функция потерь (loss function) – величина ошибки алгоритма $a$ на объекте $x$.

- В случае задачи классификации:
  - $L(a, x) = I(a(x) \neq y(x))$ – индикатор ошибки (результат работы алгоритма не совпадает с реально существующей зависимостью)
- В задаче регрессии:
  - $L(a, x) = |a(x) - y(x)|$ - абсолютное отклонение
  - $L(a, x) = (a(x) - y(x))^2$ - квадратичное отклонение
- Тогда на всей обучающей выборке $X_{train}$ (эмпирический риск):

$$Q(a, X_{train}) = \frac{1}{N} \sum_{x \in X_{train}} L(a, x)$$

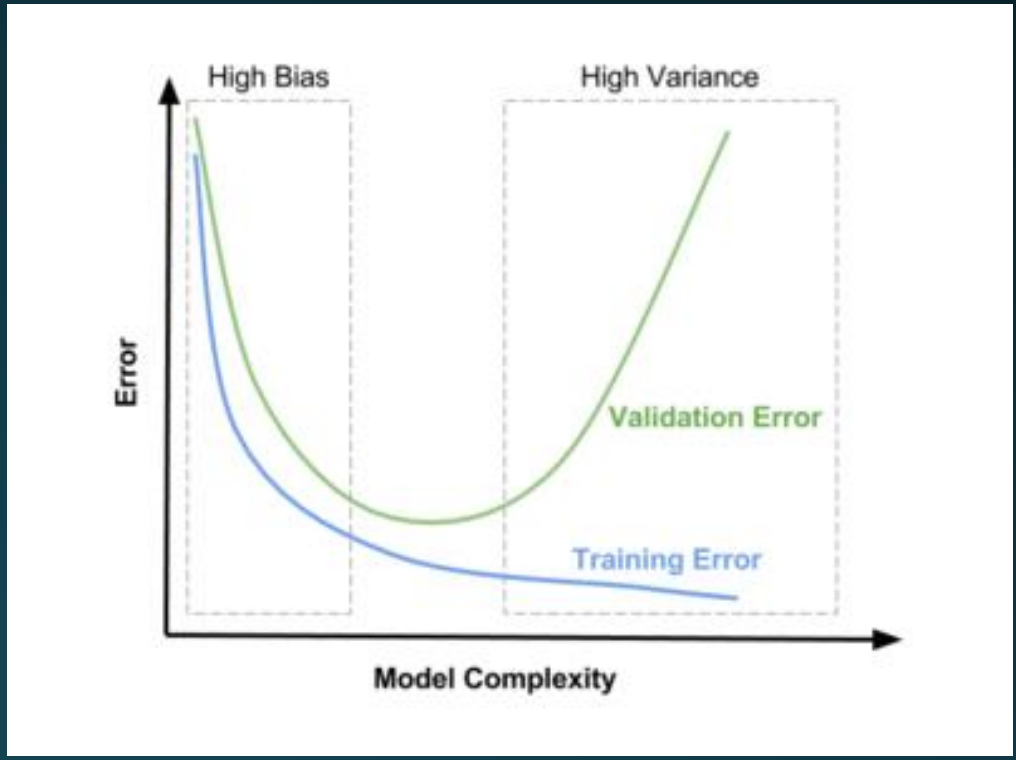
# Как оценивать качество модели?

Ставится задача о минимизации эмпирического риска.

