Введение в АД

О курсе

ThetaHat

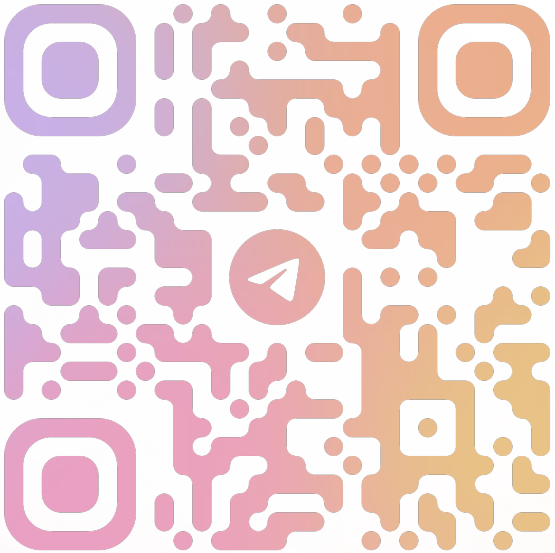
Проводимые нами учебные курсы

- Введение в анализ данных;
- DS-поток;
- Phystech@DataScience (в т.ч. статистика на ЛФИ);
- Статистика и машинное обучение на ФБМФ БТ;
- Мат. статистика на ФБМФ и ФЭФМ (семинары);
- Прикладная статистика на кафедрах в магистратуре;
- Машинное обучение в ШАД (частично);
- АБ-тестирование в ШАД.

Организационная информация

Телеграм-бот

@thetahat_ds25_bot



Код регистрации **Loss=0.193**

Сайт команды thetahat.ru

Почта thetahat@yandex.ru

Цели курса

1. **Дать представление об анализе данных;**
2. Обучить базовым инструментам анализа данных;
3. Рассказать о практическом смысле объектов теории вероятностей;
4. Помочь определиться с кафедрой.

Яндекс

В курсе предполагается участие кафедры анализа данных:

- Гостевая лекция от Яндекса;
- Работа по курсу учитывается при отборе на кафедру.

План занятий (О — обязательная, Ф — факультативная)

Дата		Тема	Лектор	Дедлайн по ДЗ
08.02	**О**	*Введение*	 Никита Волков	01.03
15.02		*Занятия нет, делаем домашки*		
22.02	**О**	*AI-инструменты*	 Валерия Я. Елизавета Д.	01.03
01.03	**О**	*Линейные модели*	 Никита Волков	15.03
8.03		*Занятия нет, делаем домашки*		
15.03	**О**	*Введение в нейросети*	 Елизавета Дахова	22.03
22.03	**Ф**	*Компьютерное зрение и генеративные модели*	 Елизавета Дахова	29.03

План занятий (О — обязательная, Ф — факультативная)

Дата		Тема	Лектор	Дедлайн по ДЗ
29.03	**Ф**	*Обработка текстов*	 Денис Титов	05.04
05.04	**Ф**	*Кластеризация и пониж. размерности*	 Никита Волков	19.04
12.04	**О**	*Гостевая лекция от Яндекса*		
19.04	**О**	*Теория вероятностей на практике*	 Дарья Плотникова	26.04
26.04	**Ф**	*Статистика*	 Дарья Плотникова	03.05
03.05	**Ф**	*Аналитика*	 Валерия Якупова	10.05


ChatGPT


Gemini


YaGPT


deepseek


GIGA
CHAT


Claude

Не можешь предотвратить — возглавь!

Использование ИИ-инструментов в курсе

При выполнении *технической* работы в домашних заданиях **рекомендуется** использовать ИИ-инструменты!

Как? Узнаете на второй лекции!

Ограничения

1. ИИ может ошибаться. Его ошибка в вашей работе — ваша ошибка. Вам необходимо понимать и перепроверять ответы ИИ.
   Аргументы "мне так сказал ИИ" не принимаются.
2. Всю содержательную работу по задаче
   вам необходимо делать самостоятельно.
3. Злоупотребление ИИ приравнивается к списыванию.
4. Ваша цель — обучиться.
   Используйте ИИ для выполнения этой цели.

Система оценивания обязательной части

Официальное название: Введение в анализ данных

Кафедра дискретной математики

Активности:

- Л — доля выполнения легких заданий (их немного);
- С — доля выполнения сложных заданий (их много);
- ЛС — доля выполнения всех домашних заданий;
- В — доля правильных ответов на вопросы в боте на занятии;
- Т — доля выполнения тестов (на "удовл", в мае);

Списывания:

- Штраф -**2 балла за каждый случай всем участникам**;
- Объяснение "мы просто общались" не прокатит;
- Злоупотребление ИИ приравнивается к списыванию.

Система оценивания обязательной части

Правила:

Ставится максимальная оценка X, для которой A & (B | C) = True.

Оценка X	Условие A	Условие B	Условие C
3	Т > 34%		
4	Т > 67%		
5	Л > 25%		
6	Л > 50%		
7	Л > 75%		
8	Л > 25% и С > 25%	ЛС > 50% и В > 50%	ЛС > 75%
9	Л > 25% и С > 25%	ЛС > 65% и В > 65%	ЛС > 85%
10	Л > 25% и С > 25%	ЛС > 80% и В > 80%	ЛС > 95%

Система оценивания факультативной части

Официальное название: Введение в анализ данных: доп. главы

Кафедра дискретной математики

Правила:

1. **О = 9*Ф+2*В**

Обозначения:

- Ф — доля выполнения домашних заданий;
- В — доля правильных ответов на вопросы в боте на занятии;
- О — итоговая оценка, округляется вверх.

Как записаться?

Просто ходить на занятия и сдавать домашние задания.

В конце семестра разошлем форму, в которой вы отметитесь, что хотите поставить оценку в ведомость.

Для кого наш курс?

Обязательная часть курса:

- ПМИ — курс обязателен.
- ПМФ, ИВТ, гр. 351, 352 — курс факультативен.
- Студенты других физтех-школ также могут сдавать курс.

  Для записи достаточно зарегистрироваться в боте.

  Для проставления оценки нужно в конце семестра взять ведомость.

Факультативная часть курса:

- Факультативно для всех.

В силу ограниченных возможностей проверяющих при большом количестве желающих возможность сдавать курс может быть ограничена.

Правила комфорта

- Постарайтесь задавать вопросы на занятии в тот момент, когда это актуально, не перебивая на полуслове.

  Другой вопрос лучше задать в перерыве или после занятия.

- Цените труд проверяющих :)

  В каком из случаев проверяющему больше захочется пойти навстречу автору вопроса?

  - "*Объясните вашу претензию, почему вы мне сняли баллы, я же все сделал, я не согласен*"
  - "*Добрый день! По такой-то задаче вы написали ..., но я считаю ..., потому что ..., и у меня в работе написано ...*"

Особенности проверки домашних заданий

Иногда к нам приходят отзывы:

"*Проверяющие проверяют рандомно,*

за одну и ту же ошибку разным студентам

сняли разное количество баллов."

Пусть на курсе 300 человек,

каждый проверяющий проверяет 30 работ.

Сколько результатов проверок нужно ему посмотреть для сравнения?

Сколько всего таких сравнений?

Оцените количество времени, которое нужно затратить.

Особенности проверки домашних заданий

Как мы решаем проблему?

- Общие критерии для всех проверяющих в табличке,
  с общим текстом и количеством баллов.
  Проверяющему достаточно поставить галочку.
- Сравнение частоты применения критерия между проверяющими
  с помощью статистического t-test'а.
- Сравнение среднего балла между проверяющими
  с помощью статистического t-test'а.
- Применяем разработанный нами ML-пайплайн,
  который ищет похожие комментарии проверяющих.

Мы стараемся, но мы не волшебники :)

Если вы заметили несправедливость проверки, пожалуйста,
напишите нам. Мы посмотрим и при необходимости поправим.
И учтем это для совершенствования наших методов проверки.

ThetaGrader

ThetaGrader — система автоматической проверки домашних заданий с помощью технологий искусственного интеллекта, разрабатываемая командой ThetaHat.

Она будет помогать проверять ваши домашки!

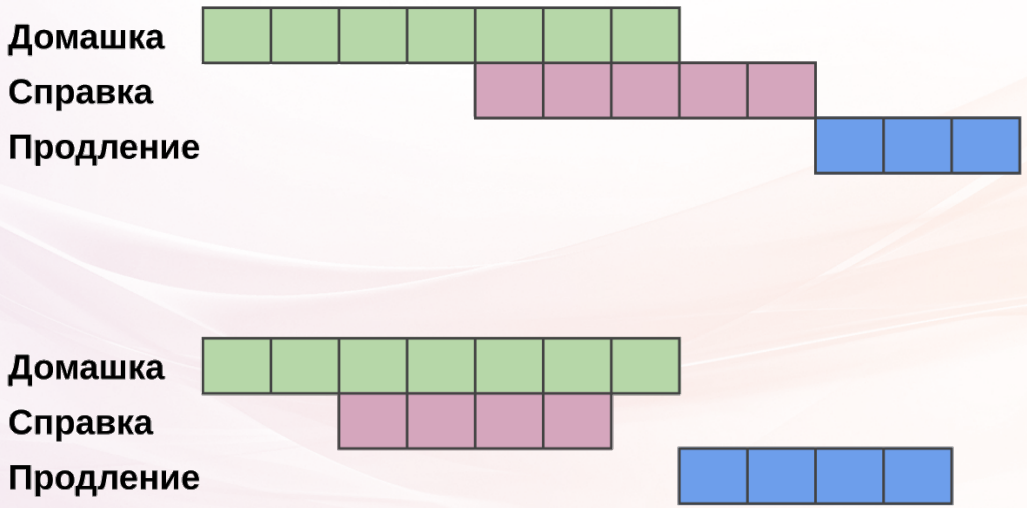
Переносы дедлайнов по уважительным причинам

Уважительные причины

- Медицинская справка с подписью и печатью.
- Приказ по институту об освобождении.

На сколько можно перенести

На количество дней пересечения интервала выполнения задания и датам по справке от даты дедлайна или окончания справки.



Тех. ассистент

Анна Бурханова

tg: @Anches_here

- Организация проверки домашних заданий
- Перенос дедлайнов по уважительным причинам
- Разные технические вопросы

DS-поток

Программа 3-4 курсов

Продвинутый анализ данных

DS-поток

Семестр	DS-поток	Основной поток ПМИ
5	Математическая статистика	Математическая статистика
	Машинное обучение	Машинное обучение
	Практика	*Практика по мат. статистике*
	Основы прикладной статистики	Курс по выбору x 2
	Курс по выбору	
6	Дискр. случ. процессы и временные ряды	Случайные процессы
	Глубокое обучение и его приложения	Вычислительная математика
	Прикладная статистика и анализ данных	Параллельные и распределенные вычисления
	Практика	Курс по выбору
Кафедра АД	Курс ШАД	Методы прикладной статистики
7	Байесовский подход в анализе данных	Курс по выбору x 2
	Практика	
8	Прикладные задачи машинного обучения	Курс по выбору x 2
	Практика	

Примечания.

1. По одним и тем же курсам лекторы разные.
2. По факту практика не является отдельным курсов.
3. В процессе обучения перейти в DS-поток невозможно.

DS-поток

Семестр	DS-поток	Основной поток ПМФ
5	Математическая статистика	Математическая статистика
	Машинное обучение	Вычислительная математика
	Практика	Курс по выбору
	Основы прикладной статистики	
6	Дискр. случ. процессы и временные ряды	Случайные процессы
	Глубокое обучение и его приложения	Квантовая механика, ч. 2
	Прикладная статистика и анализ данных	Теория и реализация языков программирования
	Практика	Курс по выбору x 2
Кафедра АД	Курс ШАД	Методы прикладной статистики
7	Байесовский подход в анализе данных	Машинное обучение
	Практика	Курс по выбору
8	Прикладные задачи машинного обучения	Курс по выбору x 2
	Практика	

Примечания.

1. По одним и тем же курсам лекторы разные.
2. По факту практика не является отдельным курсов.
3. В процессе обучения перейти в DS-поток невозможно.

DS-поток

Чему мы учим

1. В меру глубокое математическое понимание статистики и машинного обучения.
2. Применение математических моделей на реальных данных, в том числе на реальных задачах.
3. Умение составлять полноценные выводы.

Реальная практика на DS-потоке

1. Реальные примеры из практики;
2. Соревнования на Kaggle;
3. Лекторы, применяющие анализ данных на практике;
4. Разбор статей на тему анализа данных;
5. Разработка DS-проектов.

Отбор на DS-поток

Что будет учитываться?

1. **Необходимое условие:**
   оценка не менее отл(8) по обяз. части курса
   "Введение в анализ данных" и не менее хор(7) по факульт. части.
2. Работа в семестре по курсу, грамотное оформление ДЗ.
3. Оценка по теории вер., в меньшей степени — другие предметы.

Что нужно делать для отбора?

1. Трудиться в течение семестра.
2. В июне подать заявку.
3. Ждать. Результаты летом.

DS-поток адаптирован для ПМИ.Классика и ПМФ.КТ.
Студенты других напр. могут попасть на DS-поток,
но возможна доп. нагрузка и проблемы с расписанием.

Другие образовательные направления по АД

Школа анализа данных Яндекса (ШАД)
Двухгодовая программа дополнительного образования, специализирующаяся на анализе данных, разработке ML-моделей, создавать систем хранения и обработки больших данных и др..
Независима по отношению к обучению в МФТИ.
Подробнее: shad.yandex.ru

Кафедра анализа данных (Яндекс) в рамках ФПМИ

- По 2-3 курса в семестр в основном из курсов ШАДа.
- Написание и защита диплома.

Другие кафедры в рамках ФПМИ

- По 2-3 курса в семестр.
- Написание и защита диплома.

Куда пойти?

2 курс

Введение в анализ данных

3-4 курсы

DS-поток

Кафедра анализа данных

Школа анализа данных (ШАД)

1. Если анализ данных интересен, то хорошее решение: **DS-поток + кафедра анализа данных.**
2. Если выбираете кафедру анализа данных, то кафедра рекомендует пойти на DS-поток.
3. **По результатам нашего курса кафедра анализа данных может зачесть тех. собеседование при отборе на кафедру.**

Бонусы при отборе на кафедру АД

Этапы отбора:

1. Вступительный экзамен (математика, алгоритмы)
2. Техническое собеседование (математика, алгоритмы)
3. Мотивационное собеседование

Техническое собеседование засчитывается автоматом, если

1. Экзамен написан достаточно хорошо
2. Оценки ОТЛ за обе части курса Введение в АД
3. Среднее оценок по Введение в АД и ср. балла не менее 4 из 5

Если пройти на DS-поток:

1. Если оценки ОТЛ или ХОР за Введение в АД и по DS-потоку, то *техническое собеседование засчитывается*
2. Если попал в топ-5 в рейтинге до ДЗ на DS-потоке, то *экзамен и техническое собеседование засчитываются*

Кафедра по своему усмотрению может назначить собеседование.

Что такое анализ данных?

Кому нужен анализ данных

1. "*Я математик, практическое применение не интересует*".

   Скорее всего АД не нужен,
   но часто математики им начинают интересоваться.

2. "*Я математик, но хочу применять свои знания на практике*".

   АД для вас, ждем вас на DS-потоке

3. "*Я программист, и хочу писать только код*".

   Скорее всего АД в подробностях не нужен,
   но стоит понимать, чем занимаются коллеги-аналитики.

4. "*Я программист, но хочу глубоко разбираться в тонкостях математических методов*".

   АД для вас, ждем вас на DS-потоке

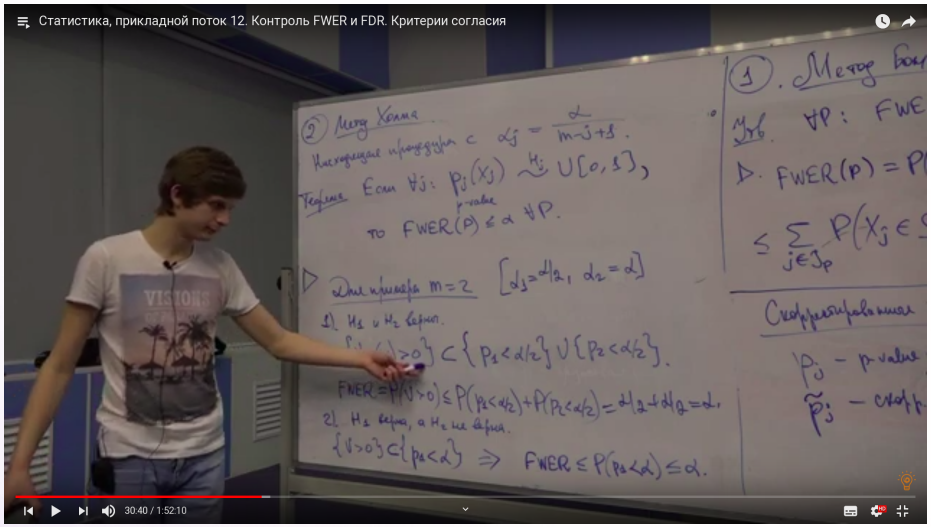
Так что, анализ данных

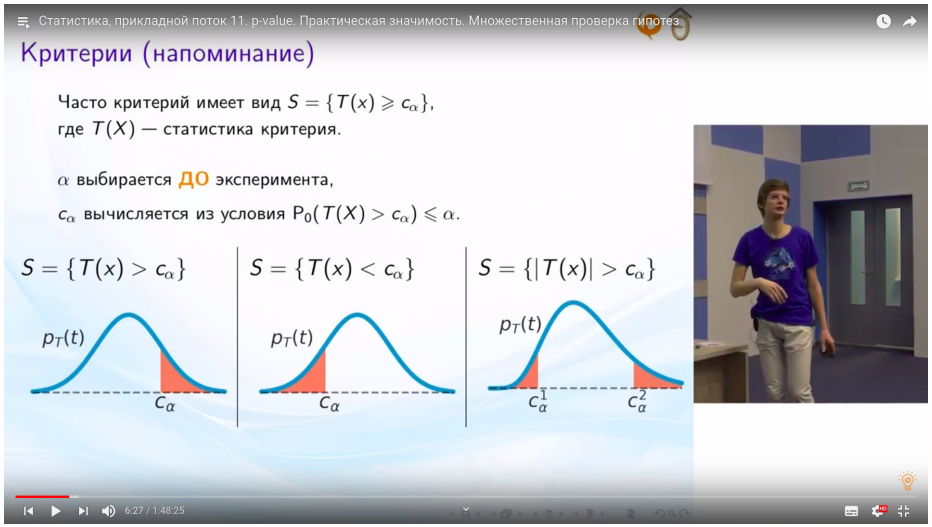
это математика

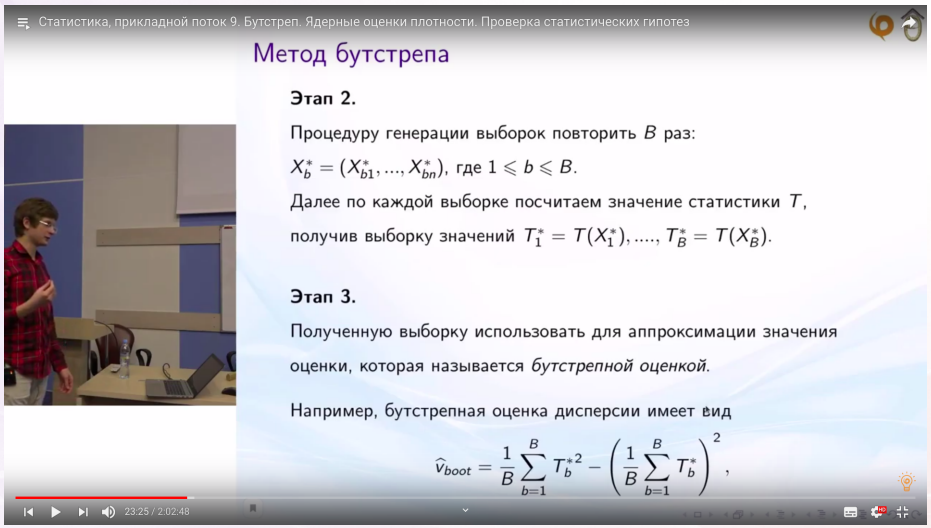
или программирование?

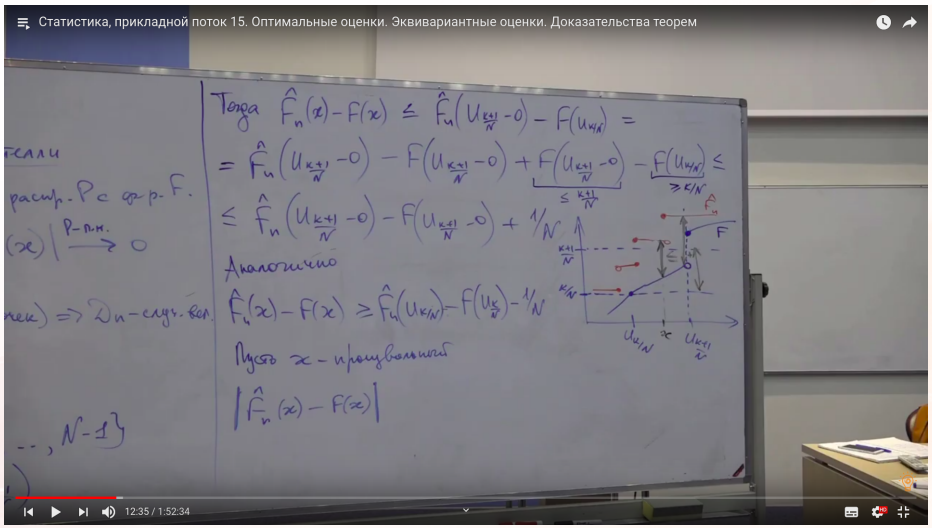
Давайте разбираться...

Посмотрим на лекции по статистике

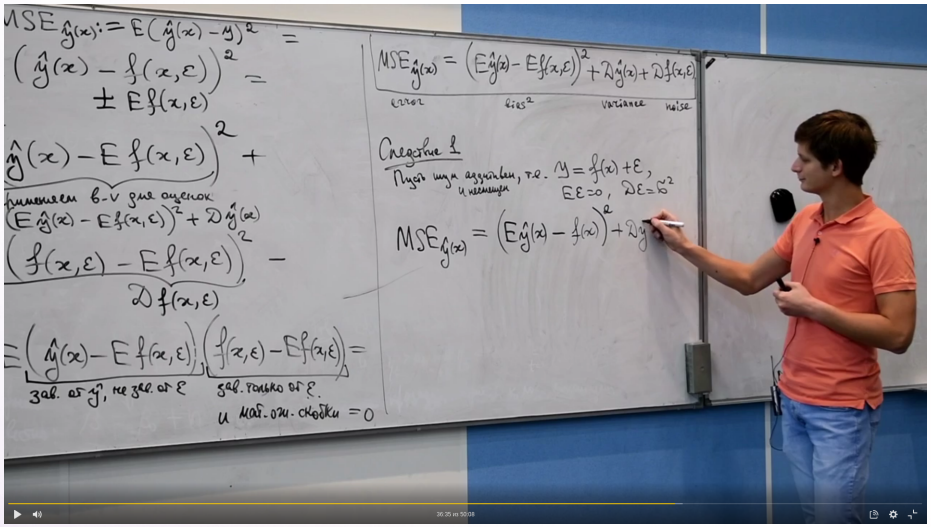








Посмотрим на лекции по машинному обучению



Двуслойная нейронная сеть

Матричное представление

Пусть $x = (x_1, x_2, ..., x_d)^T$ — элемент выборки,

$u = (u_1, u_2, ..., u_H)^T$ — выход I слоя, $y = (y_1, y_2, ..., y_M)^T$ — выход II слоя,

$W_1 = (w_{1|jh})_{jh}$ — м-ца весов I слоя, $W_2 = (w_{2|hm})_{hm}$ — м-ца весов II слоя,

$b_1 = (b_{1|1}, ..., b_{1|H})^T$ — в-р сдвигов I слоя, $b_2 = (b_{2|1}, ..., b_{2|H})^T$ — в-р сдвигов II слоя.

