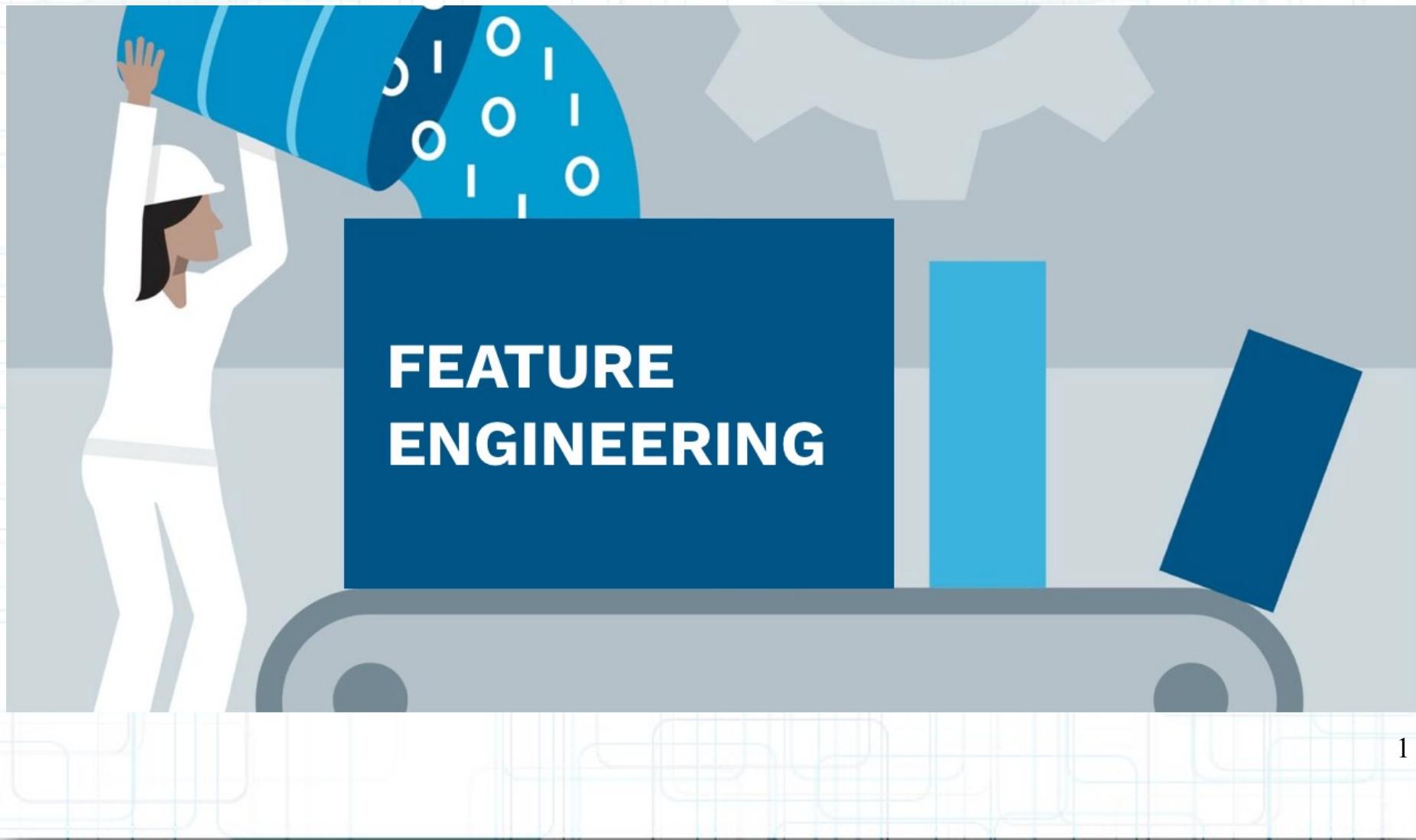


Машинное обучение

Feature engineering



Содержание лекции

- Теория и практические приемы извлечения признаков из данных разного типа (признаки для пар объектов, для групп (покупок), для кусочков временных рядов, для текстов, изображений)
- Трансформации признаков:
 - one-hot
 - дискретизация
 - выделение главных компонент (PCA, PLS)
 - другие embeddings
- Важность признаков (в т.ч. взаимная информация)

Дата и время

Дата и время

- День недели
- Месяц
- Время года
- Время суток
- Рабочие/нерабочие дни, праздники
- Рабочие/нерабочие часы
- Кластеризация
- Значения нескольких синусов с разными частотами (используется для позиционного кодирования в трансформерах)

Признаки пар разнородных объектов

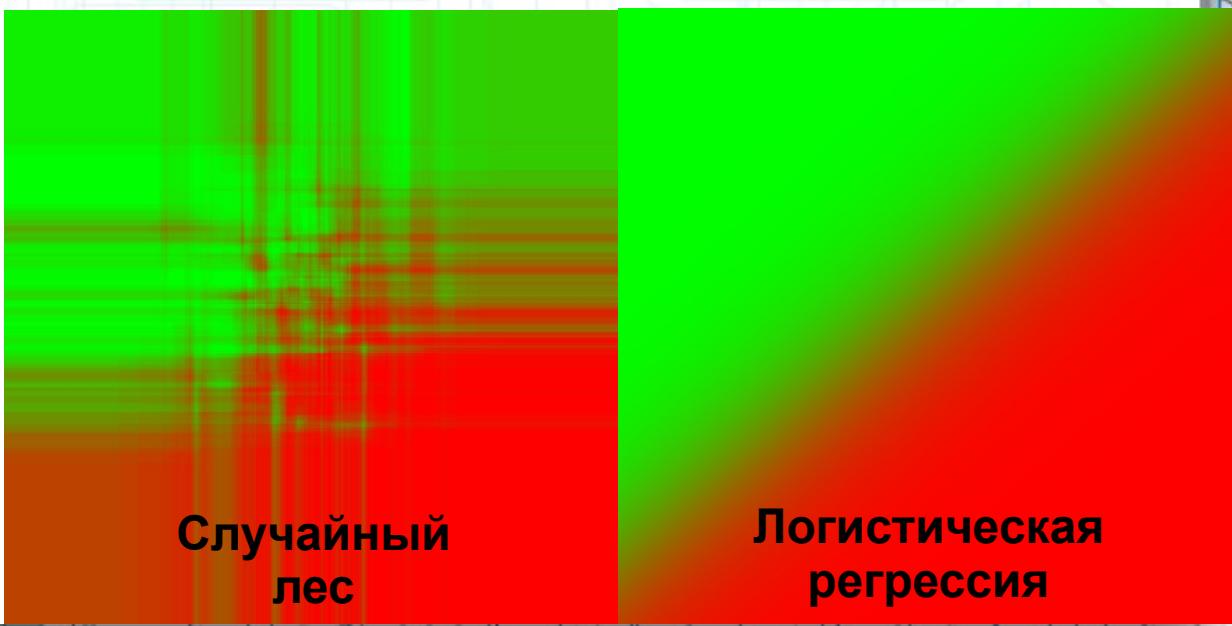
- Распространенный случай: заданы две таблицы, описывающие компоненты пары, с отношением "многие ко многим" и таблица с данными пар. Примеры:
 - (студент, дисциплина)
 - (покупатель, товар)
 - (пользователь, поисковой запрос)
 - (хеш-тег, документ)
 - (турист, гостиница)

Признаки пар разнородных объектов

- Признаки компонент пары: группировка значений из одной таблицы по ключу другой и применение какой-нибудь операции (+, min, max, mean, exists,...)
- Пример: средний балл студента, средний балл дисциплины (сложность), наличие задолжников у дисциплины, ...
- Совместные признаки компонент пары: оценка студента по предмету, отличие оценки от средней для данной дисциплины (или для данного студента)

Признаки пар однородных объектов

- Пример: пара товаров (какой понравится покупателю больше?), пара кандидатов (какой лучше?), пара студентов (кто сдаст экзамен лучше?)
- Методы, основанные на пороговых предикатах (градиентный бустинг деревьев), плохо воспринимают признаки компонент пары. Им нужны разности или отношения (вдобавок!).
- Пример:
классификация датасета с верным классом: $[x_2 > x_1]$



Признаки множеств

- Метод сумки (Bag of ...): описываем множество вектором, считая для каждого элемента количество его вхождений
- Примеры:
 - тексты – Bag of words
 - события – Bag of events
 - для покупателей – Bag of purchases

Bag of words

“This is how you get ants.”

tokenizer

```
['this', 'is', 'how', 'you', 'get', 'ants']
```

Build a vocabulary over all documents

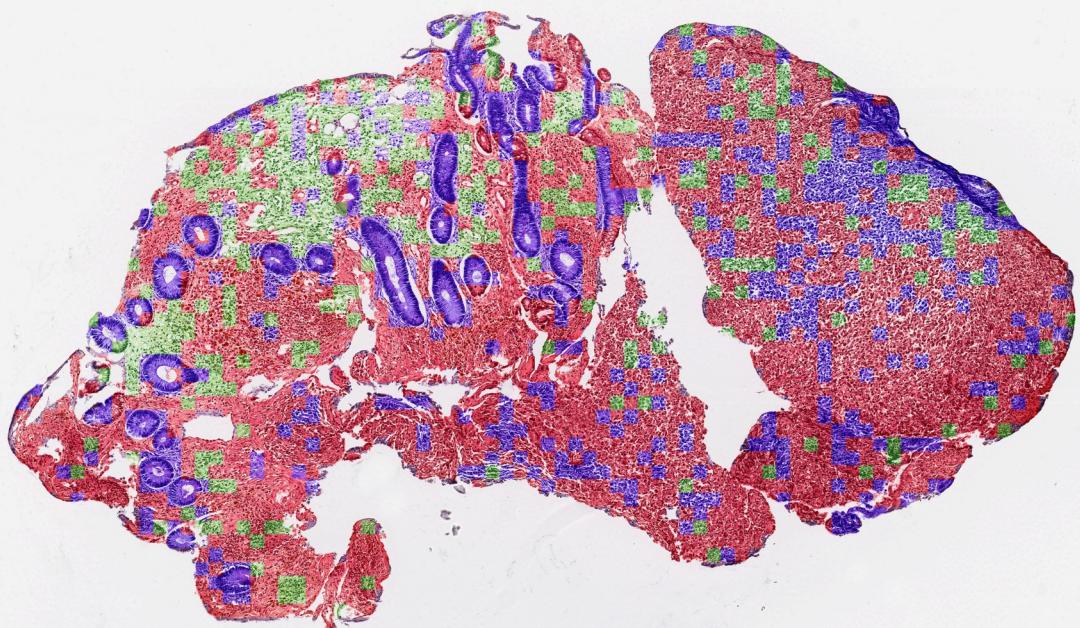
```
['aardvak', 'amsterdam', 'ants', ... 'you', 'your', 'zyxst']
```

Sparse matrix encoding

aardvak	ants	get	you	zyxst
[0, ..., 0, 1, 0, ... , 0, 1 , 0,, 0, 1, 0, , 0]				

Признаки множеств

- Распространенная проблема: большое количество разнообразных элементов, иногда даже все уникальны
- Пример: задача диагностики типа рака (основана на пропорциях патологий). Каждый кусочек изображения уникален. Выход: кластеризация элементов. После этого применяем BagOfWords (слово – это номер кластера)



TF-IDF – выделение специфичных элементов

- Элементы, встречающиеся во всех множествах не так важны, как те, которые характеризуют лишь некоторые:

$$\text{TF-IDF} = \text{TF} * \text{IDF}$$

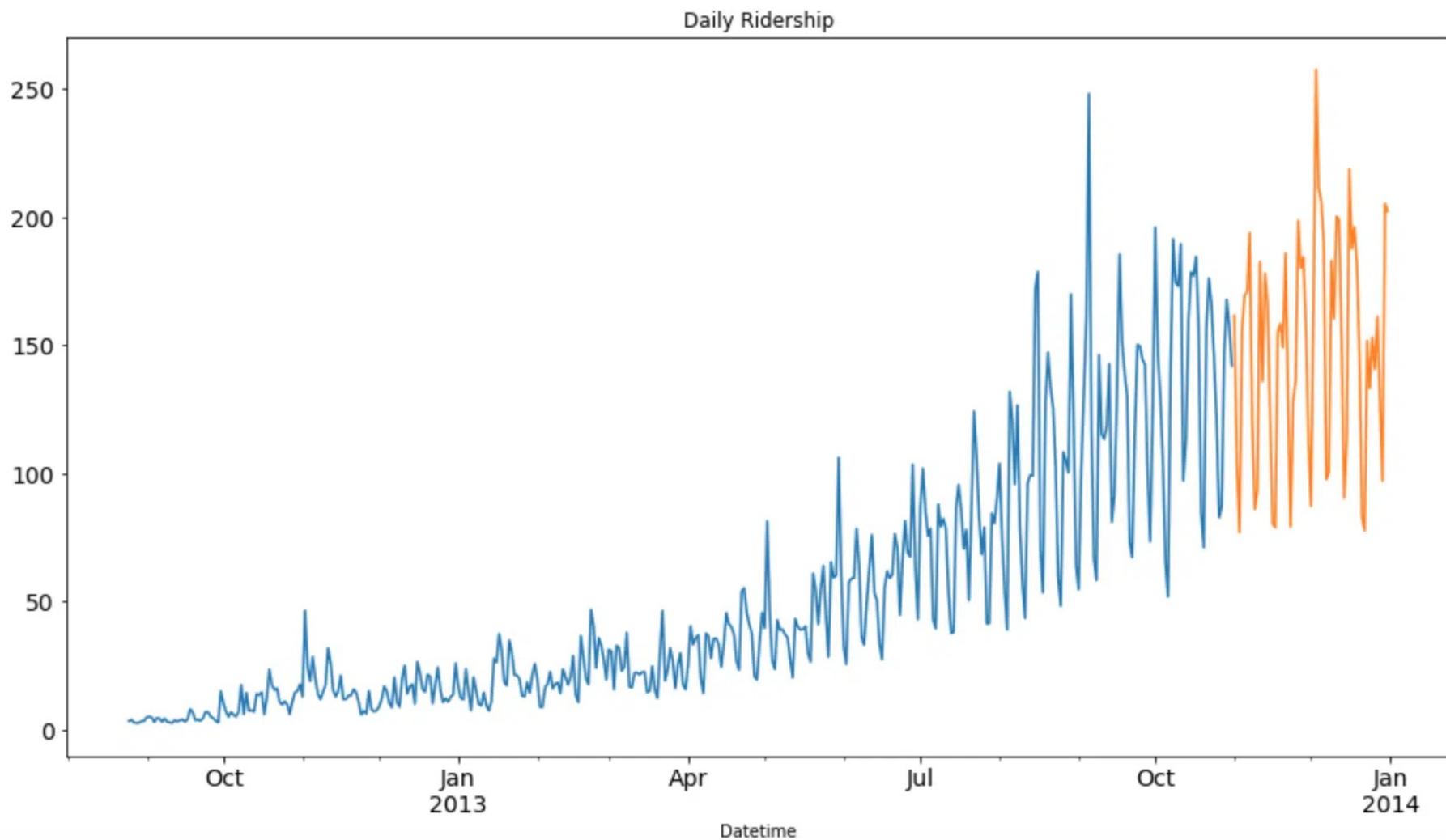
$$\text{TF} = \frac{n_i}{\sum_k n_k}$$

$$\text{IDF} = \log \frac{|D|}{|(d_i \supset t_i)|}$$

- n_i – число вхождений i -того элемента во множество
- $|d_i \supset t_i|$ - число множеств с элементом t_i
- $|D|$ - количество множеств