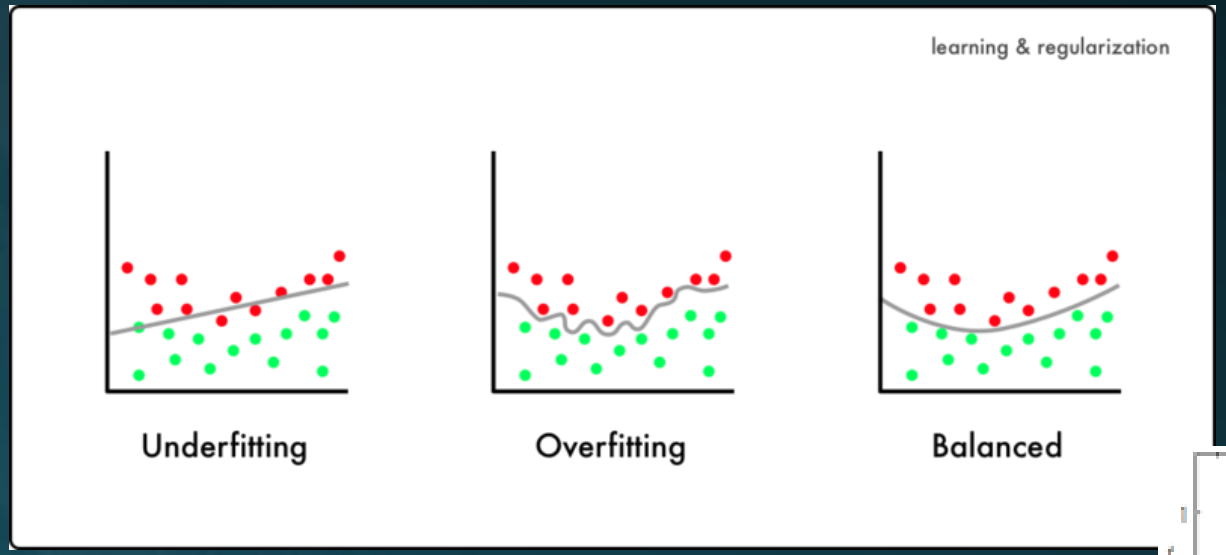
Будет ли $Q(a, X_{test})$ тоже маленьким?

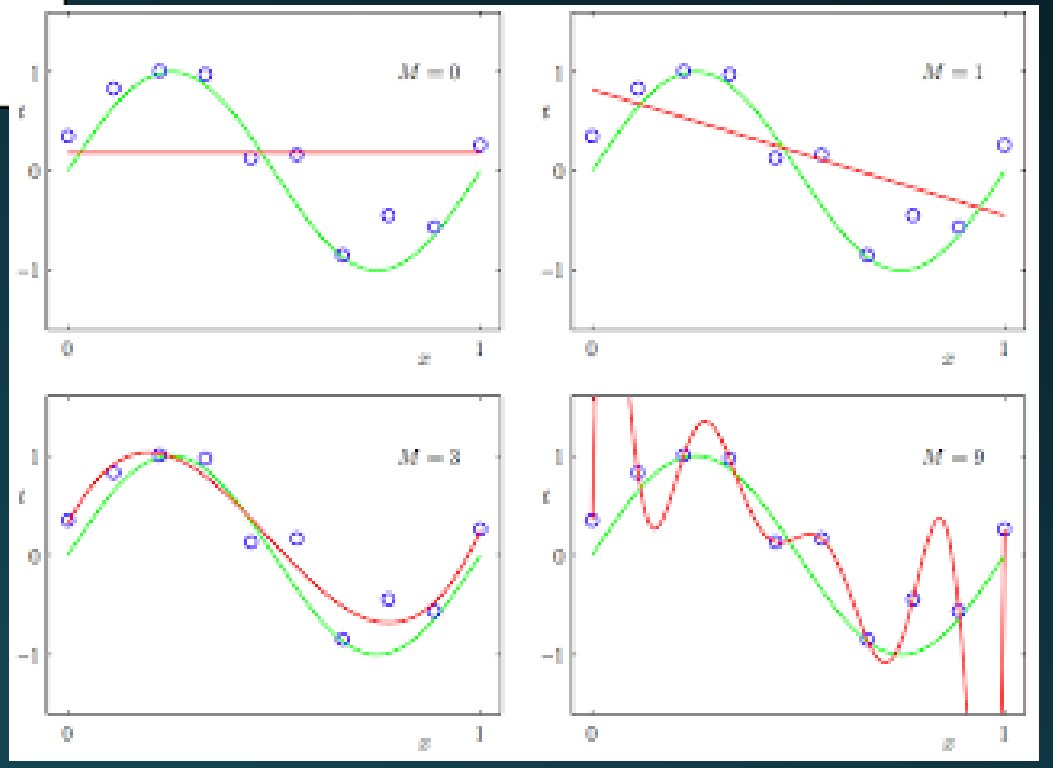
Вдруг мы подберем такой алгоритм, такие параметры модели, что он будет хорошо предсказывать данные на обучающей выборке, а на всех остальных плохо?