Тогда работу двуслойной нейронной сети можно представить как:

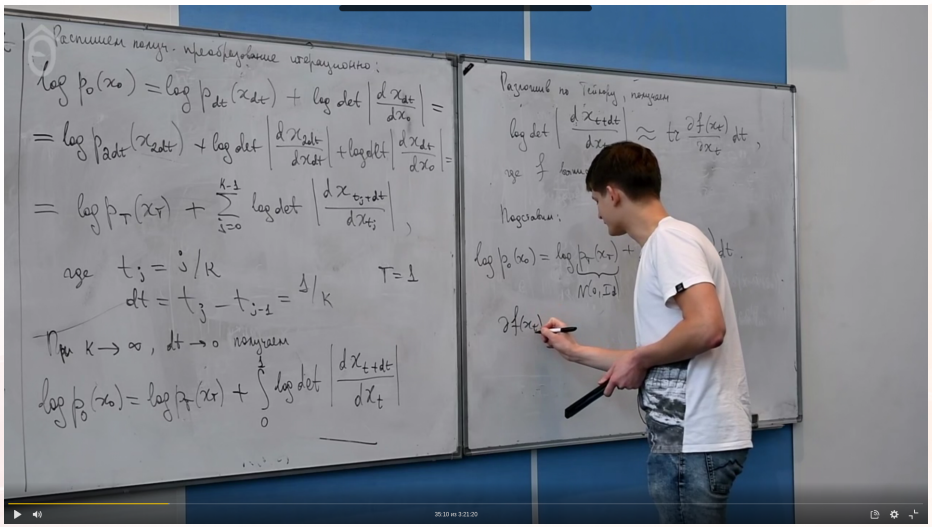
1) $u = \sigma_1(x^T W_1 + b_1)$

2) $y = \sigma_2\left(u^T W_2 + b_2\right) = \boxed{\sigma_2\left(\sigma_1\left(x^T W_1 + b_1\right)^T W_2 + b_2\right)}$

Classifier guidance

Чтобы явно включить информацию о классе в процесс диффузии, можно обучить классификатор $f_\phi(y|x_t, t)$ на зашумленном изображении x_t и использовать градиенты $\nabla_x \log f_\phi(y|x_t)$, чтобы направлять процесс диффузионного семплирования к информации об условии y путем изменения прогнозирования шума. Можно вывести, что $\nabla_x \log q(x_t) = -\frac{1}{\sqrt{(1-\overline{\alpha}_t)}}\varepsilon_\theta(x_t, t)$.

$$\nabla_{x_t} \log q(x_t, y) = \nabla_{x_t} \log q(x_t) + \nabla_{x_t} \log q(y|x_t) \simeq$$
$$-\frac{1}{\sqrt{1-\overline{\alpha}_t}}\varepsilon_\theta(x_t, t) + \nabla_{x_t} \log f_\phi(y|x_t) =$$
$$-\frac{1}{\sqrt{1-\overline{\alpha}_t}}(\varepsilon_\theta(x_t, t) - \sqrt{1-\overline{\alpha}_t}\,)\nabla_{x_t} \log f_\phi(y|x_t)$$



Посмотрим на научные статьи

McInnes, L, Healy, J, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints 1802.03426, 2018

The choice of non-expansive maps in Definition 6 is due to Spiv
note that it closely mirrors the work of Carlsson and Memoli in [1
logical methods for clustering as applied to finite metric spaces. Th
significant since pure isometries are too strict and do not provide la
Hom-sets.

In [43] Spivak constructs a pair of adjoint functors, Real and Si
the categories **sFuzz** and **EPMet**. These functors are the natural e
the classical realization and singular set functors from algebraic top
functor Real is defined in terms of standard fuzzy simplices $\Delta^n_{<a}$ as

$$\mathsf{Real}(\Delta^n_{<a}) \triangleq \left\{ (t_0, \ldots, t_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} t_i = -\log(a), t_i \geq \right.$$

similarly to the classical realization functor $|\cdot|$. The metric on R
simply inherited from $\mathbb{R}^{n+1}$. A morphism $\Delta^n_{<a} \to \Delta^m_{<b}$ exists only if
is determined by a $\mathbf{\Delta}$ morphism $\sigma : [n] \to [m]$. The action of Rea
morphism is given by the map

$$(x_0, x_1, \ldots, x_n) \mapsto \frac{\log(b)}{\log(a)} \left(\sum_{i_0 \in \sigma^{-1}(0)} x_{i_0}, \sum_{i_0 \in \sigma^{-1}(1)} x_{i_0}, \ldots, \sum_{i_0 \in \sigma^{-1}(m)} x_{i_0} \right).$$

Such a map is clearly non-expansive since $0 \leq a \leq b \leq 1$ implies that $\log(b)/\log$
1.

We then extend this to a general simplicial set X via colimits, defining

$$\mathsf{Real}(X) \triangleq \operatorname*{colim}_{\Delta^n_{<a} \to X} \mathsf{Real}(\Delta^n_{<a}).$$

Since the functor Real preserves colimits, it follows that there exists a righ
adjoint functor. Again, analogously to the classical case, we find the right adjoint
denoted Sing, is defined for an extended pseudo metric space Y in terms of its
action on the category $\mathbf{\Delta} \times I$:

$$\mathsf{Sing}(Y) : ([n], [0, a)) \mapsto \hom_{\mathbf{EPMet}}(\mathsf{Real}(\Delta^n_{<a}), Y).$$

For our case we are only interested in finite metric spaces. To correspond with
this we consider the subcategory of bounded fuzzy simplicial sets **Fin-sFuzz**. W
therefore use the analogous adjoint pair FinReal and FinSing. Formally we define
the finite fuzzy realization functor as follows:

Algorithm 2 Constructing a local fuzzy simplicial set

function LocalFuzzySimplicialSet(X, x, n)
  knn, knn-dists ← ApproxNearestNeighbors(X, x, n)
  ρ ← knn-dists[1] ▷ Distance to nearest neighbor
  σ ← SmoothKNNDist(knn-dists, n, ρ) ▷ Smooth approximator to knn-distance
  fs-set$_0$ ← X
  fs-set$_1$ ← $\{([x, y], 0) \mid y \in X\}$
  for all $y \in$ knn **do**
    $d_{x,y} \leftarrow \max\{0, \mathrm{dist}(x, y) - \rho\}/\sigma$
    fs-set$_1$ ← fs-set$_1 \cup ([x, y], \exp(-d_{x,y}))$
  return fs-set

Algorithm 3 Compute the normalizing factor for distances σ

function SmoothKNNDist(knn-dists, n, ρ)
  Binary search for σ such that $\sum_{i=1}^{n} \exp(-(\text{knn-dists}_i - \rho)/\sigma) = \log_2(n)$
  return σ

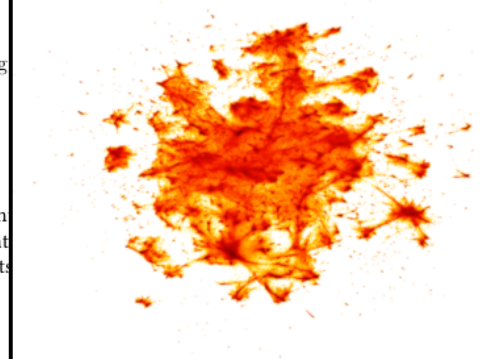


Figure 8: Visualization of the full 3 million word vectors from the GoogleNews dataset as embedded by UMAP.

is contained in U, then g is constant in B and hence $\sqrt{\det(g)}$ is constant can be brought outside the integral. Thus, the volume of B is

$$\sqrt{det(g)} \int_B dx^1 \wedge \ldots \wedge dx^n = \sqrt{det(g)} \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)},$$

re r is the radius of the ball in the ambient $\mathbb{R}^n$. If we fix the volume of the to be $\frac{\pi^{n/2}}{\Gamma(n/2+1)}$ we arrive at the requirement that

$$\det(g) = \frac{1}{r^{2n}}.$$

, since g is assumed to be diagonal with constant entries we can solve for g f as

$$g_{ij} = \begin{cases} \frac{1}{r^2} & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

geodesic distance on $\mathcal{M}$ under g from p to q (where $p, q \in B$) is defined as

$$\inf_{c \in C} \int_a^b \sqrt{g(\dot{c}(t), \dot{c}(t))} dt,$$

where C is the class of smooth curves c on $\mathcal{M}$ such that $c(a) = p$ and $c(b) = q$, and $\dot{c}$ denotes the first derivative of c on $\mathcal{M}$. Given that g is as defined in (2) we see that this can be simplified to

$$\begin{aligned} &\frac{1}{r} \inf_{c \in C} \int_a^b \langle \sqrt{\dot{c}(t), \dot{c}(t)} \rangle dt \\ =&\frac{1}{r} \inf_{c \in C} \int_a^b \langle \|\dot{c}(t), \dot{c}(t)\| dt \\ =&\frac{1}{r} d_{\mathbb{R}^n}(p, q). \end{aligned} \tag{3}$$

□

B Proof that FinReal and FinSing are adjoint

Theorem 2. *The functors* FinReal : **Fin-sFuzz** → **FinEPMet** *and* FinSing : **FinEPMet** → **Fin-sFuzz** *form an adjunction with* FinReal *the left adjoint and* FinSing *the right adjoint.*

Посмотрим на научные статьи

Diederik P Kingma, Max Welling: *Auto-Encoding Variational Bayes*, ArXiv 1312.6114, 2014

2.2 The variational bound

The marginal likelihood is composed of a sum over the marginal likelihoods of individual datapoints $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$, which can each be rewritten as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \quad (1)$$

The first RHS term is the KL divergence of the approximate from the true posterior. Since this KL-divergence is non-negative, the second RHS term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ is called the (variational) *lower bound* on the marginal likelihood of datapoint i, and can be written as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right] \quad (2)$$

which can also be written as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}$$

We want to differentiate and optimize the lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}$ parameters $\boldsymbol{\phi}$ and generative parameters $\boldsymbol{\theta}$. However, the gradien is a bit problematic. The usual (naïve) Monte Carlo gradient esti is: $\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}[f(\mathbf{z})] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}[f(\mathbf{z}) \nabla_{q_{\boldsymbol{\phi}}(\mathbf{z})} \log q_{\boldsymbol{\phi}}(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^{L} f($ $\mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$. This gradient estimator exhibits exhibits very hi and is impractical for our purposes.

2.3 The SGVB estimator and AEVB algorithm

In this section we introduce a practical estimator of the lower boun parameters. We assume an approximate posterior in the form $q_{\phi}($ technique can be applied to the case $q_{\boldsymbol{\phi}}(\mathbf{z})$, i.e. where we do not con variational Bayesian method for inferring a posterior over the parame

Under certain mild conditions outlined in section 2.4 for a chosen app can reparameterize the random variable $\tilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ using a differen of an (auxiliary) noise variable $\boldsymbol{\epsilon}$:

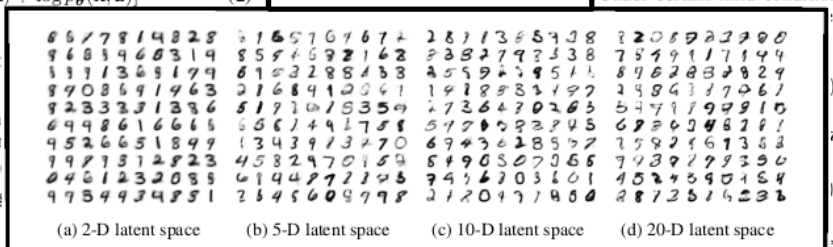
$$\tilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

See section 2.4 for general strategies for chosing such an approriate $g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$. We can now form Monte Carlo estimates of expectations $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ as follows:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}[f(\mathbf{z})] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))\right] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)}$$

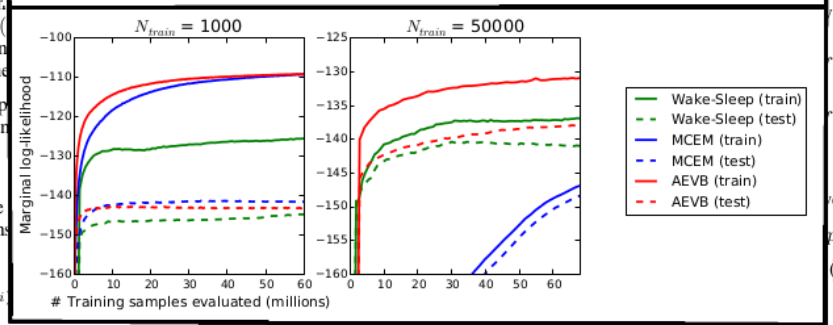
We apply this technique to the variational lower bound (eq. (2)), yielding our generic Stochastic Gradient Variational Bayes (SGVB) estimator $\tilde{\mathcal{L}}^{A}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \simeq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$:

$$\tilde{\mathcal{L}}^{A}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \frac{1}{L} \sum^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)})$$



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.



$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x})$ is the variational lower bound of the marginal likelihood of datapoint i:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) d\mathbf{z} \quad (16)$$

The expectations on the RHS of eqs (14) and (16) can obviously be written as a sum of three separate expectations, of which the second and third component can sometimes be analytically solved, e.g. when both $p_{\boldsymbol{\theta}}(\mathbf{x})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ are Gaussian. For generality we will here assume that each of these expectations is intractable.

Under certain mild conditions outlined in section (see paper) for chosen approximate posteriors parameterize conditional samples $\tilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ as

$$\tilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \quad (17)$$

and a function $g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$ such that the following holds:

$$\mathbf{z}|\mathbf{x}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) d\mathbf{z}$$

$$\left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right)\Big|_{\mathbf{z}=g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)})} d\boldsymbol{\epsilon} \quad (18)$$

approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$:

$$\tilde{\boldsymbol{\theta}} = h_{\boldsymbol{\phi}}(\boldsymbol{\zeta}) \quad \text{with} \quad \boldsymbol{\zeta} \sim p(\boldsymbol{\zeta}) \quad (19)$$

e, choose a prior $p(\boldsymbol{\zeta})$ and a function $h_{\boldsymbol{\phi}}(\boldsymbol{\zeta})$ such that the following

$$q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right) d\boldsymbol{\theta}$$

$$p(\boldsymbol{\zeta}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right)\Big|_{\boldsymbol{\theta}=h_{\boldsymbol{\phi}}(\boldsymbol{\zeta})} d\boldsymbol{\zeta} \quad (20)$$

e introduce a shorthand notation $f_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \quad (21)$$

(18), the Monte Carlo estimate of the variational lower bound, given

$$\mathcal{L}(\boldsymbol{\phi}; \mathbf{X}) \simeq \frac{1}{L} \sum_{l=1}^{L} f_{\boldsymbol{\phi}}(\mathbf{x}^{(l)}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(l)}), h_{\boldsymbol{\phi}}(\zeta^{(l)})) \quad (22)$$

where $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$ and $\zeta^{(l)} \sim p(\boldsymbol{\zeta})$. The estimator only depends on samples from $p(\boldsymbol{\epsilon})$ and $p(\boldsymbol{\zeta})$ which are obviously not influenced by $\boldsymbol{\phi}$, therefore the estimator can be differentiated w.r.t. $\boldsymbol{\phi}$.

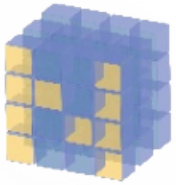
Посмотрим на примеры использования библиотек

Variational Autoencoders

https://github.com/pyro-ppl/pyro/blob/dev/examples/vae/vae.py

```python
# Copyright (c) 2017-2019 Uber Technologies, Inc.
# SPDX-License-Identifier: Apache-2.0

import argparse

import numpy as np
import torch
import torch.nn as nn
import visdom

import pyro
import pyro.distributions as dist
from pyro.infer import SVI, JitTrace_ELBO, Trace_ELBO
from pyro.optim import Adam
from utils.mnist_cached import MNISTCached as MNIST
from utils.mnist_cached import setup_data_loaders
from utils.vae_plots import mnist_test_tsne, plot_llk, plot_vae_samp


# define the PyTorch module that parameterizes the
# diagonal gaussian distribution q(z|x)
class Encoder(nn.Module):
    def __init__(self, z_dim, hidden_dim):
        super().__init__()
        # setup the three linear transformations used
        self.fc1 = nn.Linear(784, hidden_dim)
        self.fc21 = nn.Linear(hidden_dim, z_dim)
        self.fc22 = nn.Linear(hidden_dim, z_dim)
        # setup the non-linearities
        self.softplus = nn.Softplus()

    def forward(self, x):
        # define the forward computation on the image x
        # first shape the mini-batch to have pixels in the rightmost
        x = x.reshape(-1, 784)
        # then compute the hidden units
        hidden = self.softplus(self.fc1(x))
        # then return a mean vector and a (positive) square root cov
        # each of size batch_size x z_dim
        z_loc = self.fc21(hidden)
        z_scale = torch.exp(self.fc22(hidden))
        return z_loc, z_scale


# define the PyTorch module that parameterizes the
# observation likelihood p(x|z)
class Decoder(nn.Module):
    def __init__(self, z_dim, hidden_dim):
        super().__init__()
        # setup the two linear transformations used
        self.fc1 = nn.Linear(z_dim, hidden_dim)
        self.fc21 = nn.Linear(hidden_dim, 784)
        # setup the non-linearities
        self.softplus = nn.Softplus()

    def forward(self, z):
        # define the forward computation on the latent z
        # first compute the hidden units
        hidden = self.softplus(self.fc1(z))
        # return the parameter for the output Bernoulli
        # each is of size batch_size x 784
        loc_img = torch.sigmoid(self.fc21(hidden))
        return loc_img


# define a PyTorch module for the VAE
class VAE(nn.Module):
    # by default our latent space is 50-dimensional
    # and we use 400 hidden units
    def __init__(self, z_dim=50, hidden_dim=400, use_cuda=False):
        super().__init__()
        # create the encoder and decoder networks
        self.encoder = Encoder(z_dim, hidden_dim)
        self.decoder = Decoder(z_dim, hidden_dim)

        if use_cuda:
            # calling cuda() here will put all the parameters of
            # the encoder and decoder networks into gpu memory
            self.cuda()
        self.use_cuda = use_cuda
        self.z_dim = z_dim

    # define the model p(x|z)p(z)
    def model(self, x):
        # register PyTorch module `decoder` with Pyro
        pyro.module("decoder", self.decoder)
        with pyro.plate("data", x.shape[0]):
            # setup hyperparameters for prior p(z)
            z_loc = torch.zeros(x.shape[0], self.z_dim, dtype=x.dtype, dev
            z_scale = torch.ones(x.shape[0], self.z_dim, dtype=x.dtype, de
            # sample from prior (value will be sampled by guide when compu
            # decode the latent code z
            loc_img = self.decoder.forward(z)
            # score against actual images
            pyro.sample("obs", dist.Bernoulli(loc_img).to_event(1), obs=x
            # return the loc so we can visualize it later
            return loc_img

    # define the guide (i.e. variational distribution) q(z|x)
    def guide(self, x):
        # register PyTorch module `encoder` with Pyro
        pyro.module("encoder", self.encoder)
        with pyro.plate("data", x.shape[0]):
            # use the encoder to get the parameters used to define q(z|x)
            z_loc, z_scale = self.encoder.forward(x)
            # sample the latent code z
            pyro.sample("latent", dist.Normal(z_loc, z_scale).to_event(1)

    # define a helper function for reconstructing images
    def reconstruct_img(self, x):
        # encode image x
        z_loc, z_scale = self.encoder(x)
        # sample in latent space
        z = dist.Normal(z_loc, z_scale).sample()
        # decode the image (note we don't sample in image space)
        loc_img = self.decoder(z)
        return loc_img


def main(args):
    # clear param store
    pyro.clear_param_store()

    # setup MNIST data loaders
    # train_loader, test_loader
    train_loader, test_loader = setup_data_loaders(MNIST, use_cuda=args.c

    # setup the VAE
    vae = VAE(use_cuda=args.cuda)

    # setup the optimizer
    adam_args = {"lr": args.learning_rate}
    optimizer = Adam(adam_args)

    # setup the inference algorithm
    elbo = JitTrace_ELBO() if args.jit else Trace_ELBO()
    svi = SVI(vae.model, vae.guide, optimizer, loss=elbo)

    # setup visdom for visualizati
    if args.visdom_flag:
        vis = visdom.Visdom()

    train_elbo = []
    test_elbo = []
    # training loop
    for epoch in range(args.num_ep
        # initialize loss accumula
        epoch_loss = 0.
        # do a training epoch over
        # by the data loader
        for x, _ in train_loader:
            # if on GPU put mini-t
            if args.cuda:
                x = x.cuda()
            # do ELBO gradient and
            epoch_loss += svi.step

        # report training diagnost
        normalizer_train = len(tra
        total_epoch_loss_train = e
        train_elbo.append(total_ep
        print("[epoch %03d]  avera

        if epoch % args.test_frequ
            # initialize loss accu
            test_loss = 0.
            # compute the loss ove
            for i, (x, _) in enume
                # if on GPU put mi
                if args.cuda:
                    x = x.cuda()
                # compute ELBO est
                test_loss += svi.e

                # pick three rando
                # visualize how we
                if i == 0:
                    if args.visdom
                        plot_vae_s
                        reco_indic
                        for index
                            test_i
                            reco_i
                            vis.im
```

Какие вообще инструменты могут потребоваться



Вывод:

и математика

и программирование

Анализ данных это

процесс поиска закономерностей

в данных при помощи

- средств визуализации данных,
- математических методов,
- программных алгоритмов.

Искусственный интеллект

Отличительная особенность:

нет четко зафиксированного ответа на каждый входящий объект.

Что можно почитать:

Анализ данных — основы и терминология
https://habr.com/ru/post/352812/

Всё, что вам нужно знать об ИИ — за несколько минут
https://habr.com/ru/post/416889/

Сравним задачи

Алгоритмы и структуры данных

Задача: дан массив x, нужно его отсортировать.

Ровно один правильный ответ, можно
получить с помощью четких алгоритмов.

Комбинаторика

Задача: Сколько имеется способов раздать 11 разных цветков,
трём девушкам: какой-то – 5, а остальным – по 3 цветка? [ОКТЧ 2019]

Ровно один правильный ответ.

Анализ данных

Задача: Имеются данные $(x_1, y_1), ..., (x_n, y_n)$.
Восстановите по ним функцию $f : x \mapsto y$.

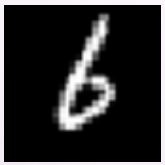
Особенности: нет четкого ответа, требуется только приближение,
но есть критерии качества.

Пример — распознавание рукописных цифр

Вход: 

Ожидается на выходе: **5**

Но как четко алгоритмически определить границу между 6 и 8?




 — 2 или 9?

 — 4 или 7?