- Пример: вычислите TF-IDF слова "нейросеть" в сообщении на форуме мехмата, если автор употребил его 2 раза в своем посте из 50 слов, а в общем на форуме оно встречается в 400 сообщениях из 10000

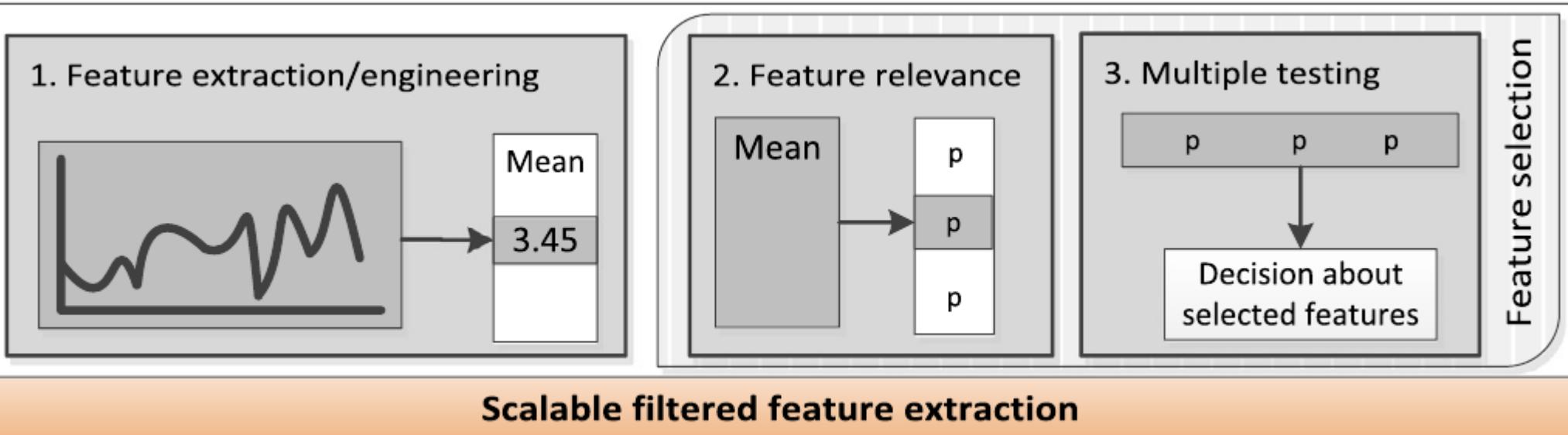
Последовательности данных во времени



Последовательности данных во времени

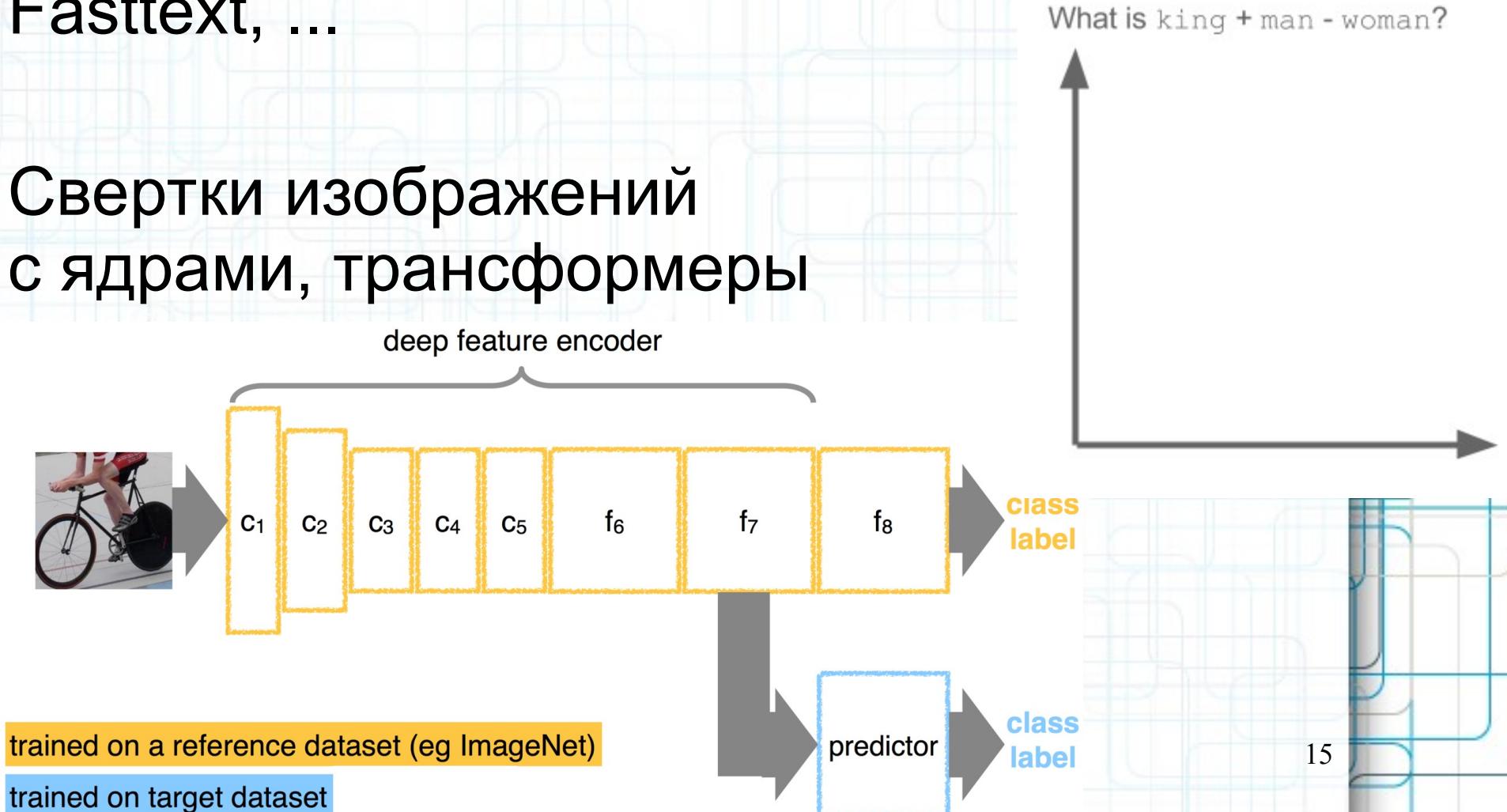
- набор последних значений
- mean/min/max за последние периоды
- экспоненциальное скользящее среднее
- тренд (коэффициенты линейной аппроксимации на последнем периоде)
- обобщение: коэф-ты авторегрессии
- количество пиков на последнем интервале
- вариация (интеграл от модуля производной)

TSFRESH – Python-библиотека для автоматического извлечения и отсея признаков временных рядов



Тексты и изображения

- Word embeddings: Word2Vec (CBOW, skip-gram), GloVe (Global Vectors), Fasttext, ...
- Свертки изображений с ядрами, трансформеры



Бинаризация признаков (One Hot encoding)

- Пусть x - единственный признак (номинальный, закодированный: 0,1,2,...k)
- Классификатор: $a(x) = \text{sign}(wx + w_0)$
- Проблема линейных алгоритмов: вес w нельзя подобрать так, чтобы классификатор был не монотонным.
- Для любых w и w_0 значения $a(x) > 0$ когда $x > w_0/w$ и $a(x) \leq 0$ в противном случае

Бинаризация признаков (One Hot encoding)

- Вместо одного номинального признака вводим k бинарных признаков.
Пример (k=5):

	x1	x2	x3	x4	x5
Азов	0	0	0	0	1
Аксай	0	0	0	1	0
Ростов	0	0	1	0	0
Новочеркасск	0	1	0	0	0
Таганрог	1	0	0	0	0

- Возможна бинаризация и количественных признаков путем предварительной дискретизации

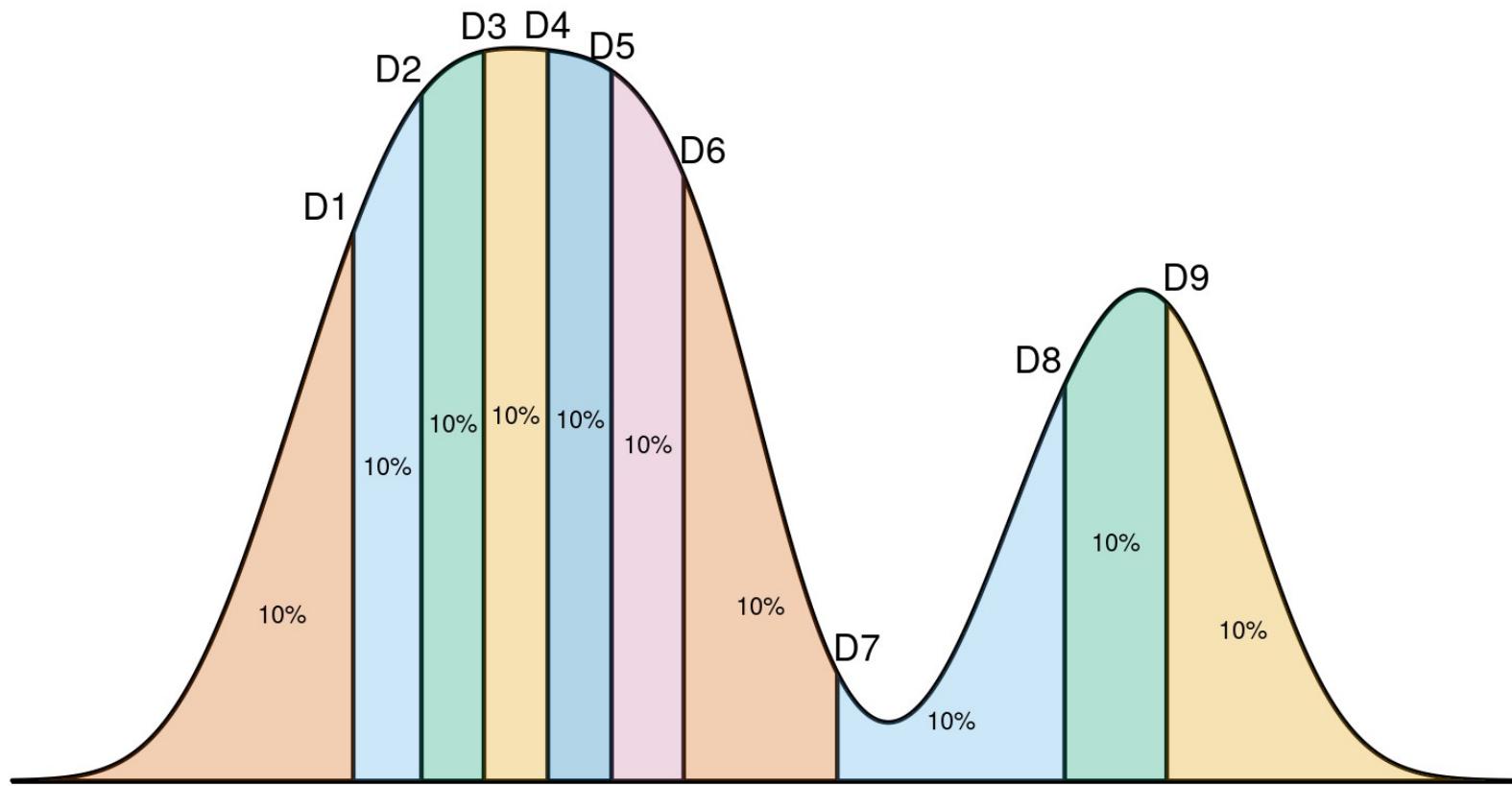
Скоринг

- Если все признаки – бинарные, то линейный классификатор удобно рассматривать как суммирование баллов (score): $\text{Sum } += w_j$, если $x_j = 1$
- Рисунок – фрагмент скоринговой карты для вопроса о выдаче кредита

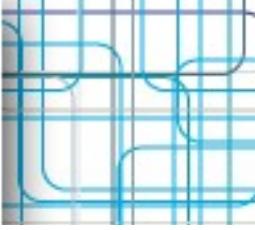
Возраст	до 25	5
	25 - 40	10
	40 - 50	15
	50 и больше	10
Собственность	владелец	20
	совладелец	15
	съемщик	10
	другое	5
Работа	руководитель	15
	менеджер среднего звена	10
	служащий	5
	другое	0
Стаж	1/безработный	0
	1..3	5
	3..10	10
	10 и больше	15
Работа_мужа /жены	нет/домохозяйка	0
	руководитель	10
	менеджер среднего звена	5
	служащий	1

Дискретизация признаков

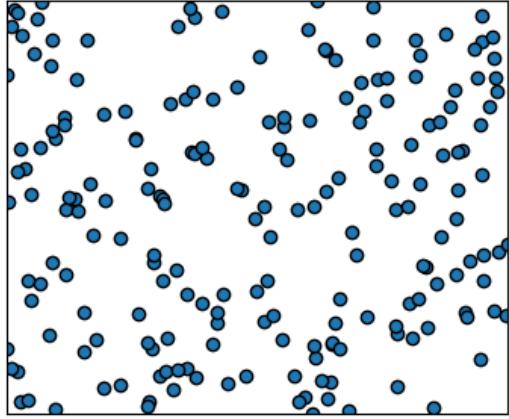
- Квантильная дискретизация сохраняет равномерность ошибки оценки плотности вероятностей



