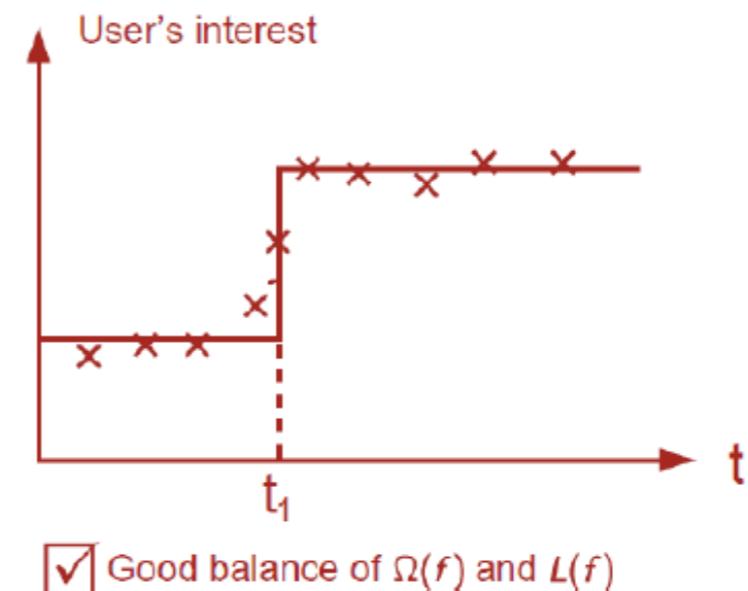
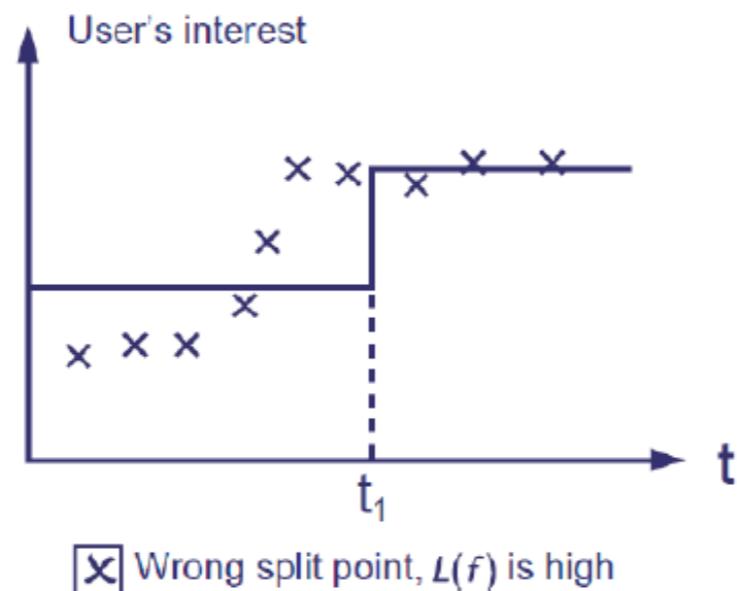
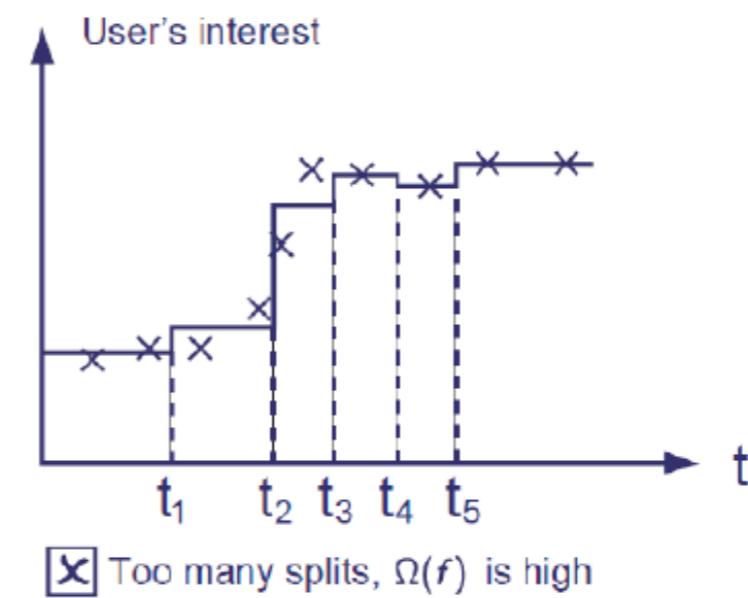
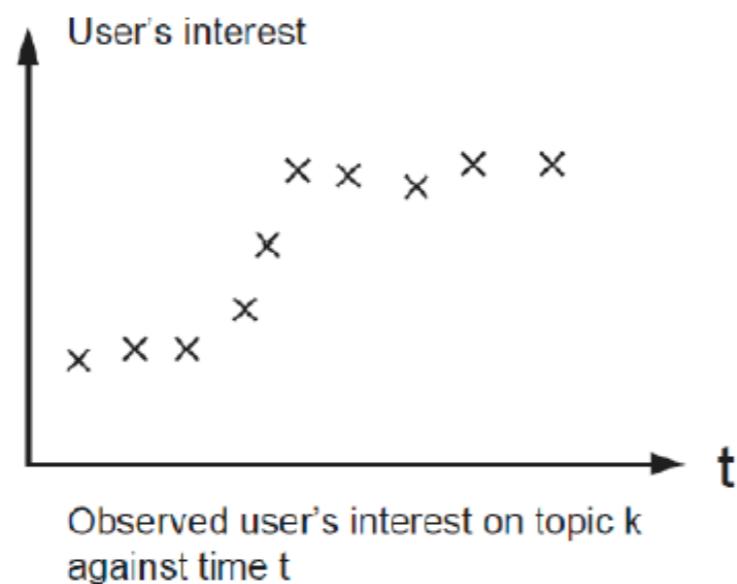


XGBoost

Компромисс между функцией потерь и регуляризатором



XGBoost

Как мы обучаем обычный градиентный бустинг

$$err(h) = \sum_{j=1}^N L(y_j, \sum_{i=1}^{T-1} b_i a_i(\mathbf{x}_j) + b \cdot a(\mathbf{x}_j))) \rightarrow \min_{b,a}$$

Соответственно для XGBoost задача оптимизации модифицируется следующим образом:

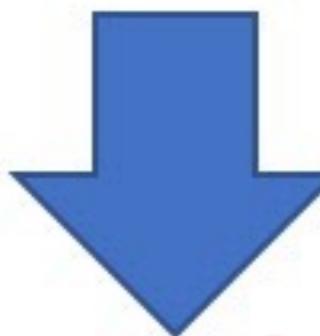
$$err(h) = \sum_{j=1}^N L(y_j, \sum_{i=1}^{T-1} b_i a_i(\mathbf{x}_j) + b \cdot a(\mathbf{x}_j))) + \sum_{i=1}^{T-1} \Omega(a_i) + \Omega(a) \rightarrow \min_{b,a}$$

LightGBM

- Гистограммный метод (ускорение, первые предложили)
- Полистовое построение дерева - leaf-wise
- Сэмплирование на основании градиента
- Способны работать с категориальными данными
- Эксклюзивная комплектация признаков

Гистограммный метод

2	3	5	9	11	12	16
---	---	---	---	----	----	----



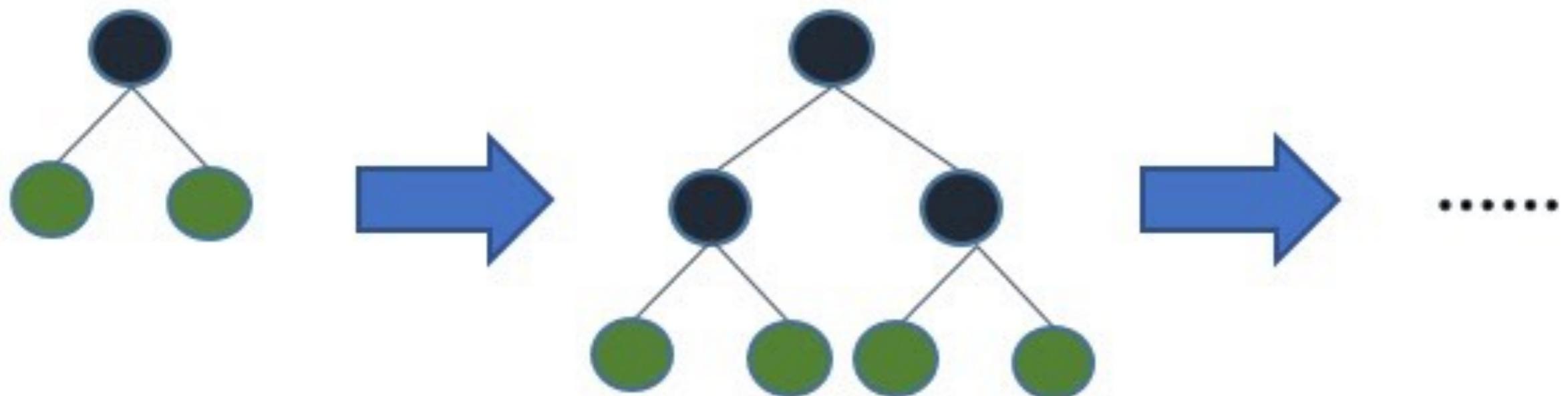
бинаризация
признаков

1	1	1	1	2	2	2
---	---	---	---	---	---	---

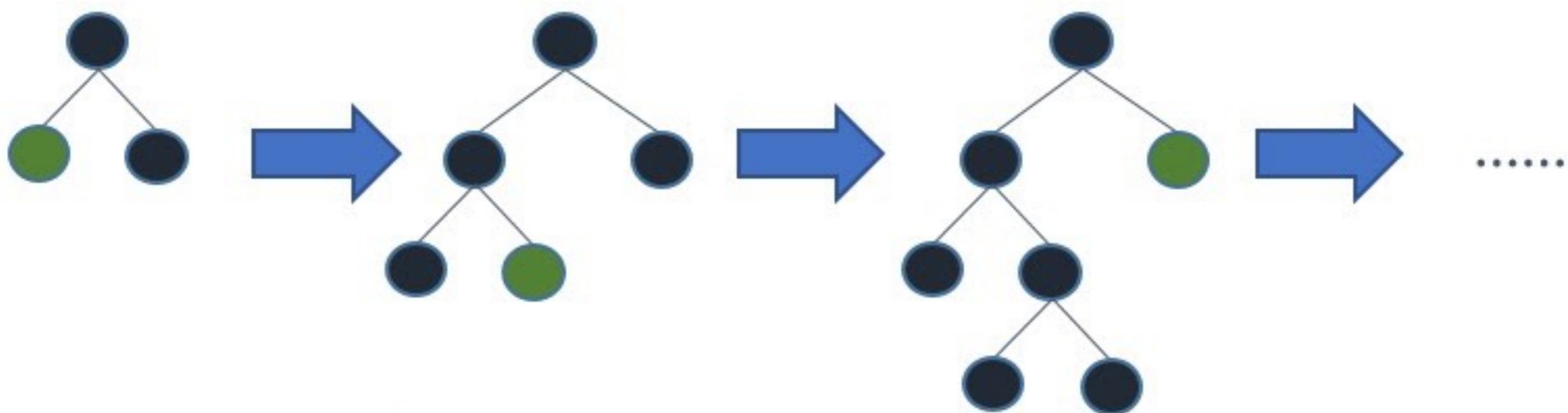
номер бина



разбиение

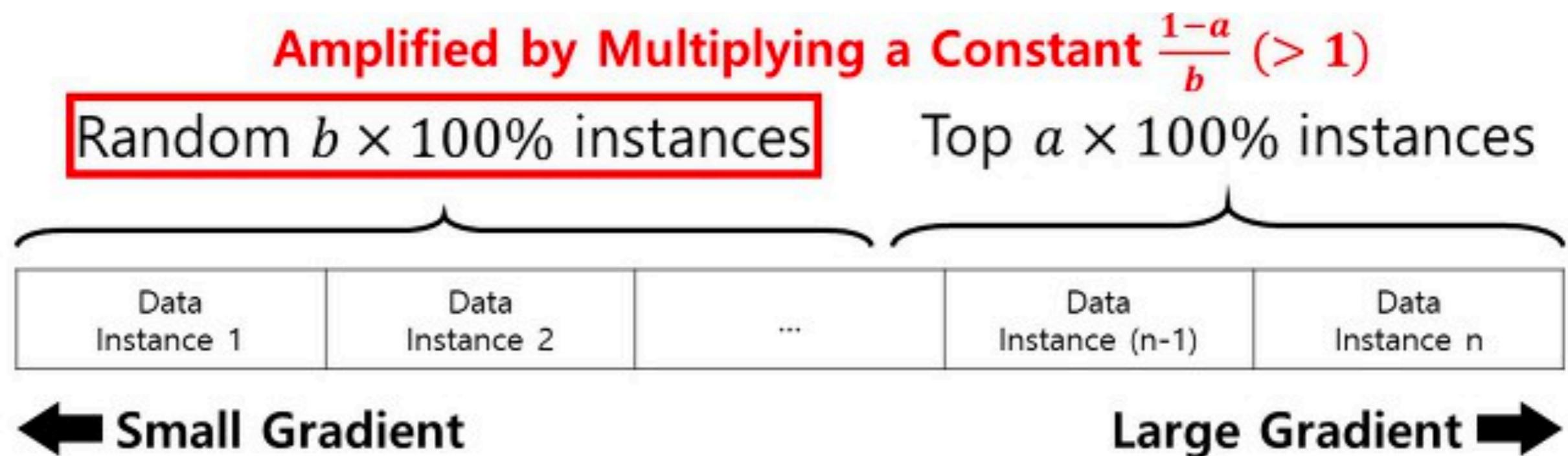


Поуроневое построение дерева



Полистовое построение дерева (первый - лучший)

Сэмплирование на основании градиента



Способы работать с категориальными данными

Как разбивать категориальные признаки?

Всего 2^N возможных разбиений

Способы работать с категориальными данными

Как разбивать категориальные признаки?

Всего 2^N возможных разбиений

Article

On Grouping for Maximum Homogeneity

Walter D. Fisher

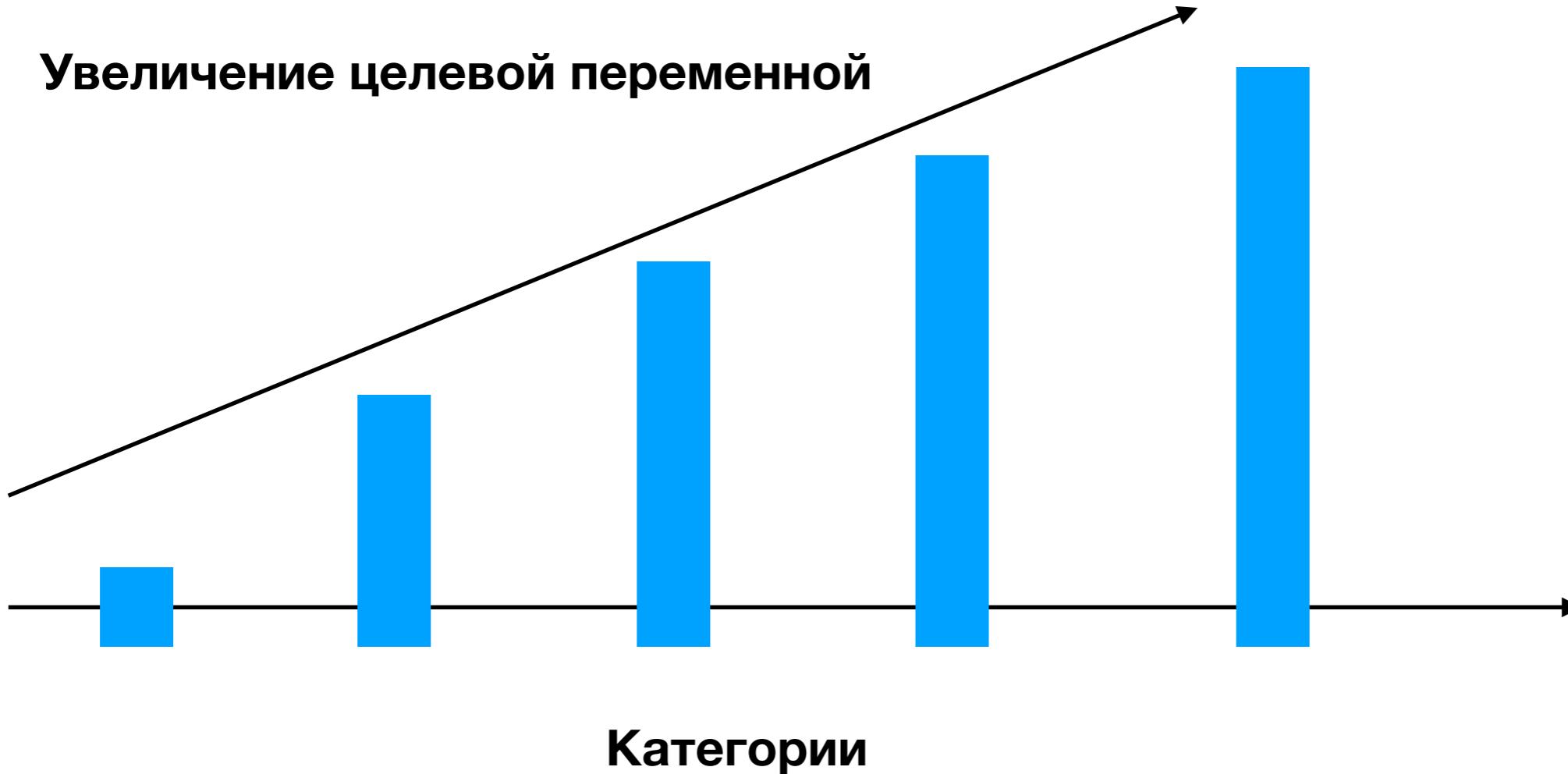
Pages 789-798 | Published online: 12 Apr 2012

“ Download citation

Можно отсортировать категории за $O(k \log k)$ и проверить
 $k-1$ разбиение

На основе чего сортируем?

Увеличение целевой переменной



Способы работать с категориальными данными

Как разбивать категориальные признаки?