Актуальность в научной среде

Число статей по запросам в Google Scholar с 2020:

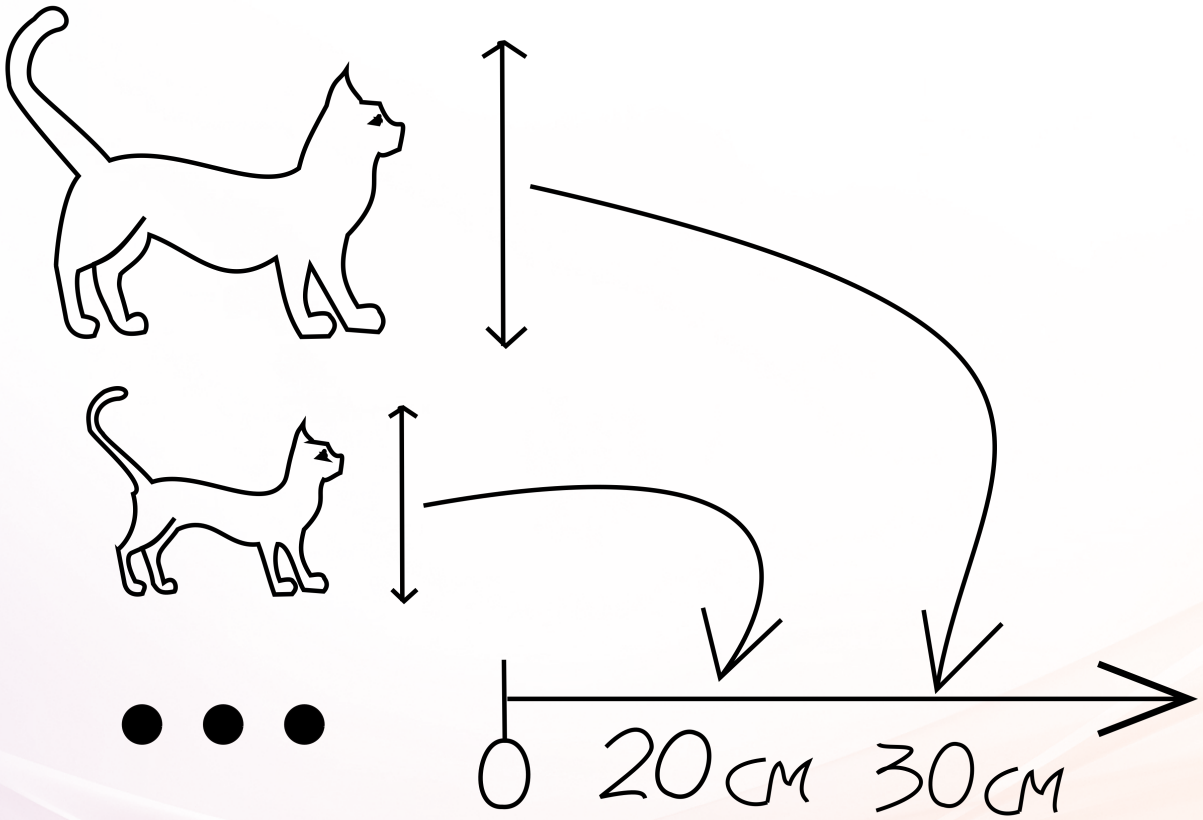
- *statistics* $\approx$ 1 240 000
- *machine learning* $\approx$ 1 040 000
- *computer vision* $\approx$ 537 000
- *deep learning* $\approx$ 236 000
- *generative models* $\approx$ 143 000
- *language models* $\approx$ 137 000
- *neural network* $\approx$ 108 000
- *artificial intelligence* $\approx$ 91 900

- *статистика* $\approx$ 14 600 статей
- *машинное обучение* $\approx$ 15 200
- *компьютерное зрение* $\approx$ 12 100
- *глубокое обучение* $\approx$ 15 900
- *генеративные модели* $\approx$ 15 800
- *языковые модели* $\approx$ 15 800
- *нейронные сети* $\approx$ 15 600
- *искусственный интеллект* $\approx$ 15 900

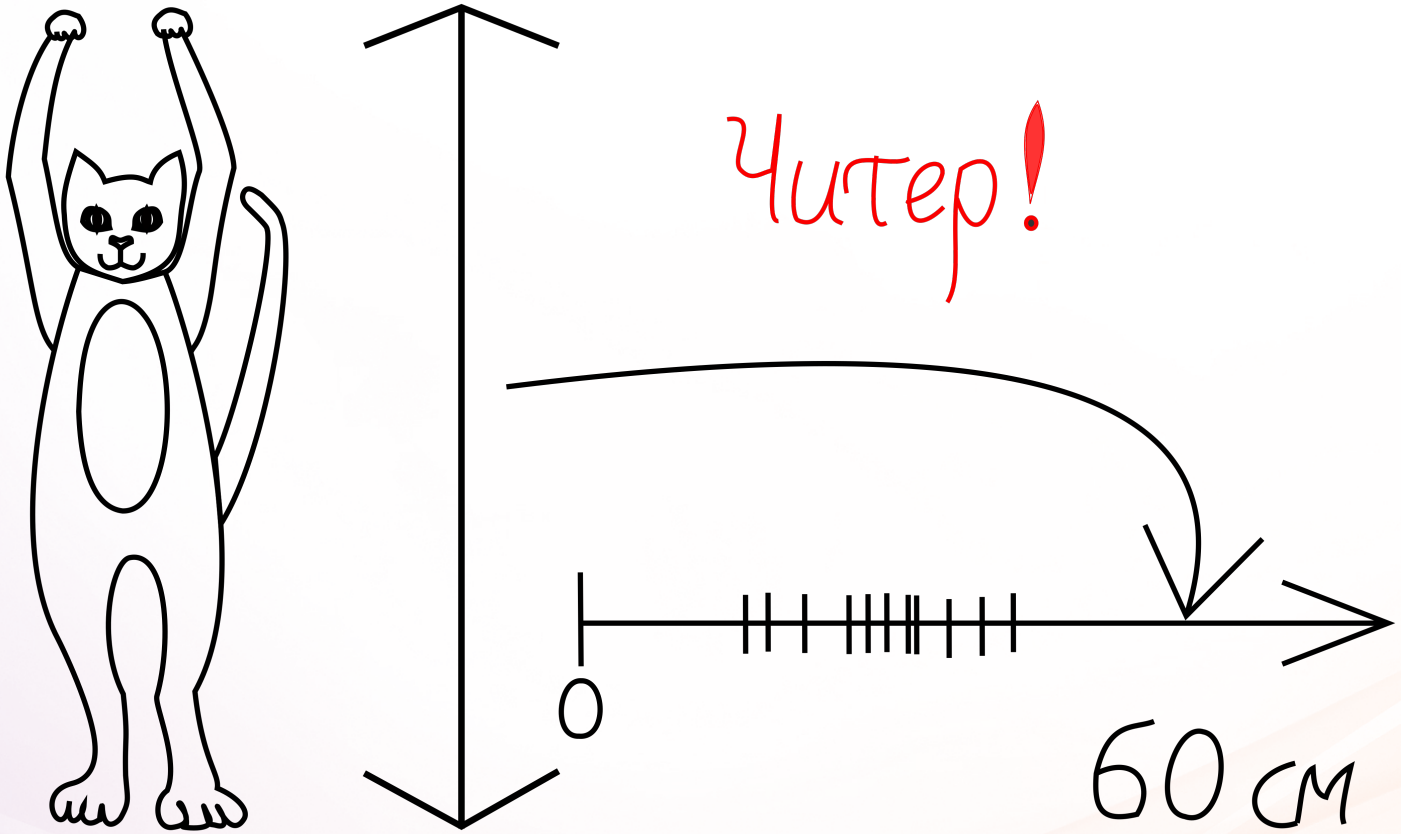
Обзор задач анализа данных

Мурмурландия

Каков средний рост котиков?

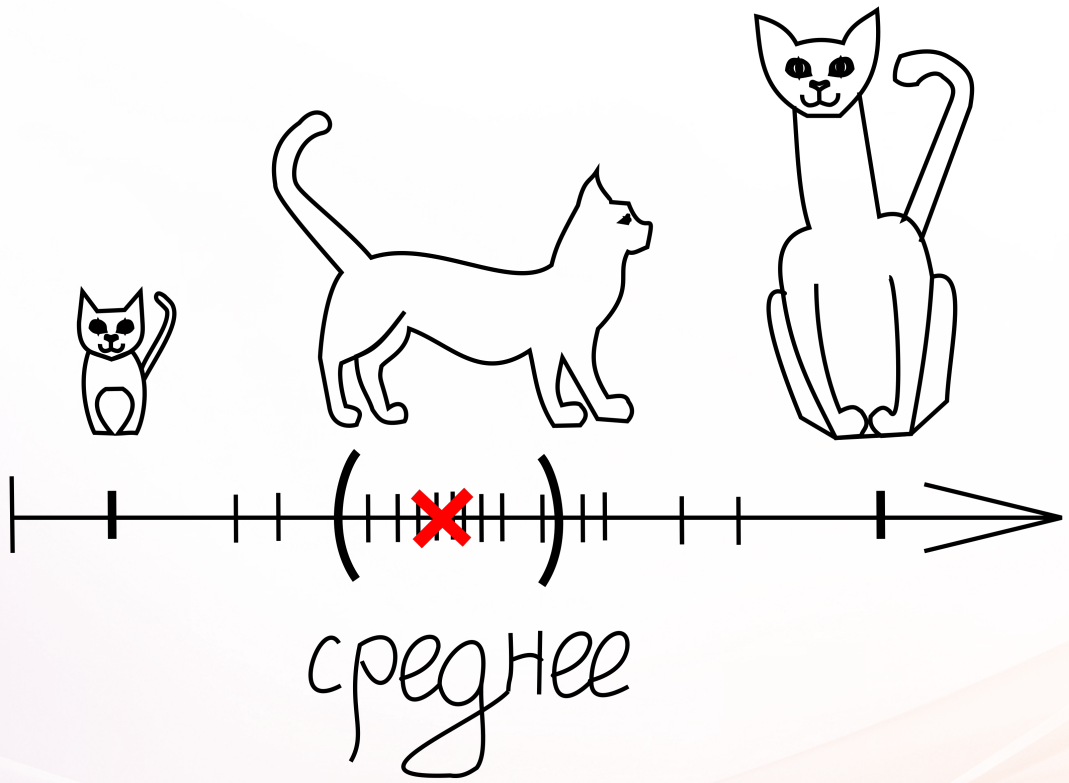


Точечное оценивание



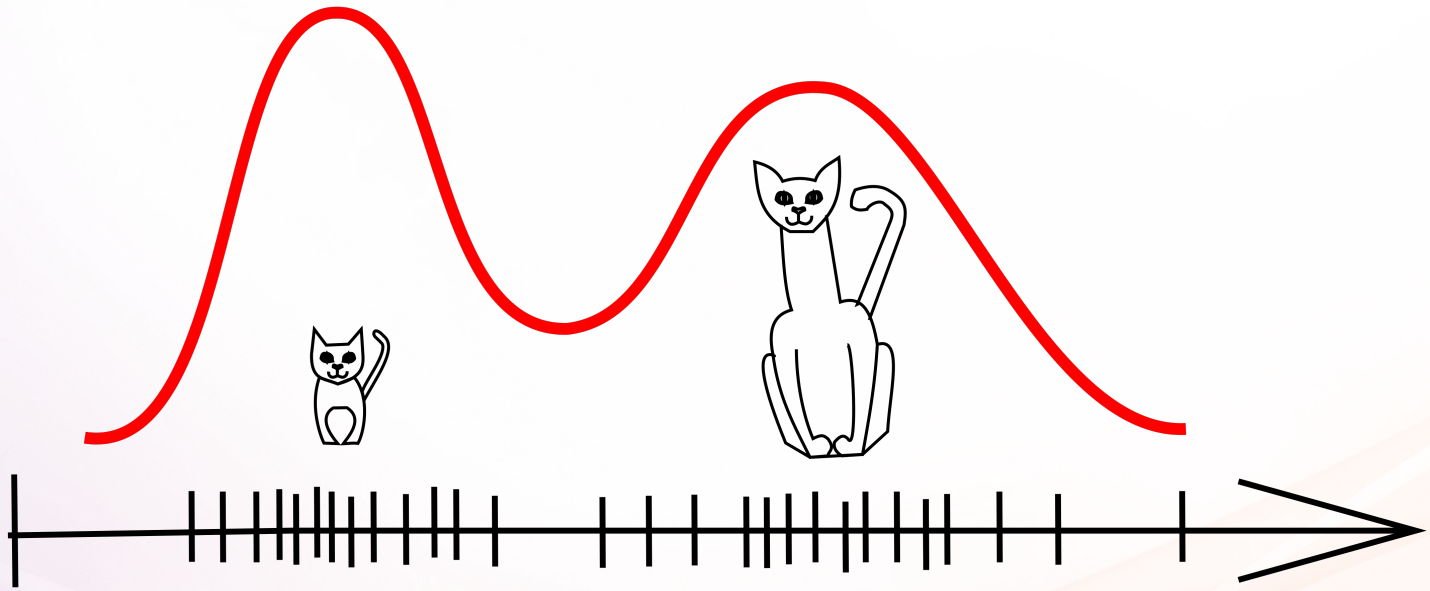
Выбросы

Среднее определяется неточно



Интервальное оценивание

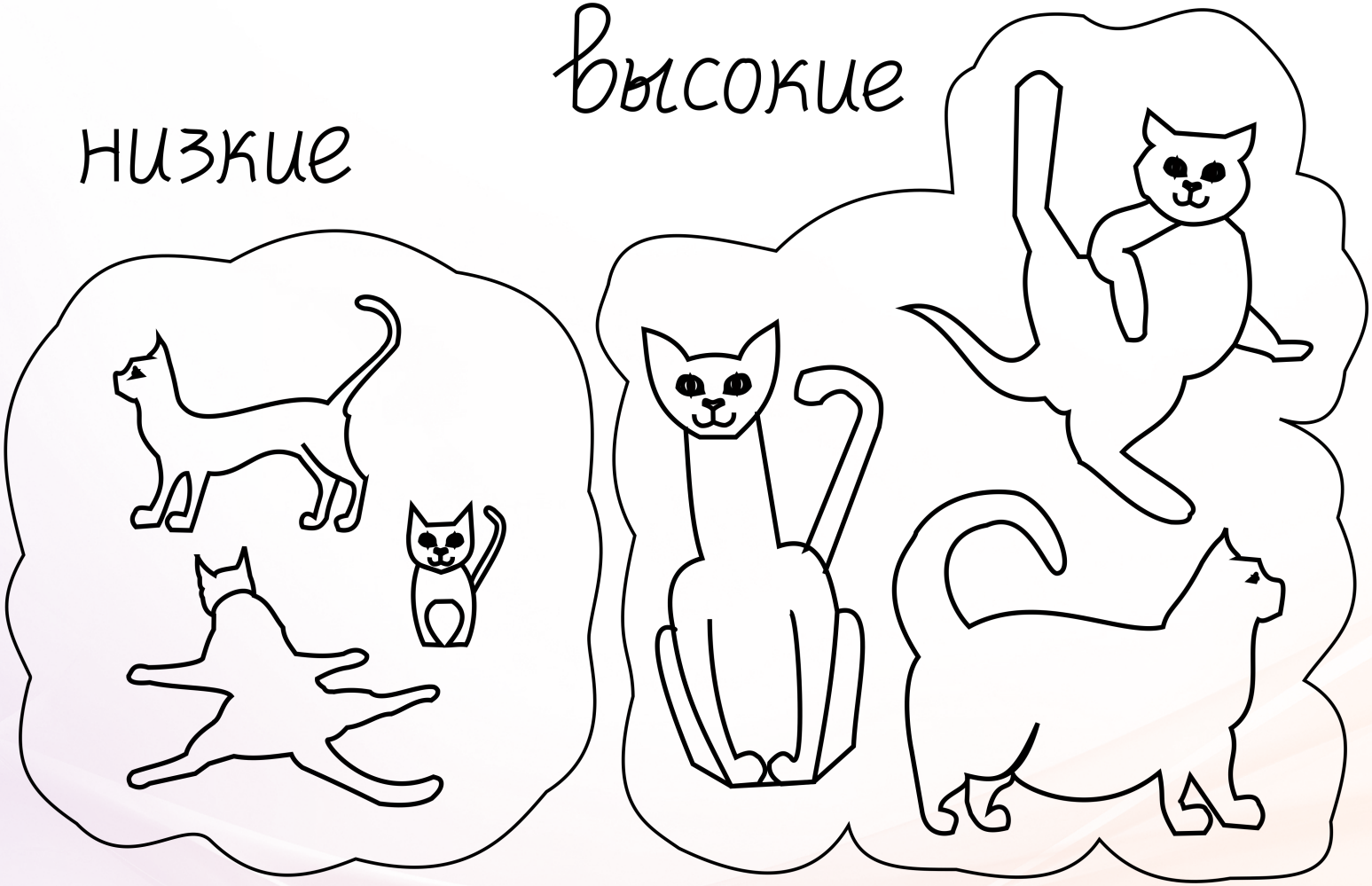
Характер распределения



Непараметрическое оценивание

низкие

высокие

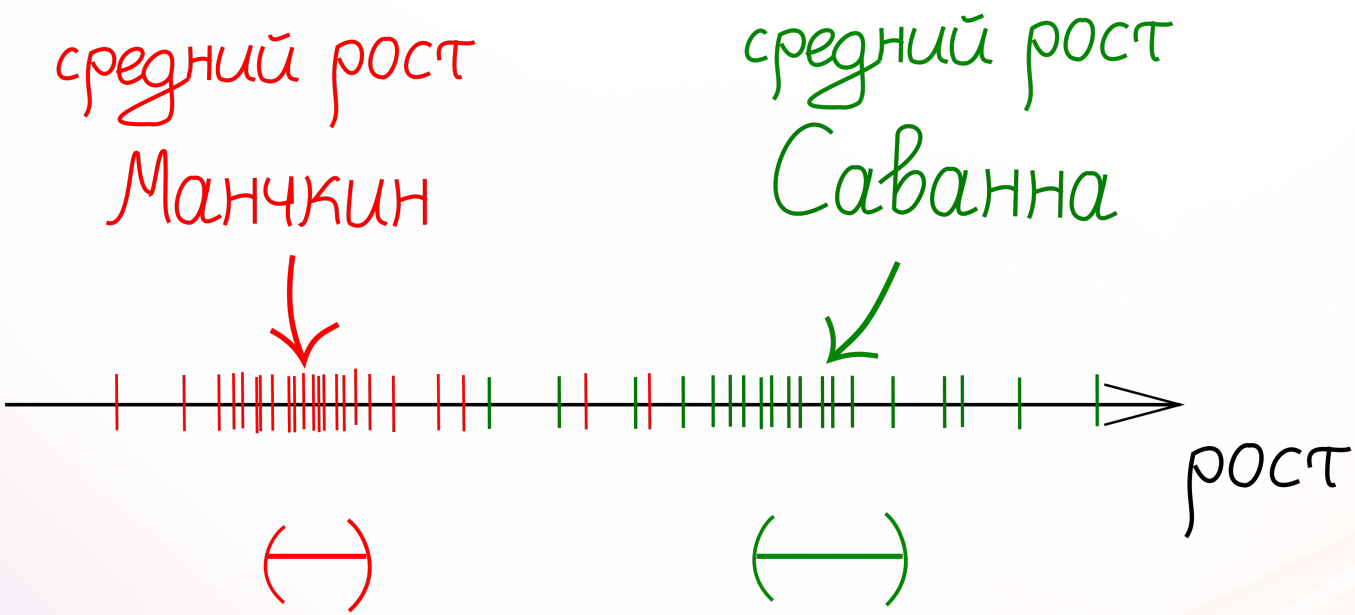


Саванна

Манчкин

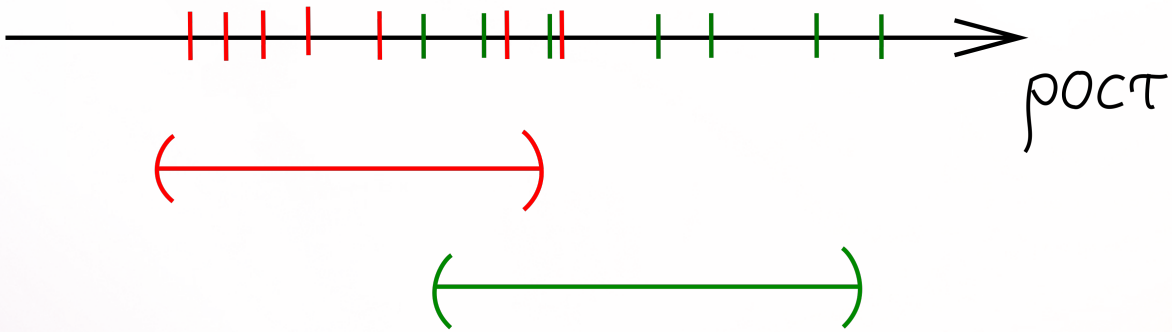
Отличается ли их средний рост?