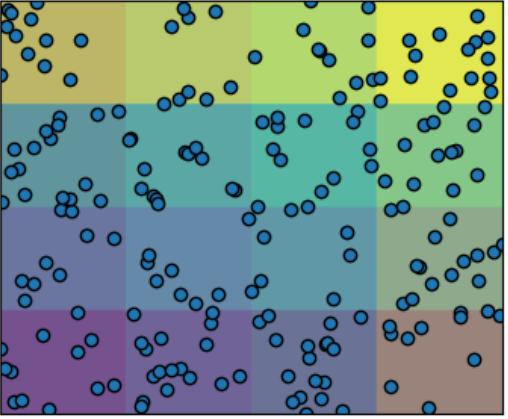
Стратегии KBinsDiscretizer



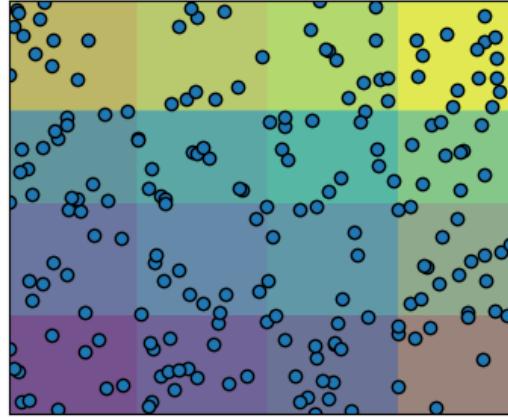
Input data



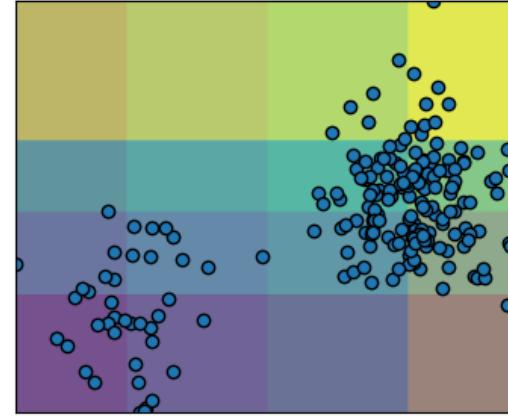
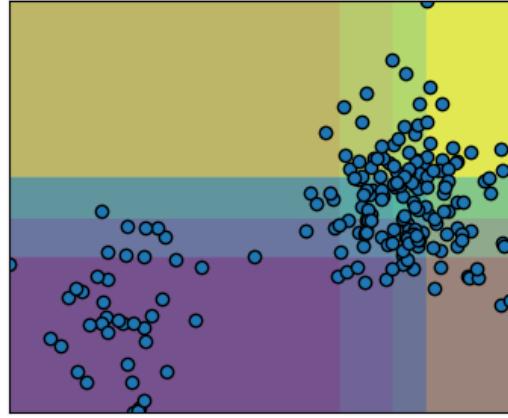
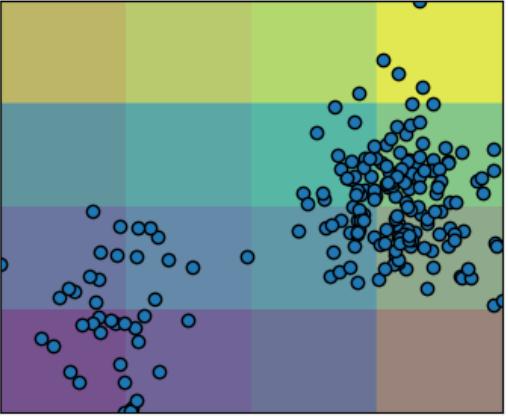
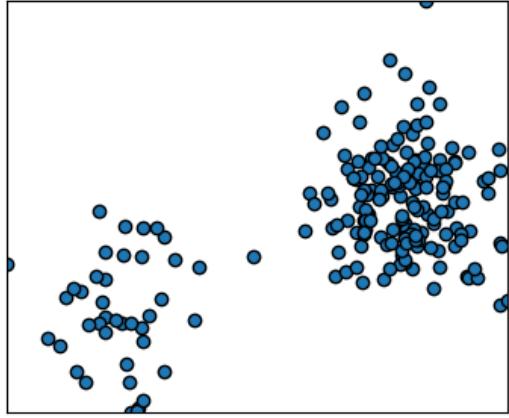
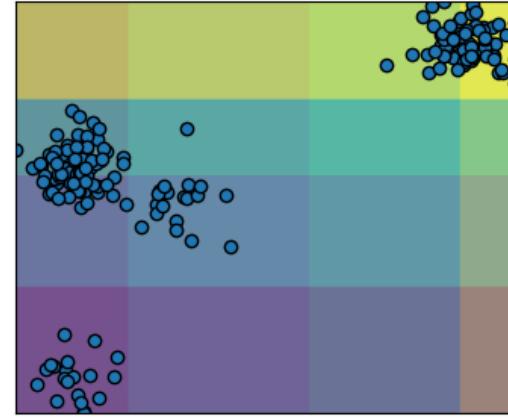
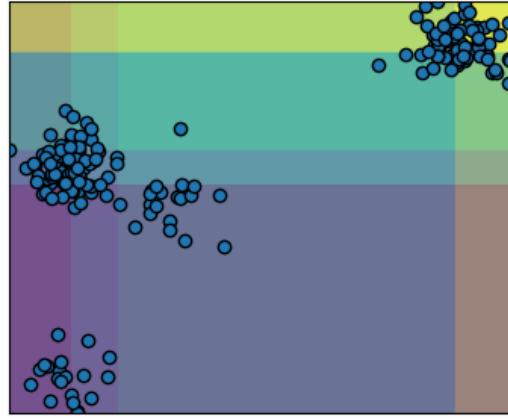
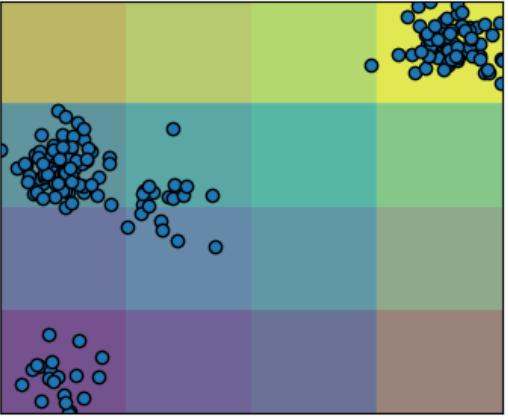
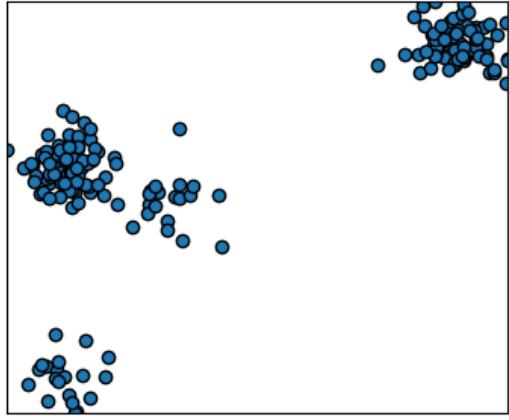
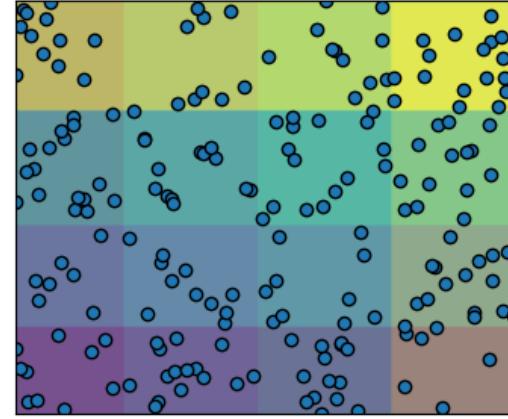
strategy='uniform'



strategy='quantile'



strategy='kmeans'



Сингулярное разложение

Произвольная $\ell \times n$ -матрица представима в виде *сингулярного разложения* (singular value decomposition, SVD):

$$F = VDU^\top.$$

Основные свойства сингулярного разложения:

- ① $\ell \times n$ -матрица $V = (v_1, \dots, v_n)$ ортогональна, $V^\top V = I_n$, столбцы v_j — собственные векторы матрицы FF^\top ;
- ② $n \times n$ -матрица $U = (u_1, \dots, u_n)$ ортогональна, $U^\top U = I_n$, столбцы u_j — собственные векторы матрицы $F^\top F$;
- ③ $n \times n$ -матрица D диагональна, $D = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$, $\lambda_j \geq 0$ — собственные значения матриц $F^\top F$ и FF^\top .

Метод главных компонент (РСА)

- $f_1(x), \dots, f_n(x)$ — исходные числовые признаки;
- $g_1(x), \dots, g_m(x)$ — новые числовые признаки, $m < n$;
- Требование: старые признаки должны линейно восстанавливаться по новым:

$$\hat{f}_j(x) = \sum_{s=1}^m g_s(x) u_{js}, \quad j = 1, \dots, n, \quad \forall x \in X$$

как можно точнее на обучающей выборке:

$$\sum_{i=1}^{\ell} \sum_{j=1}^n (\hat{f}_j(x_i) - f_j(x_i))^2 \rightarrow \min_{\{g_s(x_i)\}, \{u_{js}\}}$$

Постановка задачи РСА в матричной форме

$$\hat{F} = GU^\top \stackrel{\text{хотим}}{\approx} F$$

Найти: и новые признаки G , и преобразование U :

$$\sum_{i=1}^{\ell} \sum_{j=1}^n (\hat{f}_j(x_i) - f_j(x_i))^2 = \|GU^\top - F\|^2 \rightarrow \min_{G, U}$$

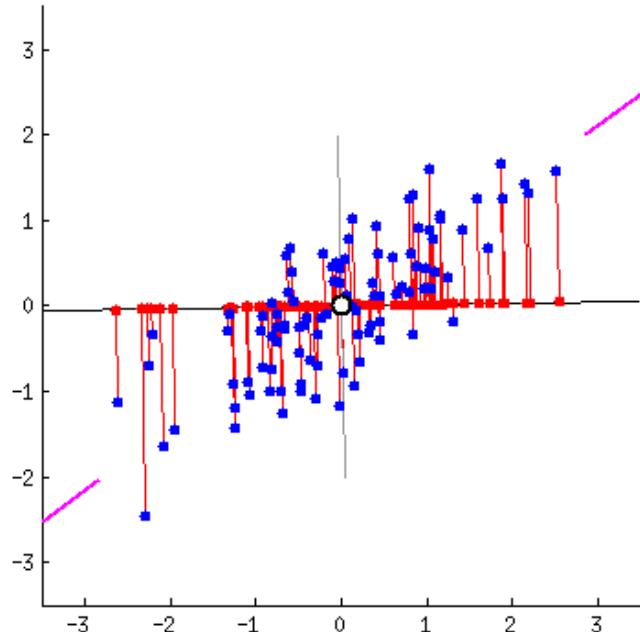
Теорема

Если $m \leq \text{rk } F$, то минимум $\|GU^\top - F\|^2$ достигается, когда столбцы U — это с.в. матрицы $F^\top F$, соответствующие m максимальным с.з. $\lambda_1, \dots, \lambda_m$, а матрица $G = FU$.

При этом:

- ❶ матрица U ортонормирована: $U^\top U = I_m$;
- ❷ матрица G ортогональна: $G^\top G = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$;
- ❸ $U\Lambda = F^\top FU$; $G\Lambda = FF^\top G$;
- ❹ $\|GU^\top - F\|^2 = \|F\|^2 - \text{tr } \Lambda = \sum_{j=m+1}^n \lambda_j$.

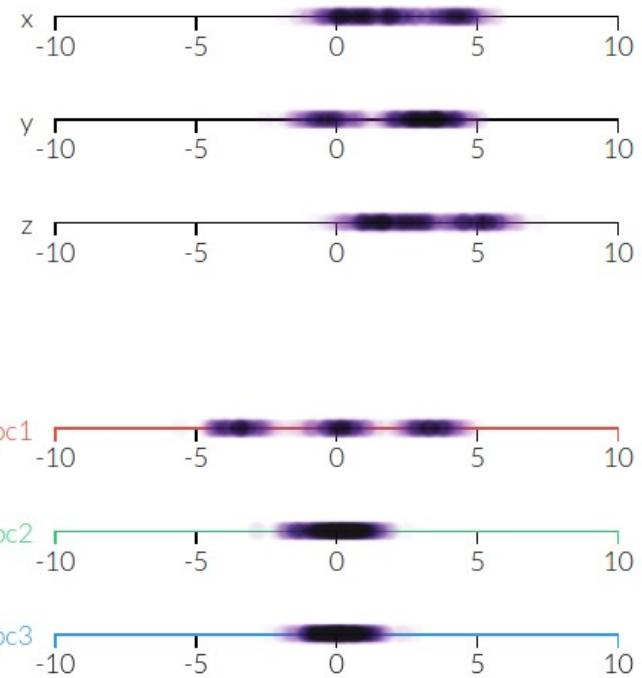
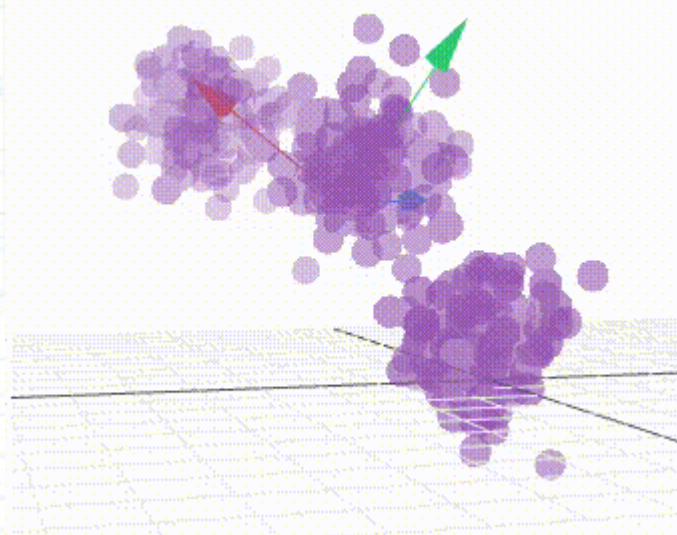
Главная компонента датасета - направление наибольшей вариации точек



$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{x}_{(i)} \cdot \mathbf{w})^2 \right\}$$

Альтернативное
определение:
главная компонента задает
прямую, среднее расстояние
точек датасета до которой
минимально