Такое явление называется переобучением (overfitting)!



# Пример переобучения



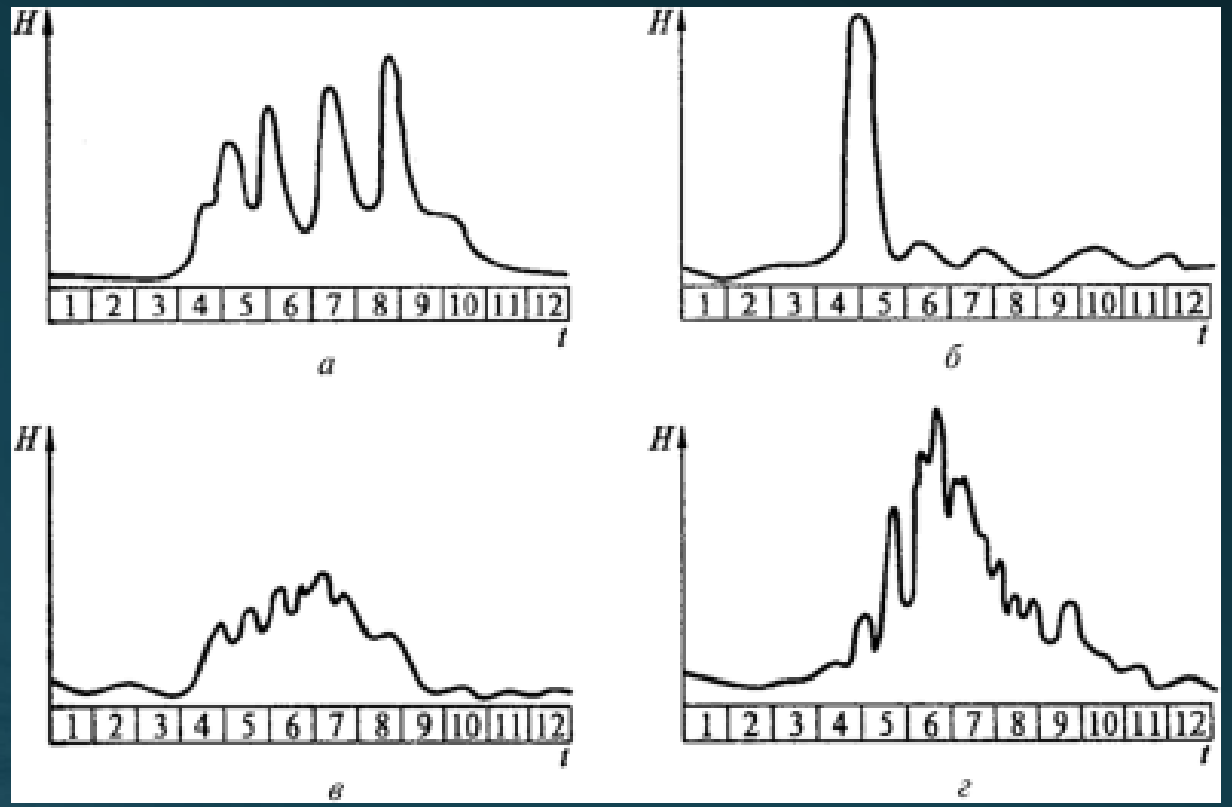
# Как бороться с переобучением?