Собираем данные

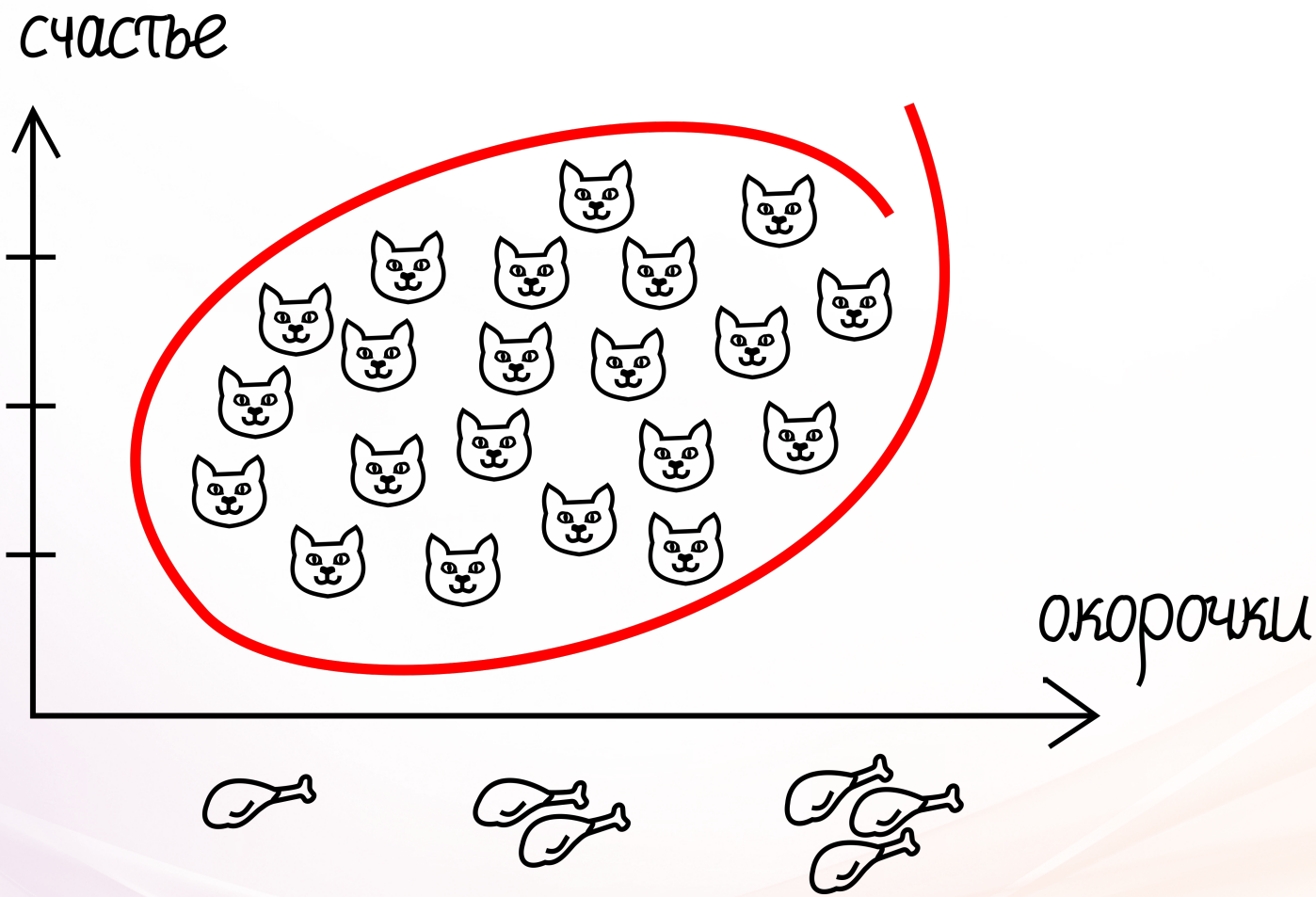


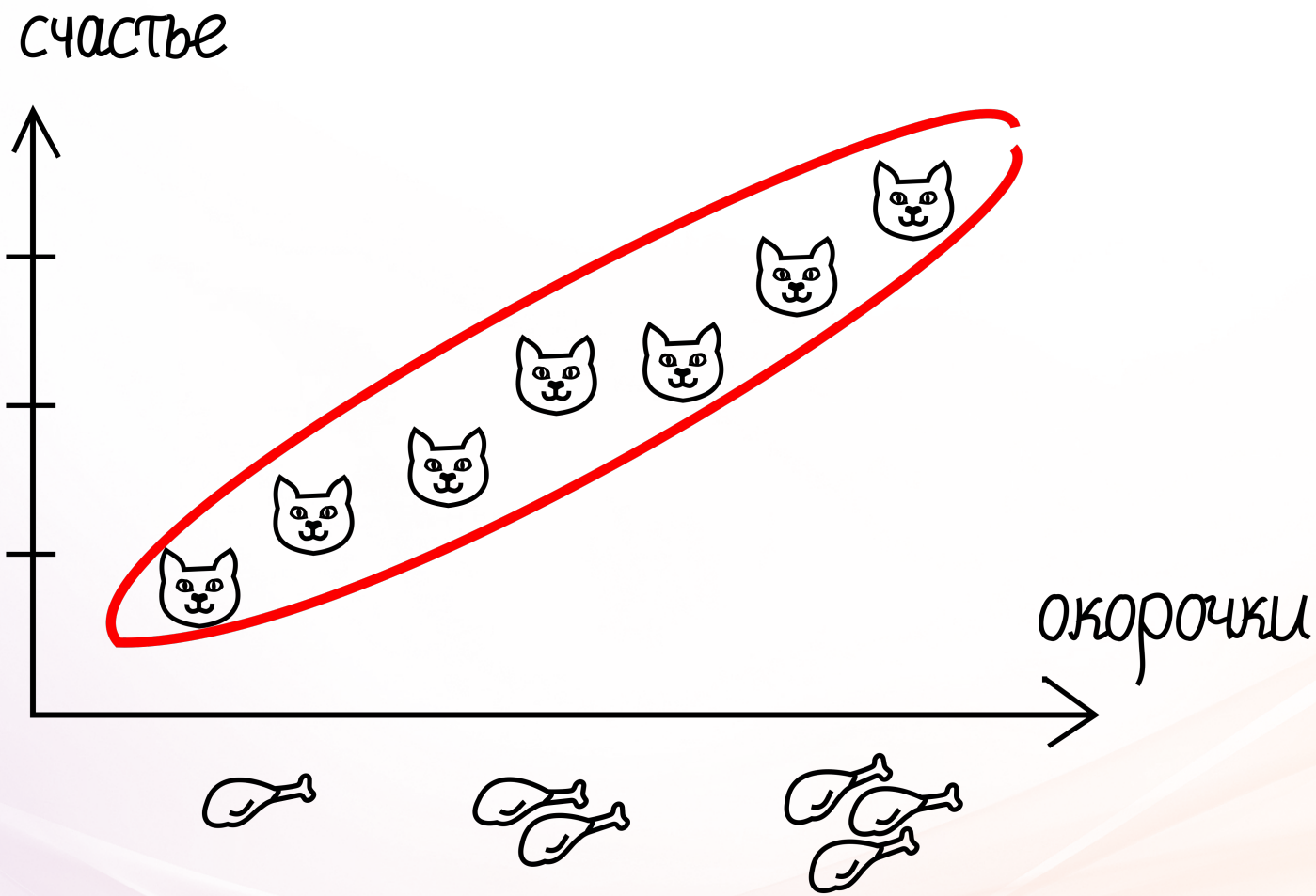
отличается

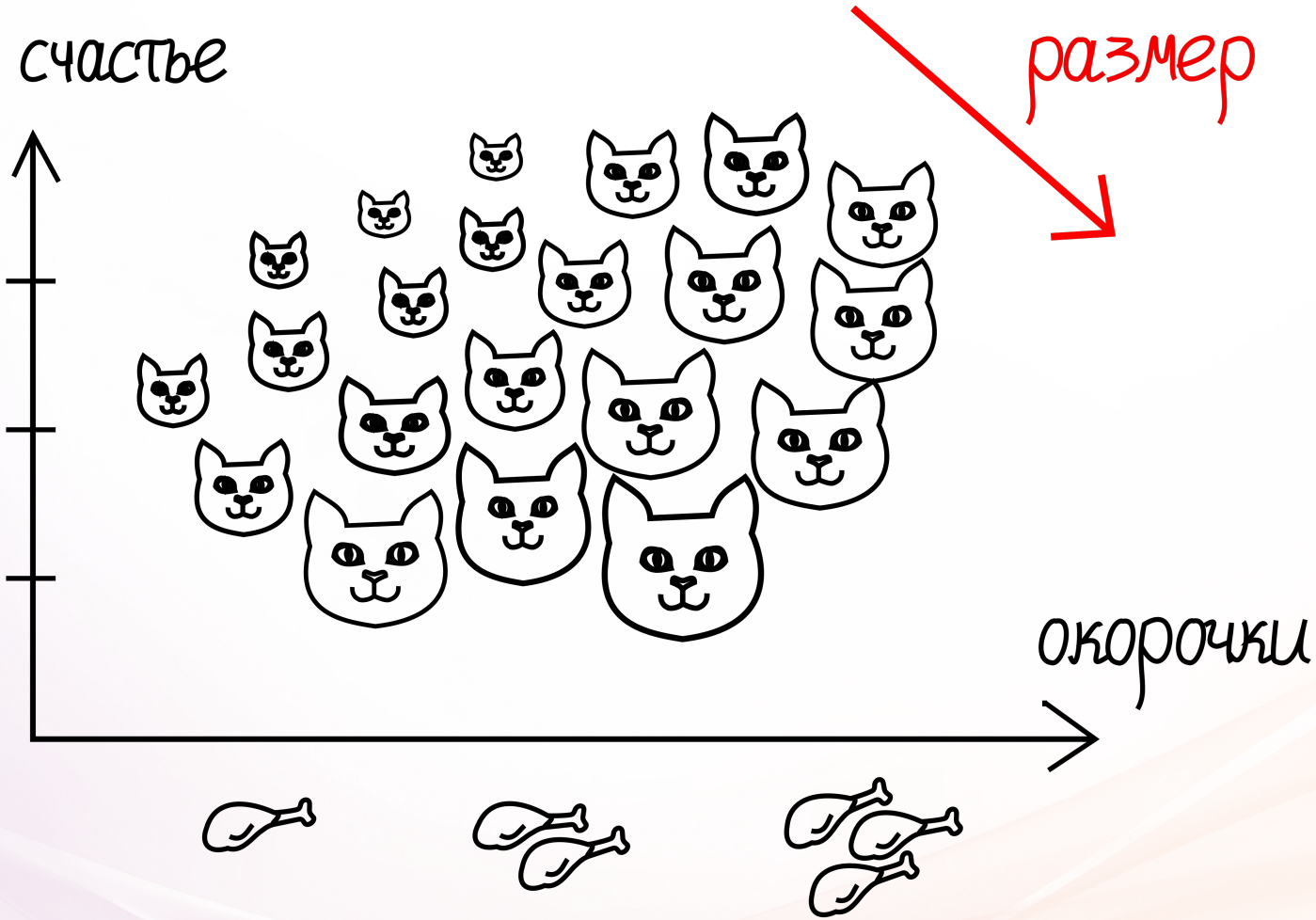
Если данных мало



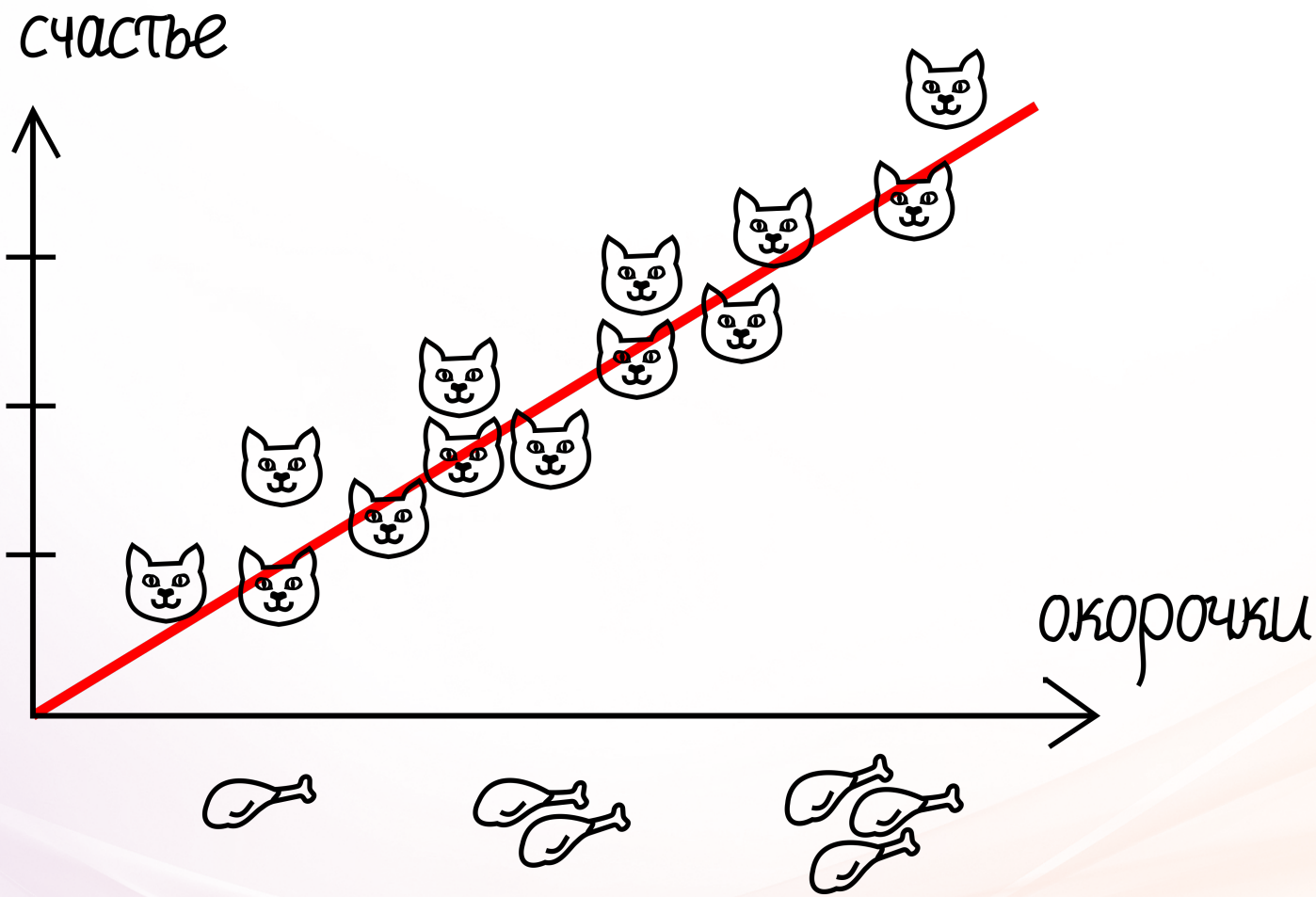
непонятно


счастье
окорочки

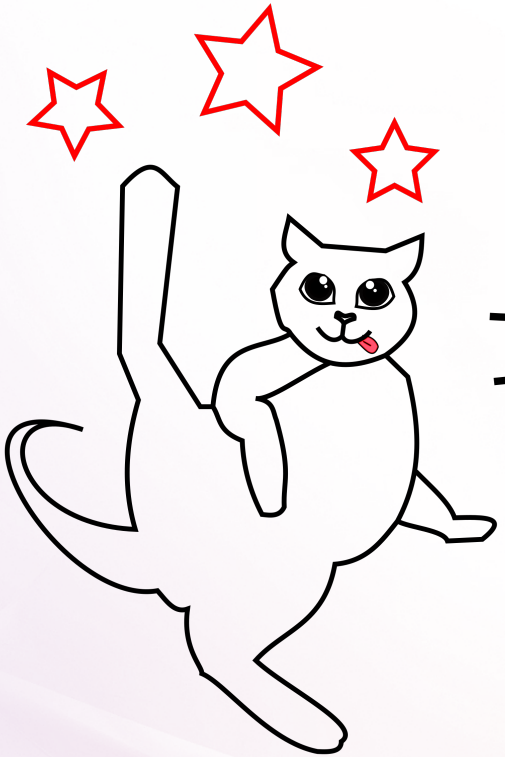

счастье
окорочки

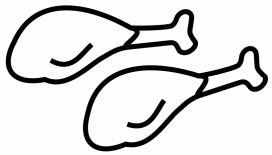


Корреляционный анализ


счастье
окорочки

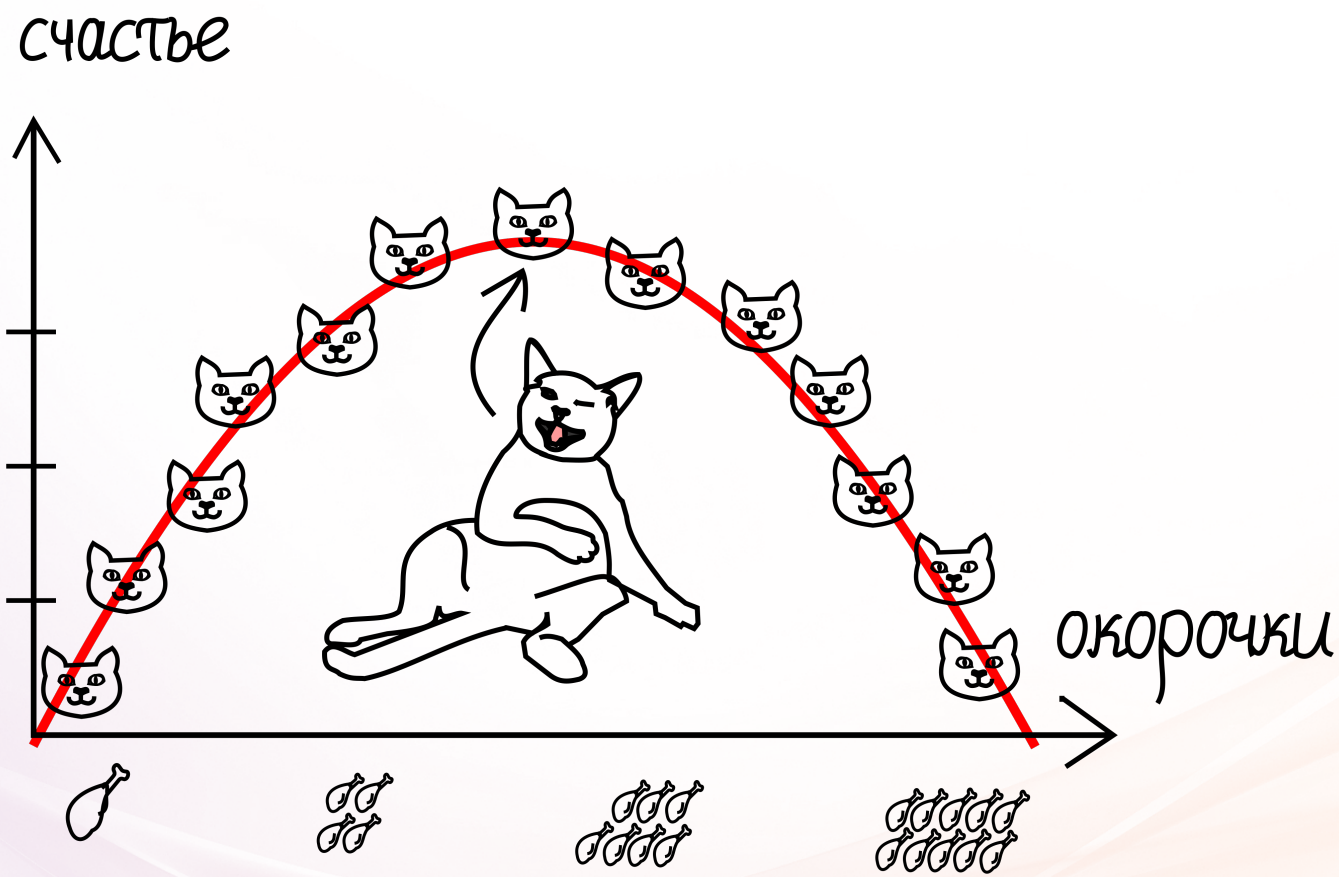
Формула счастья



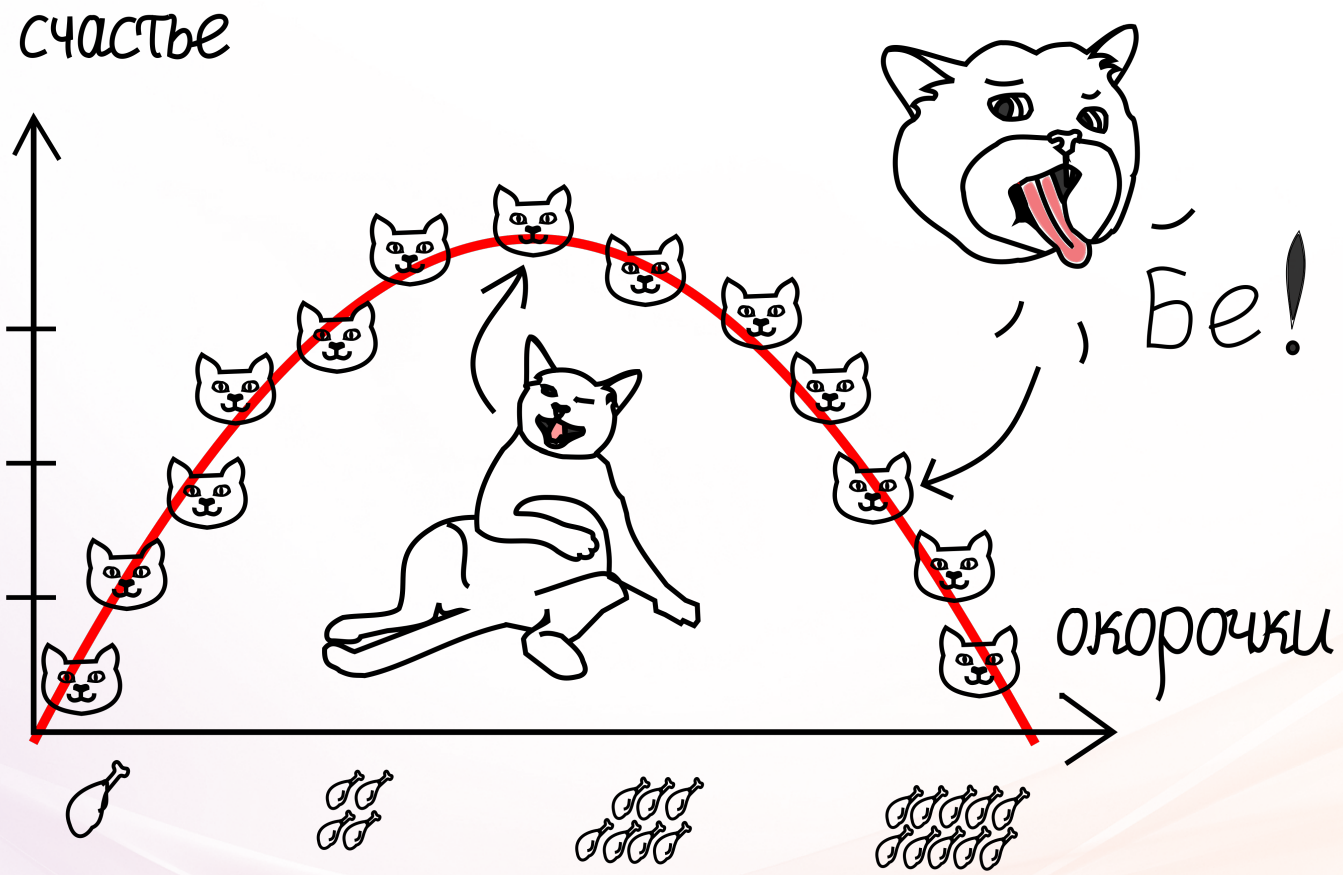
$$= \theta_0 + \theta_1 \times \text{кол-во}$$ 

$$+ \text{погрешности}$$

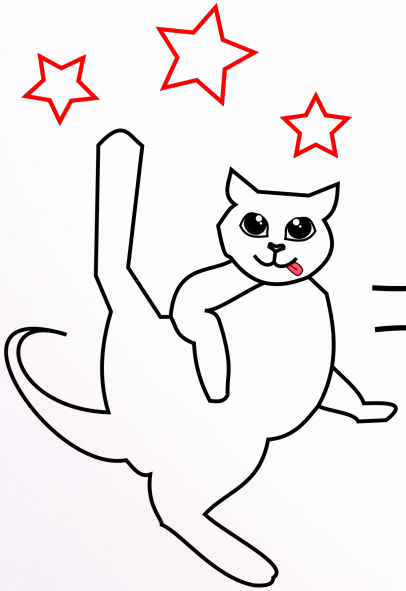
Больше окорочков



Больше окорочков



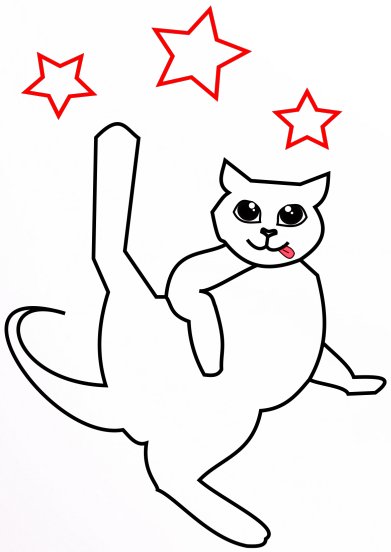
Формула счастья



$$= \theta_0 + \theta_1 \times (\text{кол-во}) - \theta_2 \times (\text{кол-во})^2$$

$$+ \text{погрешности}$$

Другие факторы



$$= \theta_0 + \theta_1 \times \text{🍗} - \theta_2 \times (\text{🍗})^2$$

$$+ \theta_3 \times \text{🧶}$$

$$+ \theta_4 \times \text{🛋}$$

$$+ \text{погрешности}$$

Регрессионный анализ

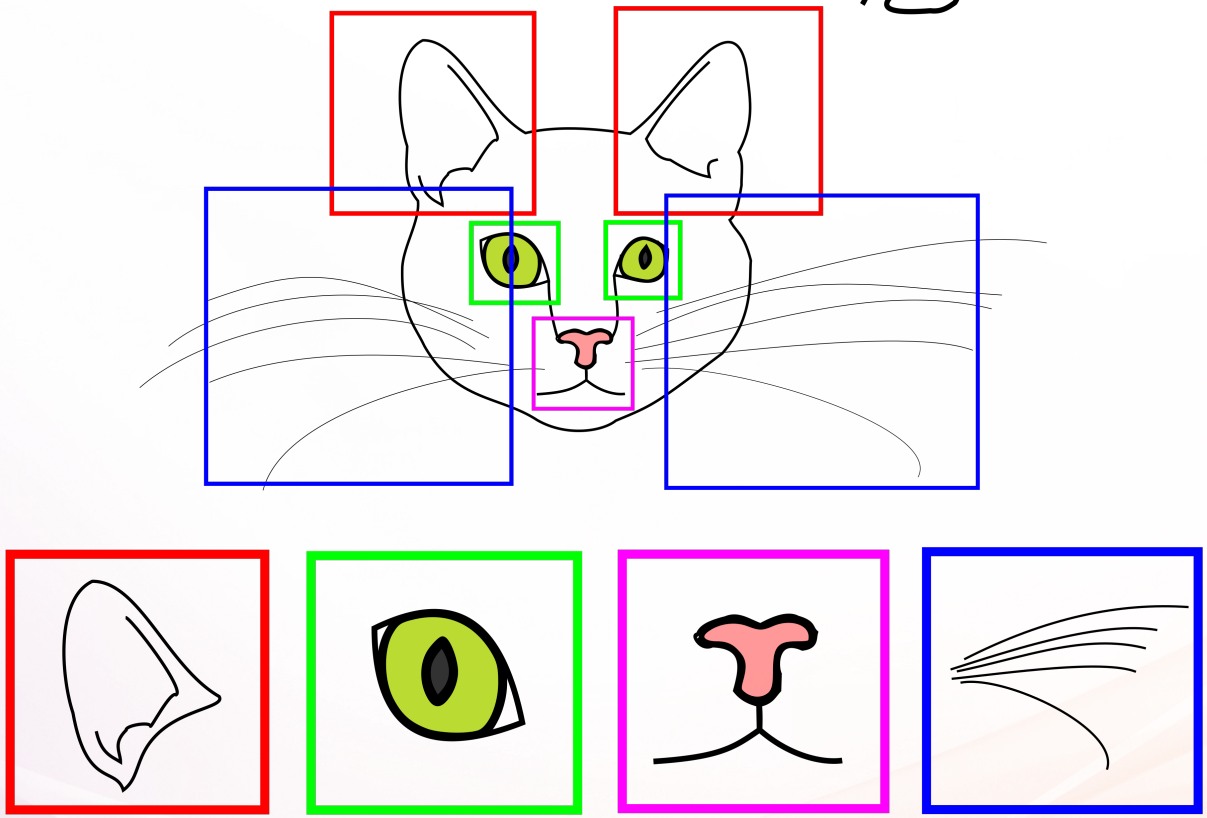
Классификация котиков

Классификация

Собираем данные

котик	порода	рост	шерсть
	Саванна	50 см	да
	Сфинкс	30 см	нет
	Манчкин	15 см	да
	Саванна	40 см	да