Всего 2^N возможных разбиений

Article

On Grouping for Maximum Homogeneity

Walter D. Fisher

Pages 789-798 | Published online: 12 Apr 2012

“ Download citation

Можно отсортировать категории за $O(k \log k)$ и проверить
 $k-1$ разбиение

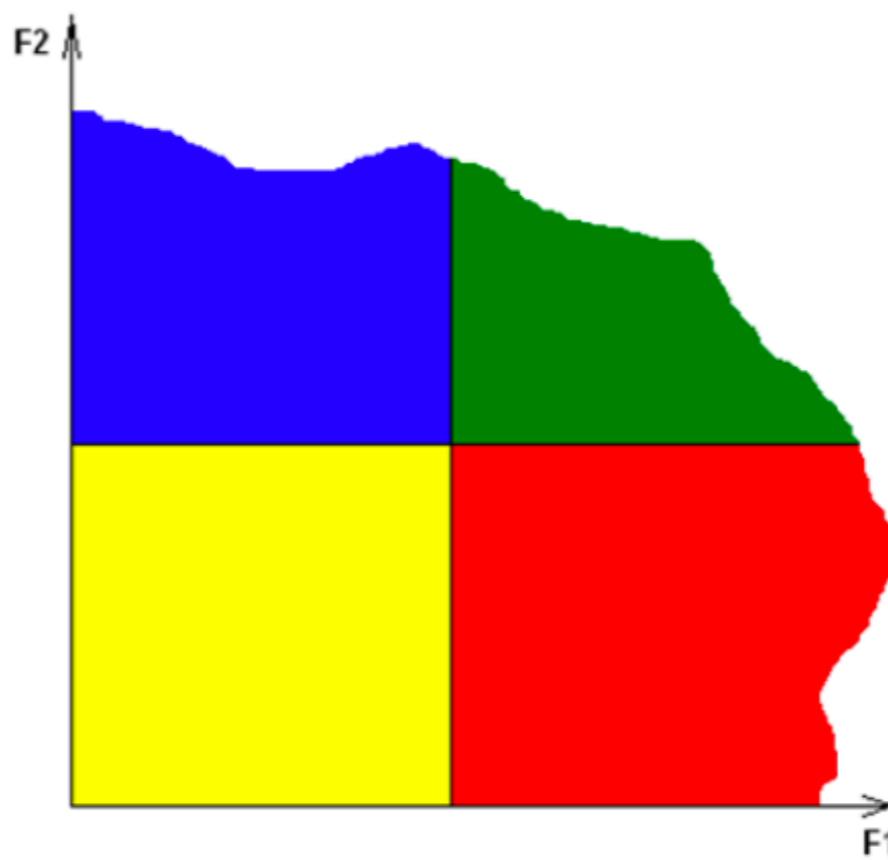
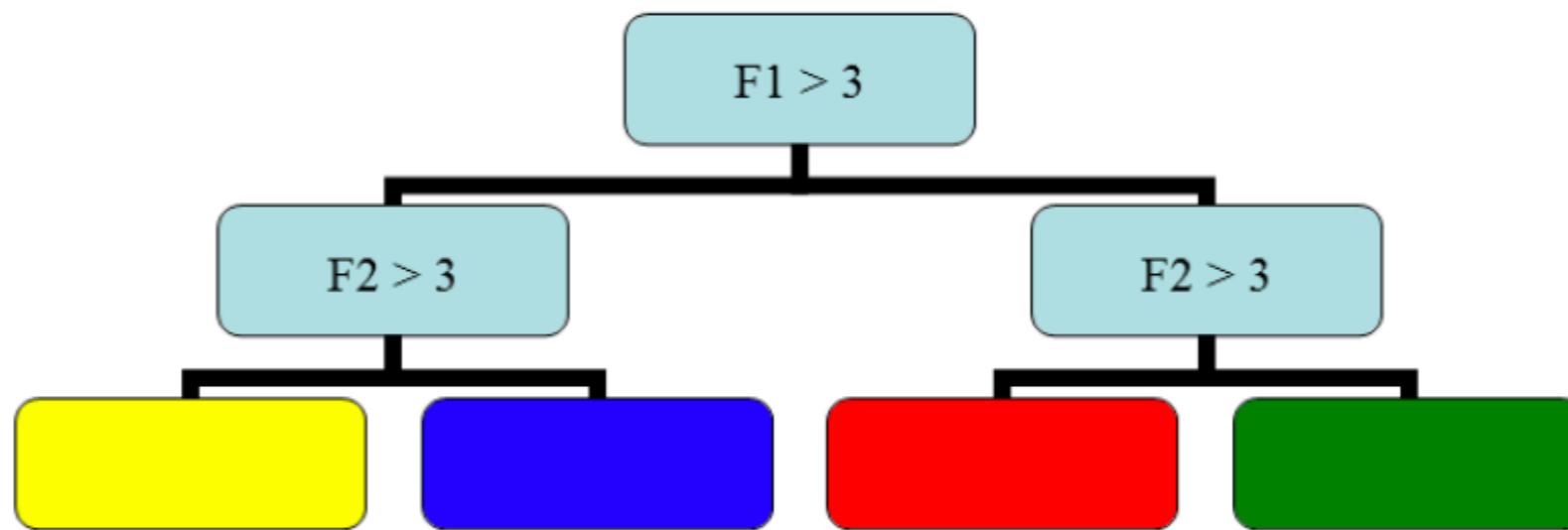
На основе чего сортируем?

На основе антиградиента

CatBoost

- Oblivious trees
- Встроенное высококачественное кодирование категориальных признаков

Oblivious Decision Tree



Работа с категориальными признаками

Кодирование меток

A/G -> 0, T/C -> 1, ...

Какой минус?

Кодирование меток

$A/G \rightarrow 0, T/C \rightarrow 1, \dots$

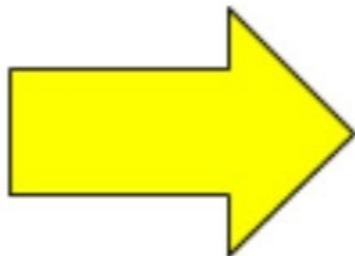
Какой минус?

**Задаем неявную информацию о том,
что $A/G > T/C$ и тд**

**Использовать только вместе с
сортировкой по предсказываемой
величине**

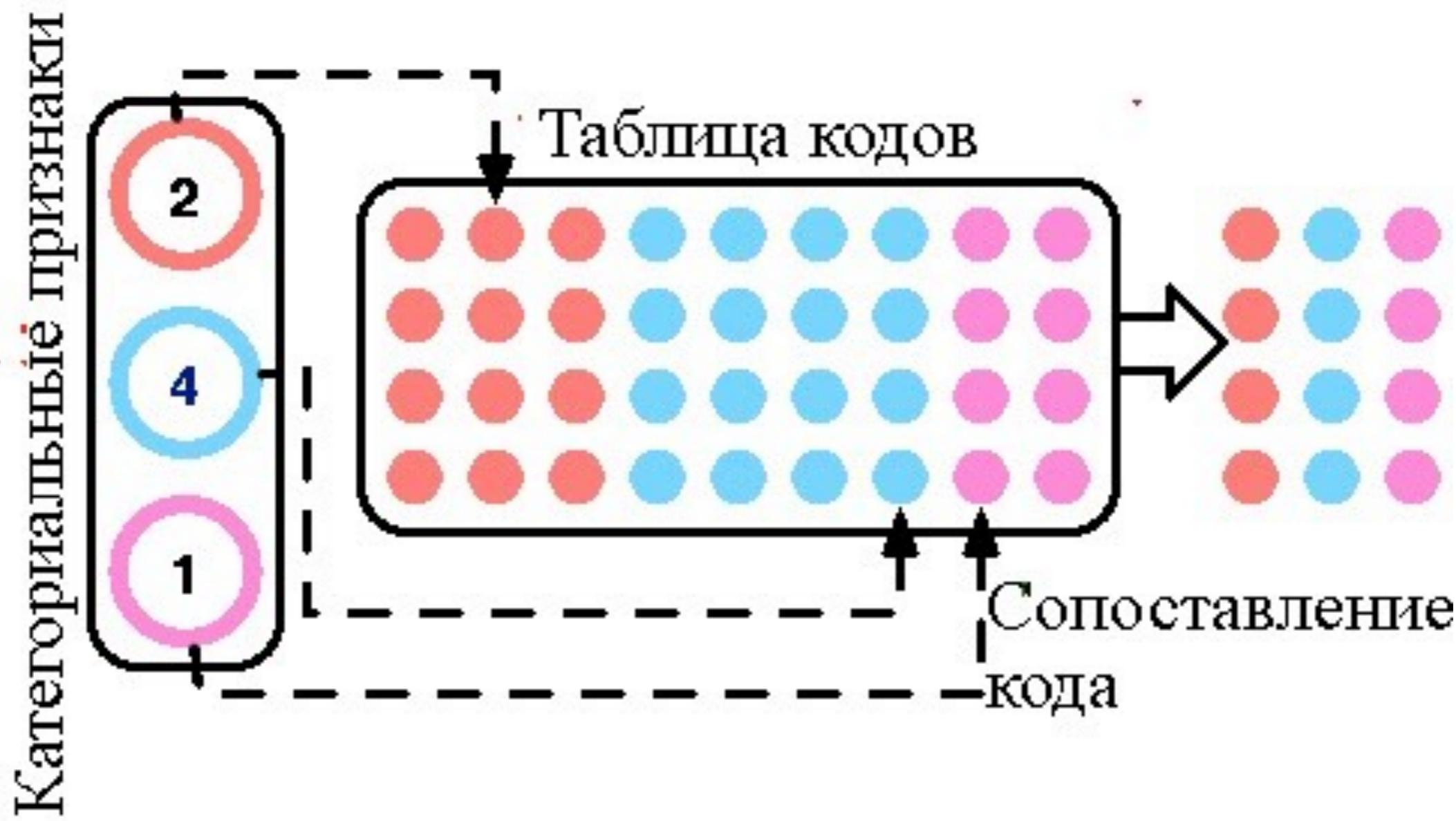
One-hot encoding

Цвет
Красный
Красный
Желтый
Зеленый
Желтый



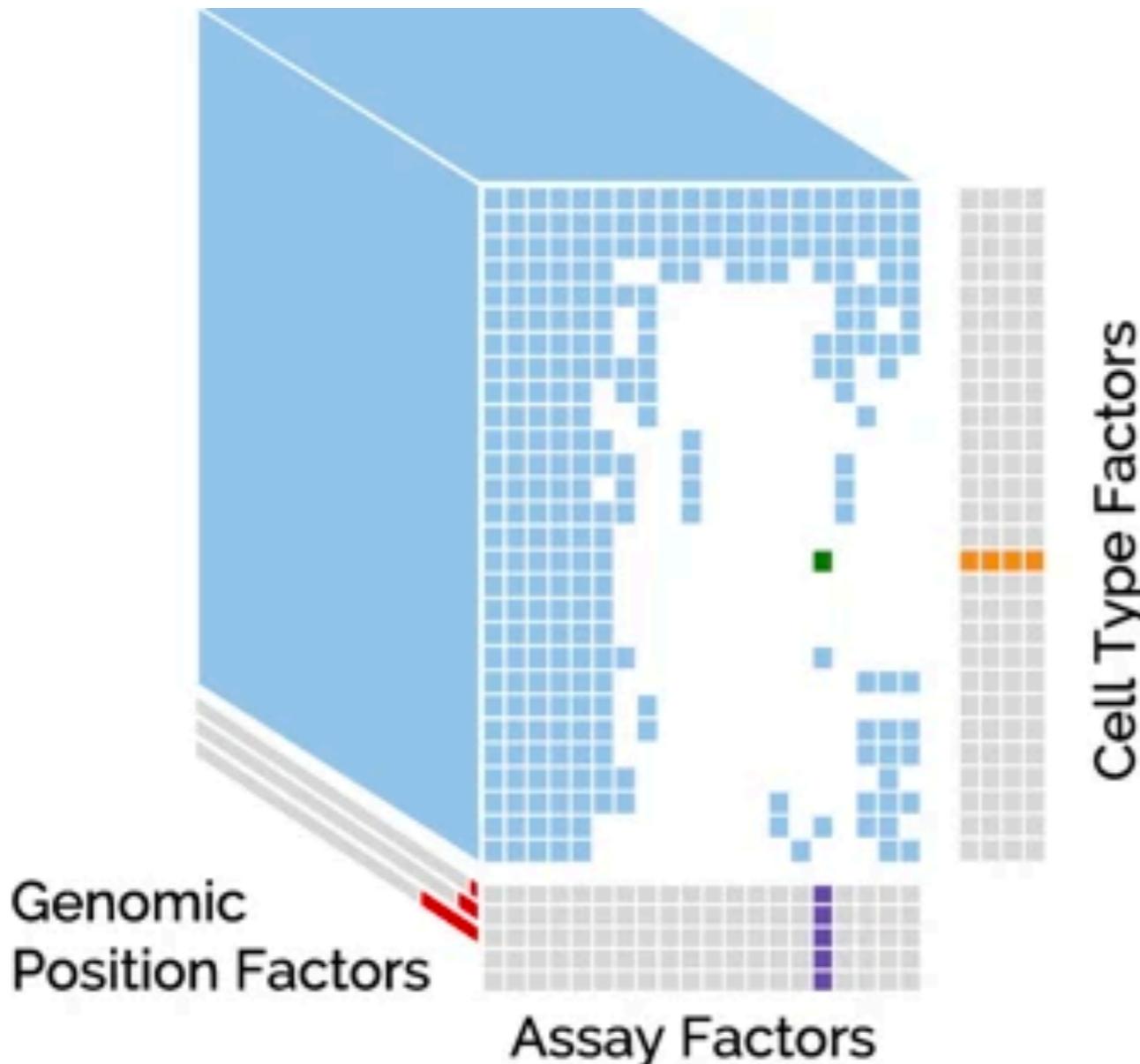
Красный	Желтый	Зеленый
1	0	0
1	0	0
0	1	0
0	0	1

Справочная таблица (lookup-table)