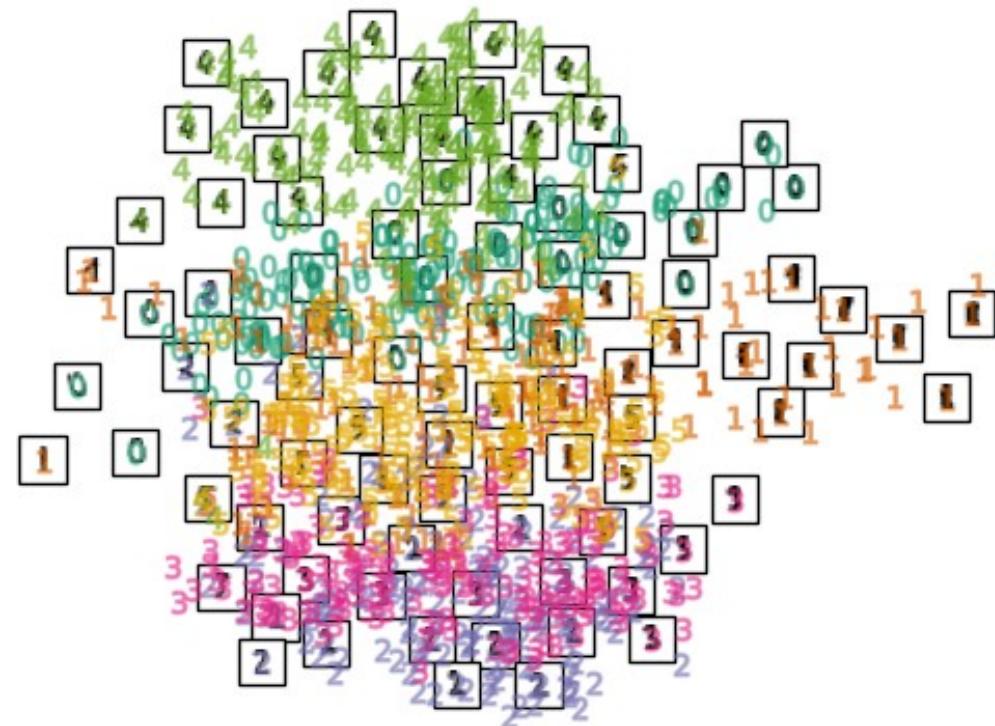
Следующие компоненты - такие же главные в ортогональном дополнении к предыдущим



Самое распространенное приложение: понижение размерности датасета

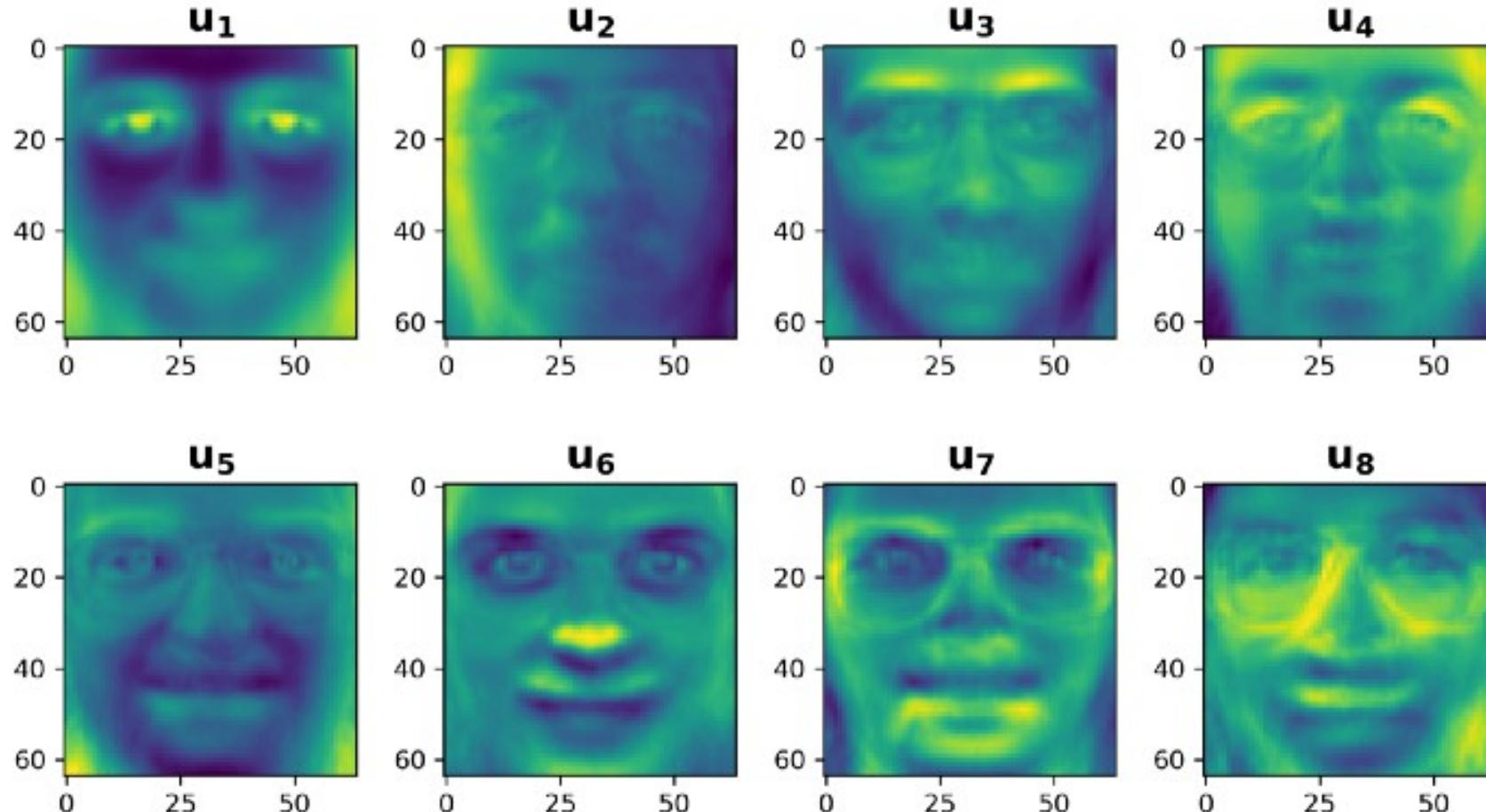
A selection from the 64-dimensional digits dataset

0	1	2	3	4	5	0	1	1	3
9	5	0	1	2	3	4	5	0	5
5	5	0	4	1	3	5	1	0	0
2	2	2	0	1	2	3	3	3	0
4	4	1	5	0	5	2	2	0	0
1	3	2	1	4	3	1	1	1	4
3	4	4	0	5	3	1	5	4	4
2	2	2	5	5	4	4	0	0	1
2	3	4	5	0	1	2	3	4	5
0	1	2	3	4	5	0	5	5	5

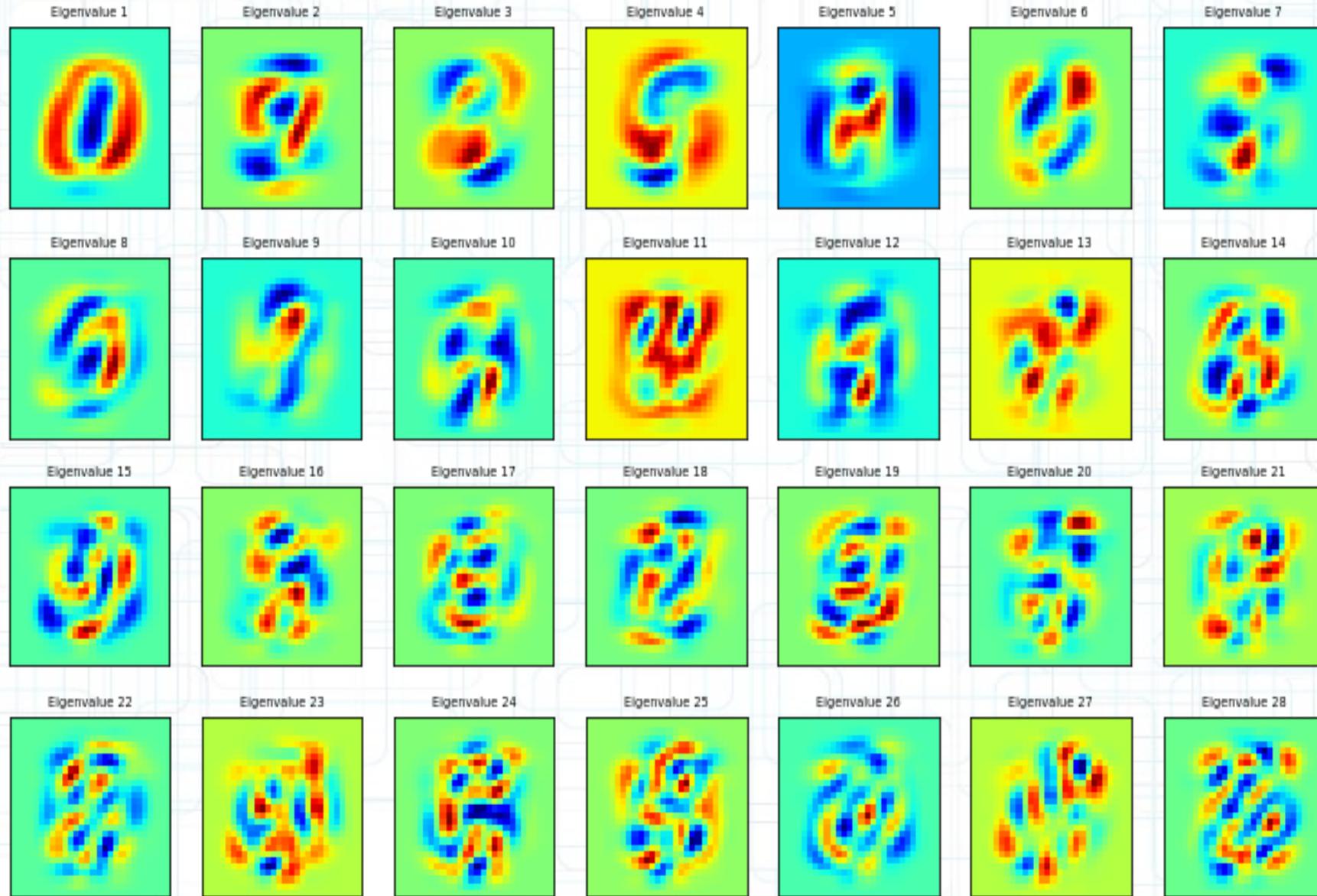


Главные компоненты датасета Olivetti faces

Показывают ортогональные направления, вдоль которых лица датасета меняются сильнее всего