Распознавание мордочек



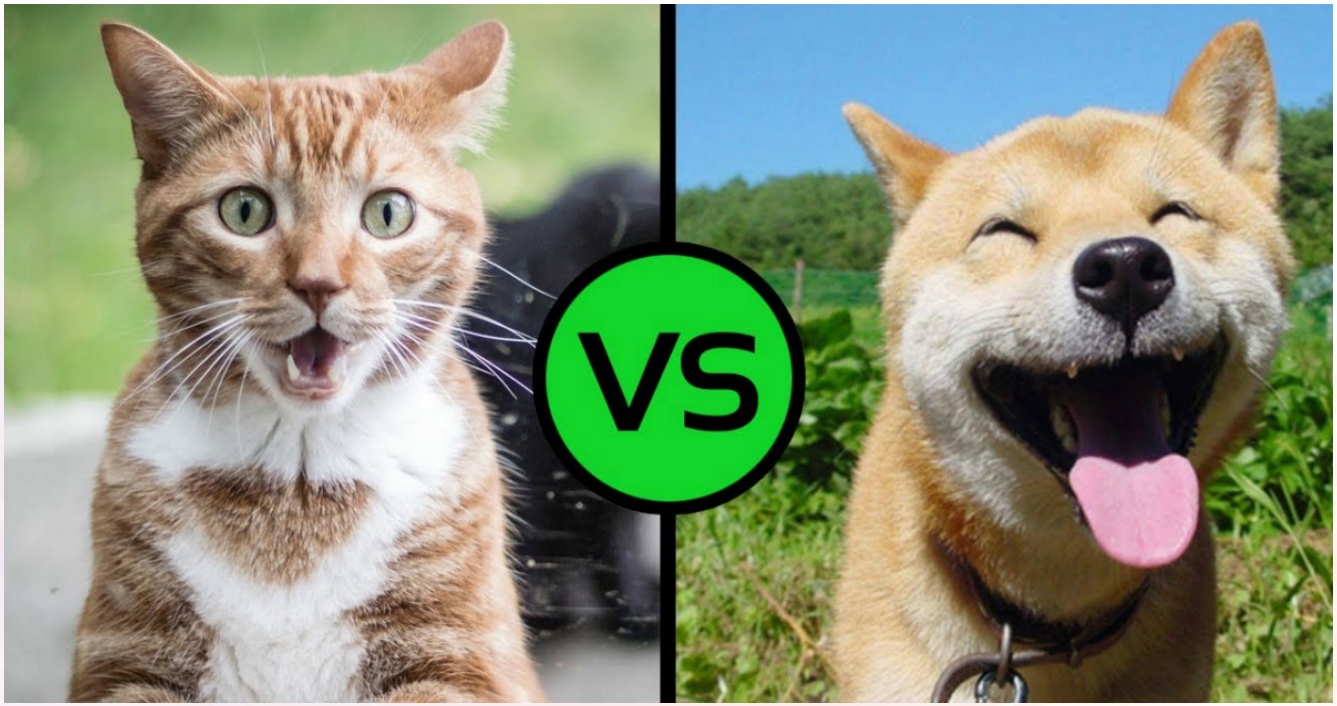
Нейронные сети

Книга с похожим содержанием

Попробуем решить задачу

А ты кто?

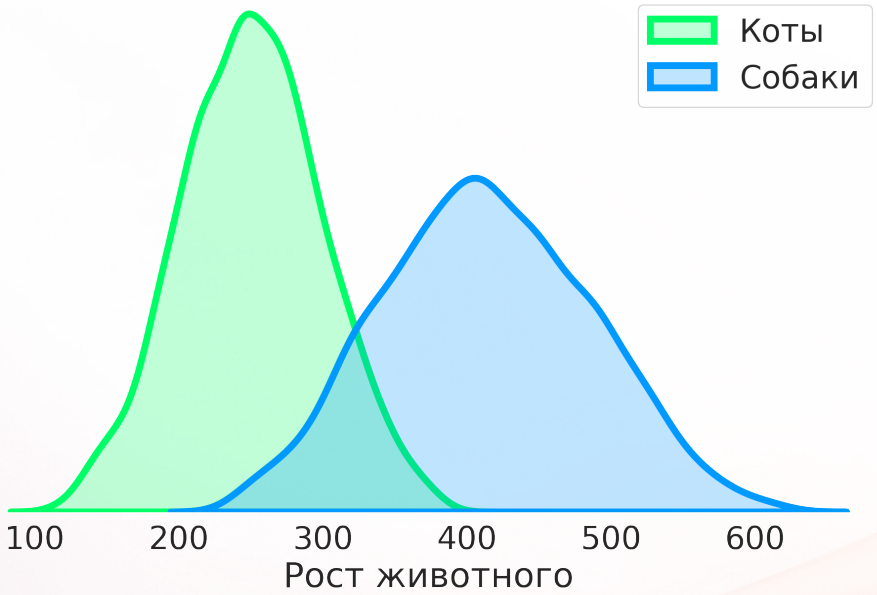
Перед нами домашнее животное. Кто это — собака или кот?



Классификация: собака vs кот

Попробуем сначала извлечь какой-то *признак*.

Построим вероятностные плотности для каждого класса.



При каких-то значениях роста мы уже можем

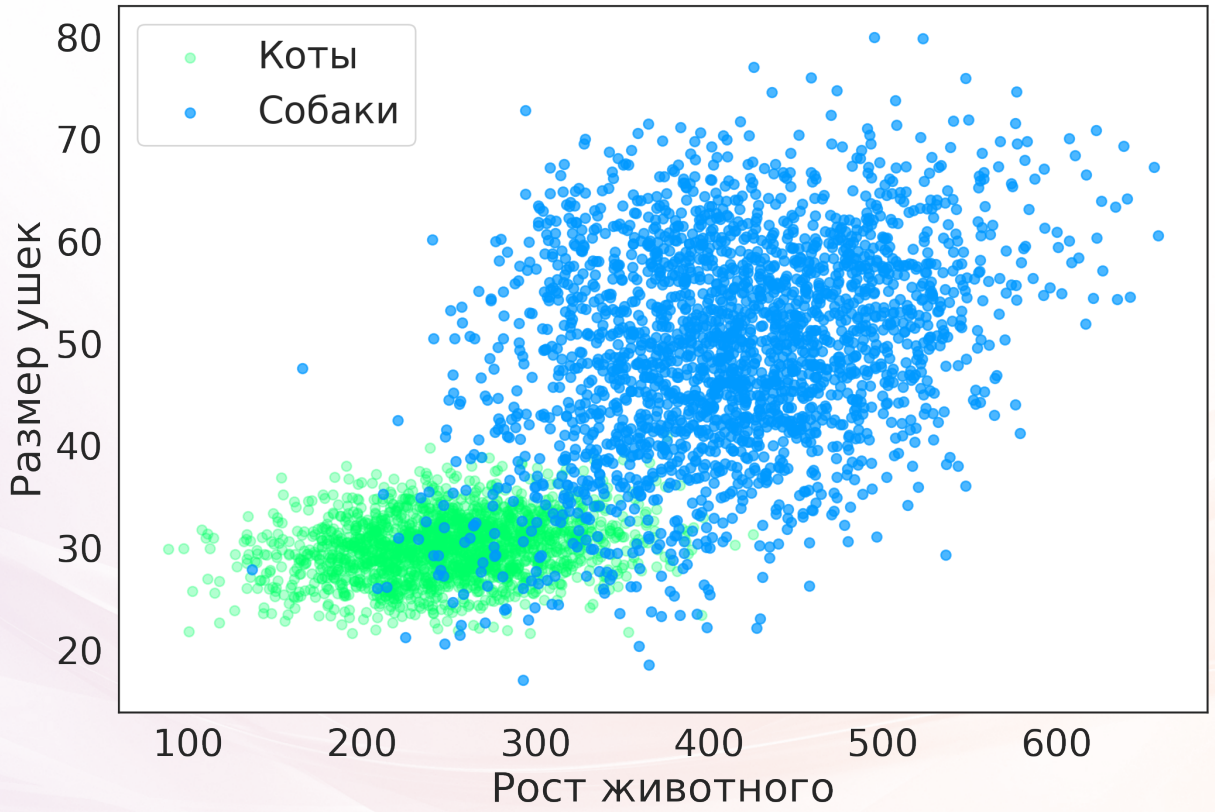
с большой уверенностью сказать ответ.

Но есть большое пересечение, это не очень здорово.

Классификация: собака vs кот

Извлечем еще один признак — размер ушек.

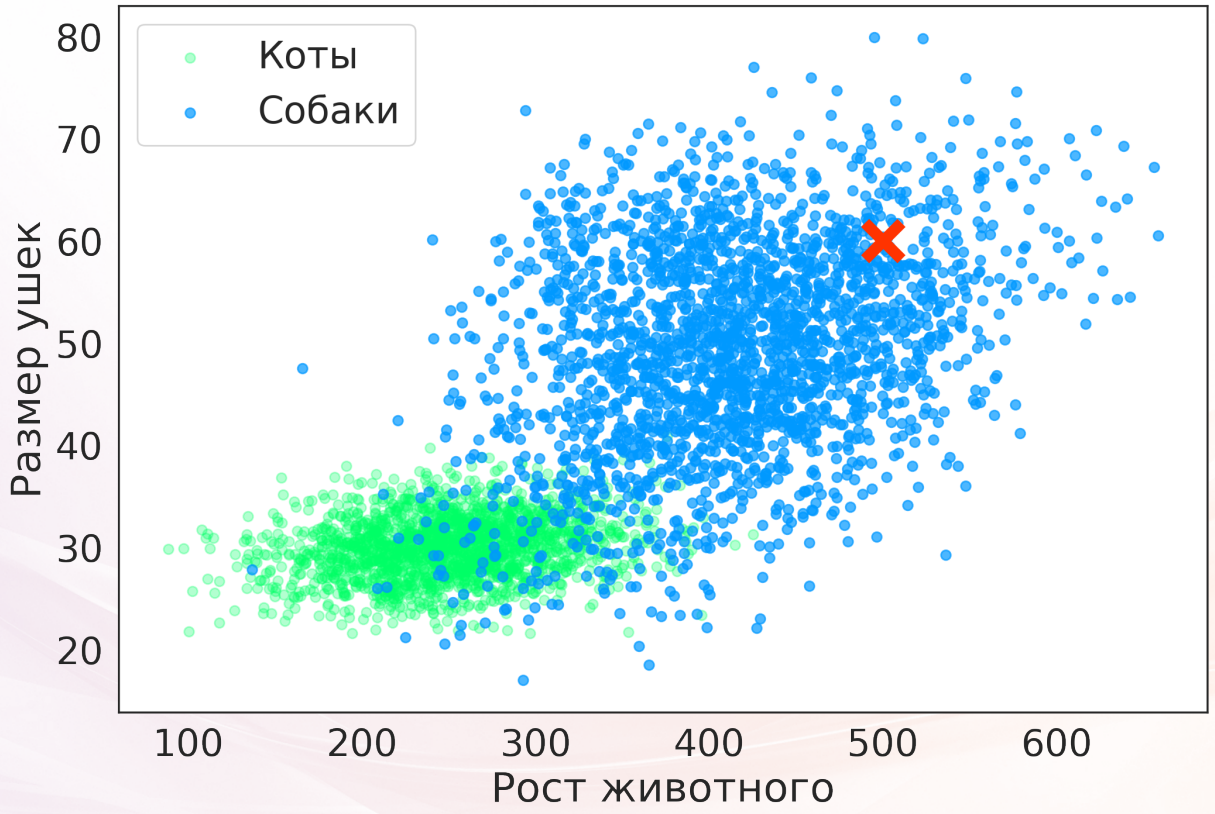
Теперь классы лучше разделяются.



Классификация: собака vs кот

Попробуем классифицировать новое животное.

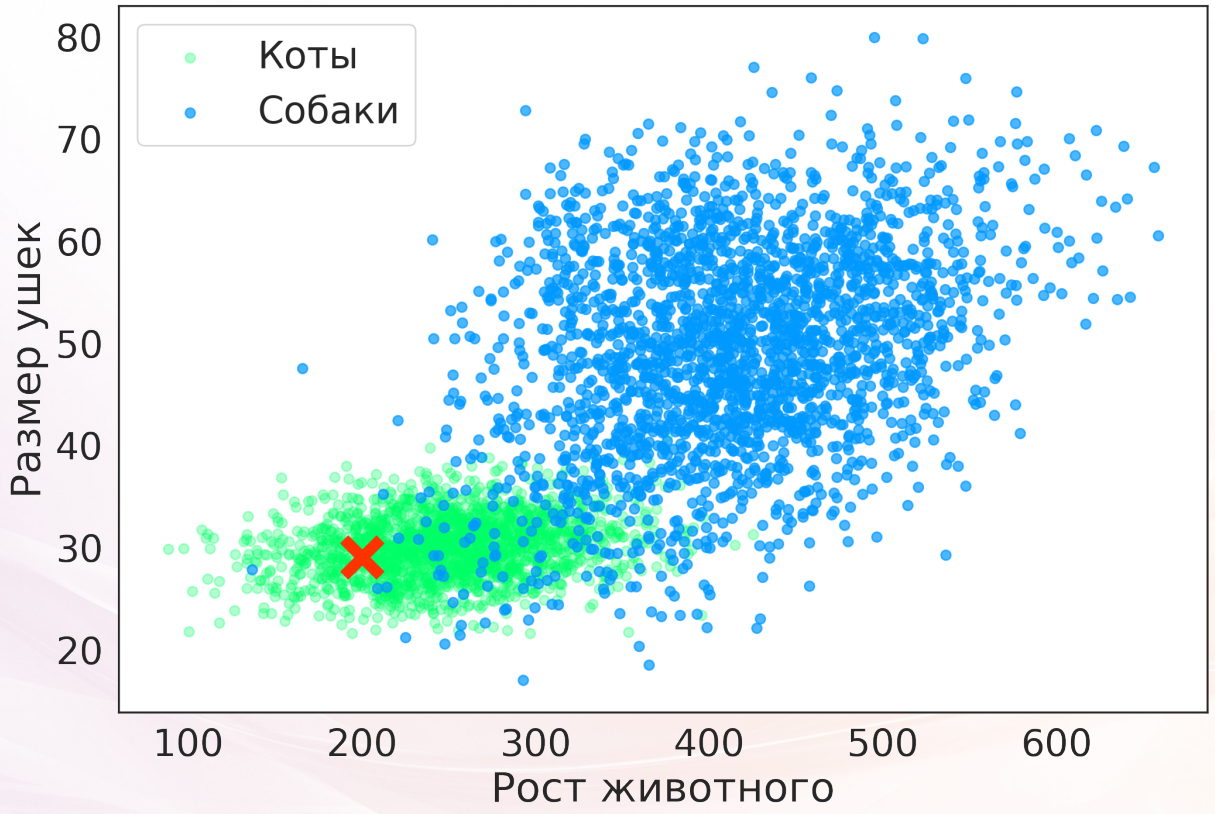
Кто отмечен красным?



Классификация: собака vs кот

Попробуем классифицировать новое животное.

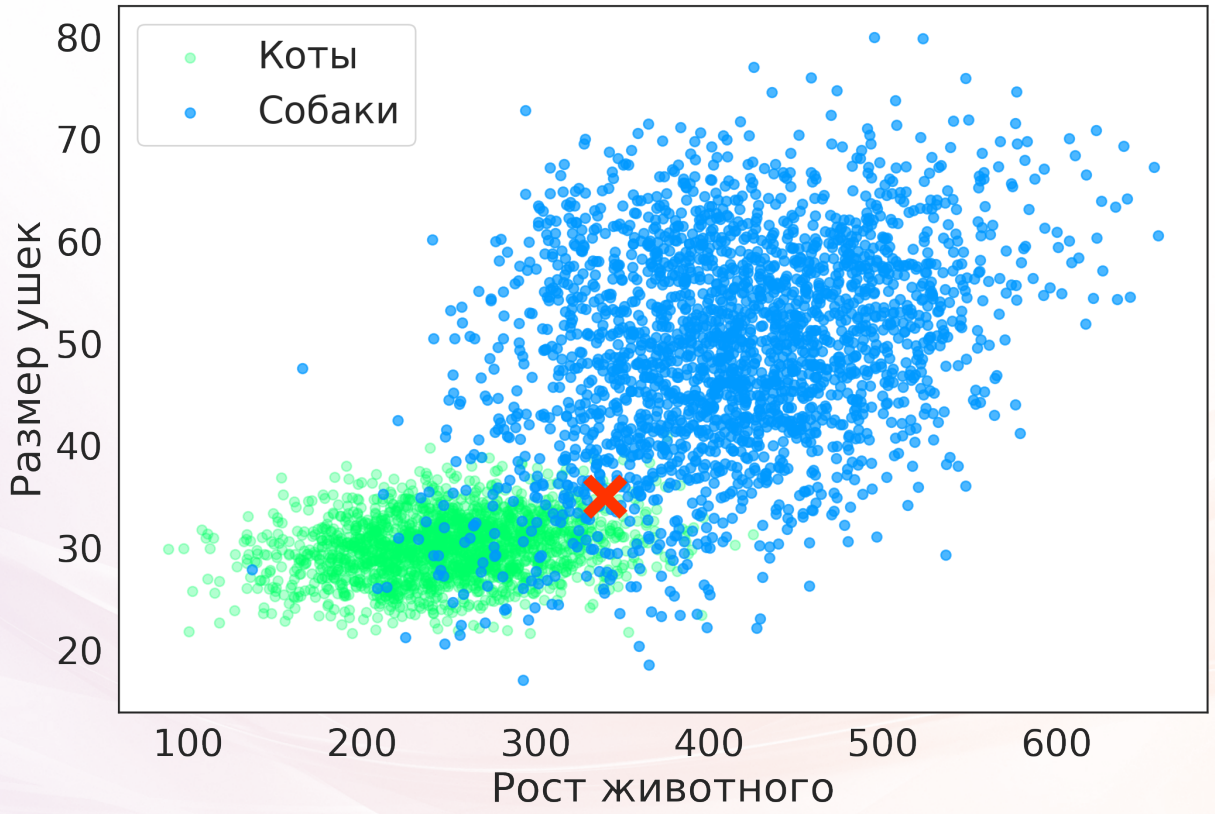
Кто отмечен красным?



Классификация: собака vs кот

Попробуем классифицировать новое животное.

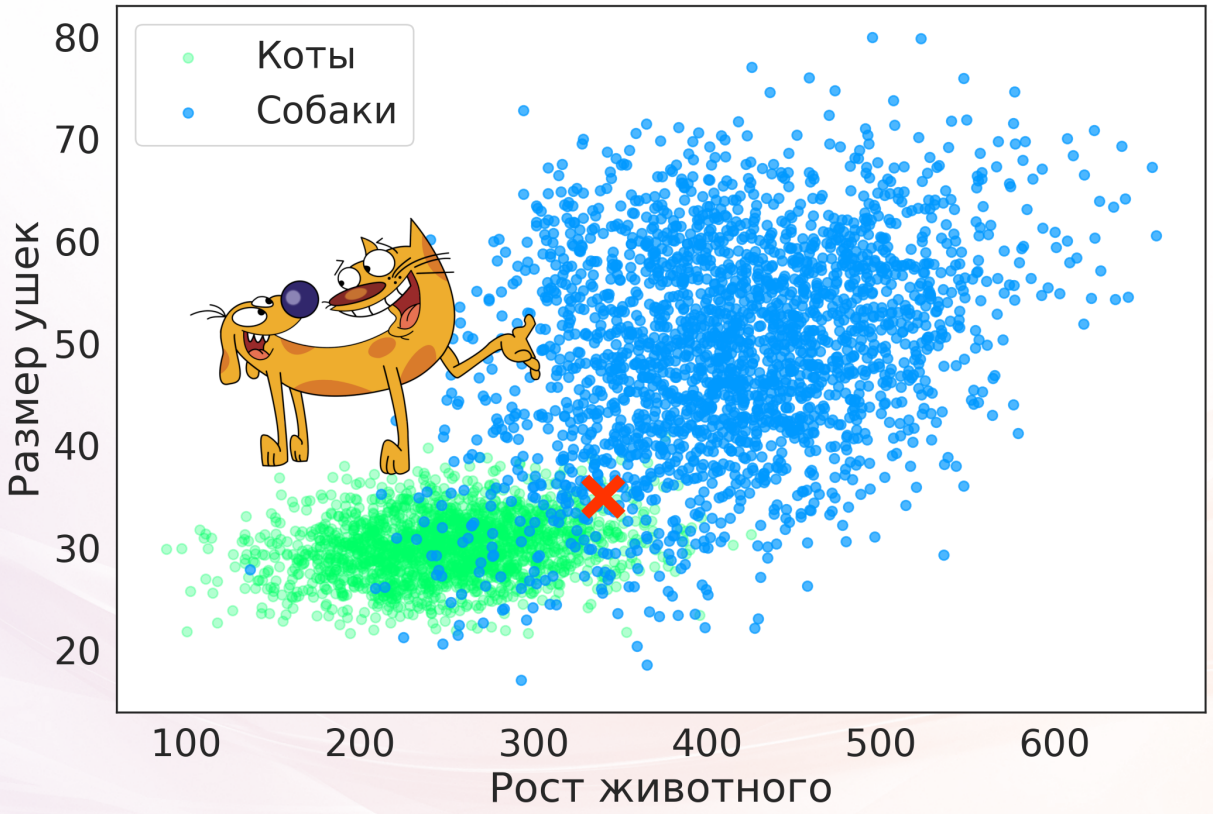
Кто отмечен красным?