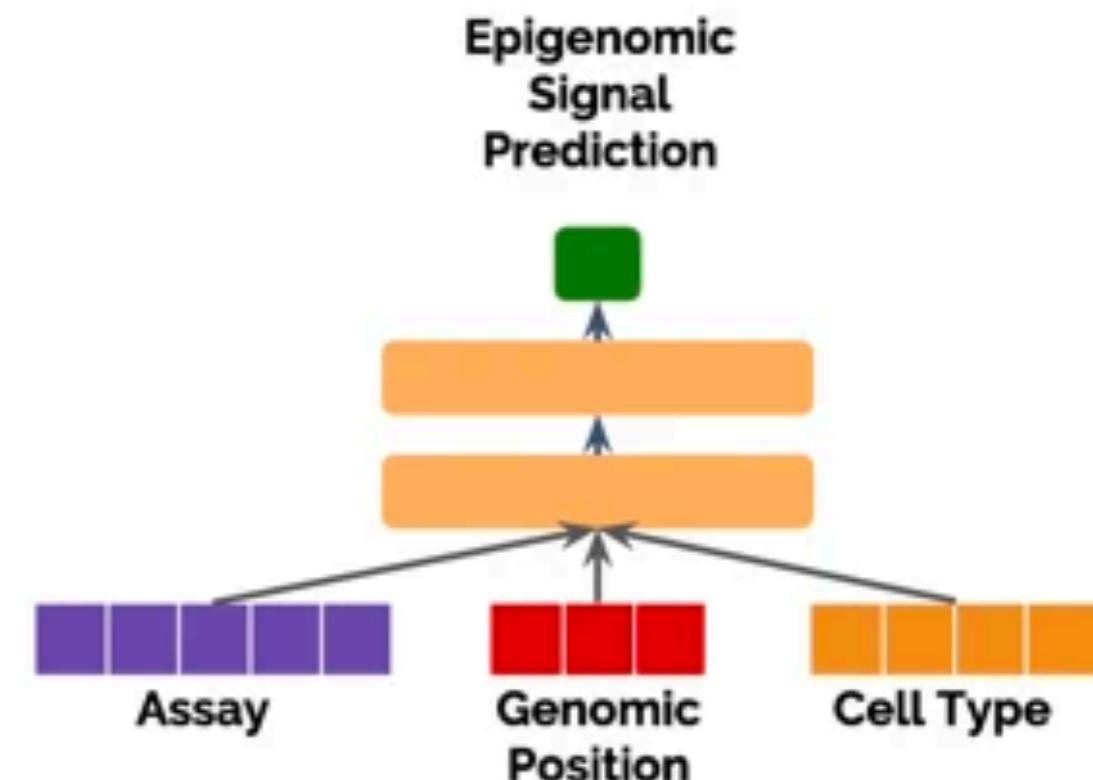


Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome

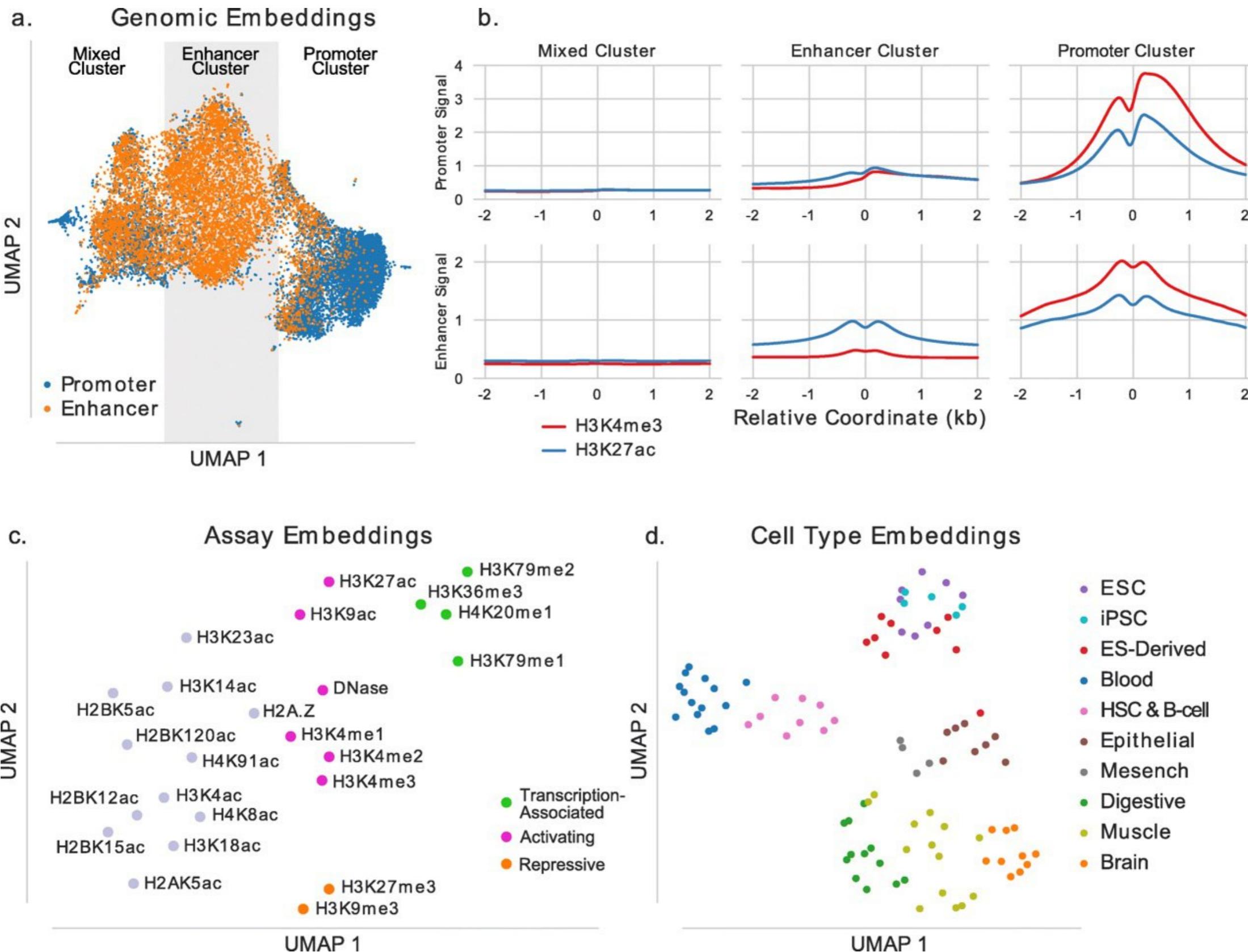
a



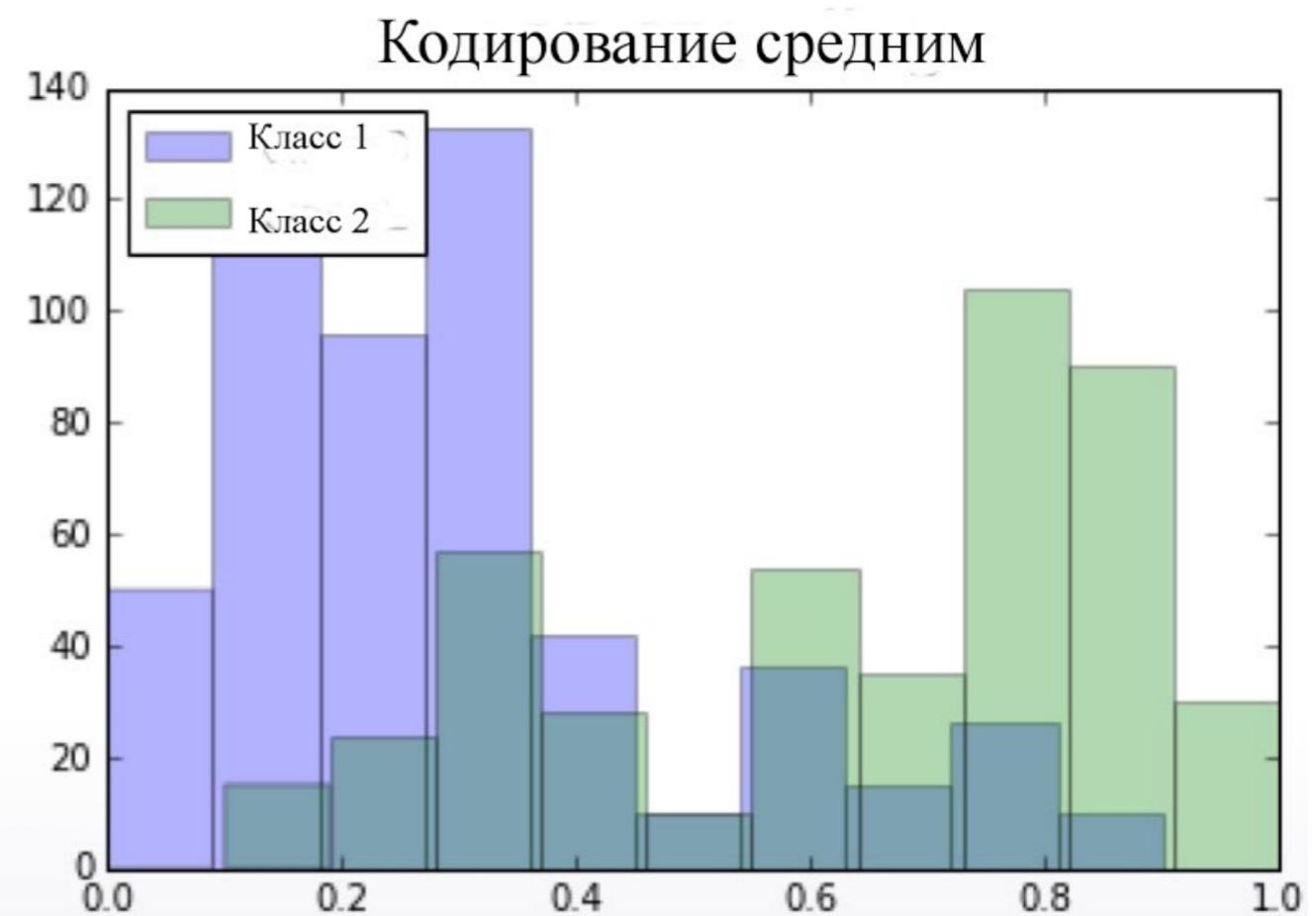
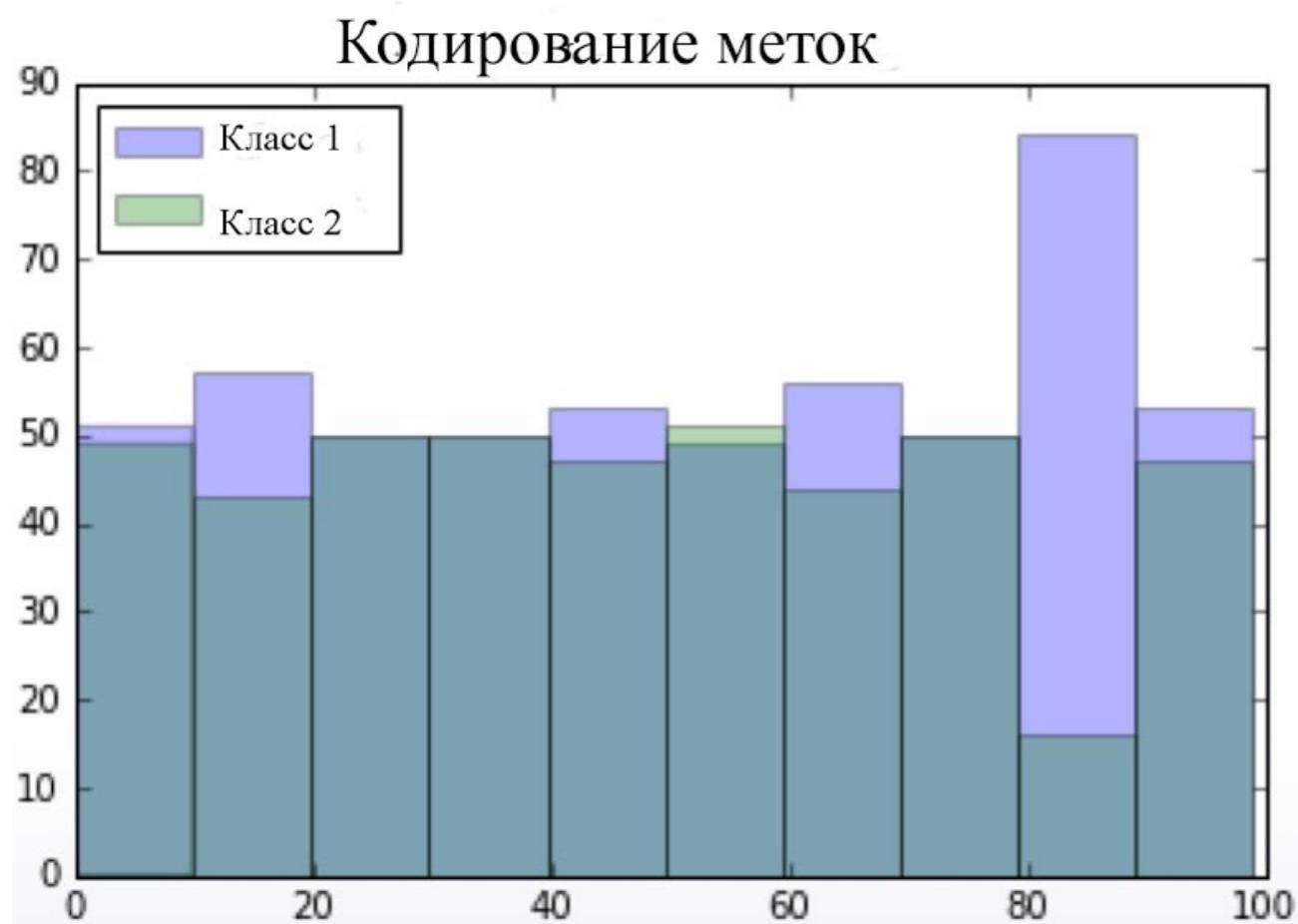
b



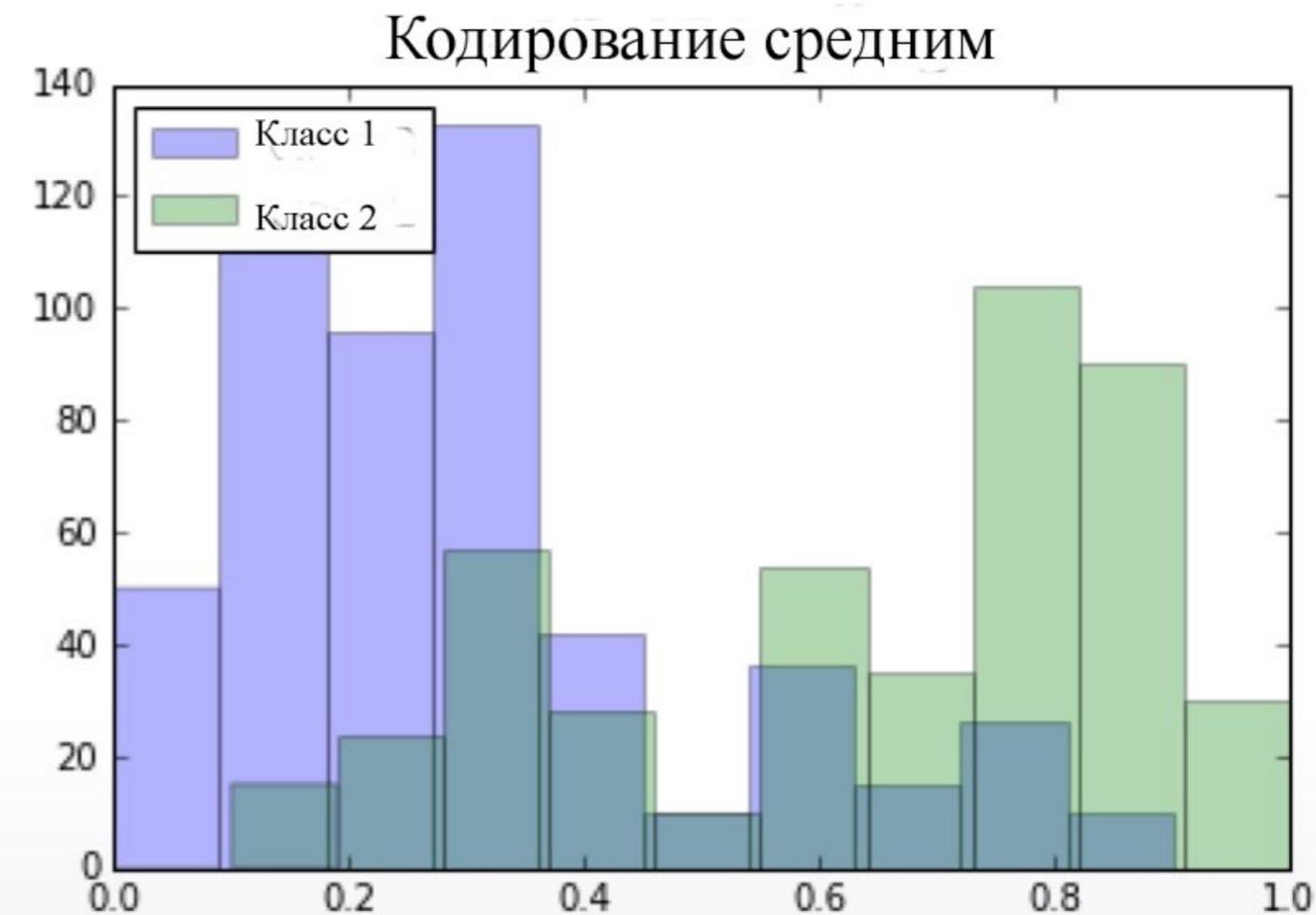
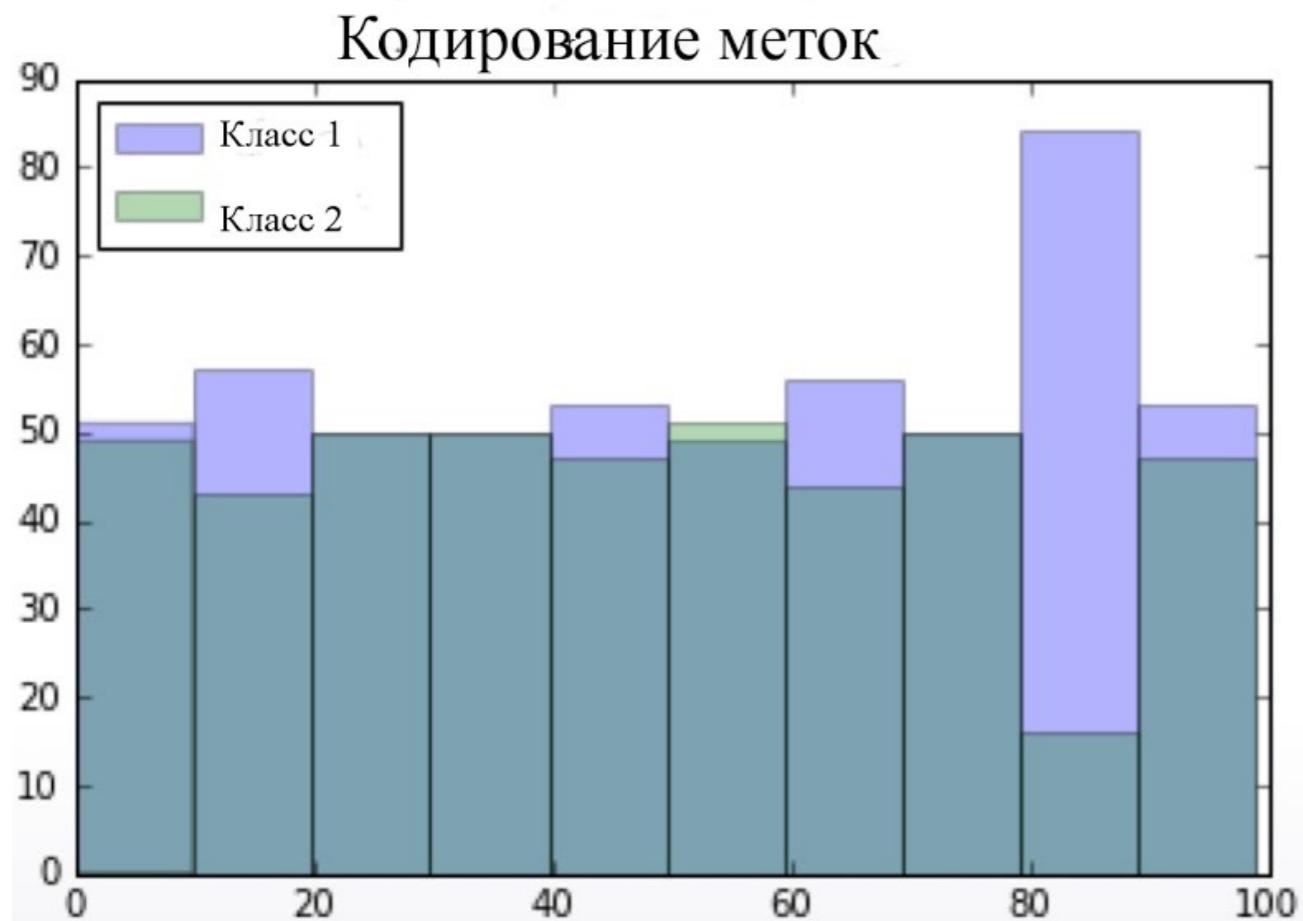
Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome



Mean encoding

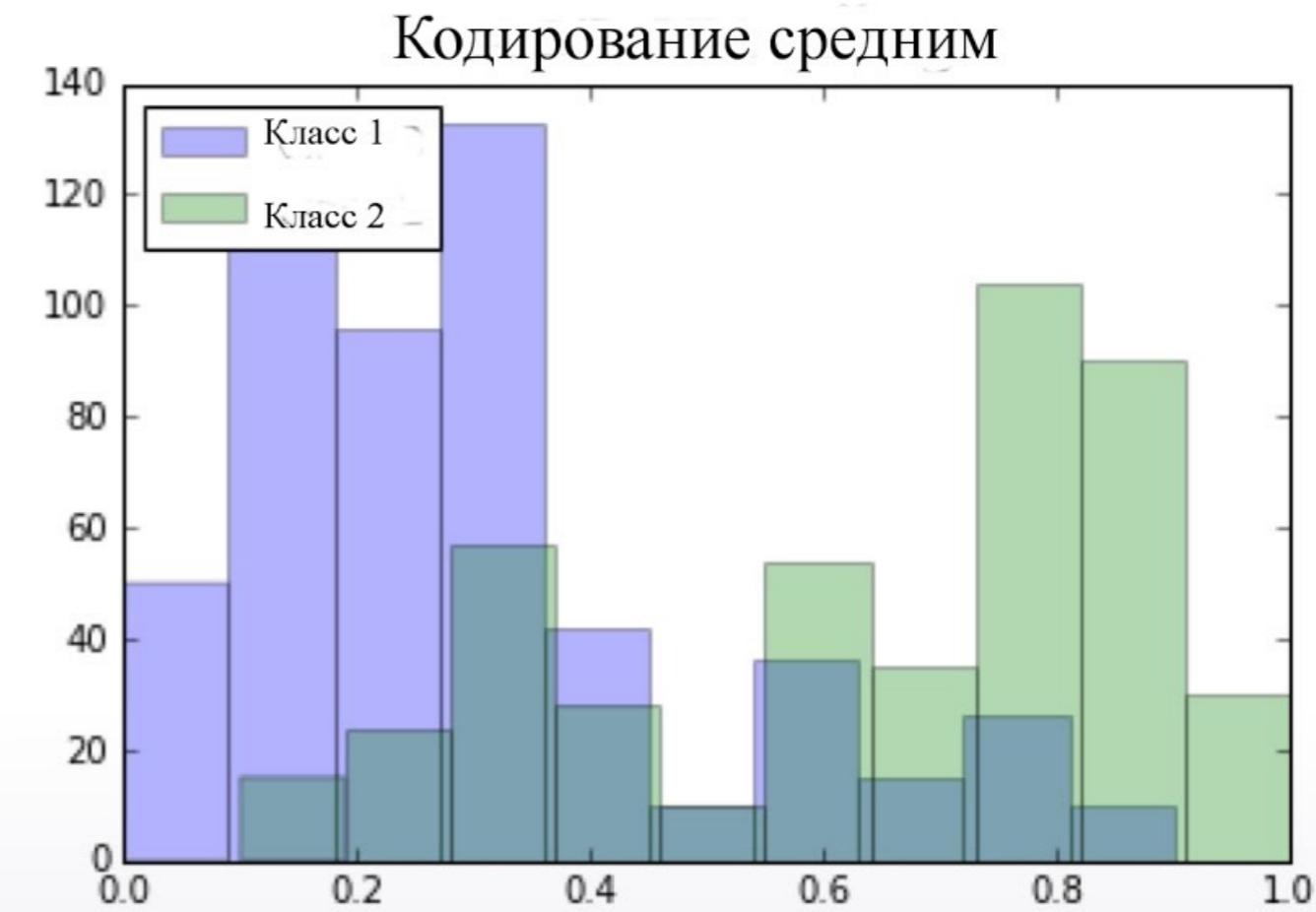
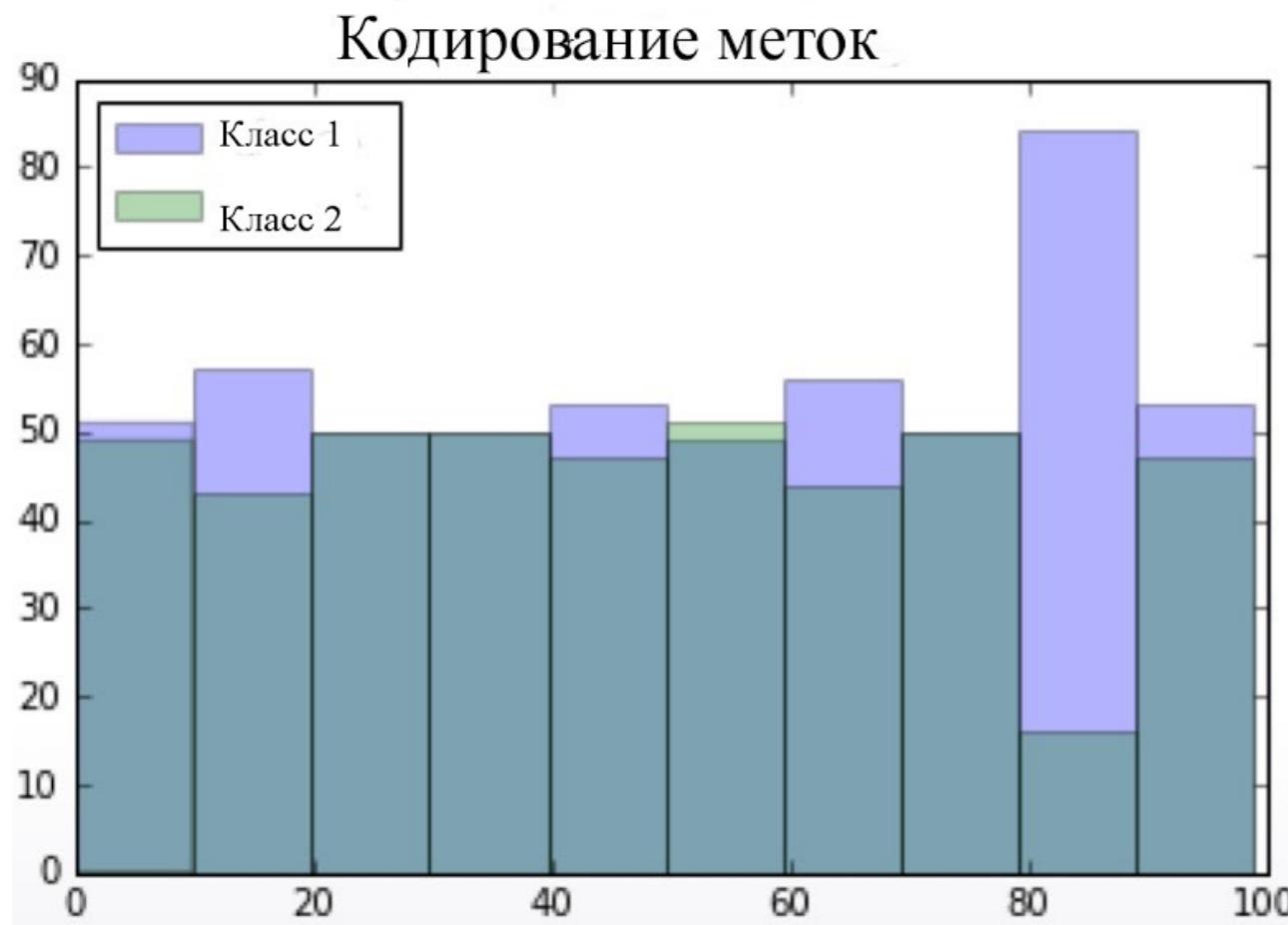


Mean encoding



Кто-то может предположить, какая проблема возникает?

Mean encoding

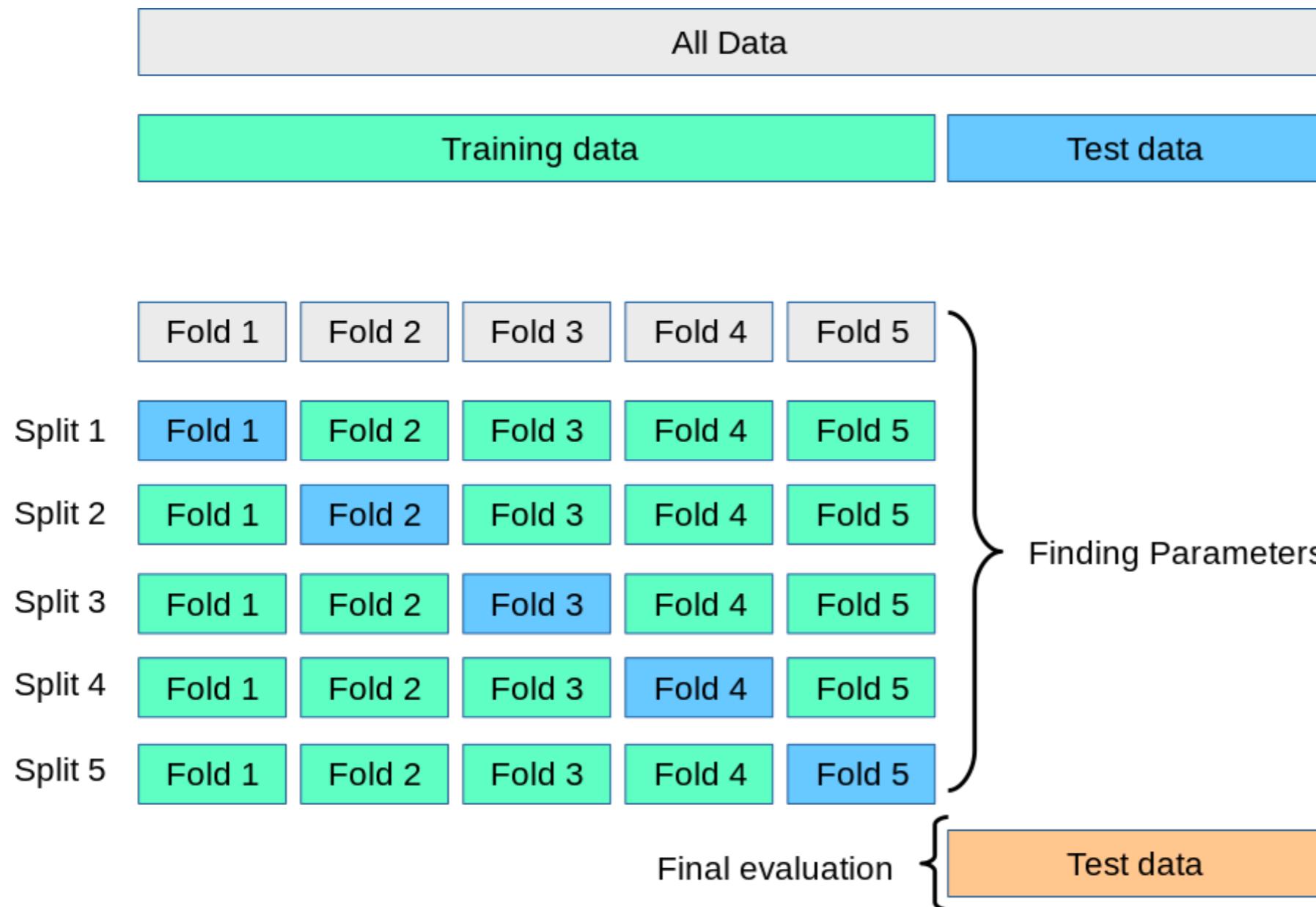


Кто-то может предположить, какая проблема возникает?

Редкие значения категорий могут в силу статистических причин соответствовать только 0 классу в обучающей выборке, являясь идеальными признаками с точки зрения модели, и только 1 в тестовой выборке, что приводит к неверным предсказаниям на teste.

Mean encoding

Как бороться?



Для подсчета mean encoding для k-го блока используется остальные блоки

Mean encoding

Сглаживание

$$\frac{\text{mean}(\text{target}, \text{category}) \cdot nrow + \text{globalmean} \cdot \alpha}{nrows \cdot \alpha}$$

Expanding mean

Пусть наши объекты каким-то образом упорядочены в обучающей выборке. Для объекта n будем считать среднее только по объектам от 1 до $n-1$. В большинстве случаев это упорядочение не имеет смысла, потому можно обучить несколько моделей на разных перестановках объектов в датасете и выдавать среднее предсказание.

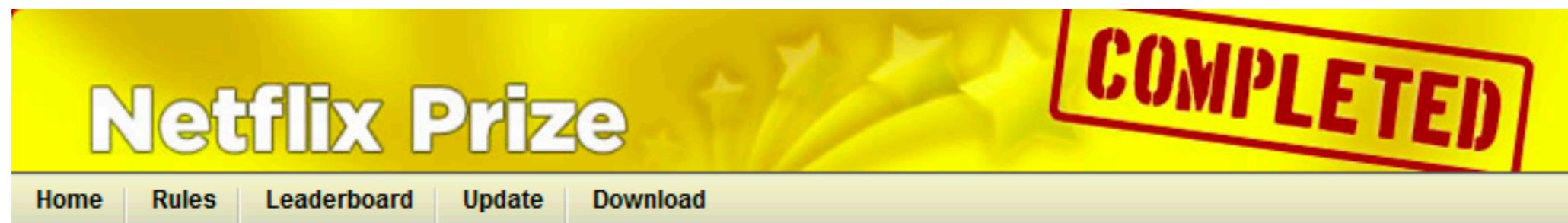
Этот способ использует catboost

Смесь экспертов

$$h(x) = \sum w(x) \cdot a(x)$$

**Вес конкретной модели в предсказании зависит от объекта,
для которого мы предсказываем**

Объединение моделей



Leaderboard

Showing Test Score. [Click here to show quiz score](#)

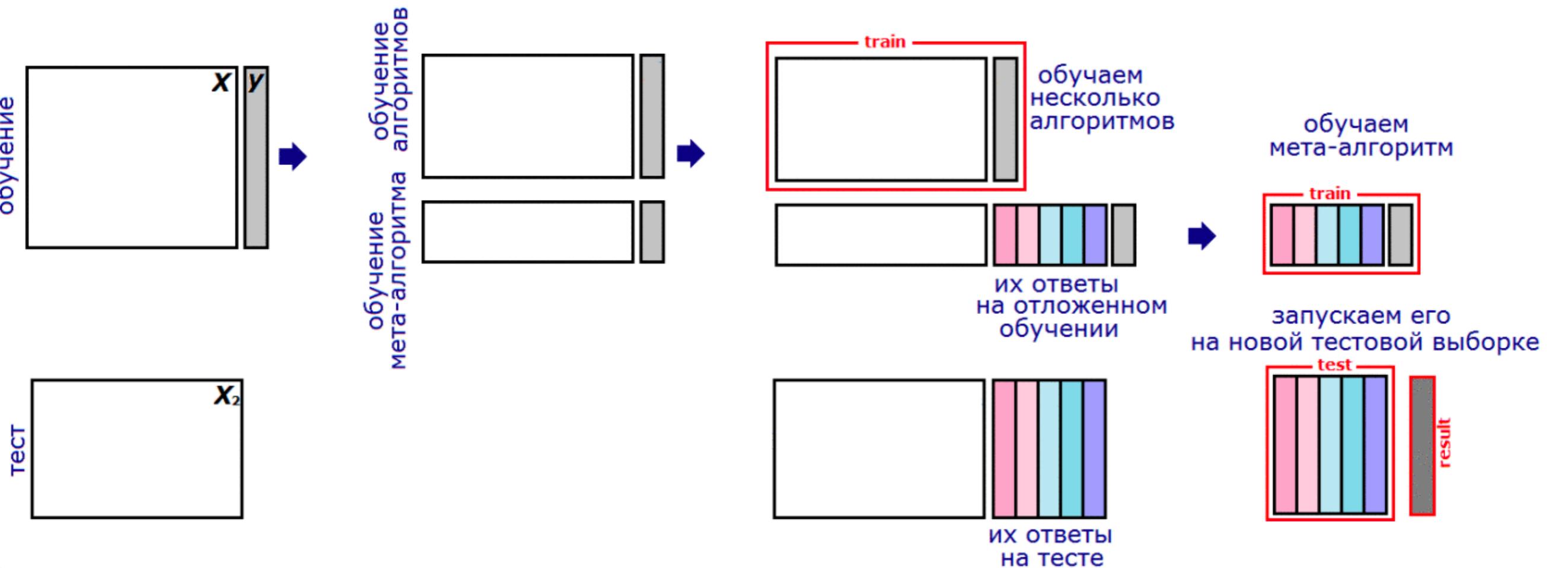
Display top leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43

Блендинг

1. Разбить тренировочную выборку на новую тренировочную и валидационную выборки
2. Обучить модели на тренировочном датасете
3. Сделать предсказания на валидационном и тестовом датасете
4. На валидационном датасете обучить модель предсказывать целевую переменную, используя предсказания построенных моделей
5. Использовать эту модель, чтобы сделать финальное предсказание

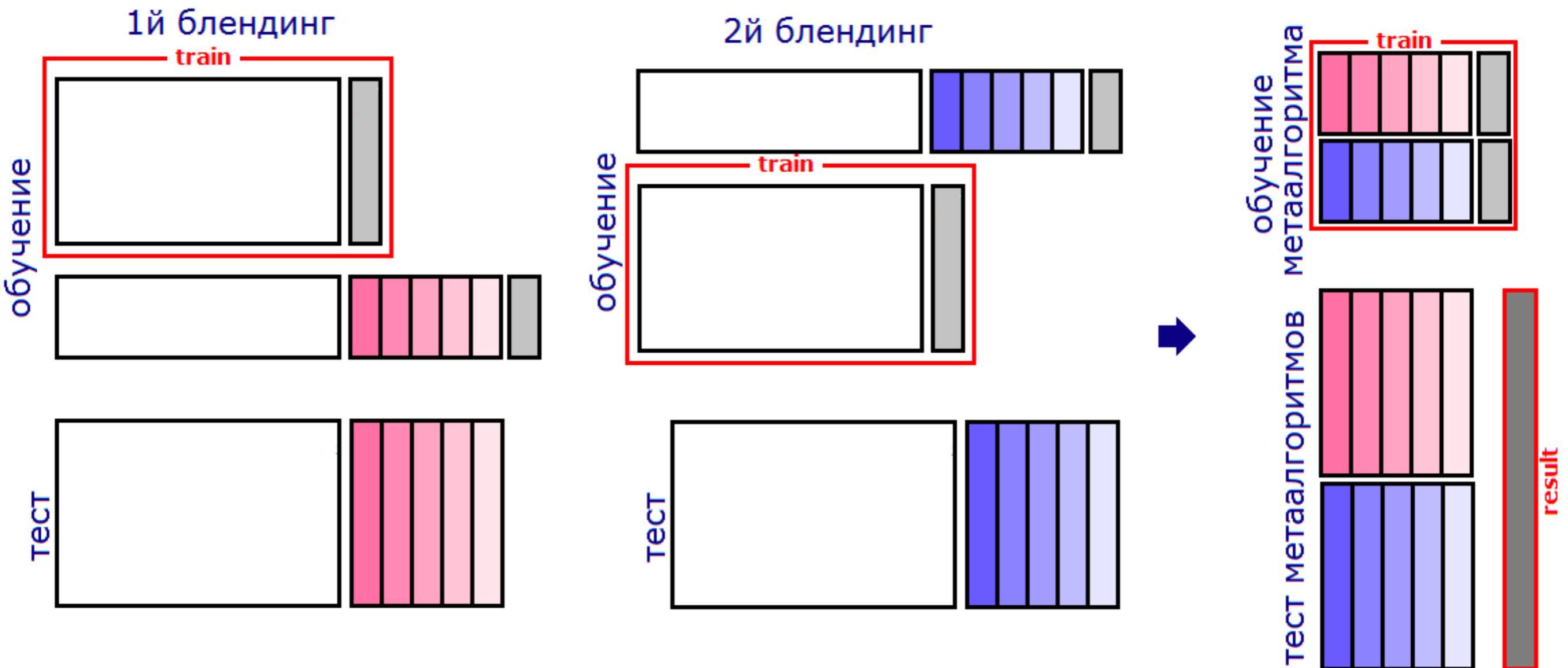
Блендинг 1



<https://dyakonov.org/2017/03/10/>

[c%D1%82%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3-stacking-%D0%B8-%D0%B1%D0%BB%D0%B5%D0%BD%D0%BD%D0%B4%D0%B8%D0%BD%D0%B3-blending/](https://dyakonov.org/2017/03/10/c%D1%82%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3-stacking-%D0%B8-%D0%B1%D0%BB%D0%B5%D0%BD%D0%BD%D0%B4%D0%B8%D0%BD%D0%B3-blending/)