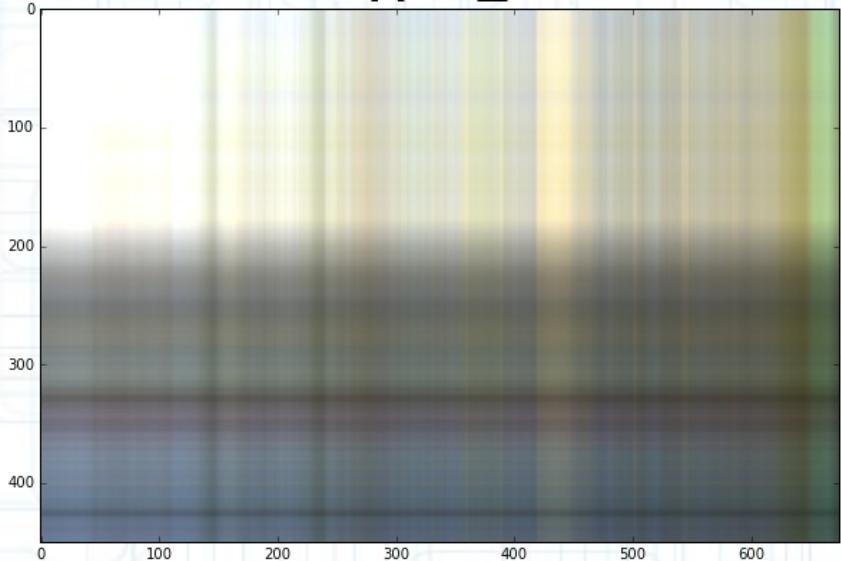


Главные компоненты датасета рукописных цифр



Применение РСА к сжатию изображений

$n=1$



$n=10$



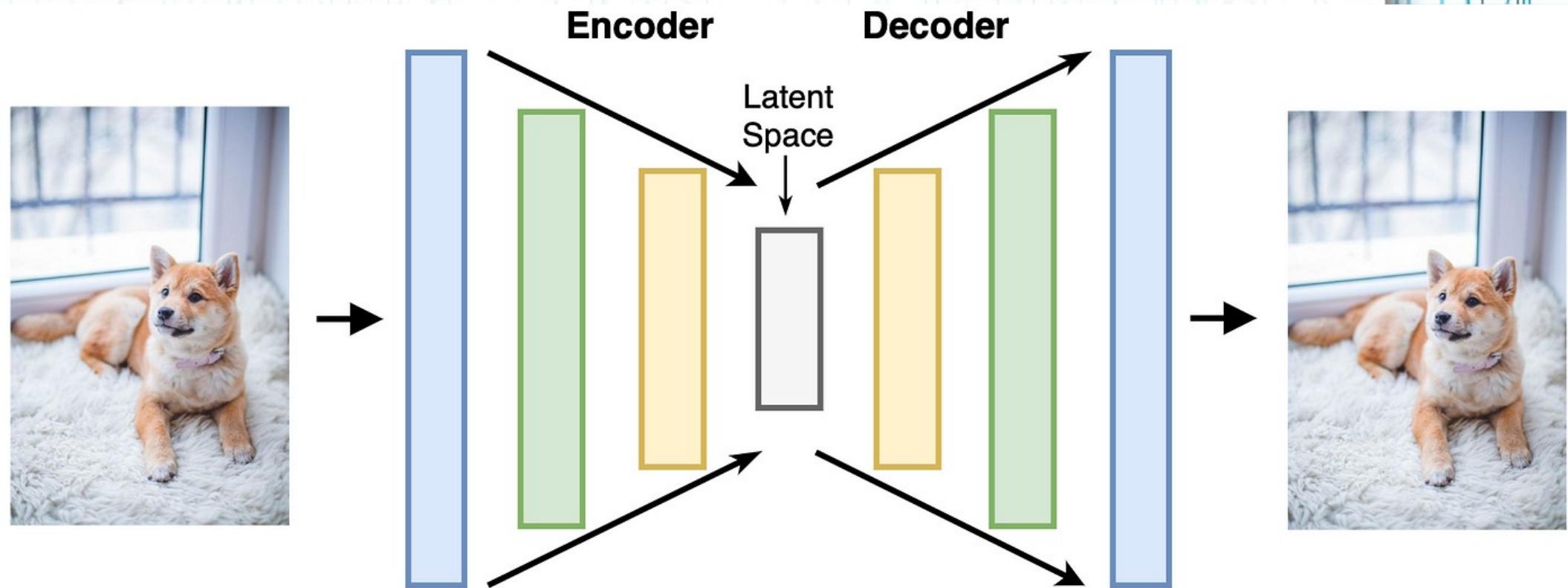
$n=30$



$n=100$

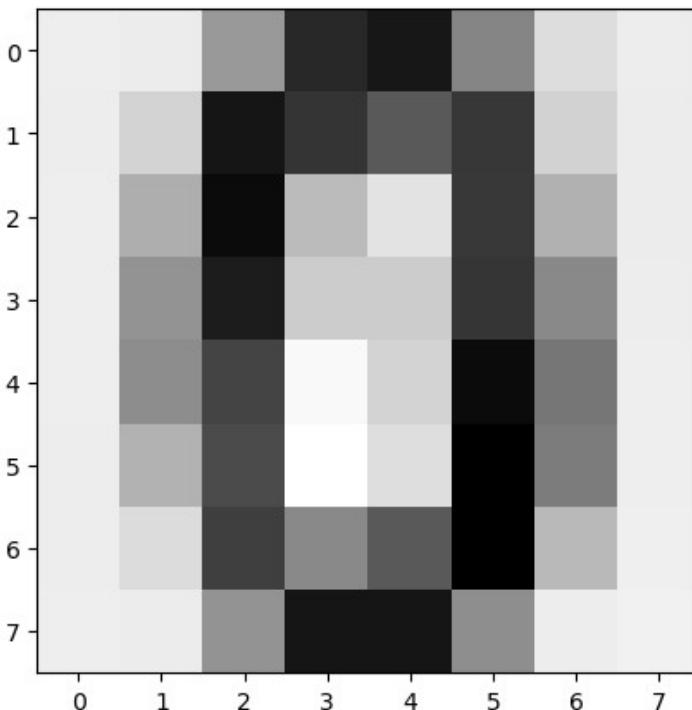
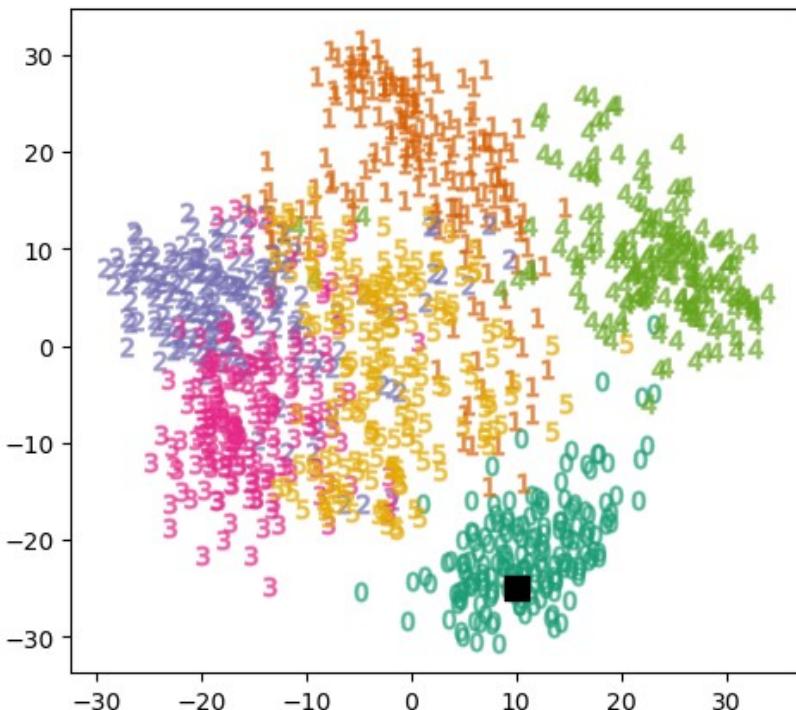


PCA – простейший пример трансформера



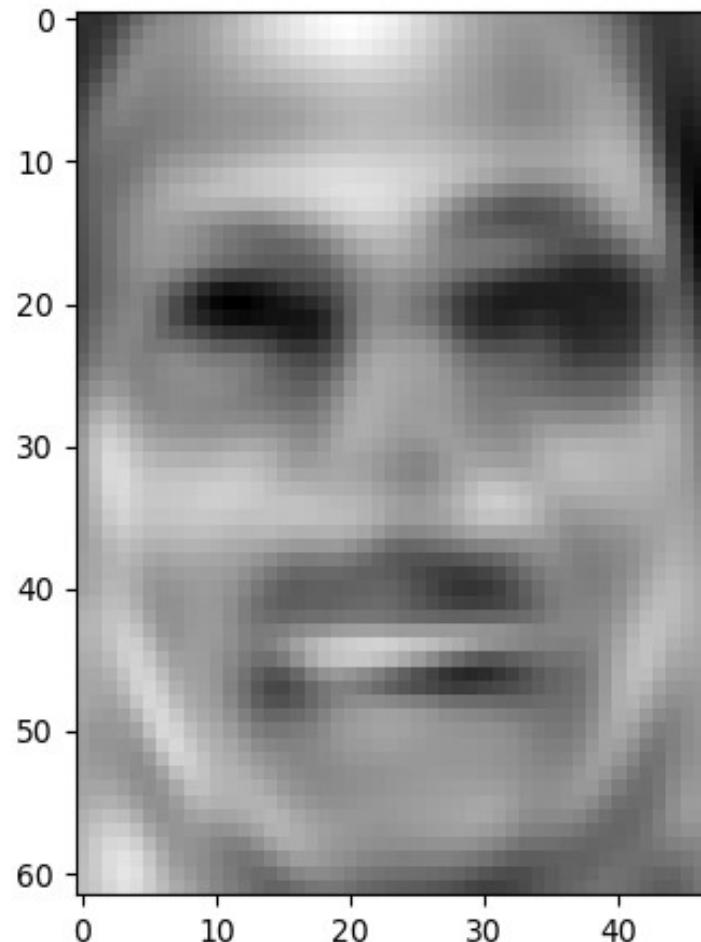
PCA - простейший линейный трансформер

- Латентное пространство PCA - линейная оболочка, натянутая на первые несколько главных компонент
- Выбирая точки в латентном пространстве и применяя декодер, можно генерировать “искусственные” объекты, отсутствующие в датасете

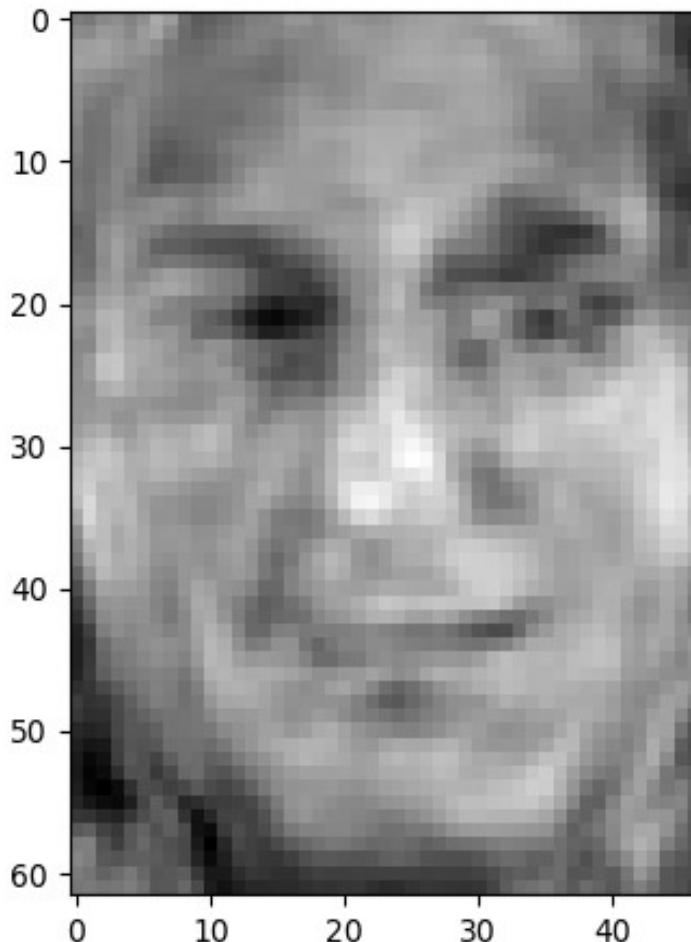


Генерация фото с помощью РСА

150 компонент

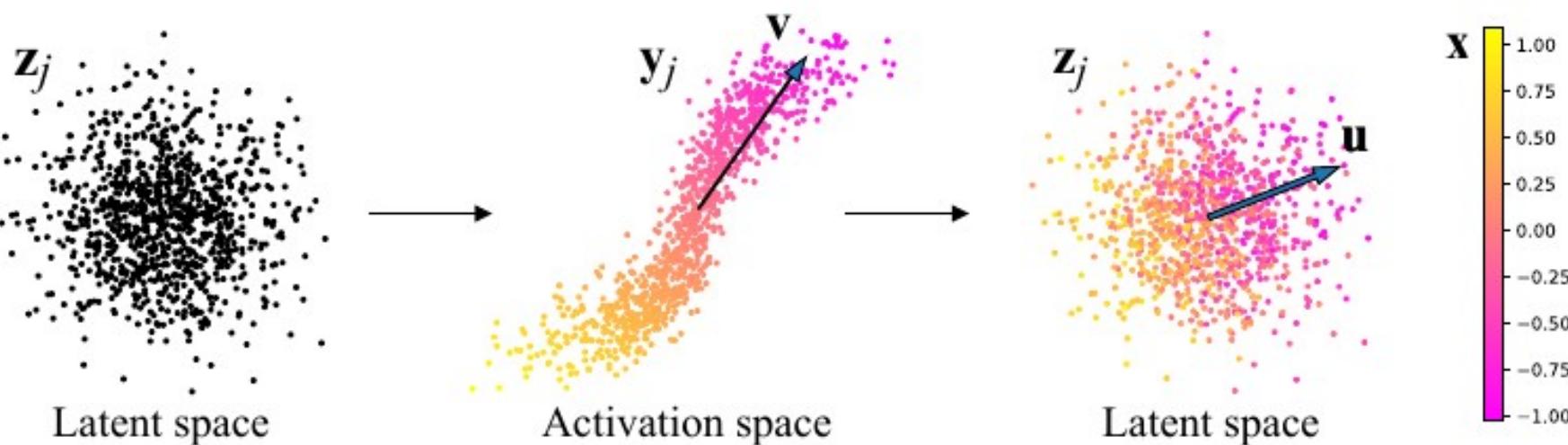


1500 компонент



Управление генерацией в GAN

- Для генерации изображений с помощью GAN берут случайные точки в латентном пространстве
- Оказывается, что после трансформации нормально распределенной выборки на одном скрытом слое, главные компоненты в следующих слоях поддаются интерпретации