Классификация: собака vs кот

Попробуем классифицировать новое животное.

Кто отмечен красным?



На основе чего вы сделали все выводы?

Метод ближайших соседей (kNN)

Дано:

$X_1, ..., X_n$ — набор размеченных объектов.

$Y_1, ..., Y_n$ — соответствующие метки класса.

Задача:

Пусть x — исследуемый объект. Какого он класса?

Решение:

Будем смотреть на свойства k ближайших соседей.

$x_{(1)}, ..., x_{(k)}$ — k его соседей в порядке удаления от x.

$Y_1, ..., Y_k$ — соответствующие им классы.

Ответ — наиболее часто встречаемый класс среди $x_{(1)}, ..., x_{(k)}$.

Свойства:

1. k — гиперпараметр модели;
2. Не редко на практике показывает хорошие результаты.
3. **Дорогое применение:**
   для каждого x результат вычисляется за $O(n \ln n)$.

Взвешенный метод ближайших соседей

Пусть x — исследуемый объект.

$x_{(1)}, ..., x_{(k)}$ — k его соседей в порядке удаления от x.

$Y_1, ..., Y_k$ — соответствующий класс.

$w_1, ..., w_k$ — вклад k-го соседа, определяемый пользователем.

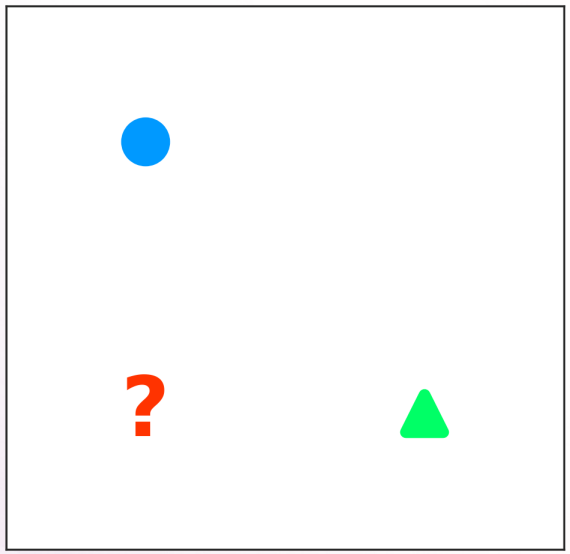
Способы определения веса:

- $w_j = 1 - j/k$ — зависящий от номера соседа;
- $w_j = \|x - x_{(j)}\|^{-1}$ — зависящий от расстояния до соседа.

$$\widehat{y}(x) = \arg\max_y \sum_{j=1}^{k} w_j I\{Y_j = y\} \text{ — классификация}$$

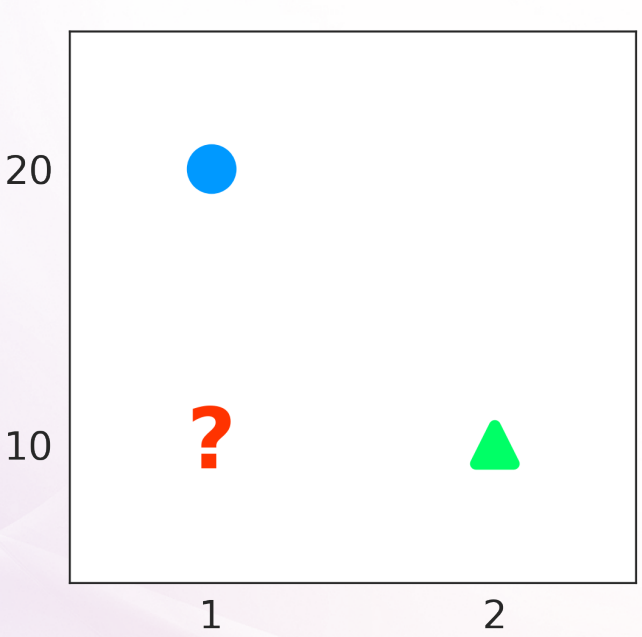
Особенности

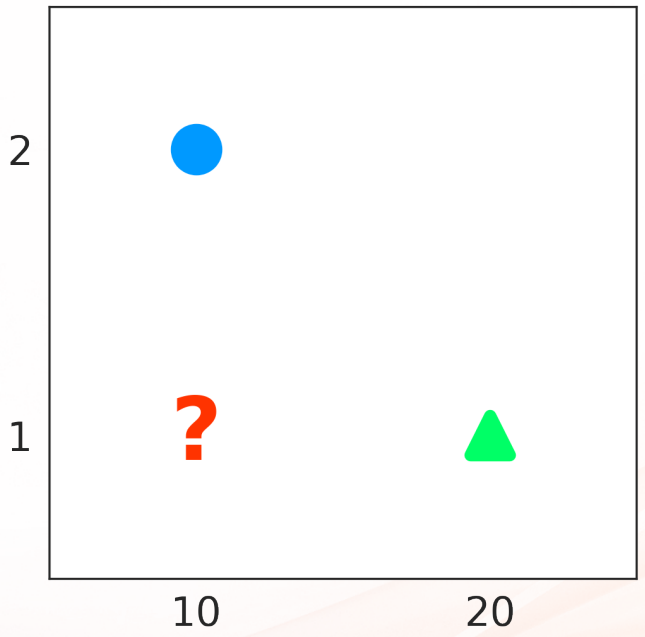
Классифицируйте объект "?".



Особенности

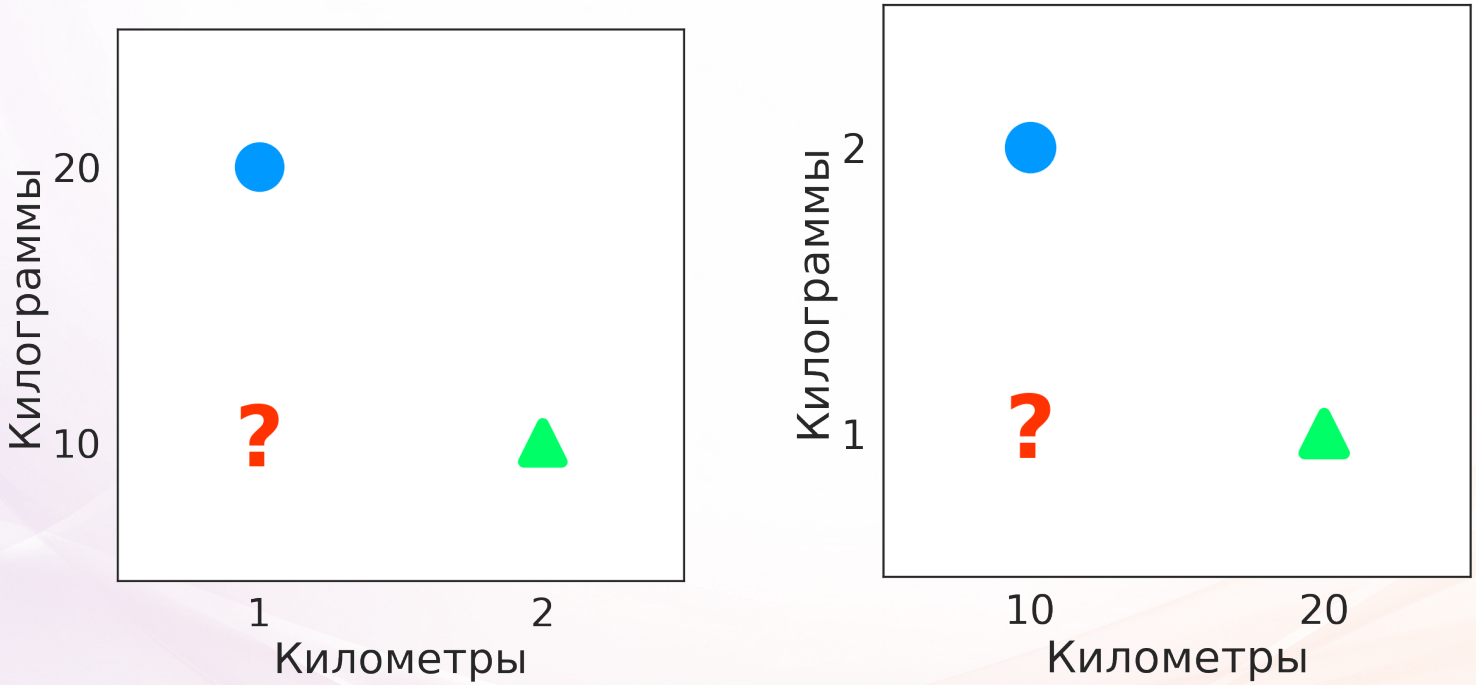
Классифицируйте объект "?".





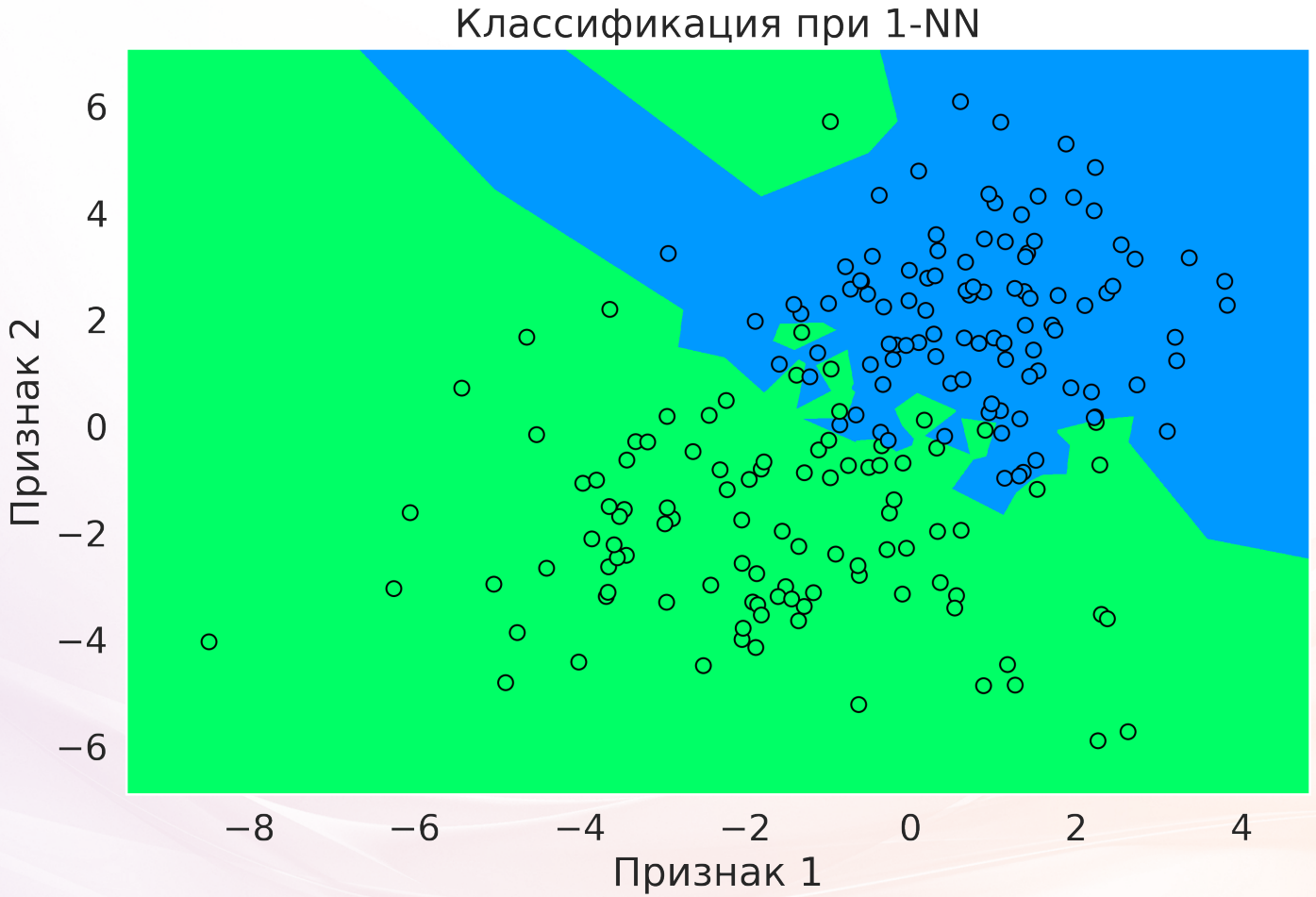
Особенности

Классифицируйте объект "?".

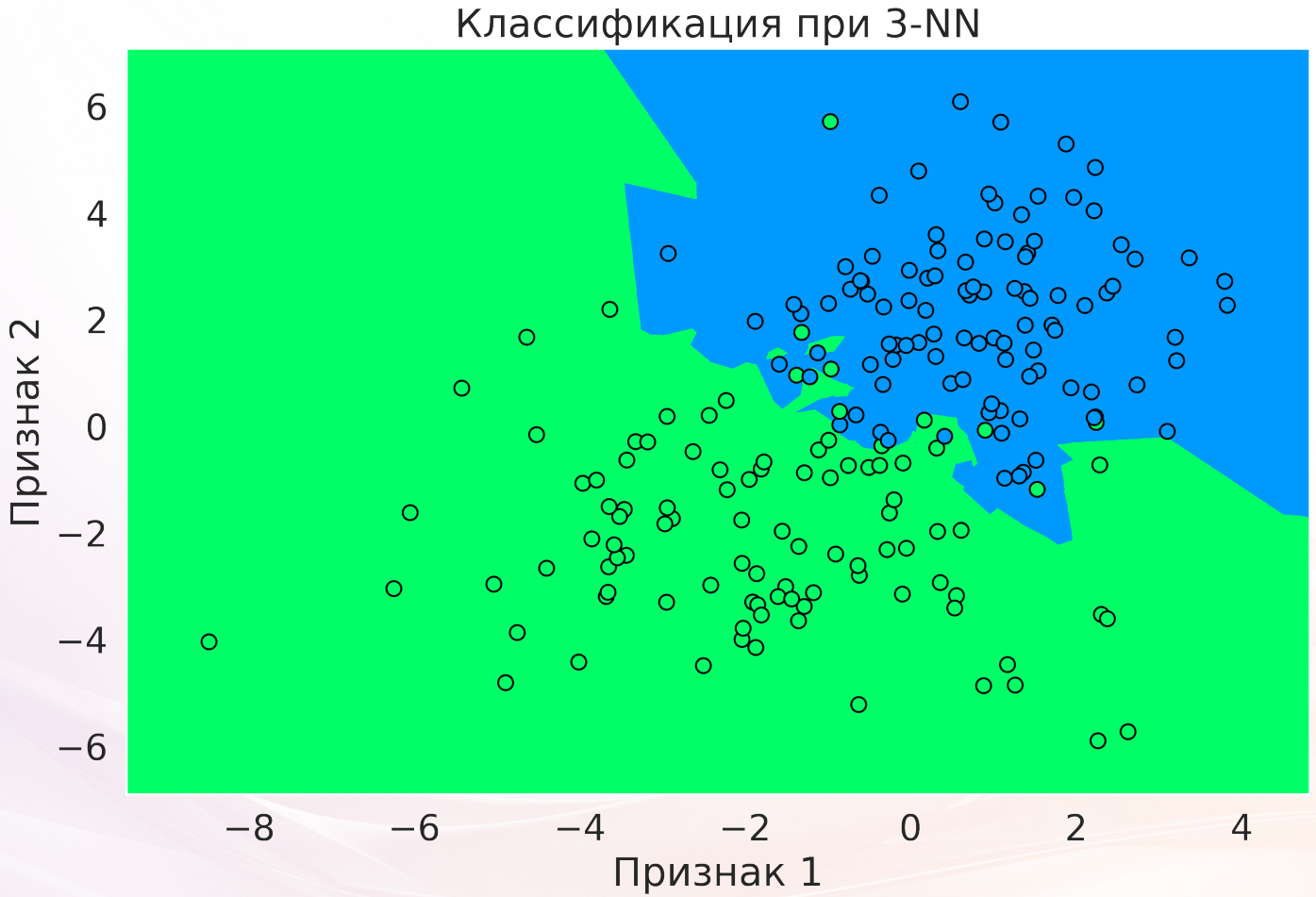


Вывод: результат сильно зависит от используемой метрики между точками в пространстве. Не складывайте *кг* с *км*!

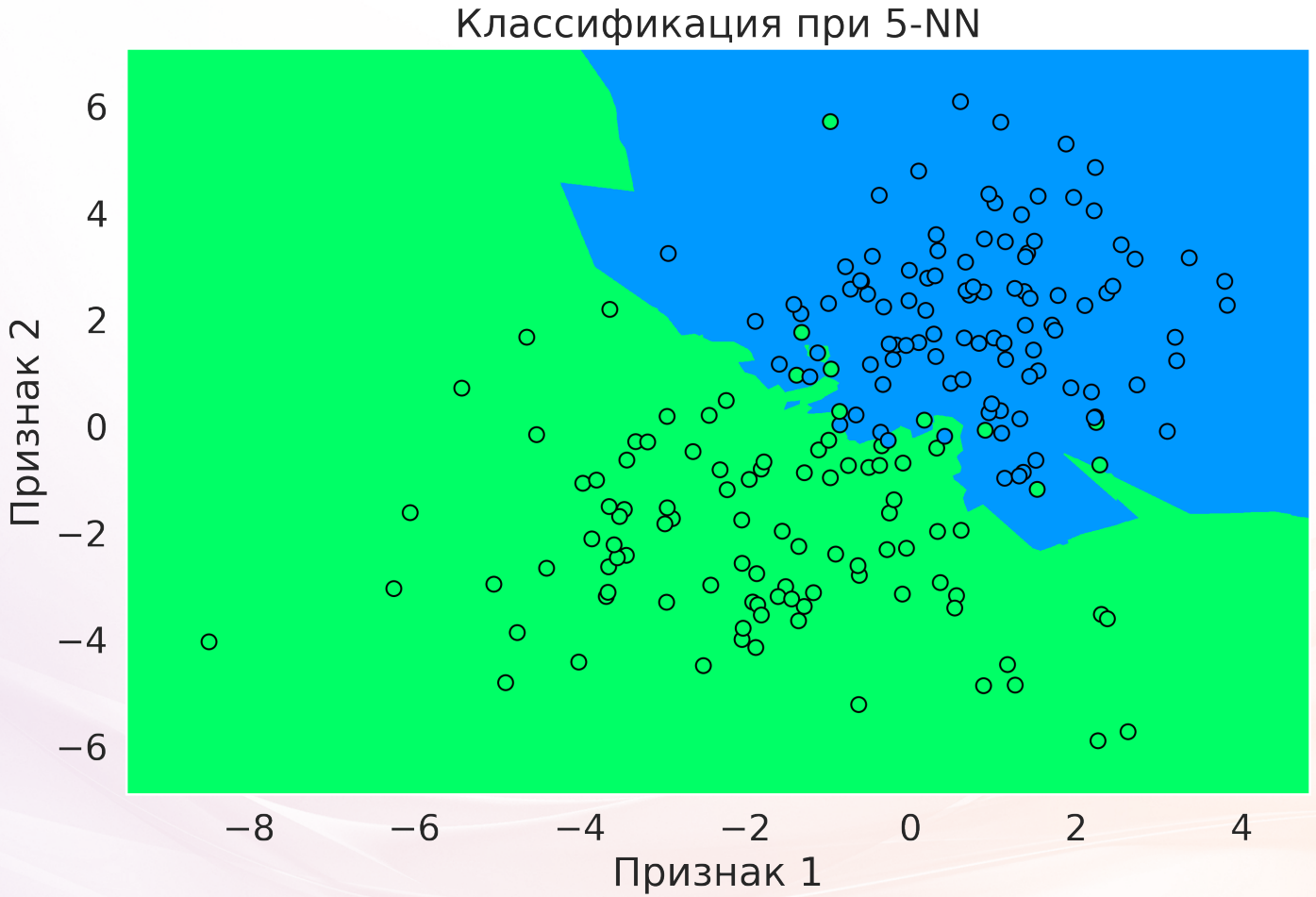
Что происходит при разных k?



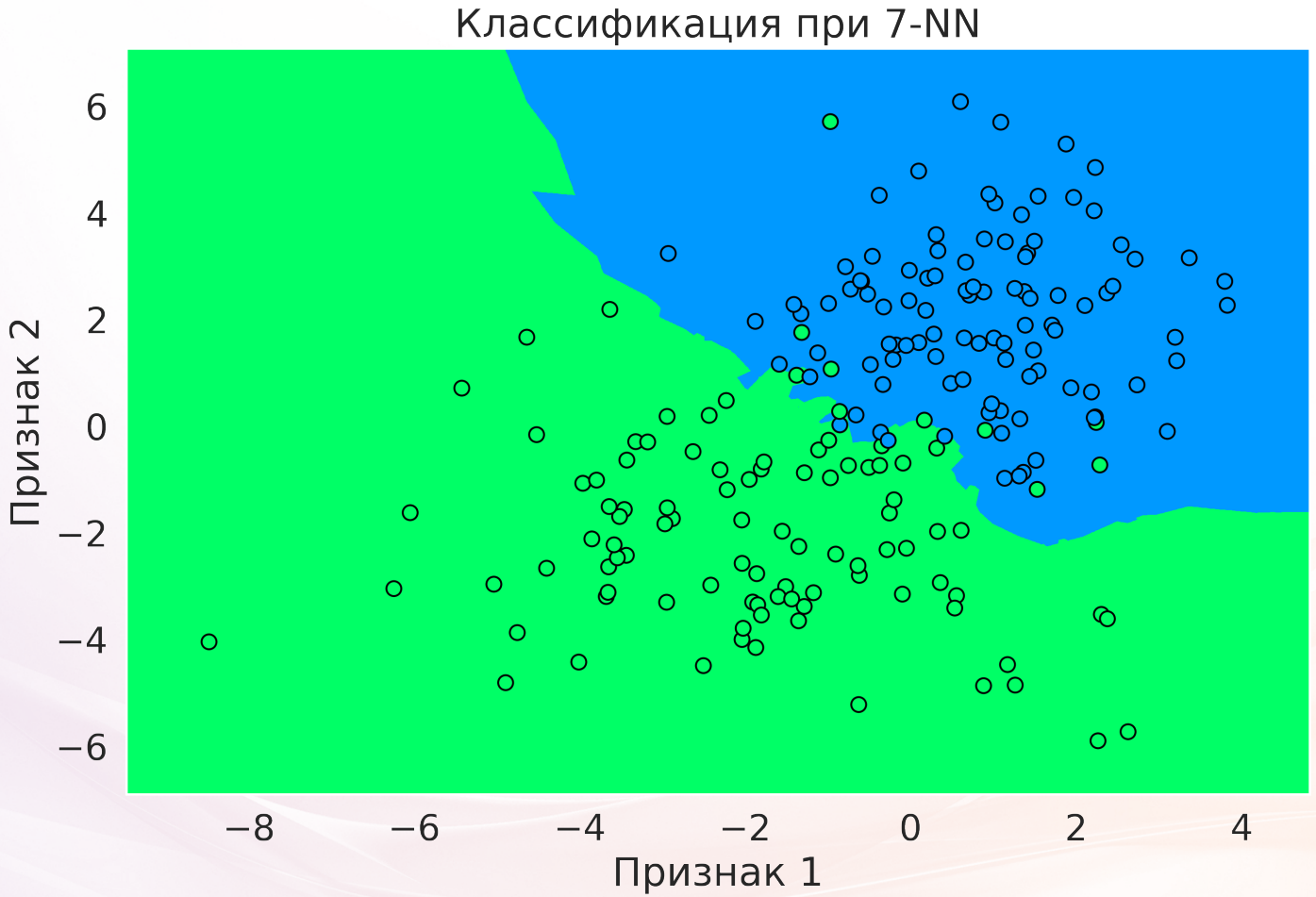
Что происходит при разных k?