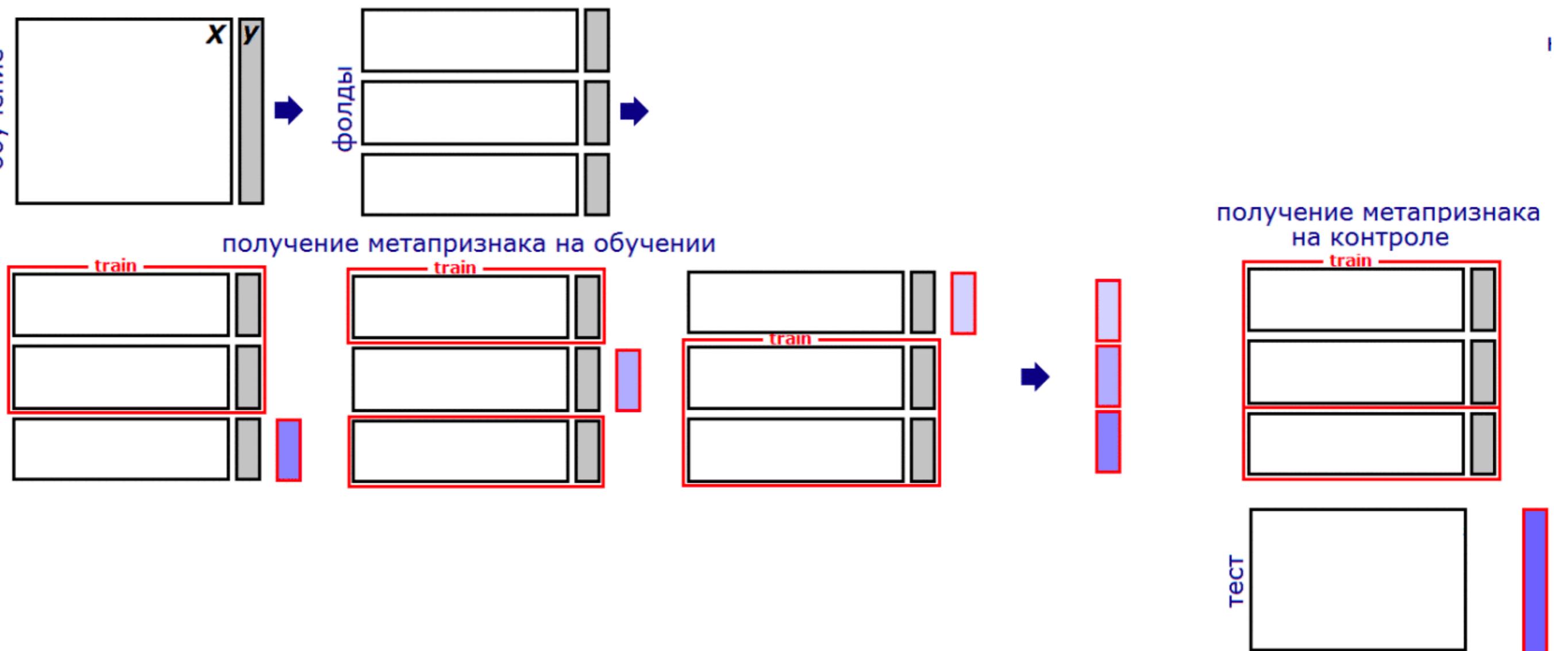
Блендинг 2



Стэкинг

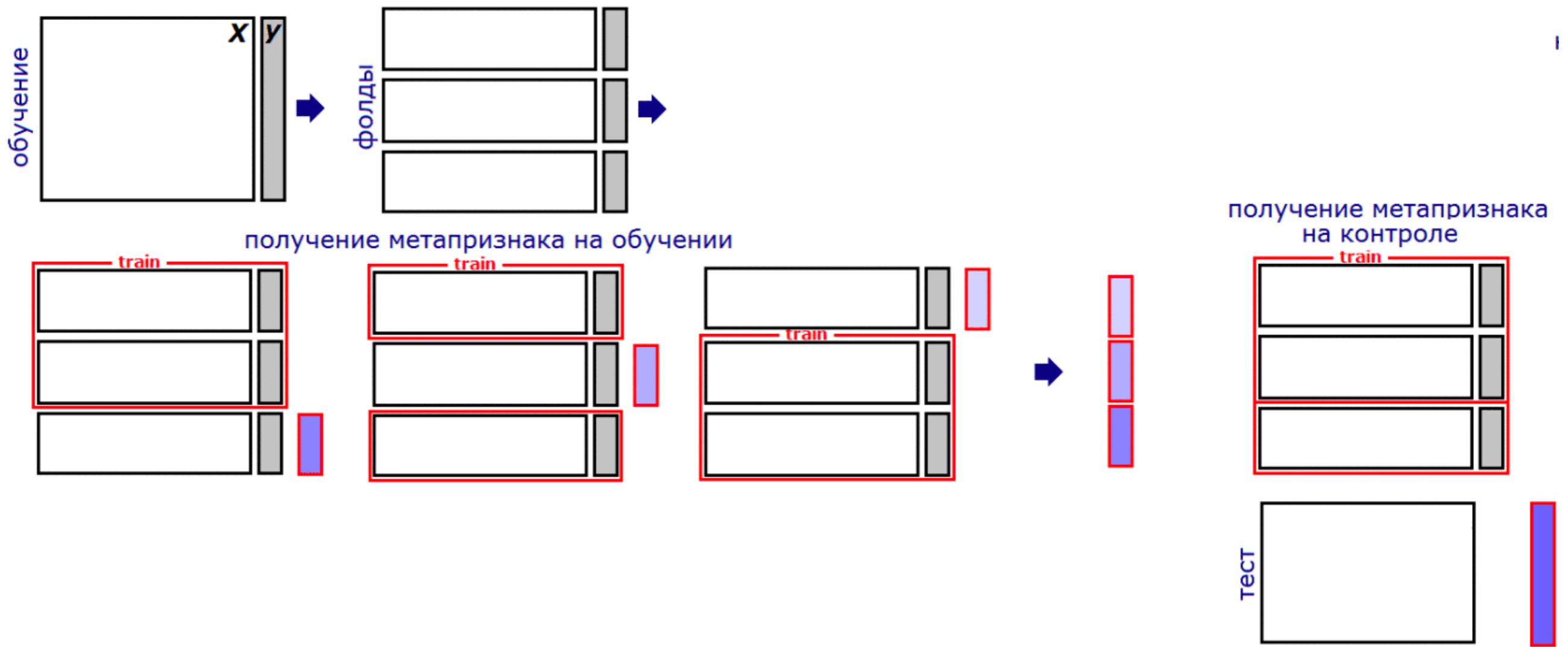
- 1) Разбить тренировочный датасет *train* на два - *train_a* и *train_b*;
- 2) Обучить базовые модели на *train_a* и сделать предсказания на *train_b*;
- 3) Обучить базовые модели на *train_b* и сделать предсказание на *train_a*;
- 4) Обучить базовые модели на всем *train* и сделать предсказания для *test*; Либо усреднить предсказания для *test* от моделей обученных на этапах 2-3;
- 5) Обучить модель первого уровня (называемую *stacker*) на вероятностях, получаемых от базовых моделей (возможно, добавляя исходные признаки).

Стэкинг



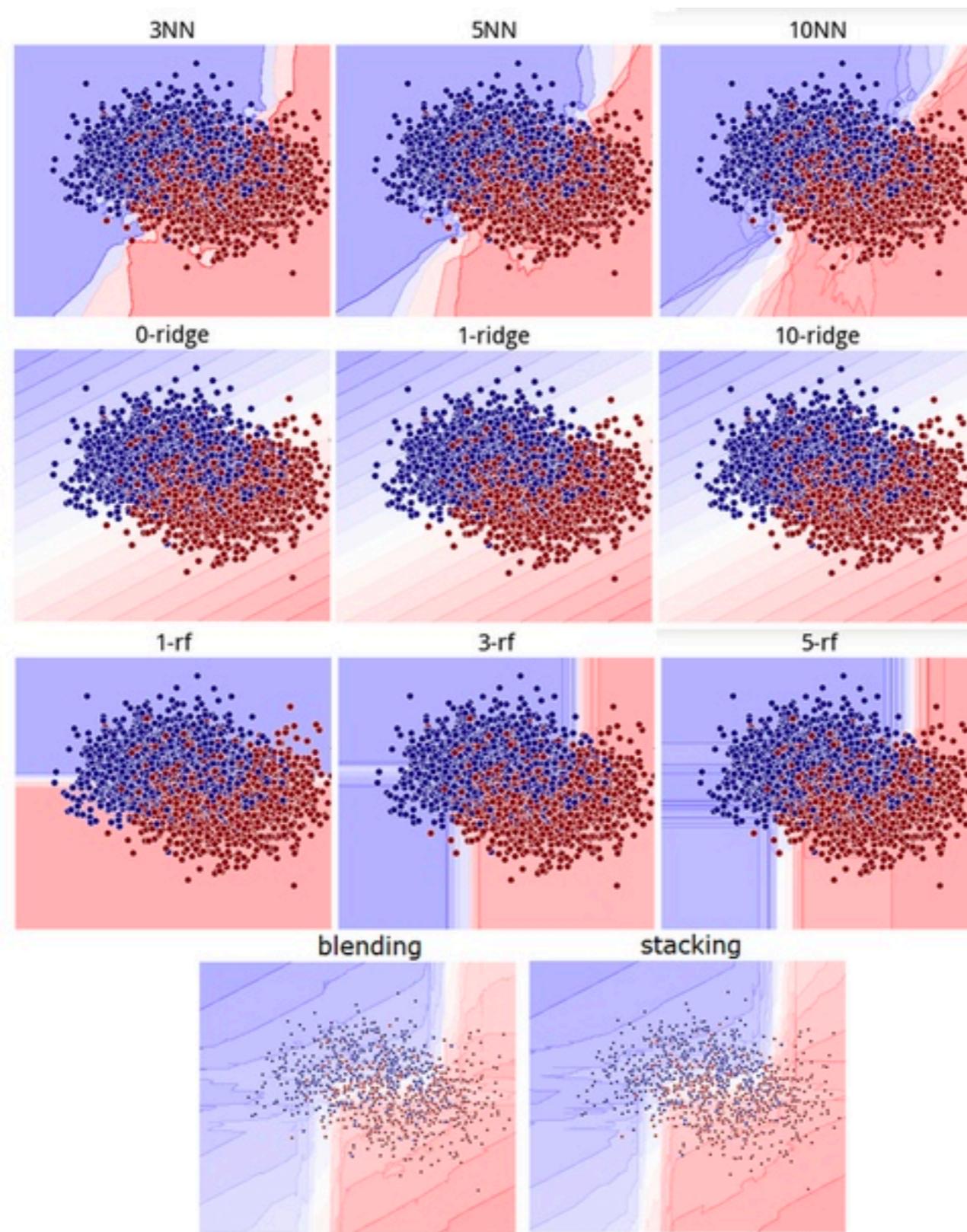
Какая проблема?

Стэкинг

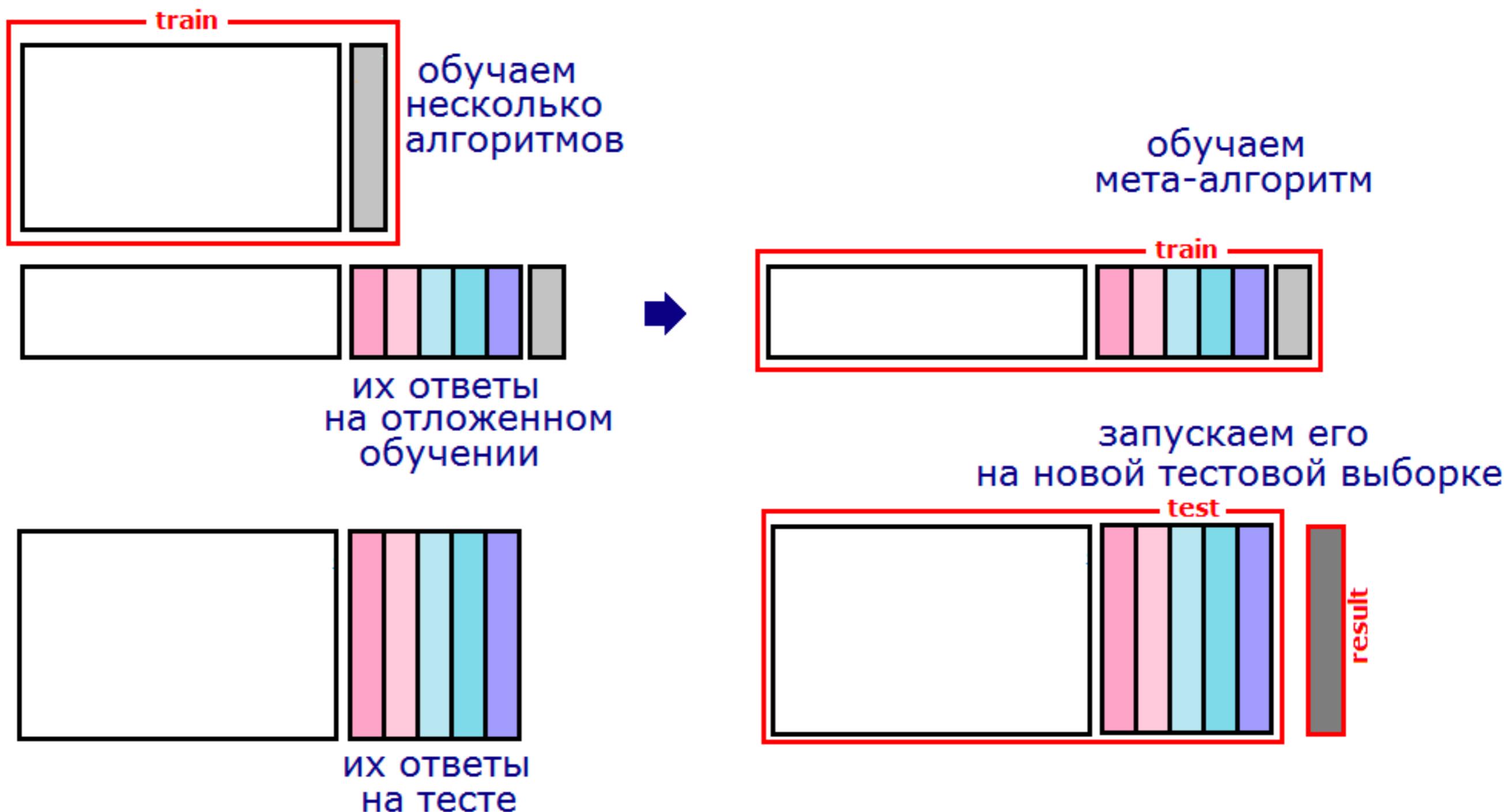


Какая проблема?