- Вводить ограничения на параметры модели (ограничение классов моделей, регуляризация)
- Метод выкинутой точки (Leave one out, LOO)
- Метод кросс-валидации (cross-validation, CV)


Полученные оценки качества модели усредняются, тем самым получается среднее качество модели на обучающей выборке

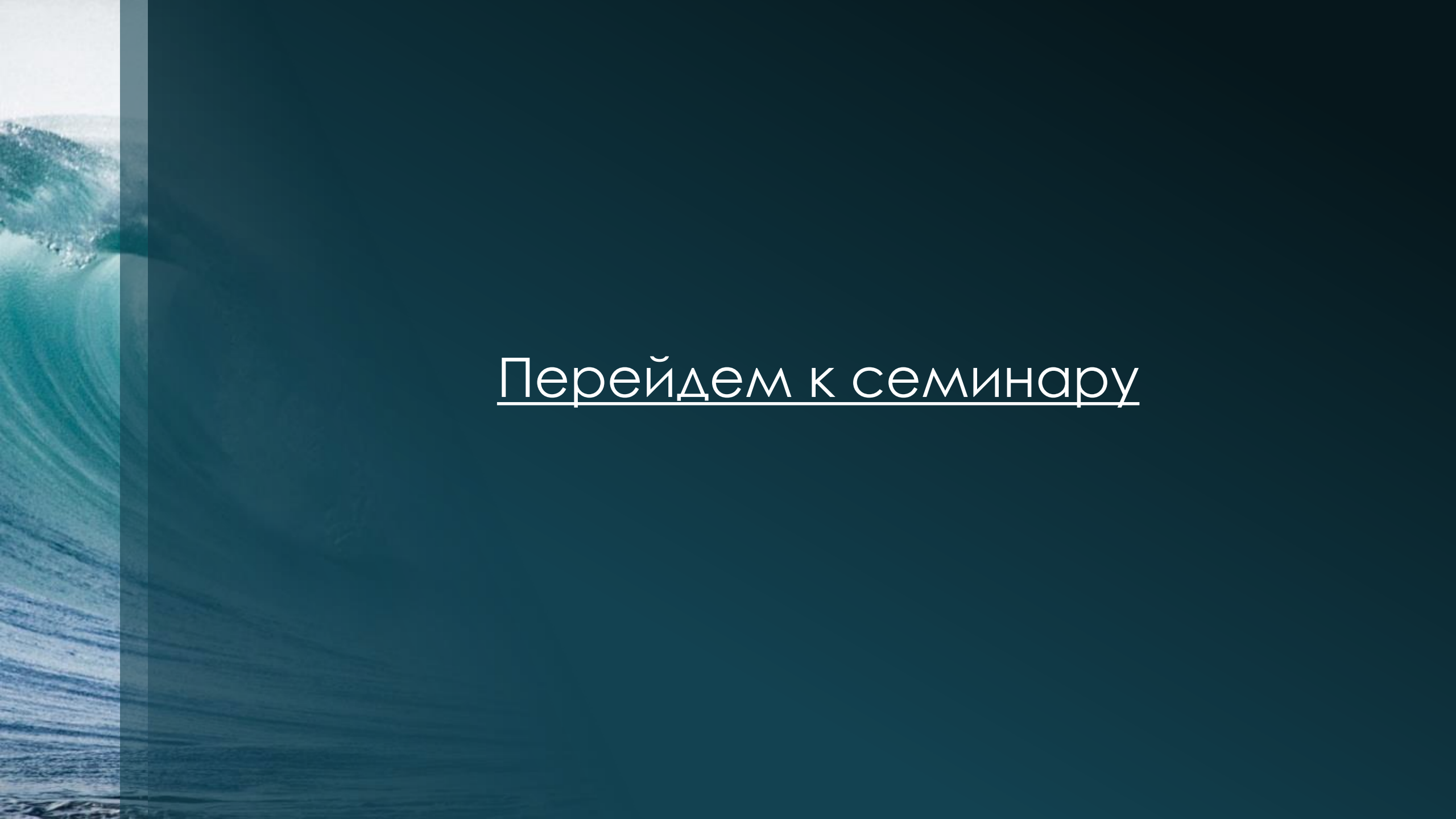
# Задачи классификации

1. Предсказание образования ледового затора
2. Определение характера питания реки по гидрографу



# Задачи регрессии

1. Предсказание желаемой переменной по метеорологическим и метеорологическим наблюдениям
   - Предсказание сезонно меженного стока по максимальному уровню грунтовых вот в зимний период (Draper and Smith, 1966)
   - Моделирование толщины снежного покрова (Айзель Г.В. 2016)
   - Зависимость слоя стока от слоя атмосферных осадков (Иофин З.К. 2018)


Перейдем к семинару