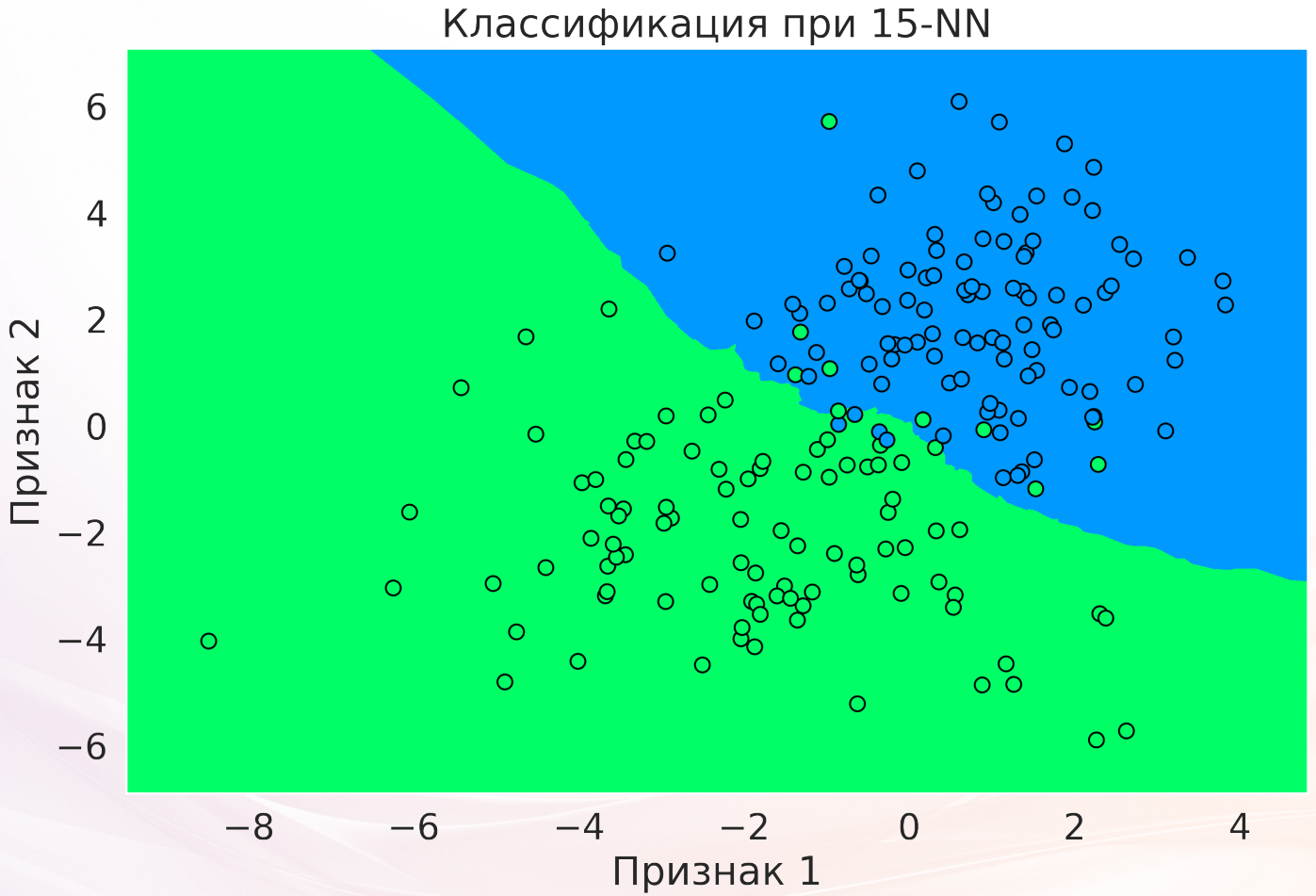
Что происходит при разных k?



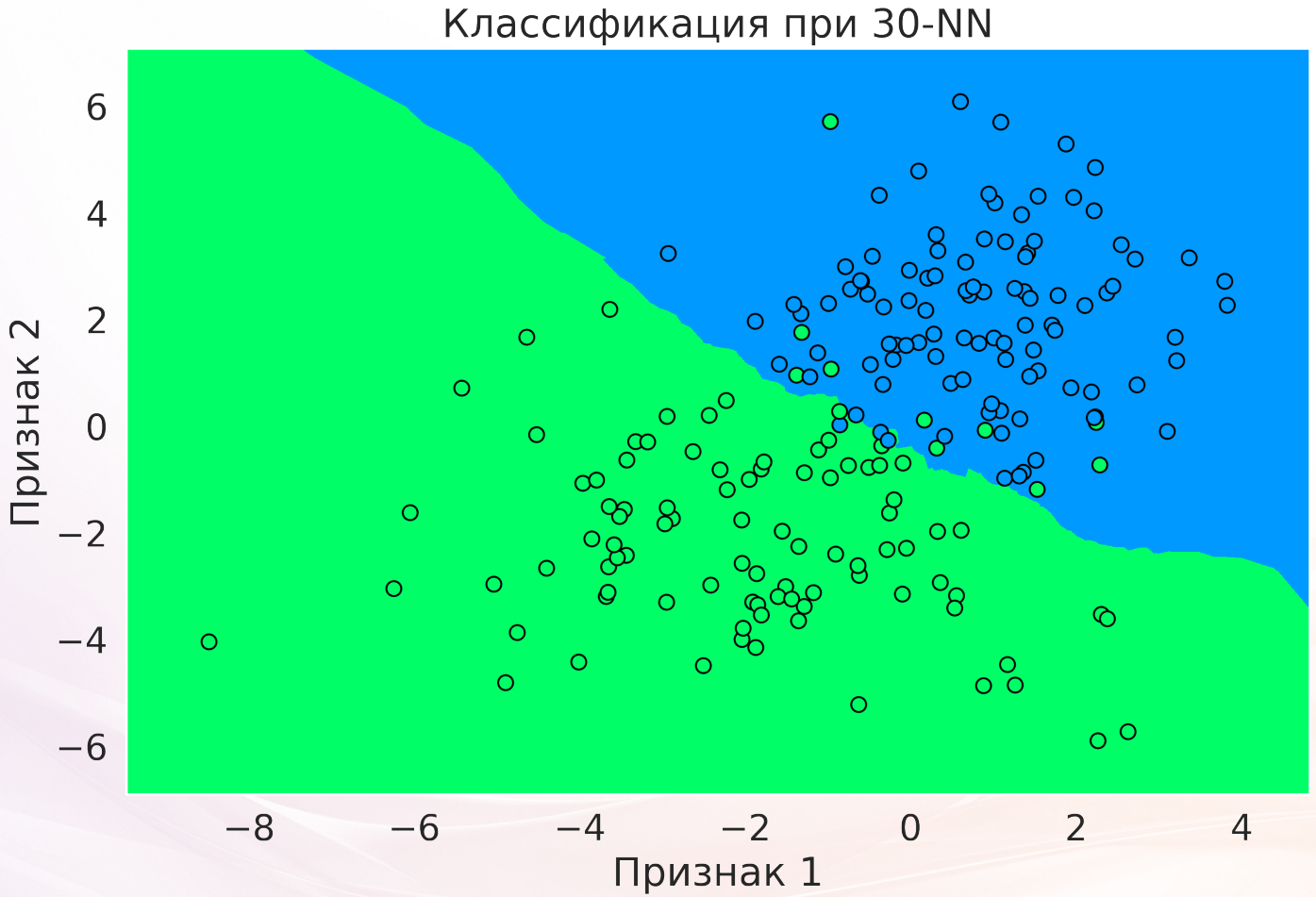
Что происходит при разных k?



Что происходит при разных k?



Что происходит при разных k?



Как оценить качество классификации?

Пусть $\widehat{y}(x)$ — оценка класса для объекта x.

Можем посчитать **точность** — доля правильно угаданных классов

$$A = \frac{1}{n}\sum_{i=1}^{n} I\{Y_i = \widehat{y}(x_i)\}$$

Оценка качества называется **метрикой** (*не путать с метр. пр-вами*).

Какое число соседей оптимизирует эту метрику?

Ответ: $k = 1$, т.к. при вычислении $\widehat{y}(x_i)$ берем сам Y_i.

Поэтому данные делят случайно на **две непересекающие части**:

1. на одной определяют правило классификации,
2. на другой — считают оценку качества классификации.

Точность 90% это много или мало?

Кажется, круто. А если в данных 85% котов? Тогда отвечая всегда "кот" сможем добиться точности 85%, и 90% уже не так круто...

А что если по картинке?

Хорошо, но что если объект — изображение кота или собаки?

Изображение 100×100 состоит из 10^4 пикселей,

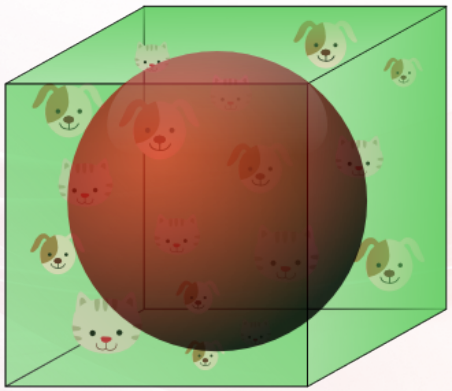
в каждом по 3 числа. Какой размерности получается объект?

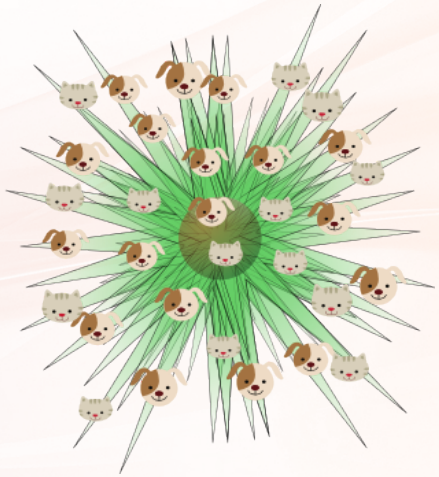
Ответ: $100 \times 100 \times 3 = 30\,000$ чисел в одной картинке.

Проблема:

в пр-ве больших размерностей расстояния неинформативны.

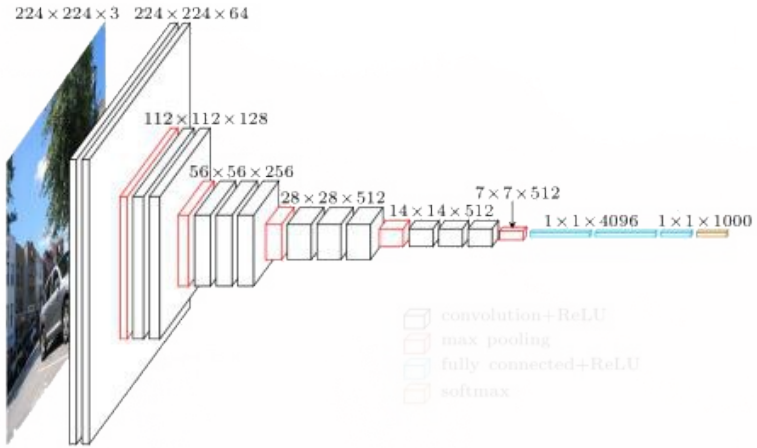
Например, среди фиксированного количества случайных точек
в единичном кубе в пространстве большой размерности
почти все точки будут лежать около границы куба.

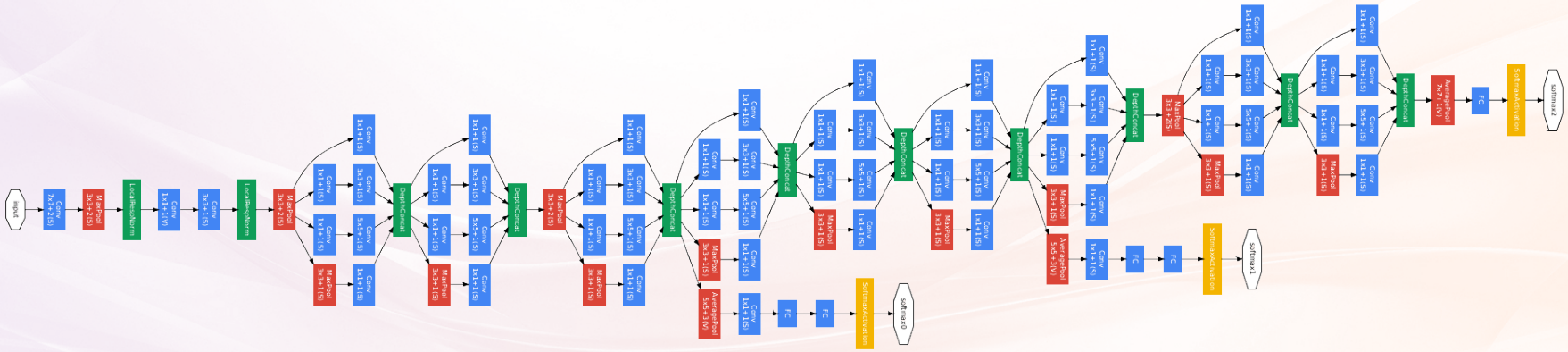




А что в сложных случаях?

Нейросети! Но об этом позже :)





Примеры прикладных задач

АБ-тестирование

Распознавание лиц

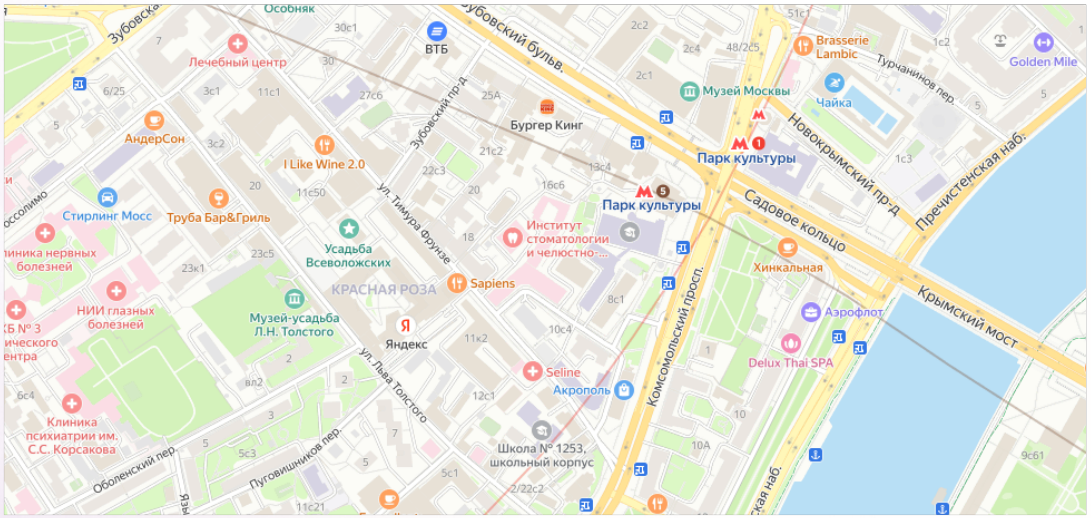
Генерация изображений

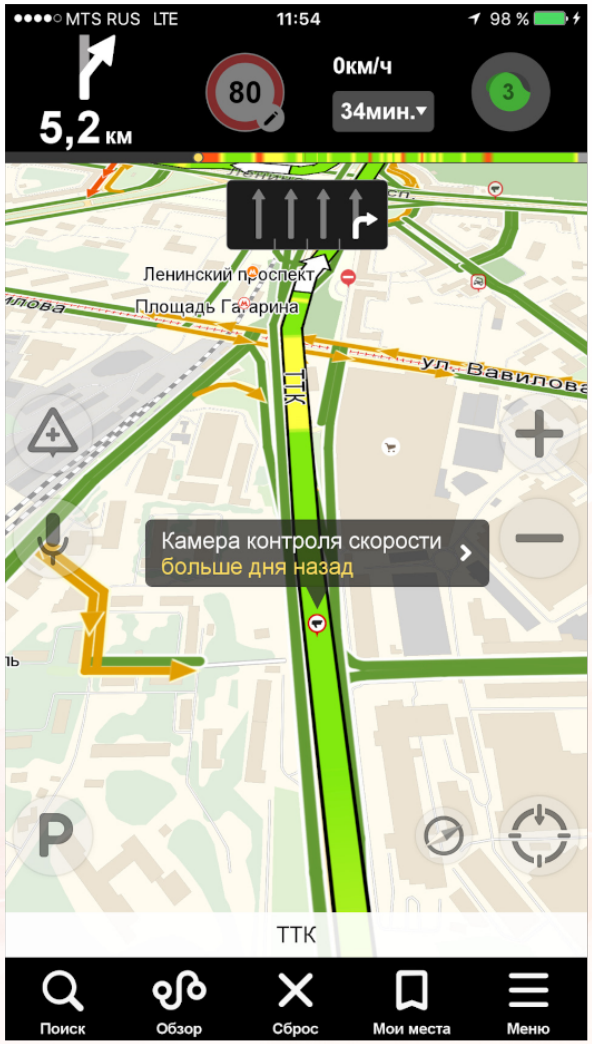
Синтез речи

Обучение с подкреплением

Где еще?

Какой стиль карт удобнее пользователям?





Какая концепция магазина приносит больше выручки?

Контрольная группа
Группа A

Пользователи видят
прежнюю версию сервиса

Тестовая группа
Группа B

Пользователи видят
новую версию сервиса

Классический способ проверки

Пусть $X_1, ..., X_n$ и $Y_1, ..., Y_m$ — значения целевой метрики (выручка, клики, рейтинг и т.д.) для контрольной и тестовой групп.

Постановка задачи:

$\mathrm{H}_0 : \mathrm{E}X_1 = \mathrm{E}Y_1$ *vs.* $\mathrm{H}_1 : \mathrm{E}X_1 \ \{<, \neq, >\} \ \mathrm{E}Y_1$

Справедлива сходимость

$$T(X, Y) = \frac{\overline{X} - \overline{Y}}{\sqrt{S_X^2 / n + S_Y^2 / m}} \xrightarrow{d_0} \mathcal{N}(0, 1).$$

Статистический критерий $S = \{|T(X, Y)| > z_{1-\alpha/2}\}$

Доверительный интервал для $\mathrm{E}X_1 - \mathrm{E}Y_1$ уровня доверия $1 - \alpha$

$$\left(\overline{X} - \overline{Y} \pm z_{1-\alpha/2}\sqrt{S_X^2 / n + S_Y^2 / m}\right).$$

При наличии дополнительных данных

Введем обозначения

$$Y = \begin{pmatrix} Y_{11} \\ \dots \\ Y_{n1} \\ Y_{12} \\ \dots \\ Y_{m2} \end{pmatrix}, \quad X = \begin{pmatrix} 1 & 0 \\ \dots & \\ 1 & 0 \\ 1 & 1 \\ \dots & \\ 1 & 1 \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \dots \\ \varepsilon_{n+m} \end{pmatrix}.$$

Линейная регрессия предполагает зависимость $Y = X\theta + \varepsilon$.

Тогда $Y_{i1} = \theta_0 + \varepsilon_i$ и $Y_{i2} = \theta_0 + \theta_1 + \varepsilon_{n+i}$, следовательно

- θ_0 — среднее в группе A,
- θ_1 — эффект от эксперимента.

Вывод: для проверки AB-теста нужно построить интервал для θ_1 и проверить гипотезу $H_0 : \theta_1 = 0$ критерием Стьюдента.

Важно использовать оценку дисперсии, устойчивую к гетероскедастичности, т.к. группы могут иметь разную дисперсию.

Примеры прикладных задач

АБ-тестирование

Распознавание лиц

Генерация изображений

Синтез речи

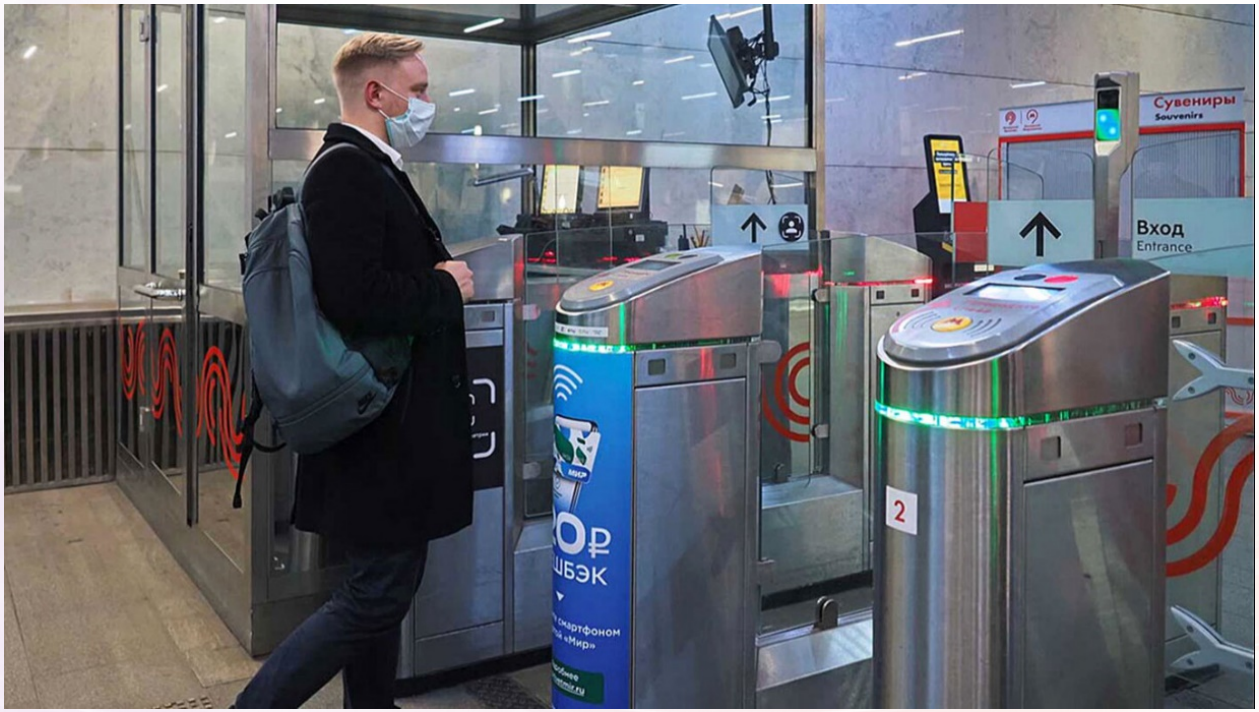
Обучение с подкреплением

Где еще?

Распознавание лиц

Фото человека → Модель детекции лиц → Координаты лица→

Обработка фото → Сиамская сеть → Идентификатор человека



Распознавание лиц

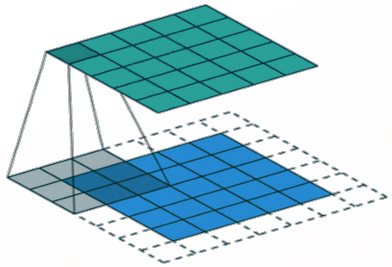
Пример **модели детекции лиц**: TinaFace.

- Основу модели составляют сверточные слои.