На самом деле, модели, на которых учился метаалгоритм и те, предсказания которых он объединяет для теста - разные



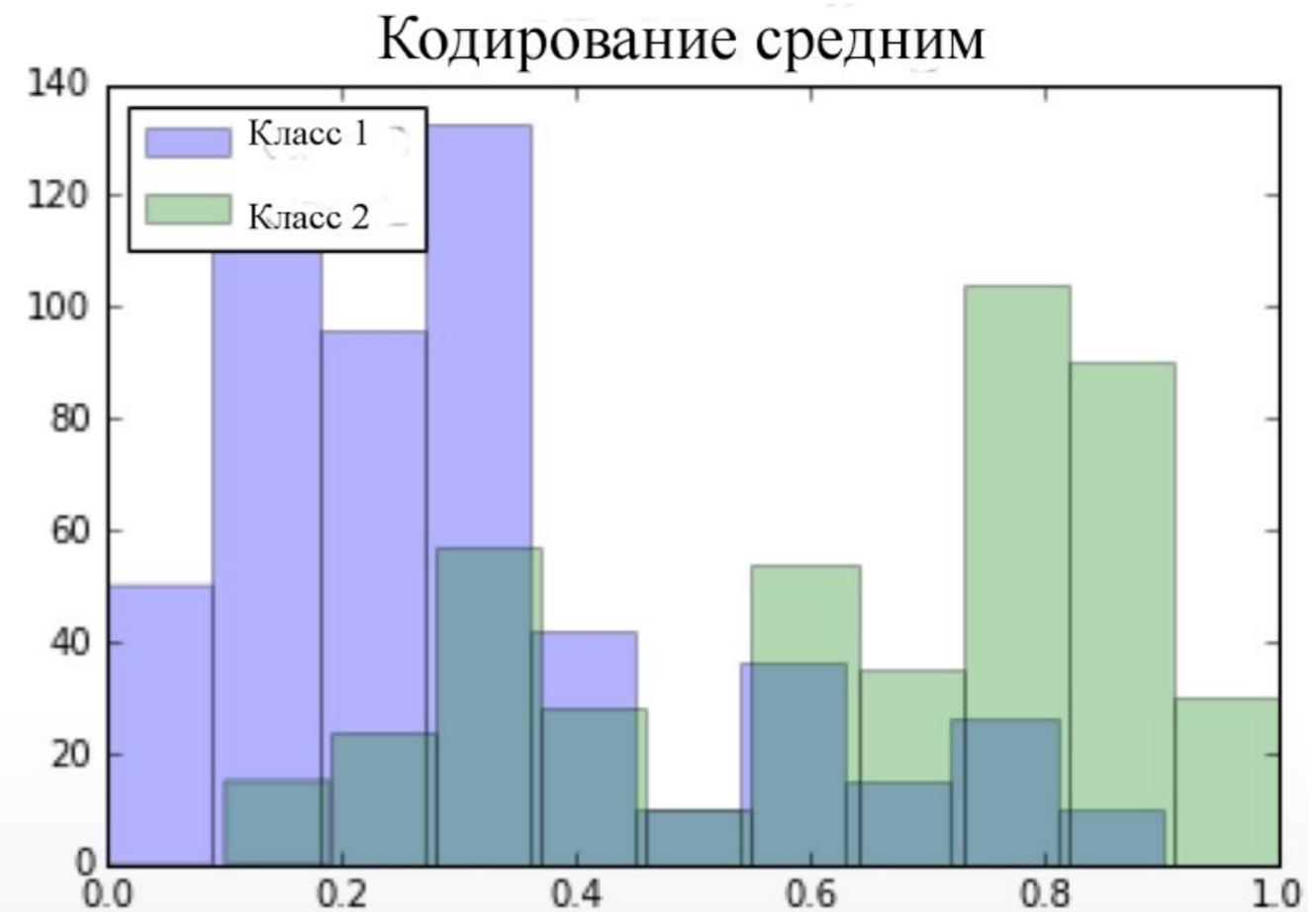
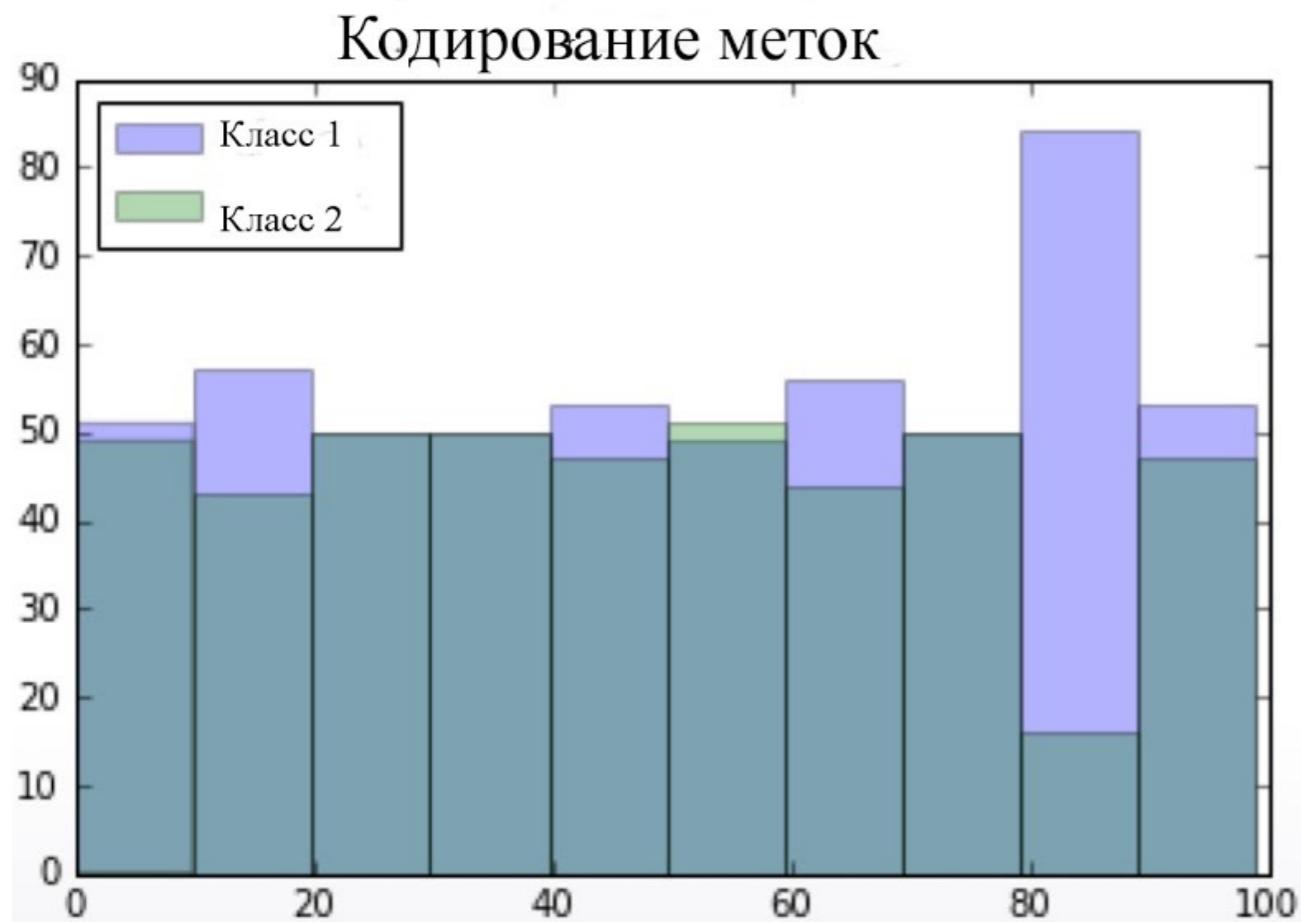
Использование признаков вместе с метапризнаками



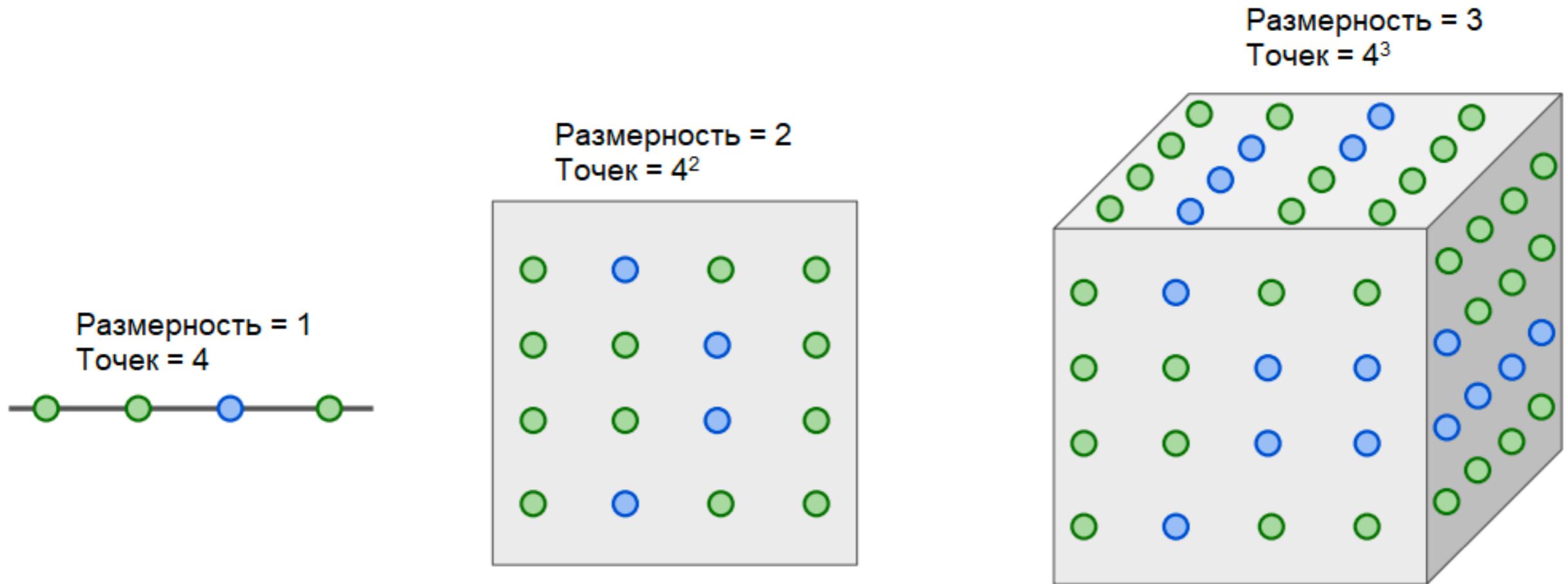
Деформация признаков

- Используем не просто предсказания моделей, а производные от них, например, попарные произведения

Mean encoding - разновидность стэкинга/блэндинга



Проклятье размерности

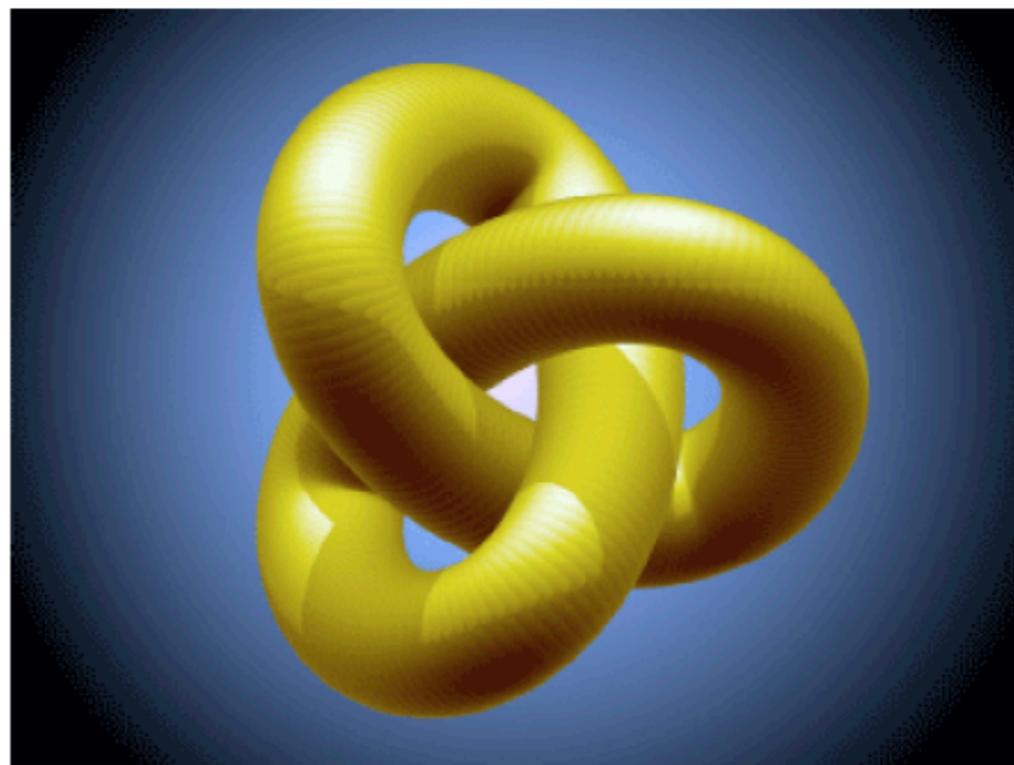


Чем больше размерность - тем больше точек нужно, чтобы покрыть все пространство -> тем больше точек нужно, чтобы быть уверенным в том, что наша модель правильно оценила распределение данных

Manifold assumption

Идея

Будем предполагать, что данные лежат на некотором многообразии меньшей размерности, включенном в Евклидово пространство

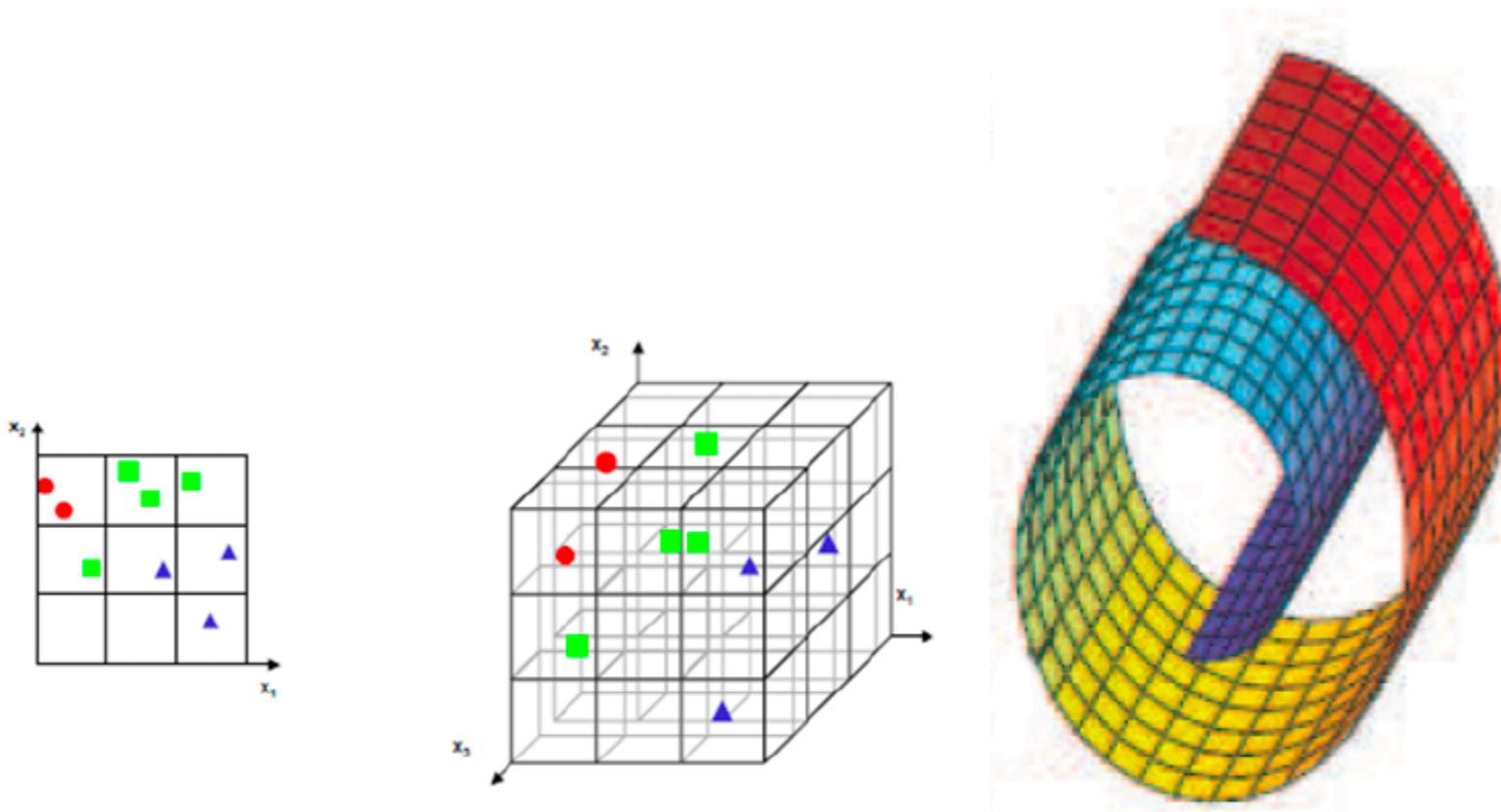


Для простоты можно представлять данные многообразия, как некоторые гладкие поверхности заключенные в Евклидово пространство

Manifold assumption

В чем смысл?