Управление генерацией в GAN



Управление генерацией в GAN

StyleGAN2
Cars



Initial image



E(\mathbf{u}_{22} , 9-10)
change color



E(\mathbf{u}_{41} , 9-10)
add grass



E(\mathbf{u}_0 , 0-4)
rotate

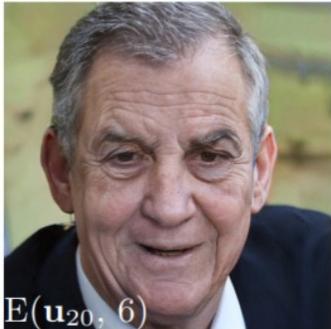


E(\mathbf{u}_{16} , 3-5)
change type

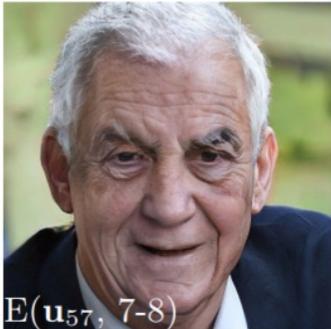
StyleGAN2
FFHQ



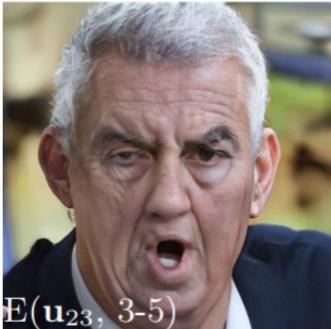
Initial image



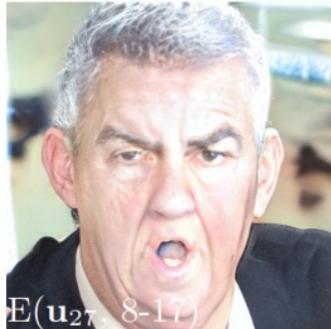
E(\mathbf{u}_{20} , 6)
add wrinkles



E(\mathbf{u}_{57} , 7-8)
hair color



E(\mathbf{u}_{23} , 3-5)
expression



E(\mathbf{u}_{27} , 8-17)
overexpose

BigGAN512-deep
Irish setter



Initial image



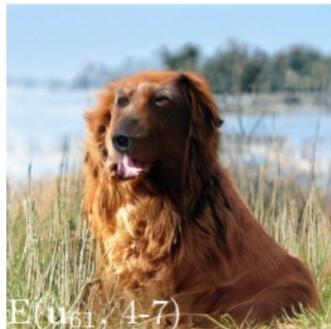
E(\mathbf{u}_3 , all)
rotate



E(\mathbf{u}_{12} , all)
zoom out



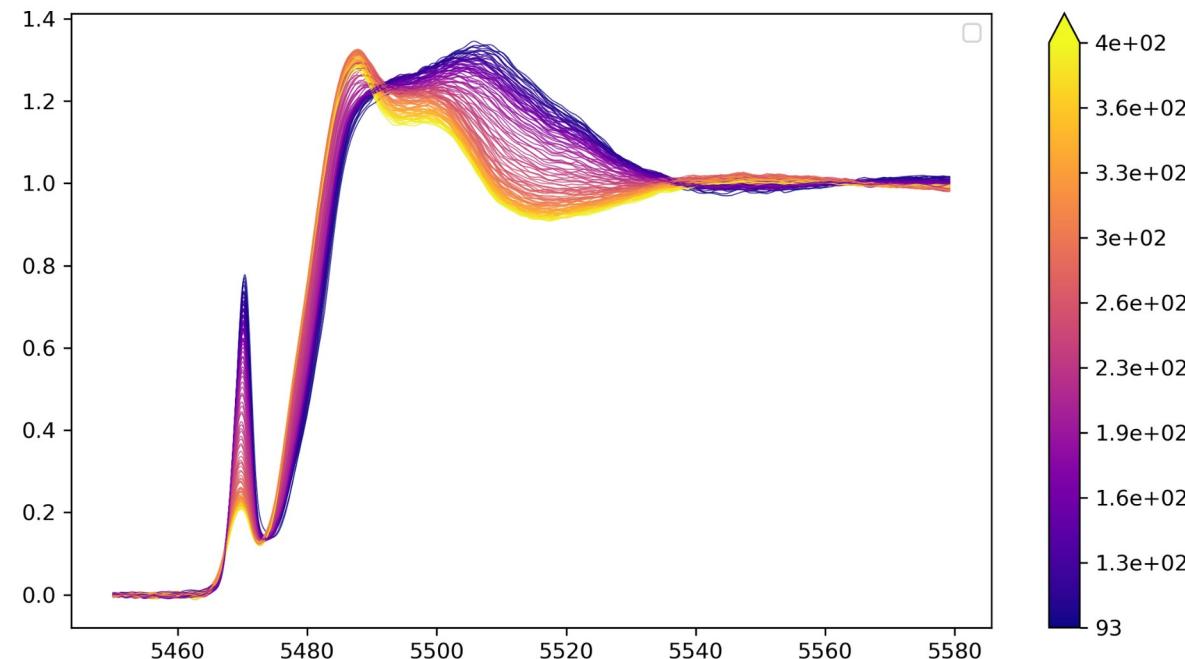
E(\mathbf{u}_{15} , 1-5)
show horizon



E(\mathbf{u}_{61} , 4-7)
change scenery

Поиск компонент смесей

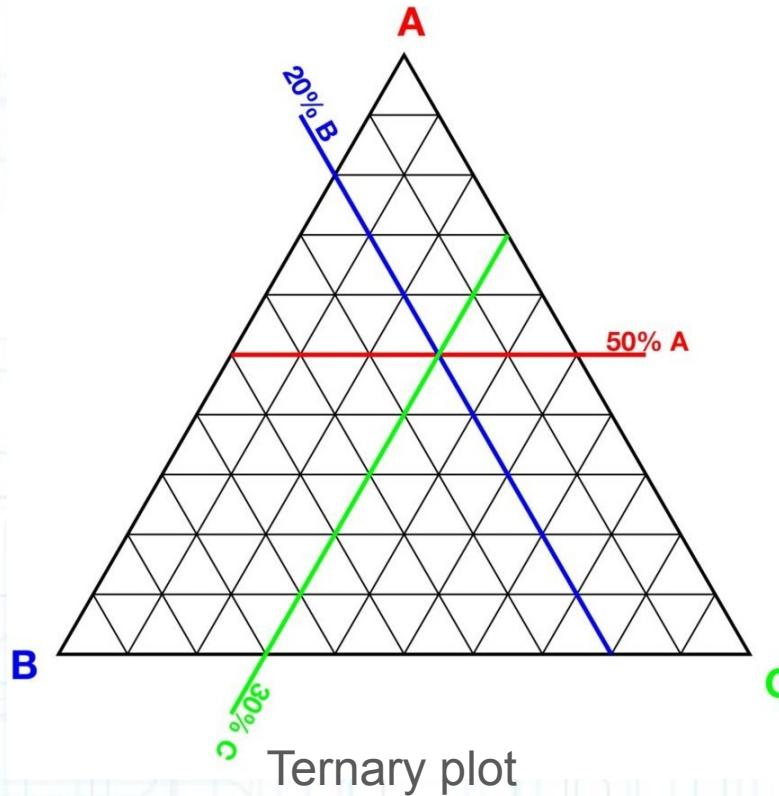
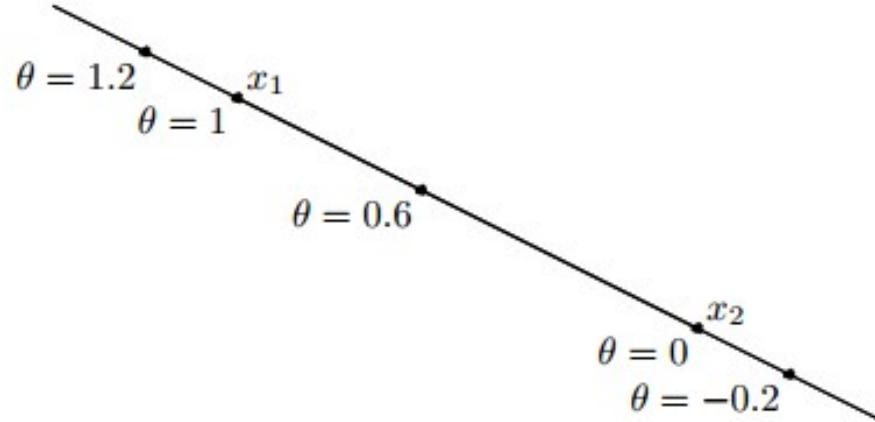
- Дано: серия спектров для смеси веществ в процессе проведения некоторого эксперимента (изменяется температура/ давление/... и в результате меняются концентрации компонент)
- Найти: спектры чистых компонент и зависимость концентрации от меняющегося параметра эксперимента



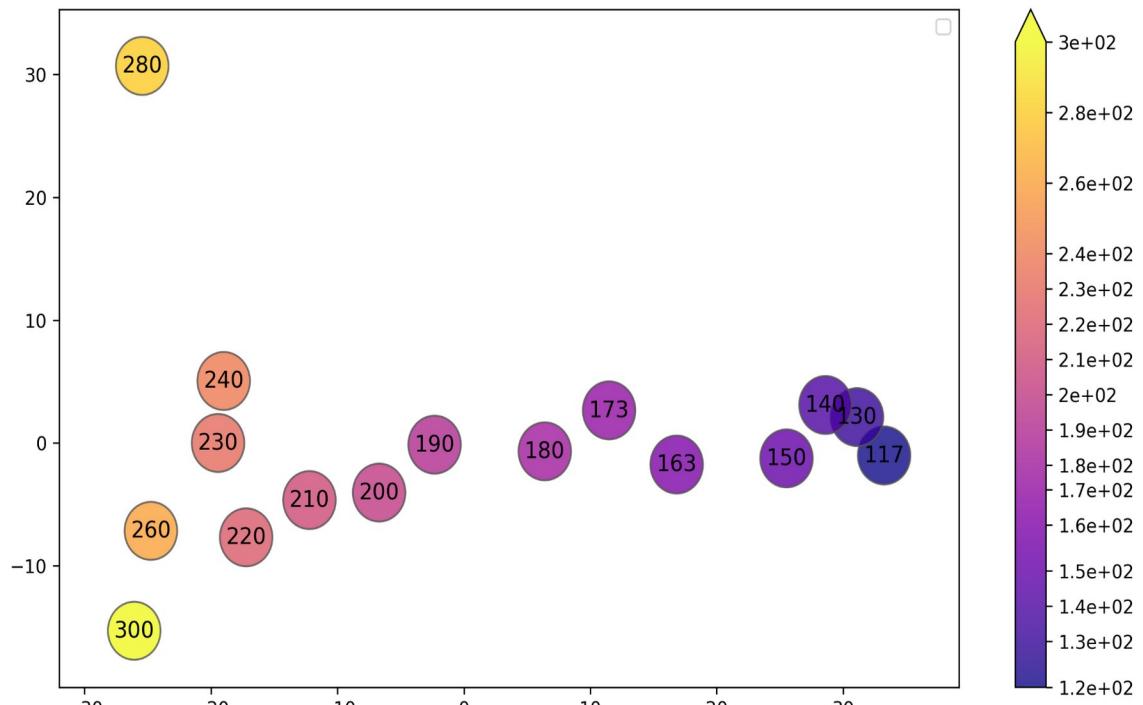
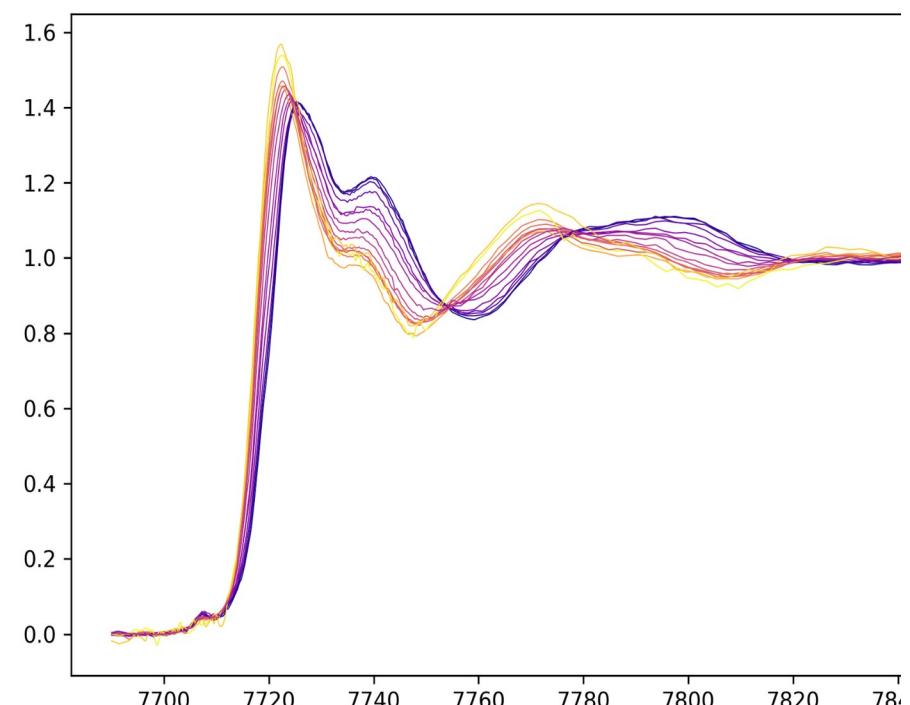
Подход на основе РСА

- Благодаря линейности преобразования в латентном пространстве смеси тоже являются линейными комбинациями точек и имеют вид (для 2-х компонент):
 $c1(t) \cdot \text{Spectr1} + c2(t) \cdot \text{Spectr2}$
- Что за фигура получается в результате изменения температуры t ?
- А если компонент три:
 $c1(t) \cdot \text{Spectr1} + c2(t) \cdot \text{Spectr2} + c3(t) \cdot \text{Spectr3}$?

Смеси двух и трёх компонент в латентном пространстве



Пример Со ТЕМРО

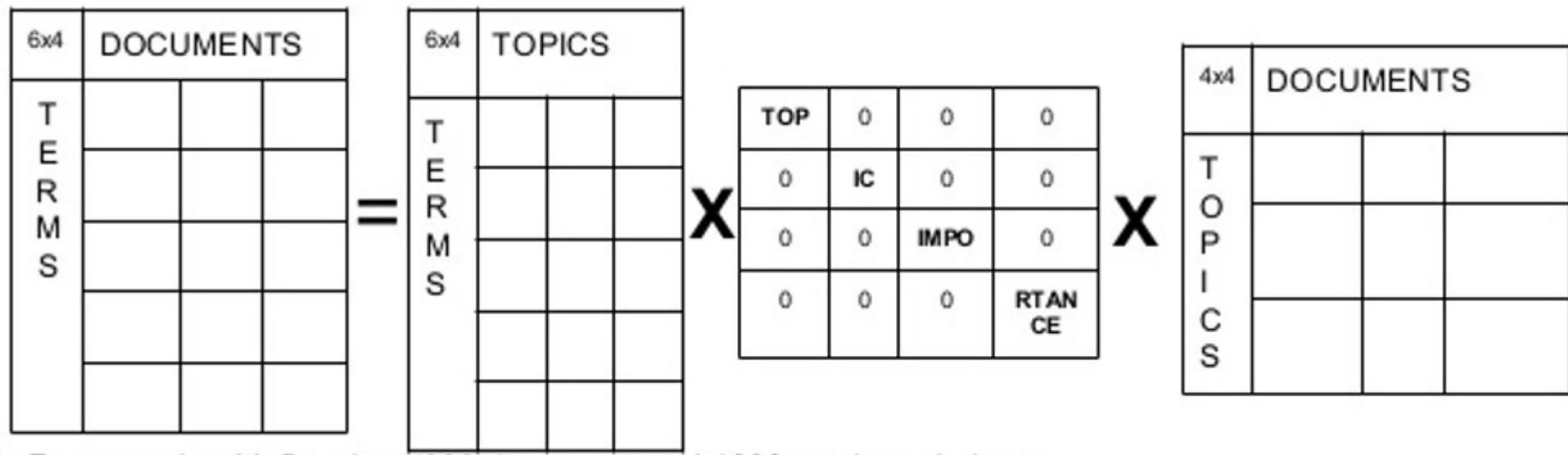


Применение к наборам текстов

$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix}$$

	D1	D2	D3	D4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4

SVD-разложение Term-Document матрицы



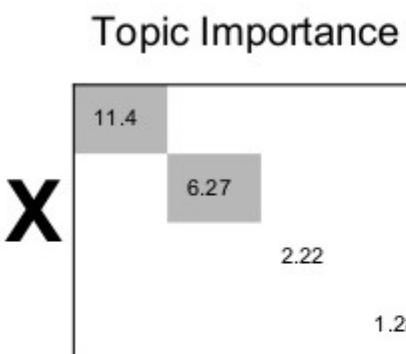
Числа в диагональной матрице имеют смысл
“важностей” тем в нашей коллекции документов

Пример

3	4	1	0
4	3	0	1
3	4	4	3
0	1	4	3
2	0	3	3
0	1	3	4

=

	To1	To2	To3	To4
Te1	-0.33	-0.53	0.37	-0.14
Te2	-0.32	-0.54	-0.49	0.35
Te3	-0.62	-0.10	0.26	-0.14
Te4	-0.38	0.42	0.30	-0.24
Te5	-0.36	0.25	-0.68	-0.47
Te6	-0.37	0.42	0.02	0.75



X

X

	D1	D2	D3	D4
To1	-0.42	-0.48	-0.57	-0.51
To2	-0.56	-0.52	0.45	0.46
To3	-0.65	0.62	0.28	-0.35
To4	-0.30	0.34	-0.63	0.63

Оставили только главные компоненты

Word assignment to topics

3	4	1	0
4	3	0	1
3	4	4	3
0	1	4	3
2	0	3	3
0	1	3	4

=

	IT	cars
linux	-0.33	-0.53
modem	-0.32	-0.54
the	-0.62	-0.10
clutch	-0.38	0.42
steering	-0.36	0.25
petrol	-0.37	0.42