  Формула свертки для изображения X

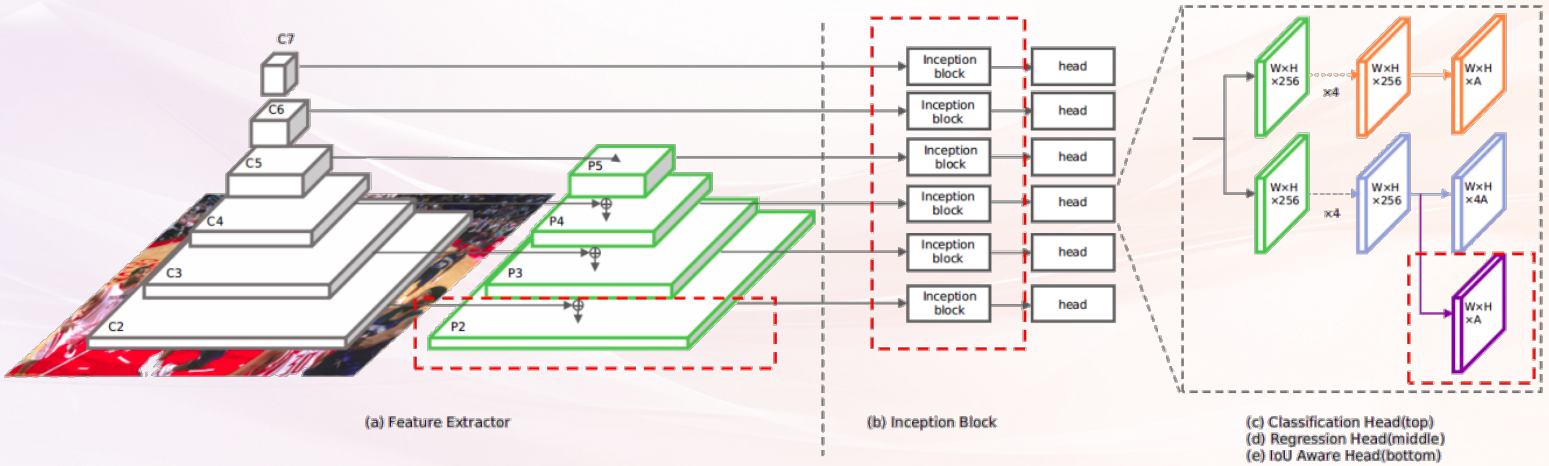
  и фильтра с весами W и сдвигом b:

$$\sum_{i=1}^{M}\sum_{j=1}^{M} w_{ij} \cdot x_{m+i-1,n+j-1} + b$$



- Архитектура модели содержит множество блоков

  из сверточных слоев, функций активаций и т.д.



Распознавание лиц

- **Сиамская сеть** для двух объектов X_1 и X_2 определяет, принадлежат ли они одному классу, оценивая близость между ними.
- Архитектура модели представляет собой сверточную сеть.
- Для оптимизации параметров модели минимизируется **contrastive loss**.

Пусть Y_1 и Y_2 — классы объектов X_1 и X_2 соотв.,
d — функция расстояния,
L_{sim} и L_{dissim} — функции штрафующие за близость объектов одного класса и дальность объектов разных классов соотв.
Тогда лосс равен:

$$L(X_1, X_2, Y_1, Y_2) = I\{Y_1 = Y_2\} L_{sim}\Big(d\left(f(X_1), f(X_2)\right)\Big)$$

$$+ I\{Y_1 \neq Y_2\} L_{dissim}\Big(d\left(f(X_1), f(X_2)\right)\Big)$$

Примеры прикладных задач

АБ-тестирование

Распознавание лиц

Генерация изображений

Синтез речи

Обучение с подкреплением

Где еще?

Генерация изображений

Задача — научиться генерировать разнообразные правдоподобные изображения, например котов.

Генерация изображений

Для этого построим **диффузионную модель**.

Она моделирует 2 процесса:

- Прямой процесс — постепенно добавляем шум ко входу.
- Обратный процесс — модель постепенно восстанавливает данные из шума.

Генерация изображений

Прямой диффузионный процесс

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\varepsilon, \text{ где } \varepsilon \sim \mathcal{N}(0, \mathrm{I})$$

$$X_t | X_{t-1} \sim \mathcal{N}(\sqrt{1-\beta_t} X_{t-1}, \beta_t \mathrm{I})$$

Генерация изображений

Обратный диффузионный процесс

Обучается нейросетевая модель таким образом,

чтобы минимизировать *ELBO*:

$$ELBO = \mathsf{E}_q \log \frac{p_\theta(X_0, \ldots, X_T)}{q(X_1, \ldots, X_T | X_0)}$$

$$\simeq const - \sum_{t=2}^{T} \frac{\widetilde{\alpha}_{t-1}\beta_t^2}{2\widetilde{\beta}_t(1-\alpha_t)^2} \mathsf{E}_q \|X_0 - x_\theta(X_t, t)\|^2$$

Примеры прикладных задач

АБ-тестирование

Распознавание лиц

Генерация изображений

Синтез речи

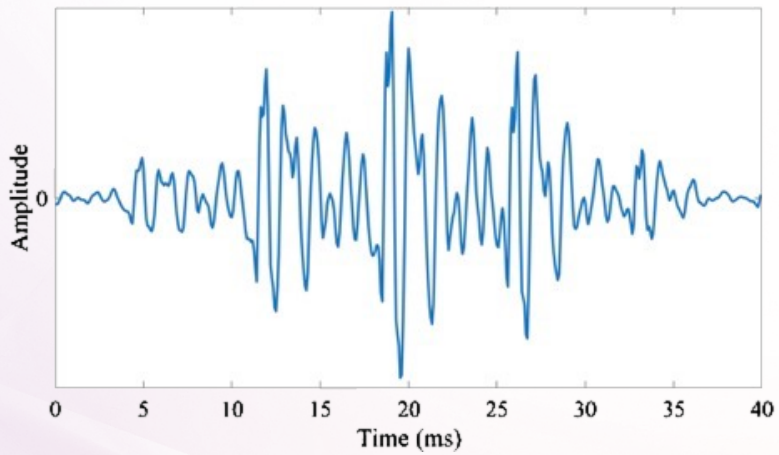
Обучение с подкреплением

Где еще?

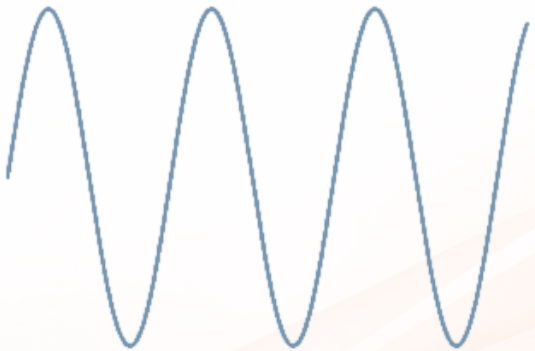
Синтез речи

Что такое звук?

Звук



Волна



Звук — композиция волн с разными амплитудами и частотой.

Волна — периодич. ф-ия, имеющая амплитуду, период и частоту.

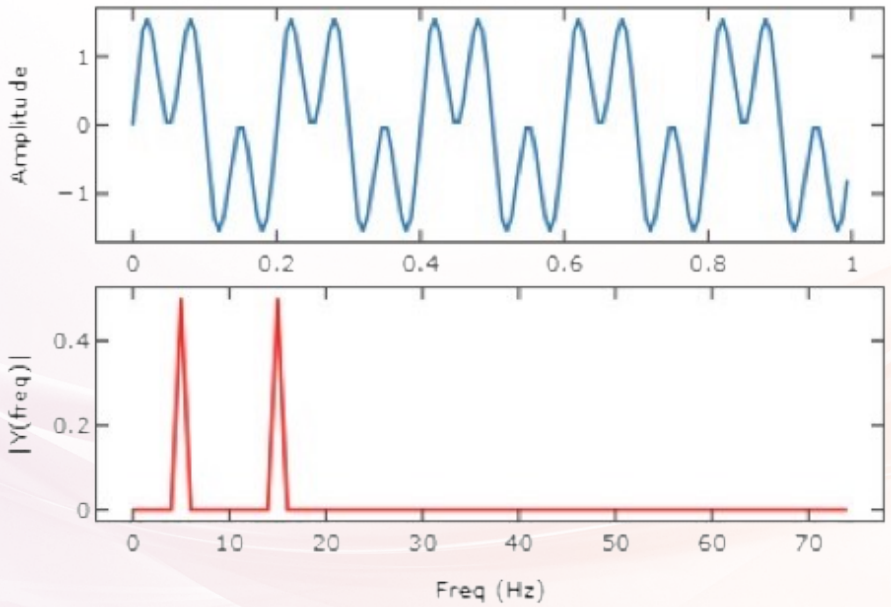
Синтез речи

Как работать со звуком?

Посмотрим на распределение частот волн в звуке.

Для этого применяется дискретное преобразование Фурье:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N} kn} = \sum_{n=0}^{N-1} x_n \left[\cos\left(\frac{2\pi}{N} kn\right) - i \sin\left(\frac{2\pi}{N} kn\right) \right]$$



Пример для звука из двух волн

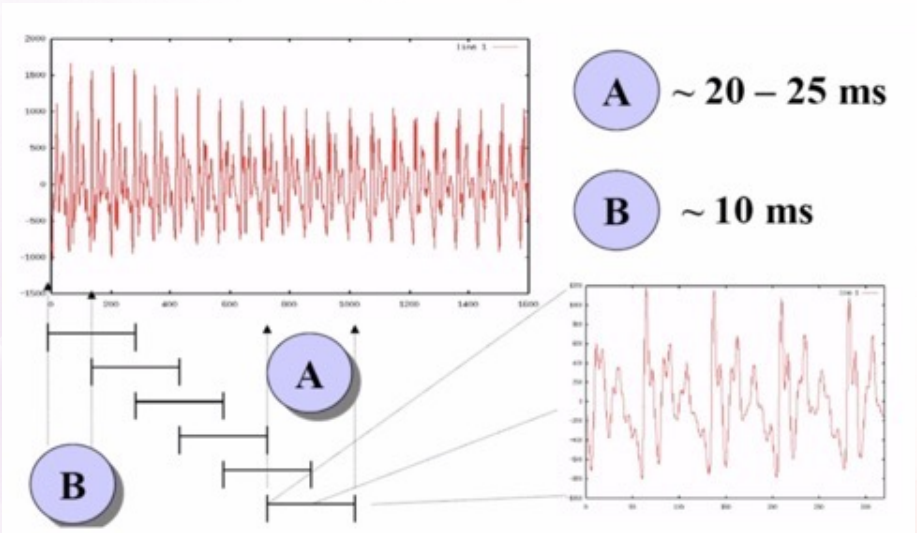
Синтез речи

Как работать со звуком?

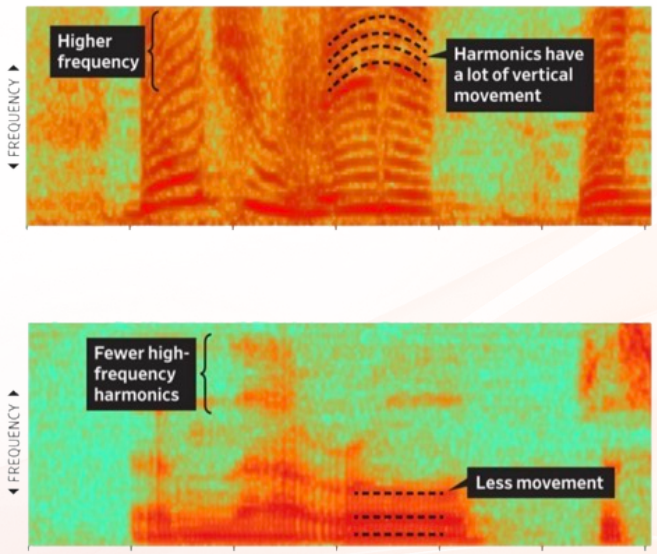
У результата преобразования Фурье ко всем данным нет времени.

Посчитаем распределение на окнах из звука.

Окна



Спектограмма



- Берем маленькие окна (20-25 мс) от исходных данных.
- К каждому окну применяем преобразования Фурье.
- Стакаем распределения частот вместе, получаем спектрограмму.

Синтез речи

1. Акустическая модель по тексту

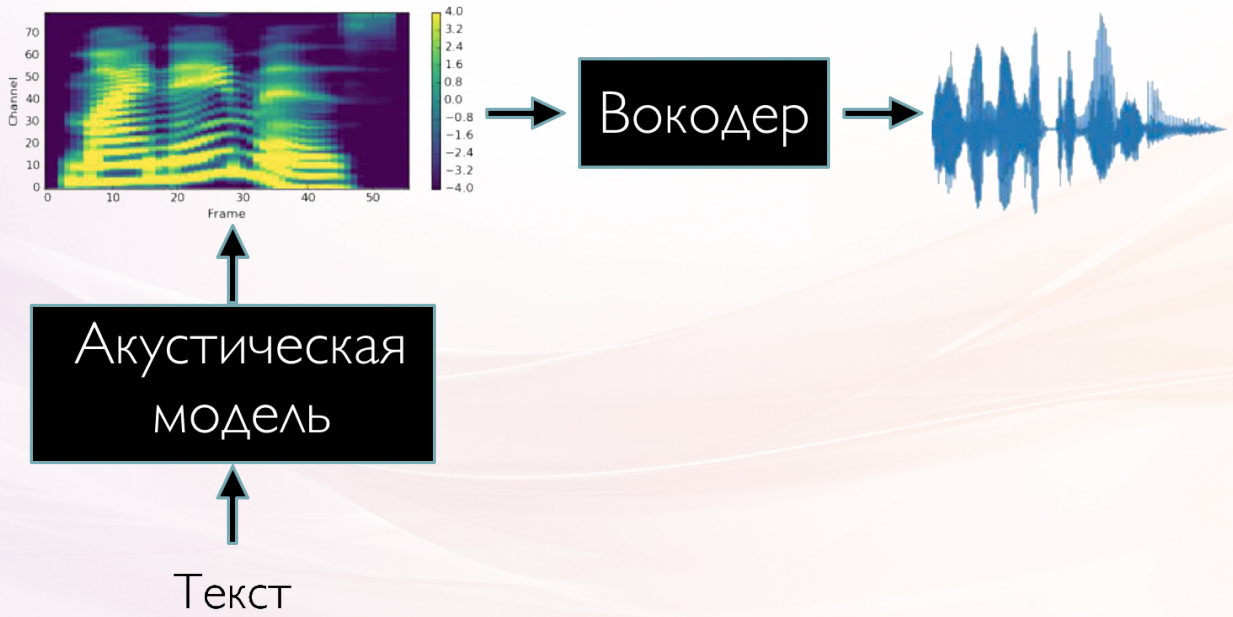
   предсказывает мел-спектограмму.

   Акустическая модель — это какая-то нейросеть.

2. Вокодер по мел-спектограмме предсказывает аудио.

   Вокодер может быть как алгоритмом, так и нейросетью.

   Некоторые известные вокодеры используют байесовские методы.



Примеры прикладных задач

АБ-тестирование

Распознавание лиц

Генерация изображений

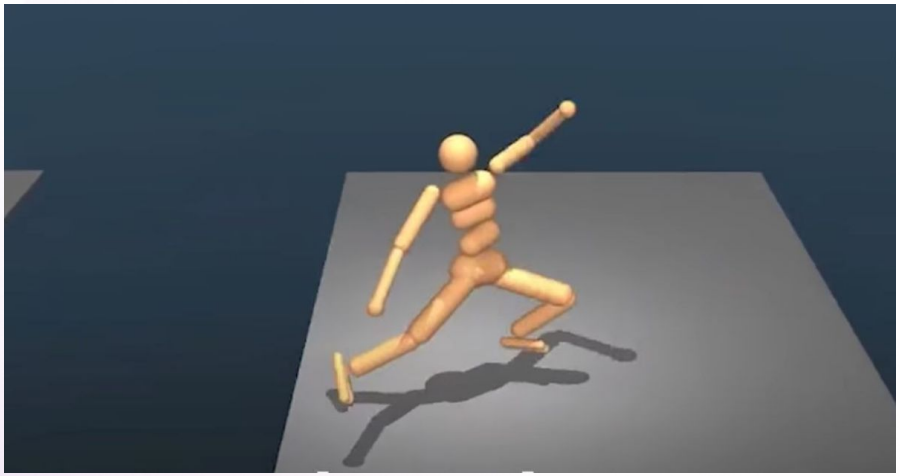
Синтез речи

Обучение с подкреплением

Где еще?

Учимся ходить

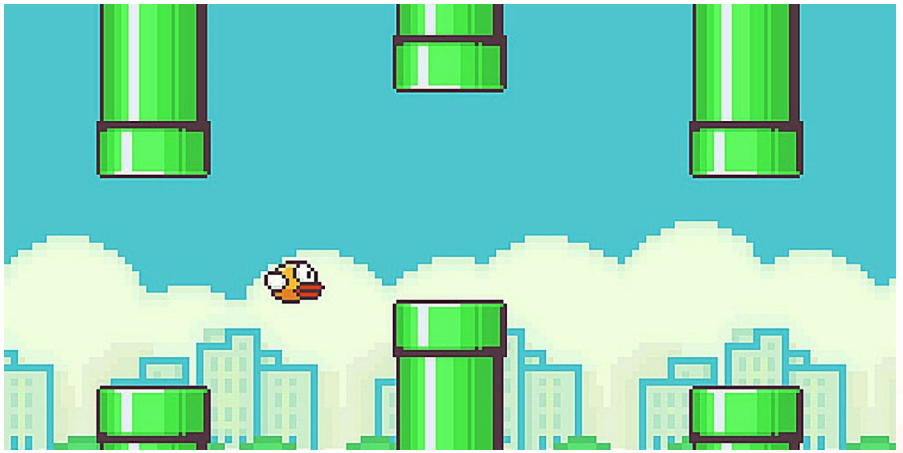
Хотим научить компьютерную симуляцию человека ходить.



Как человек учится ходить?
Делаем много попыток,
постоянно улучшая свои навыки.

Играем в игры

Хотим научить компьютер играть в игру Flappy Bird.



Как бы мы сами учились играть?
Играем много раз, в каждый из которых улучшаем свои навыки игры.

Похожим образом устроен **Reinforcement learning**.

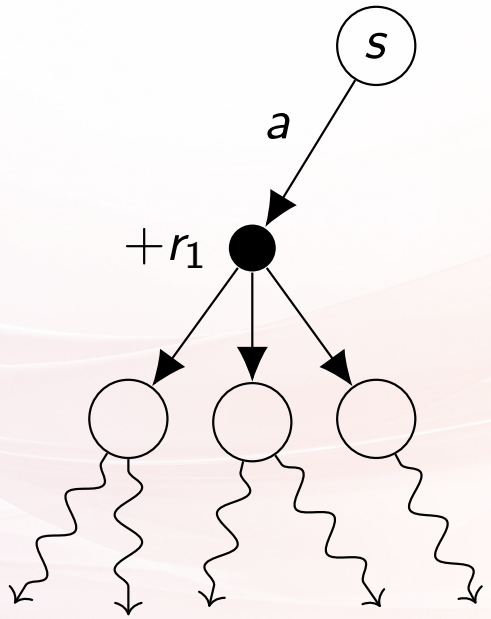
Важные определения RL

Стратегия π — способ выбора **действия** a в **среде** (стиль игры).

Награда $R(s, a)$ — величина поощрения в действии a и состоянии s.

Q-функция — ожидаемый итоговый выигрыш в будущем при условии состоянии s и совершения действия a при следовании стратегии π.

$$Q_\pi(s, a) = \mathrm{E}_{\pi,P,R}(R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \mid S_t = s, A_t = a)$$



Свойства Q-функции

Теорема (об оптимальности). Существует π^* такая, что она даёт выигрыш не хуже любой другой стратегии π:

$$Q_{\pi^*}(s, a) = \max_{\pi} Q_{\pi}(s, a) = Q_{opt}(s, a)$$

Теорема (уравнение Беллмана). В произвольной среде и для произвольной стратегии π выполнено соотношение

$$Q_{\pi}(s, a) = \mathrm{E}R(s, a) + \gamma \sum_{s' \in \mathcal{S}, a' \in \mathcal{A}} P(s'|s, a)\, \pi(a'|s')\, Q_{\pi}(s', a').$$

Уравнение показывает, как Q-функция обновляется: текущая ценность действия складывается из непосредственной награды и лучших возможных будущих действий.

Q-learning

Если знаем оптимальную Q-функцию, можем определить **лучшую стратегию действий**. Воспользуемся итерационной процедурой:

$$Q(s_t, a_t) := Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

Этот процесс называется **Q-learning**:

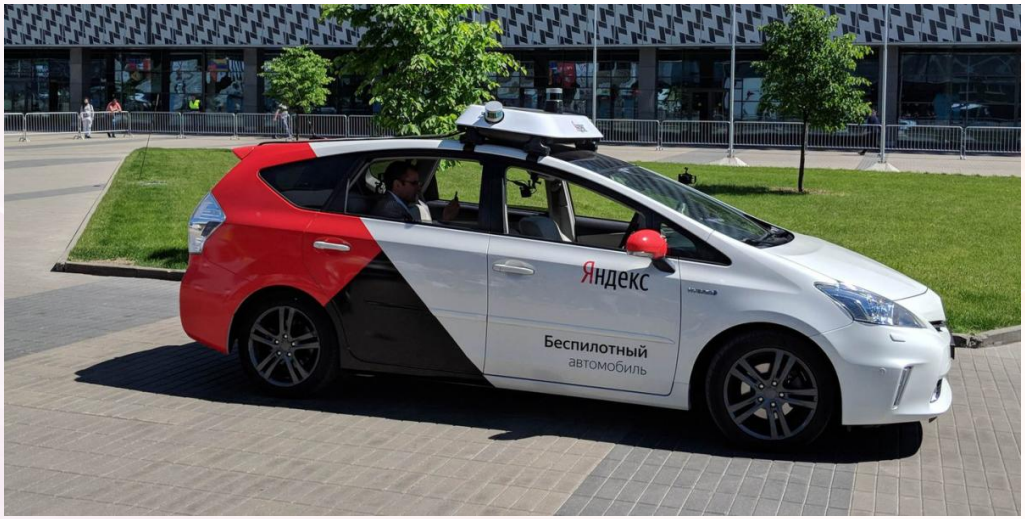
- Агент **совершает действие** и получает **награду**.
- **Обновляет** Q-значение, учитывая **лучшие будущие действия**.
- Со временем стратегия становится всё более оптимальной.

Пример: Учимся играть в новую игру. Сначала ты пробуем случайные действия, но постепенно запоминаем, какие из них приводят к победе.

Q-learning

Теорема. В случае конечного MDP в алгоритме Q-learning $Q(s, a)$ сходится к $Q_{opt}(s, a)$ с вероятностью 1, если

1. $\forall s \in \mathcal{S}, a \in \mathcal{A}$ $\mu(a|s) \neq 0$, μ — стратегия для сессий в обучении
2. $\sum_{k=1}^{\infty} \alpha_k = \infty$, где α_k — степень **exploration** среды на шаге k
3. $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$.



Примеры прикладных задач

АБ-тестирование

Распознавание лиц

Генерация изображений

Синтез речи

Обучение с подкреплением

Где еще?

- Беспилотные автомобили
- Ранжирование результатов в поисковой системе
- Поиск по картинкам и видео на основе содержания
- Распознавание спама/фрода
- Прогноз погоды
- Прокладывание маршрутов в навигаторах
- AI-камеры в телефонах
- Генерация картин подобно художникам
- Оплата проезда в метро взглядом
- Решение о выдаче кредита
- Персонализированная реклама
- Определение причин оттока клиентов
- Прогнозирование спроса на товар
- Определение месторасположения новой торговой точки
- Расстановка полок в магазинах
- Автоопределение свежести товаров в магазинах
- Распознавание патологий на медицинских снимках
- Персонализированная медицина
- Прогнозирование поломок оборудования
- Автоматическая система оптимального управления оборудованием

Слово студентам DS-потока

Вероника Прохорова

3 курс, DS-поток

tg: @poopppoop2

ВСЁ!