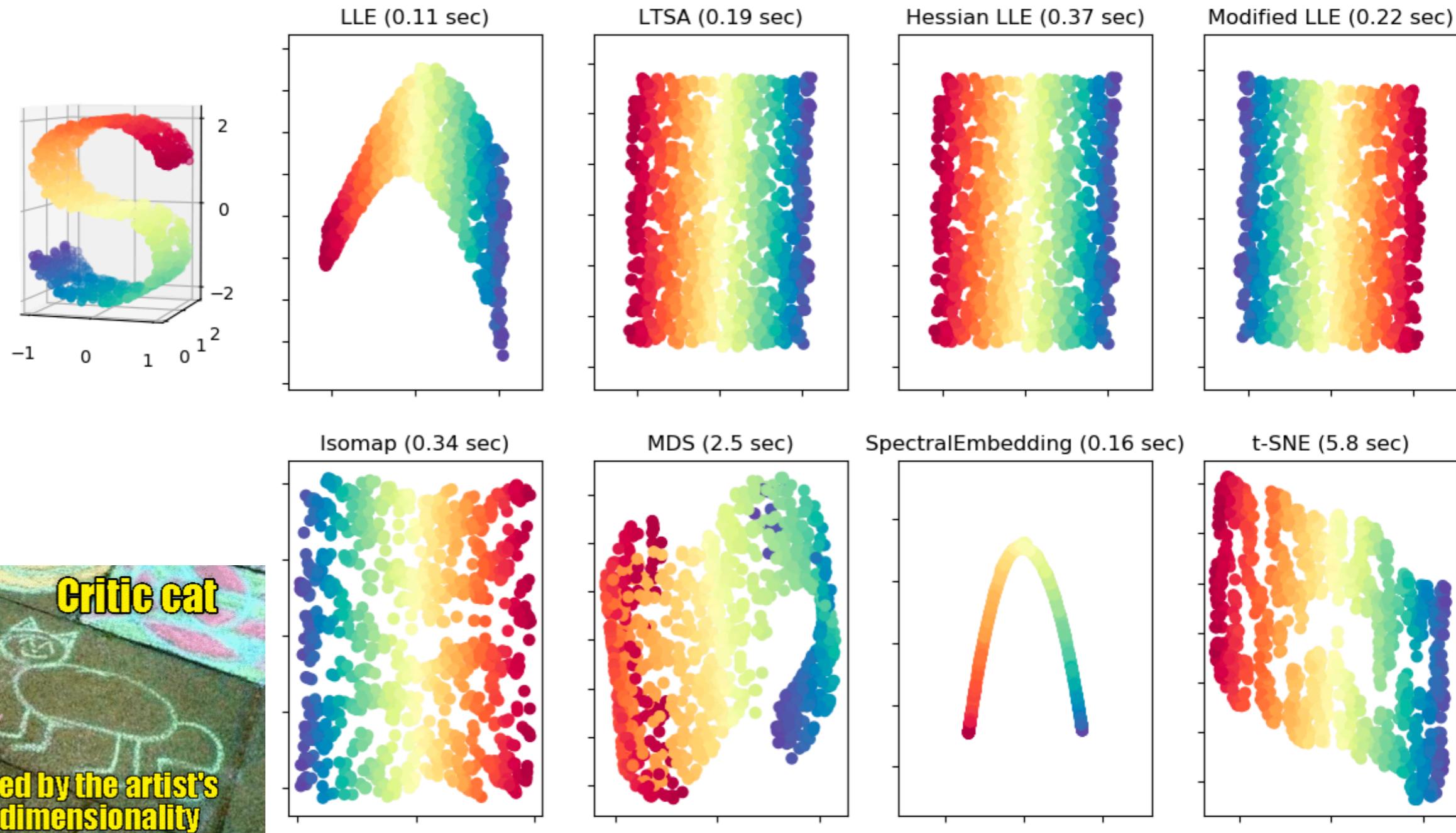
Хотим добиться “гладкости” решения в соответствии с внутренней структурой данных



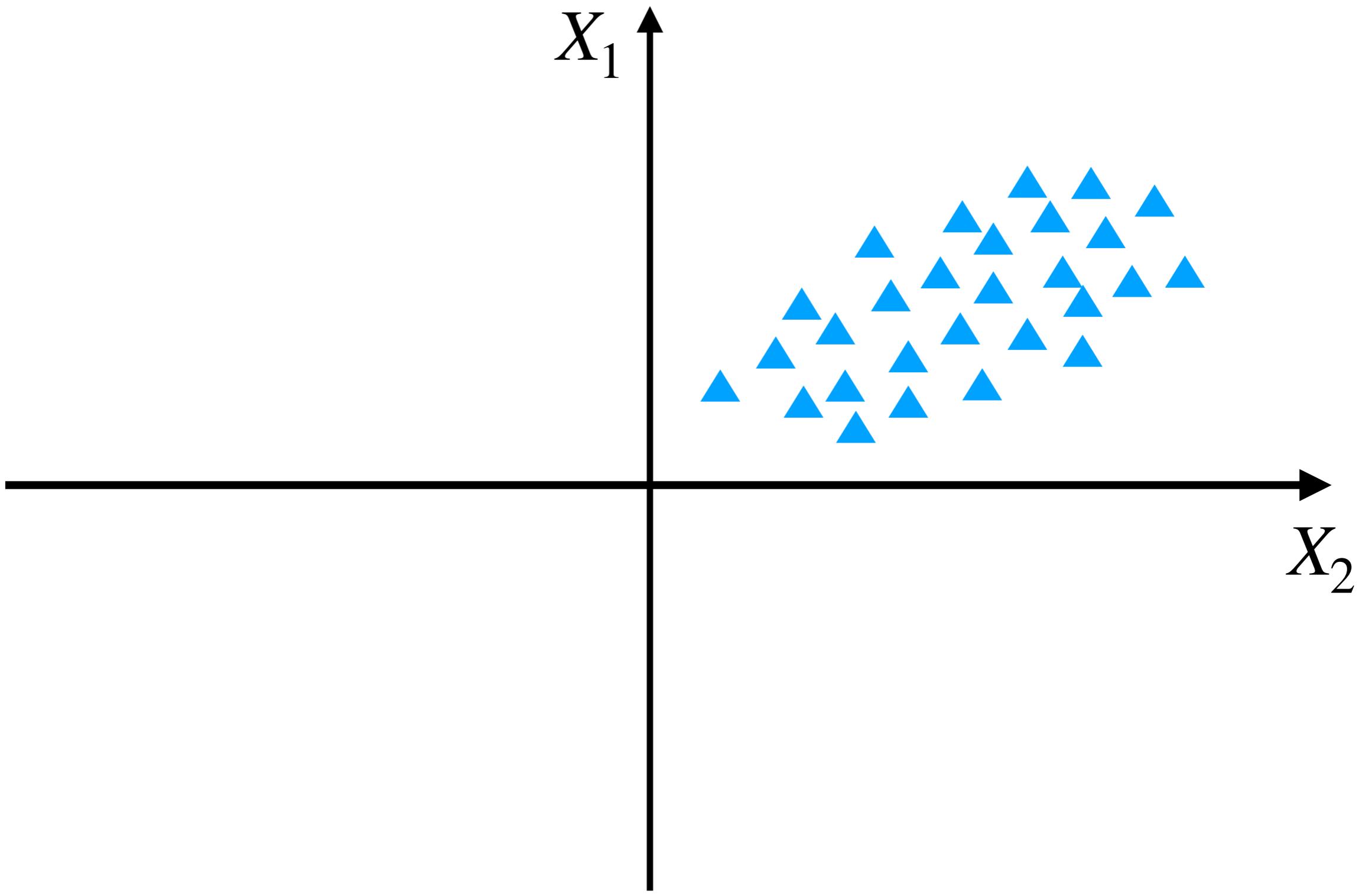
- ▶ Расстояния в многомерных пространствах неинформативны (проклятие размерности)
- ▶ Работаем с расстояниями в пространстве меньшей размерности, отражающим внутреннюю структуру данных

Снижение размерности

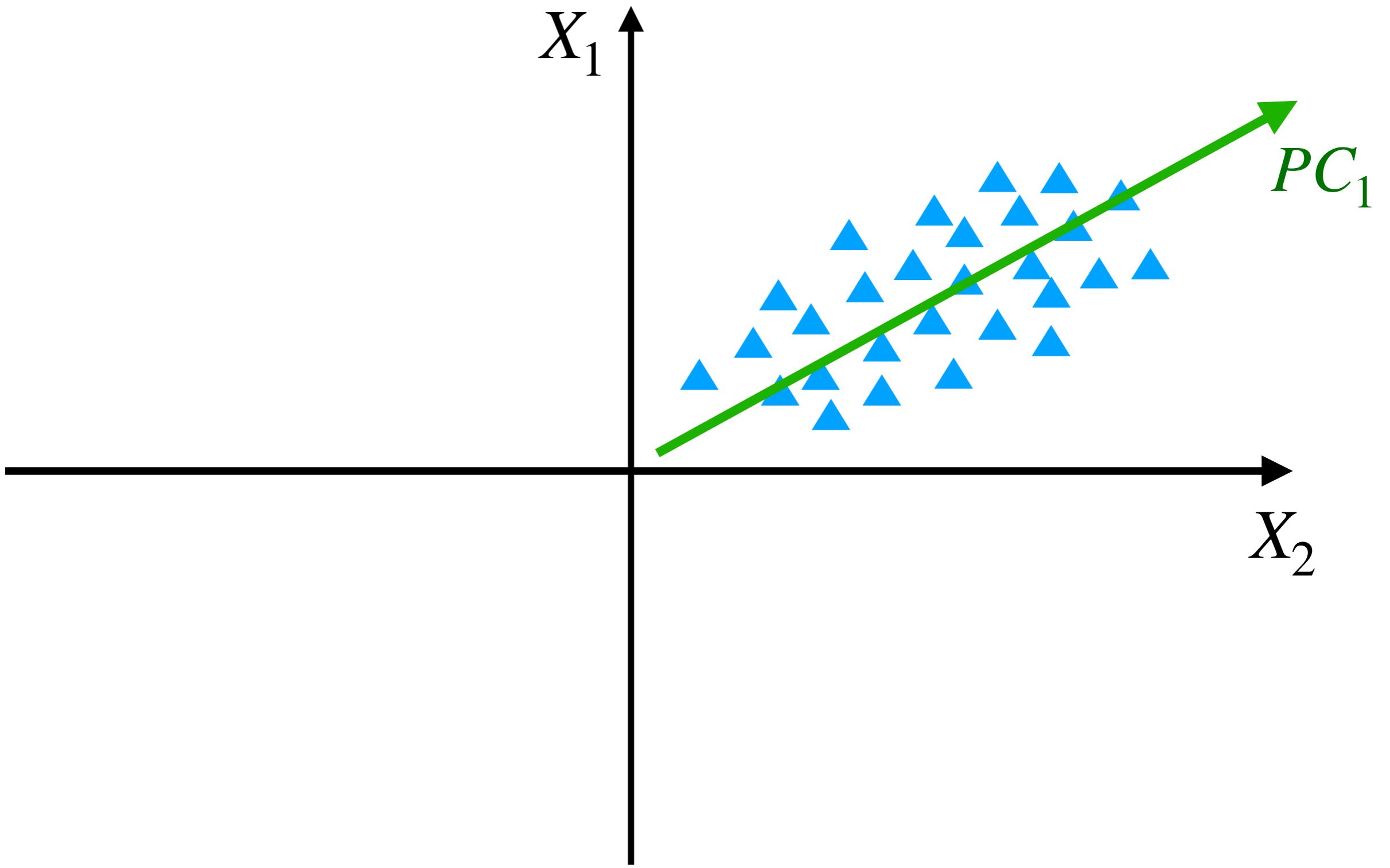
Manifold Learning with 1000 points, 10 neighbors



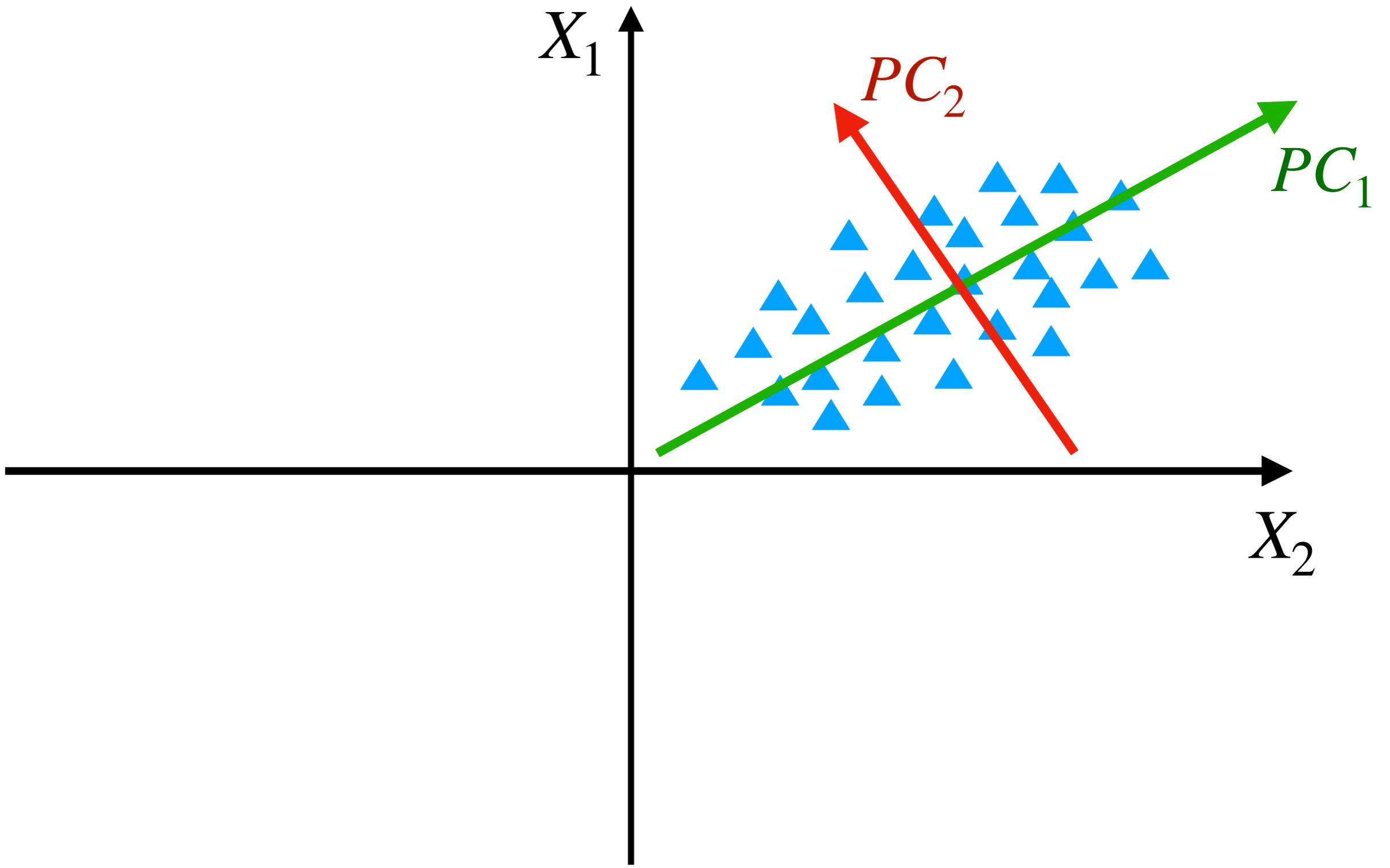
PCA



PCA



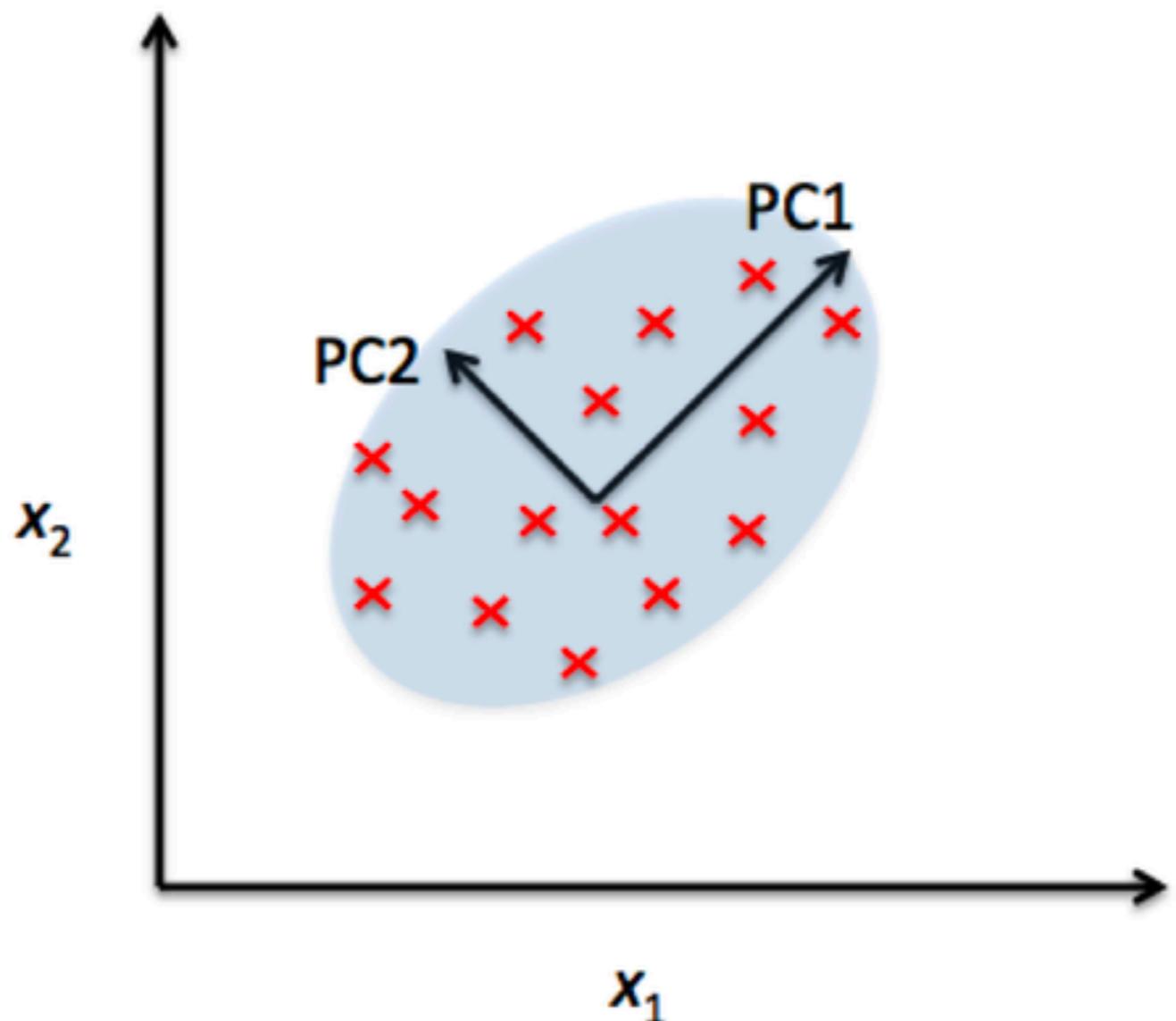
PCA



Говорят ли о чем-то расстояния?