Topic Importance

11.4	
	6.27

X

Topic distribution acr documents

	D1	D2	D3	D4
-0.42	-0.48	-0.57	-0.51	
-0.56	-0.52	0.45	0.46	

Пример работы Самые весомые слова в полученных темах новостей

music
band
songs
rock
album
jazz
pop
song
singer
night

book
life
novel
story
books
man
stories
love
children
family

art
museum
show
exhibition
artist
artists
paintings
painting
century
works

game
Knicks
nets
points
team
season
play
games
night
coach

show
film
television
movie
series
says
life
man
character
know

theater
play
production
show
stage
street
broadway
director
musical
directed

clinton
bush
campaign
gore
political
republican
dole
presidential
senator
house

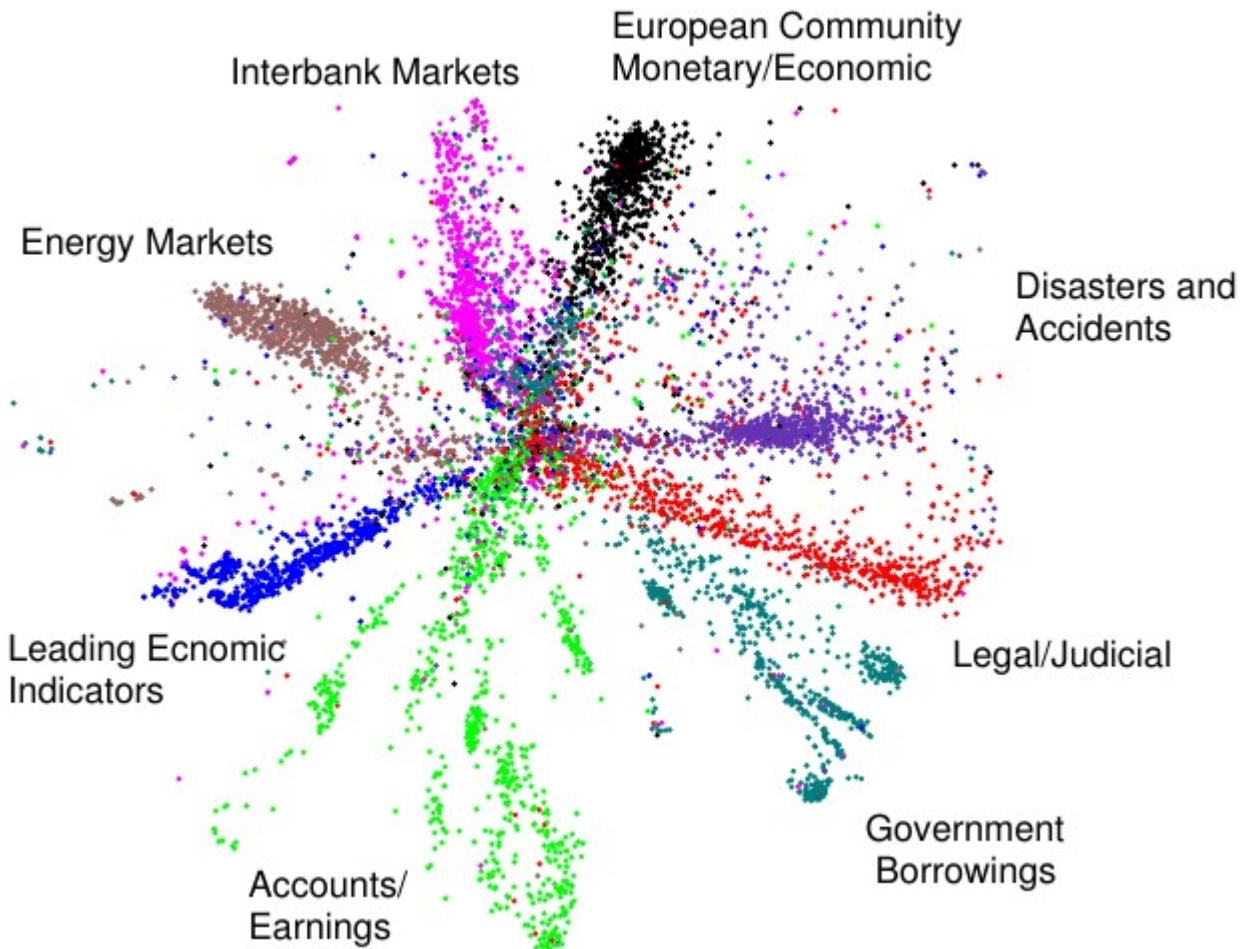
stock
market
percent
fund
investors
funds
companies
stocks
investment
trading

restaurant
sauce
menu
food
dishes
street
dining
dinner
chicken
served

budget
tax
governor
county
mayor
billion
taxes
plan
legislature
fiscal

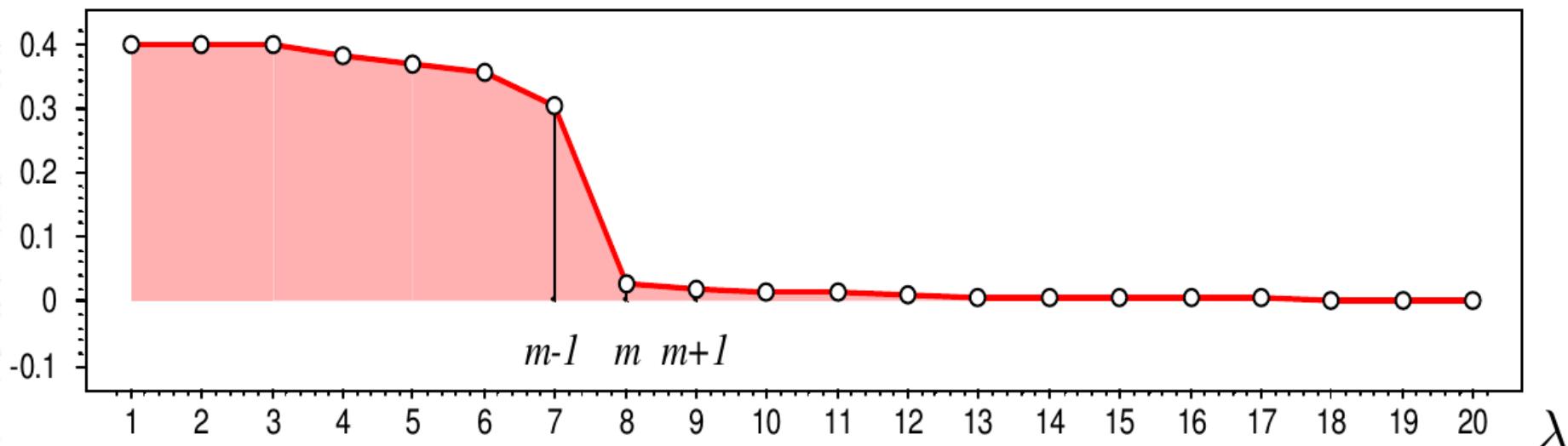
Пример работы 2

Визуализация документов



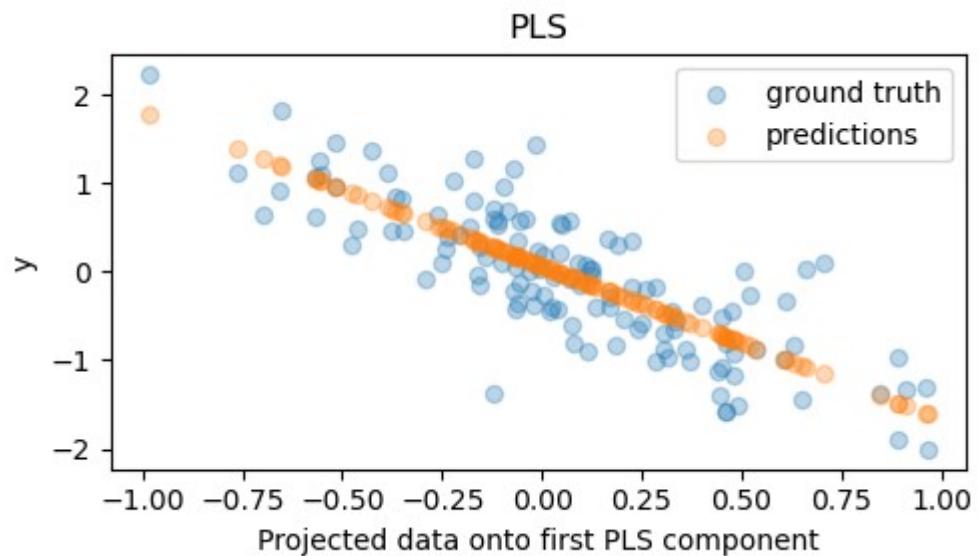
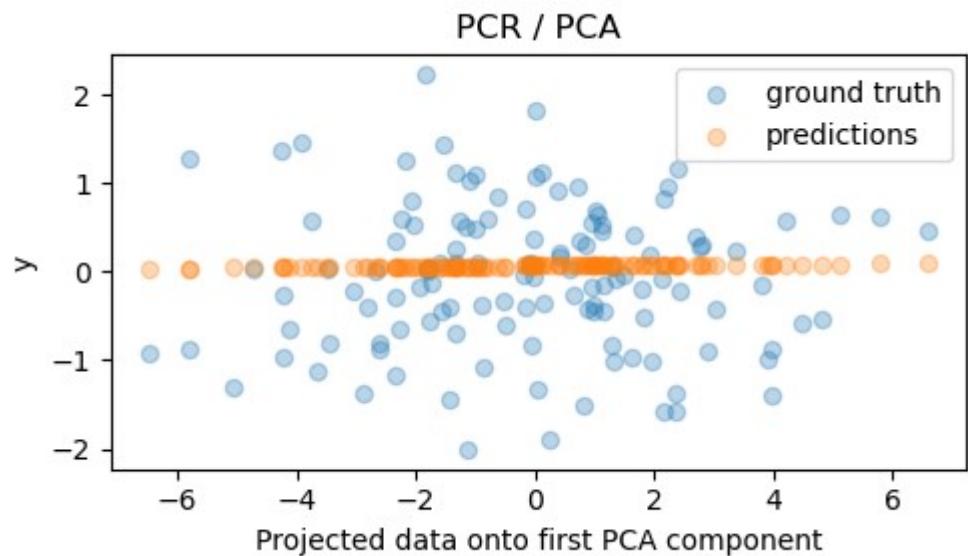
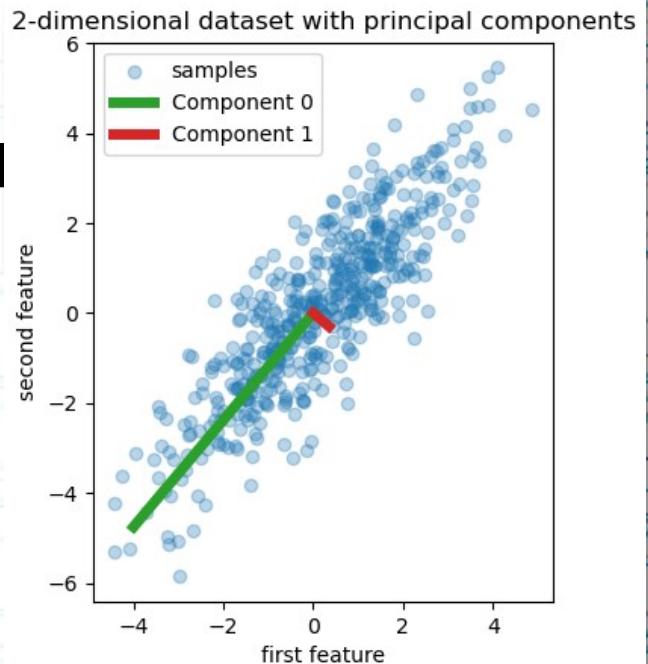
Сколько главных компонент брать?

- Критерий “крутого склона”:



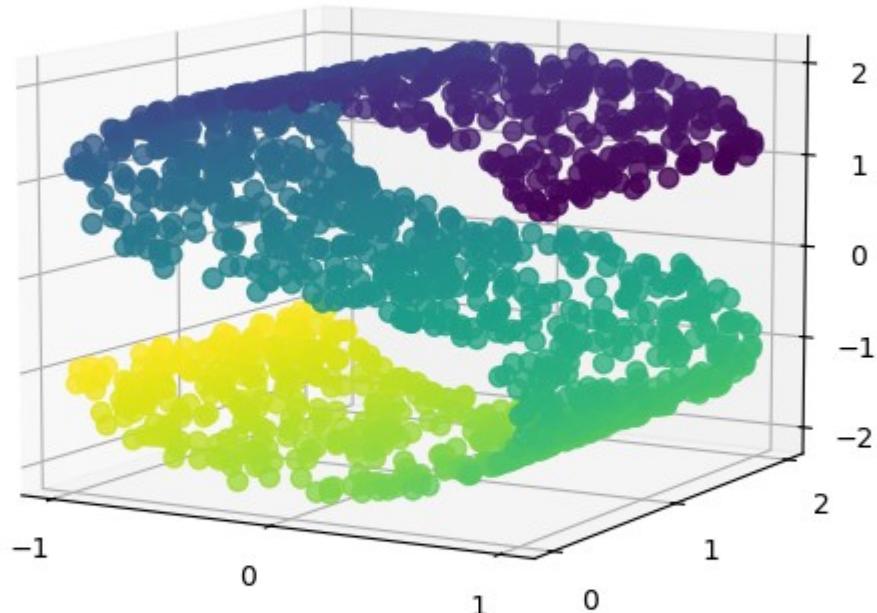
PLS - Partial Least Squares

- PCA находит компоненты, хорошо аппроксимирующие матрицу X , но не связанные с y
- Компоненты PLS ищутся из условия максимизации корреляций с y