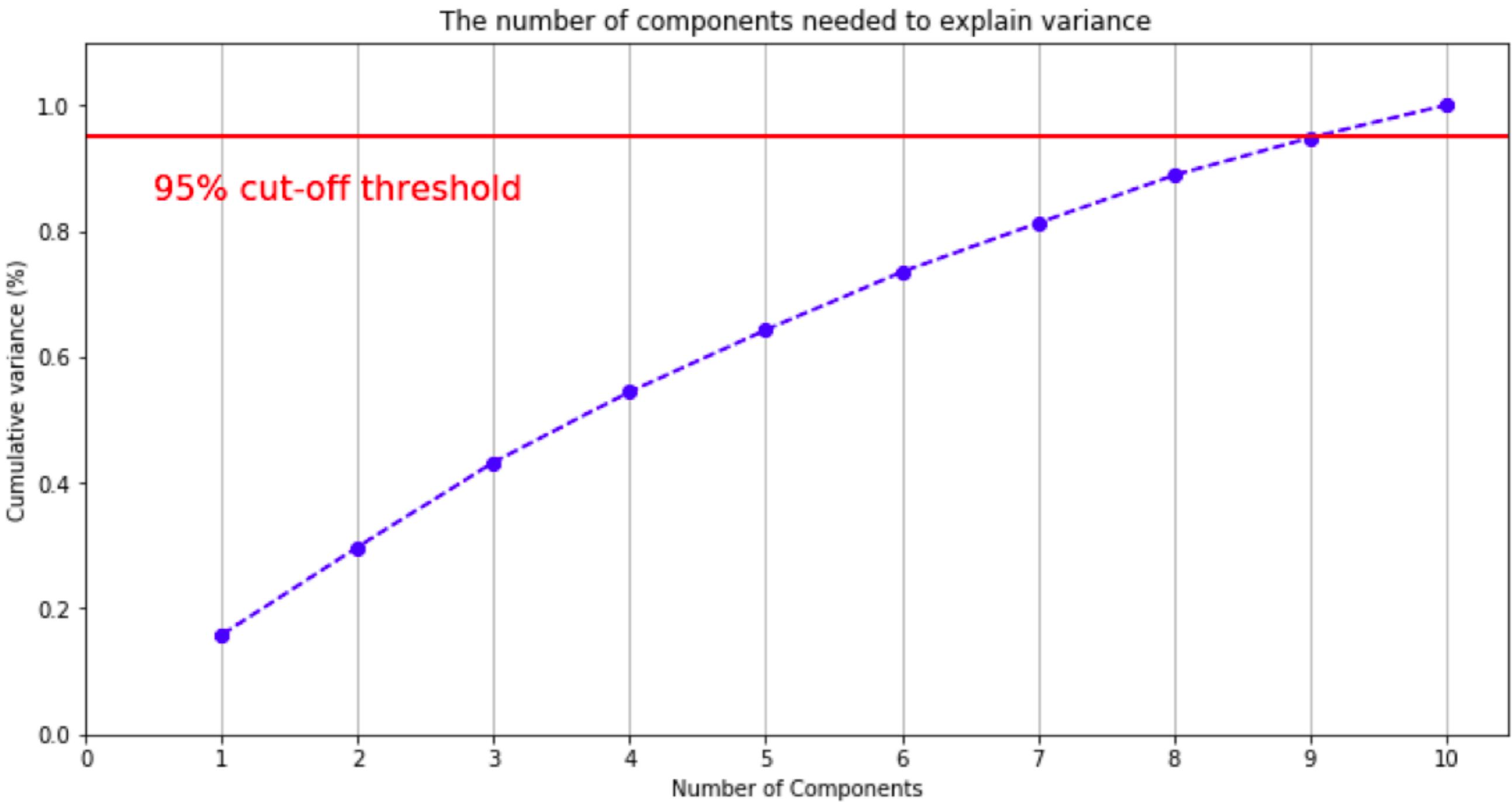
Расстояния между точками
сохраняют свой смысл.

Близкие точки были
близки в исходном
пространстве, далекие -
далеки



Как подбирать число компонент?

Сколько дисперсии объясняет?

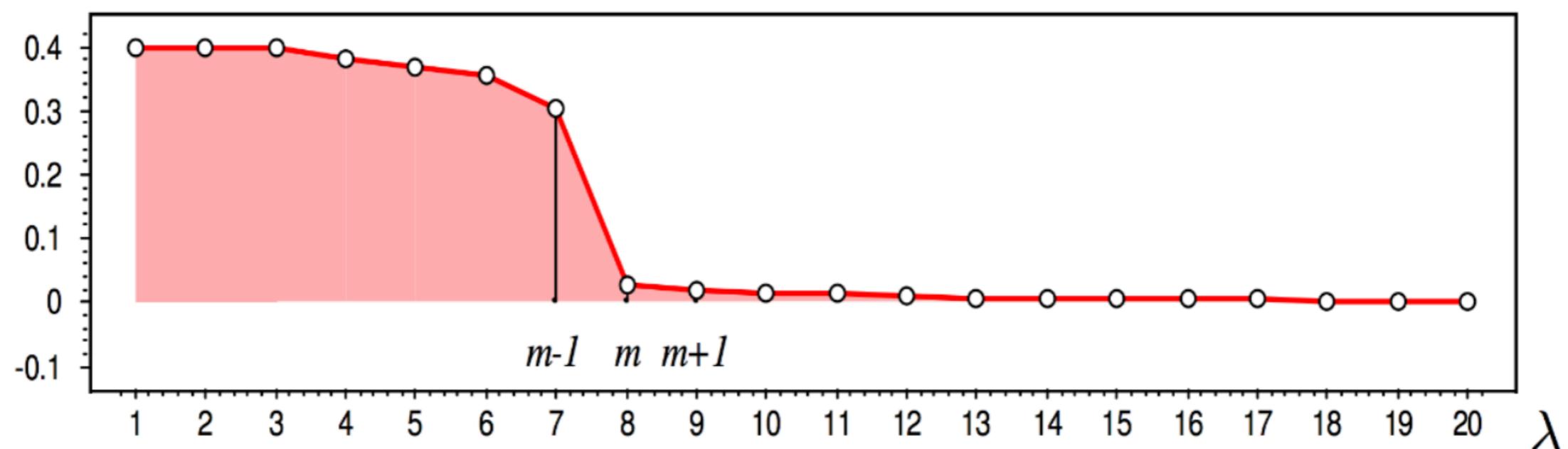


По формуле...

Каждой компоненте соответствует собственное число. При этом первой компоненте соответствует максимальное собственное число. И тд

$$E_m = \frac{\lambda_{m+1} + \dots + \lambda_n}{\lambda_1 + \dots + \lambda_n}$$

Критерий «крутого склона»: находим m : $E_{m-1} \gg E_m$:

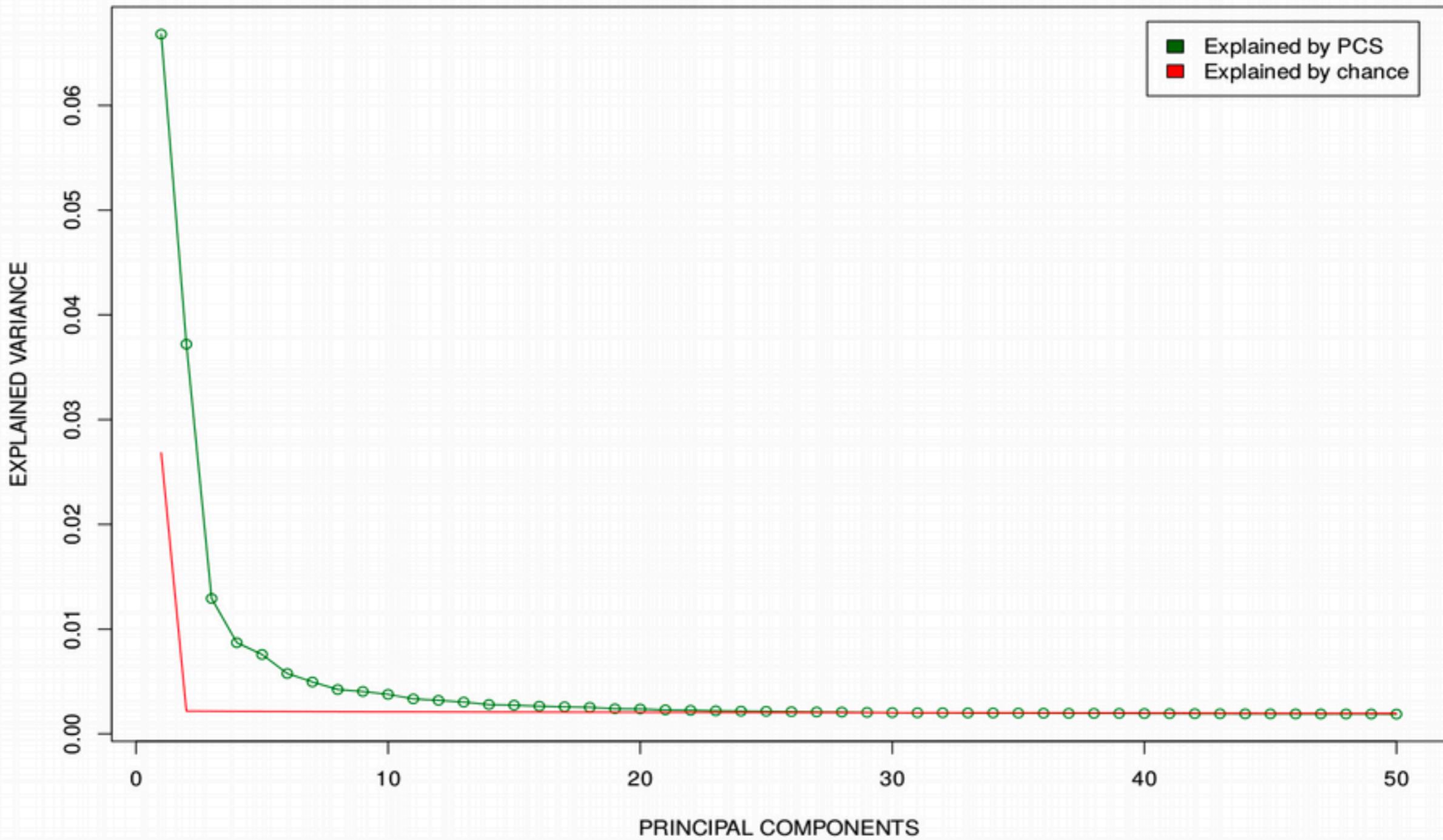


Permutation

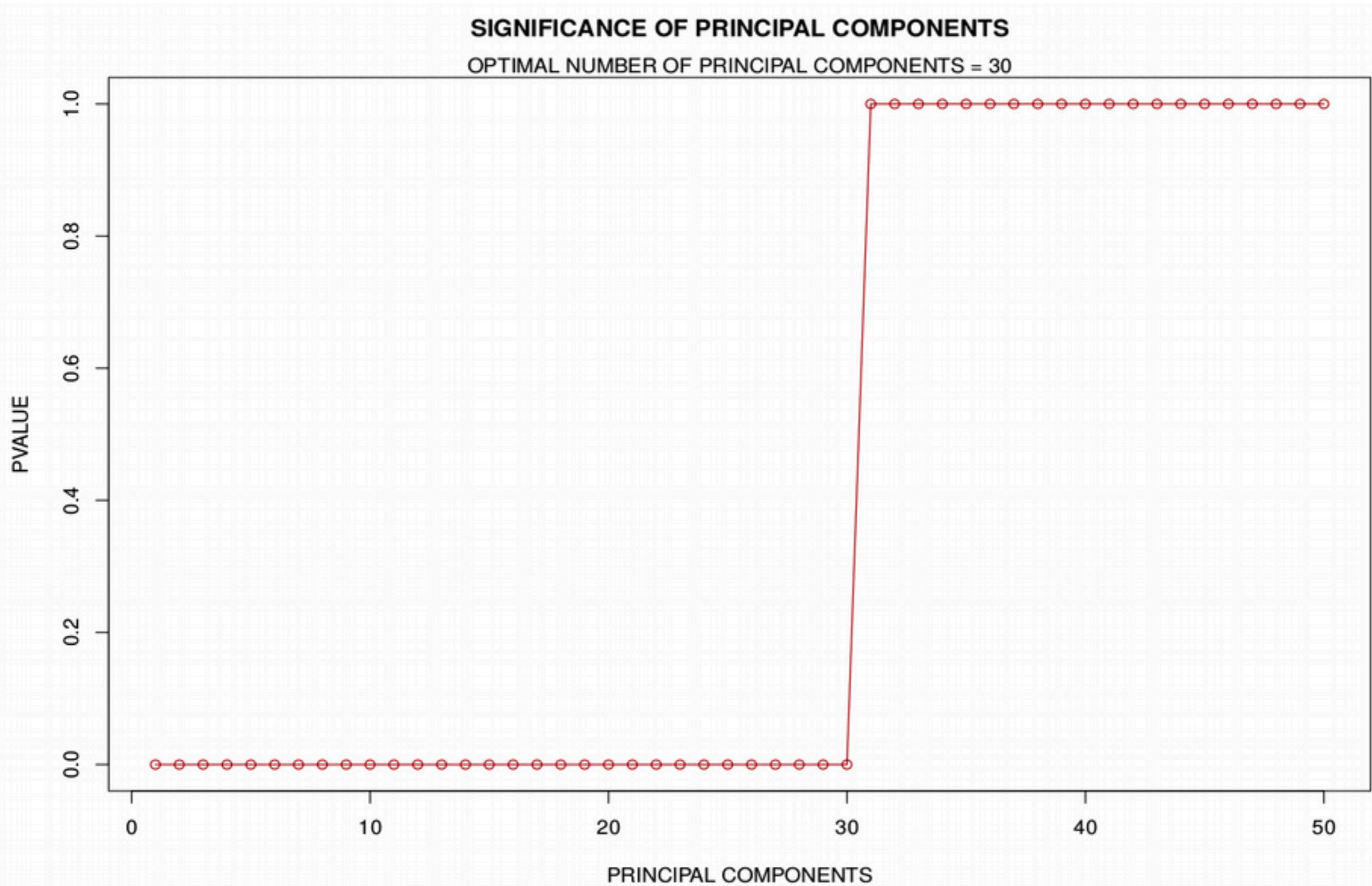
- 1) Перемешиваем значения каждого признака.
- 2) Получаем матрицу признаков, которая не содержит никакой информации.
- 3) Делаем PCA
- 4) Любая *explained variance* - просто из-за природы данных
- 5) Делаем так 100-1000 раз
- 6) Пусть на реальных данных k -я компонента объясняет $n\%$ дисперсии.
- 7) Смотрим на распределение доли дисперсии, объясняемой k -компонентой для случайных данных (полученных перемешиванием).
- 8) Можем сравнить и принять решение, объясняет ли k -я компонента что-то реальное, или просто шум

Permutation

VARIANCE EXPLAINED BY PRINCIPAL COMPONENTS



Permutation

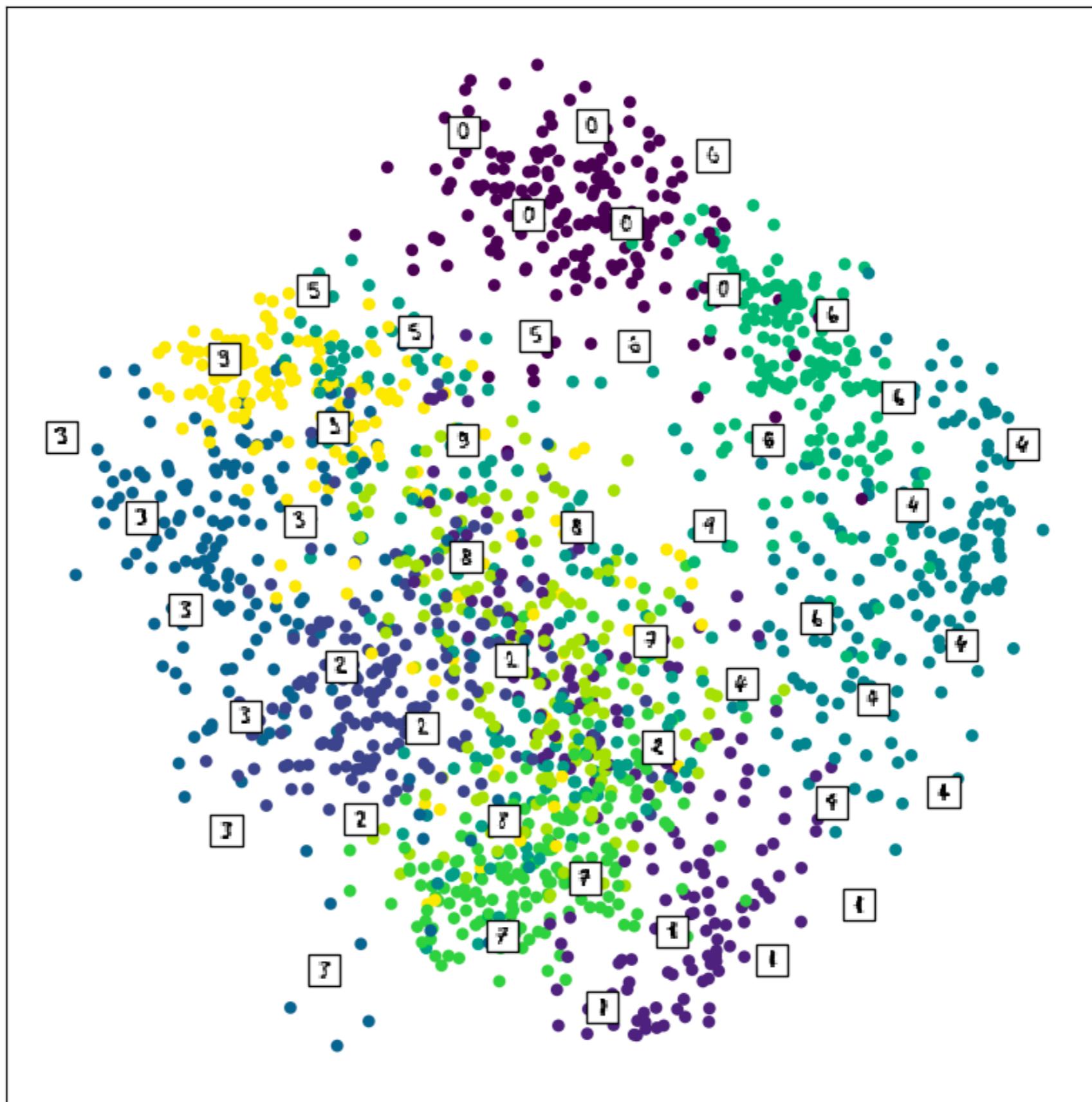


Для чего используют?

1. Визуализация данных
2. Анализ данных (можно посмотреть, какие изначальные признаки вносят вклад в важные компоненты)
3. Очищение данных от шума
4. В частности - очищение данных от шума, чтобы далее передать другому методу
5. Снизить размерность, чтобы ускорить работу каких-то методов, медленно работающих для большого числа признаков

MNIST

PCA



Kernel PCA

Используем kernel-trick, чтобы делать PCA в нелинейных пространствах