Другие embeddings

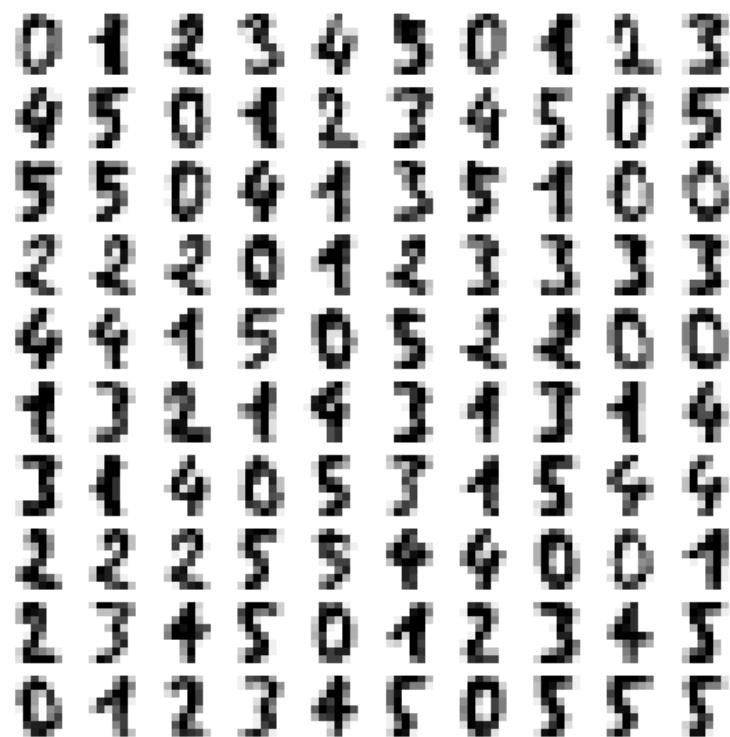
- Многомерность многих датасетов искусственно завышена. Пример: наблюдаемый набор данных физического эксперимента, в котором все зависит от двух варьирующихся параметров



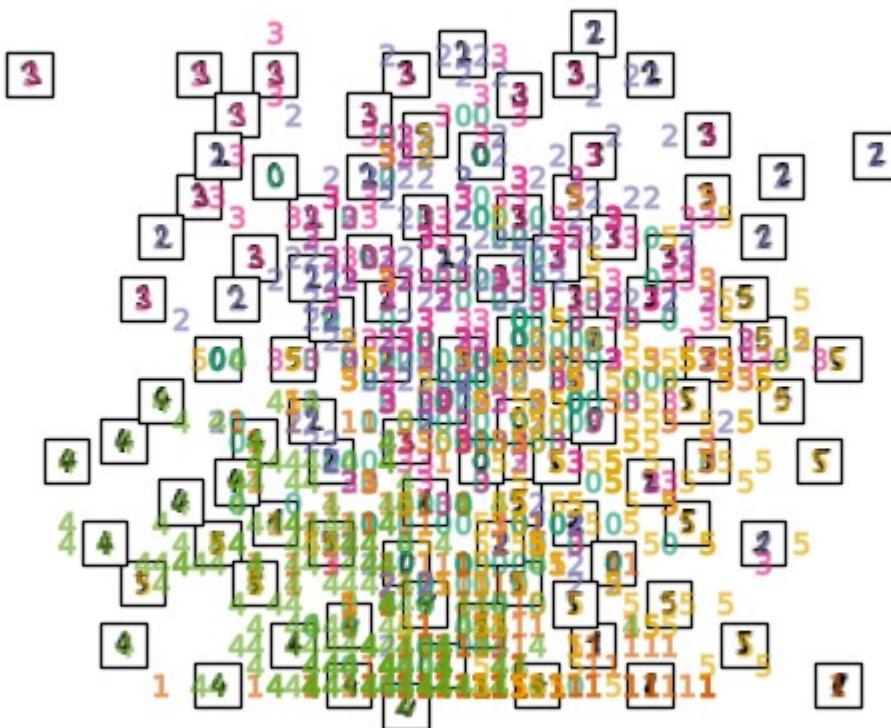
Другие embeddings

- Проекция датасета на случайные ортогональные вектора – не самый хороший способ уменьшения размерности

A selection from the 64-dimensional digits dataset



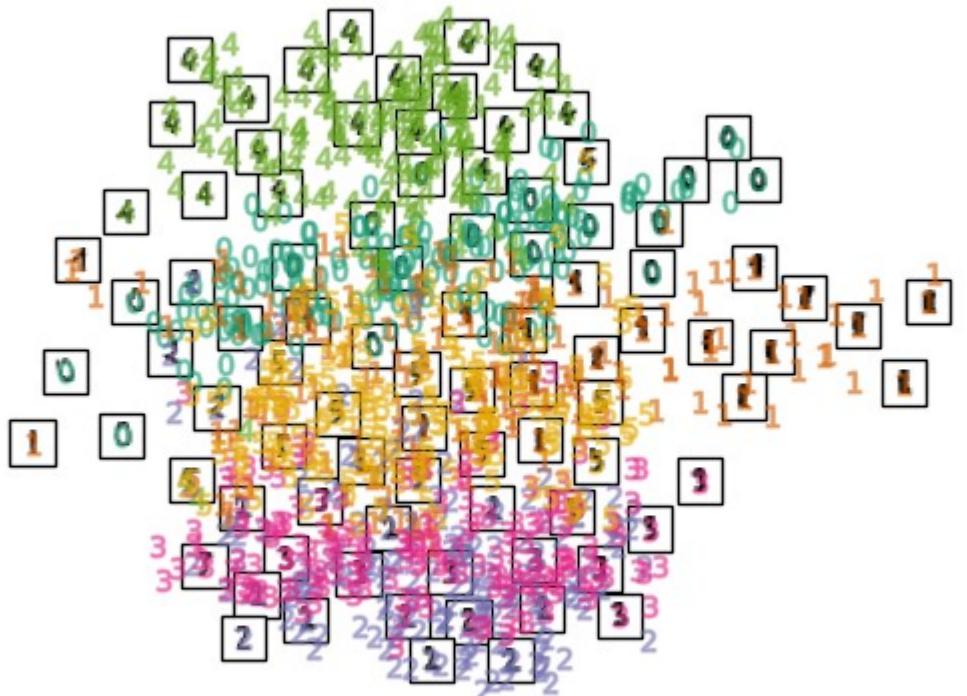
Random projection embedding (time 0.001s)



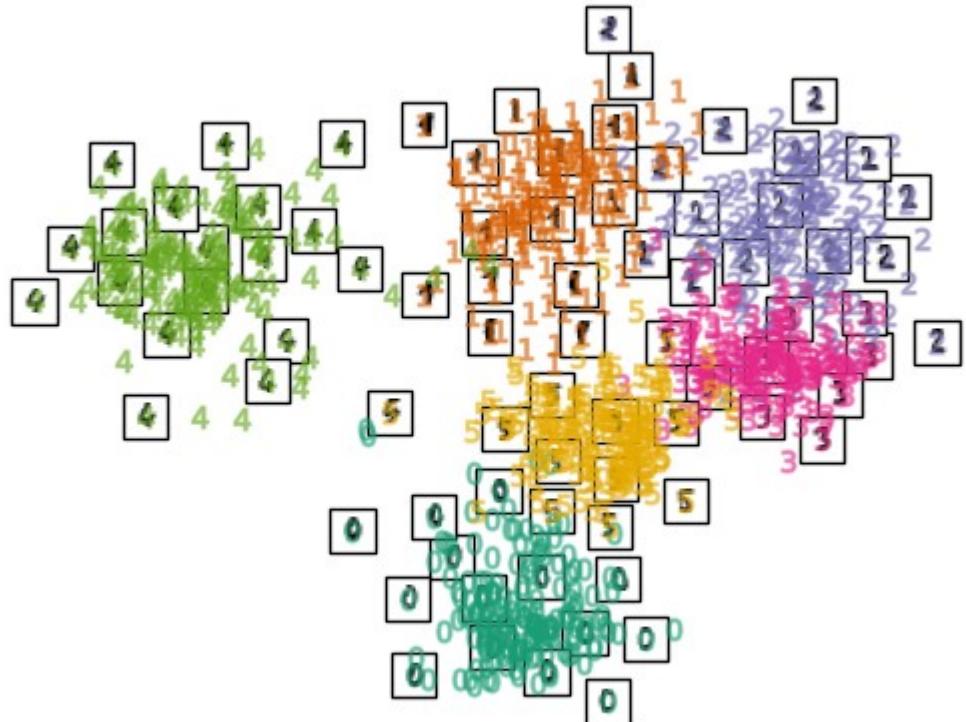
Другие embeddings

- Выбор PCA/PLS или LDA направлений позволяет получить более "интересные" результаты

Truncated SVD embedding (time 0.003s)



Linear Discriminant Analysis embedding (time 0.006s)



LDA – линейный дискриминантный анализ

- Моделирование распределения объектов в классах многомерными нормальными распределениями с общей ковариационной матрицей

$$P(x|y = k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k)\right)$$

- В качестве "главных направлений" выбираются вектора, проекции μ_k на которые имеют максимальную вариацию

MDS – многомерное шкалирование

- Находит преобразование пространства объектов в пространство меньшей размерности, сохраняющее расстояния между объектами.
- Многие методы оптимизируют функционал стресса:

$$S(X^\ell) = \sum_{(i,j) \in D} w_{ij} (d_{ij} - R_{ij})^2 \rightarrow \min$$

R_{ij} – исходные расстояния

d_{ij} – расстояния в пространстве меньшей размерности

w_{ij} – неизвестные веса

tSNE – стохастическое вложение соседей с t-распределением

- Похожесть точки данных x_j точке x_i является условной вероятностью $p_{j|i}$, что для x_i будет выбрана x_j в качестве соседней точки, если соседи выбираются пропорционально их гауссовой плотности вероятности с центром в x_i

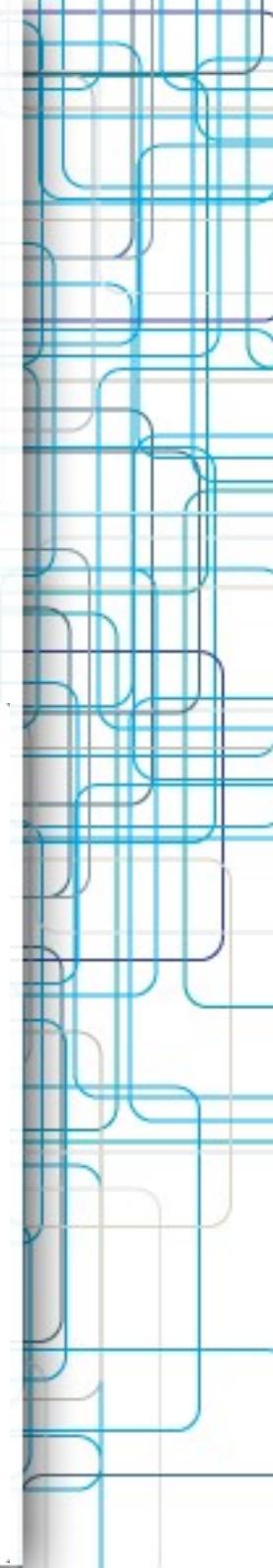
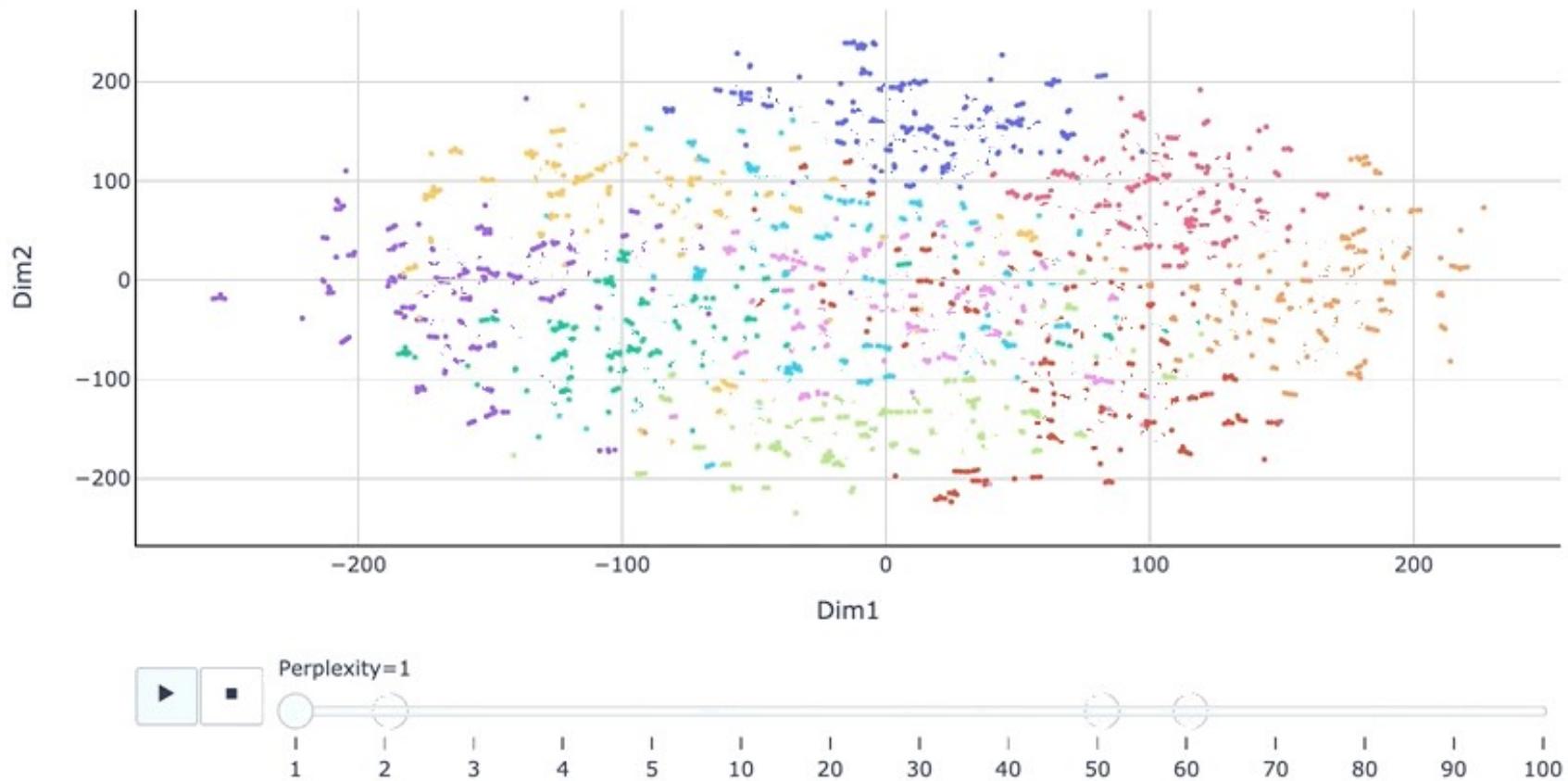
$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)},$$

- В пространстве меньшей размерности для расчета похожести $q_{j|i}$ применяется t-распределение Стьюдента
- Расположение точек в пространстве меньшей размерности итерационно подбирается минимизацией расстояния Кульбака — Лейблера

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

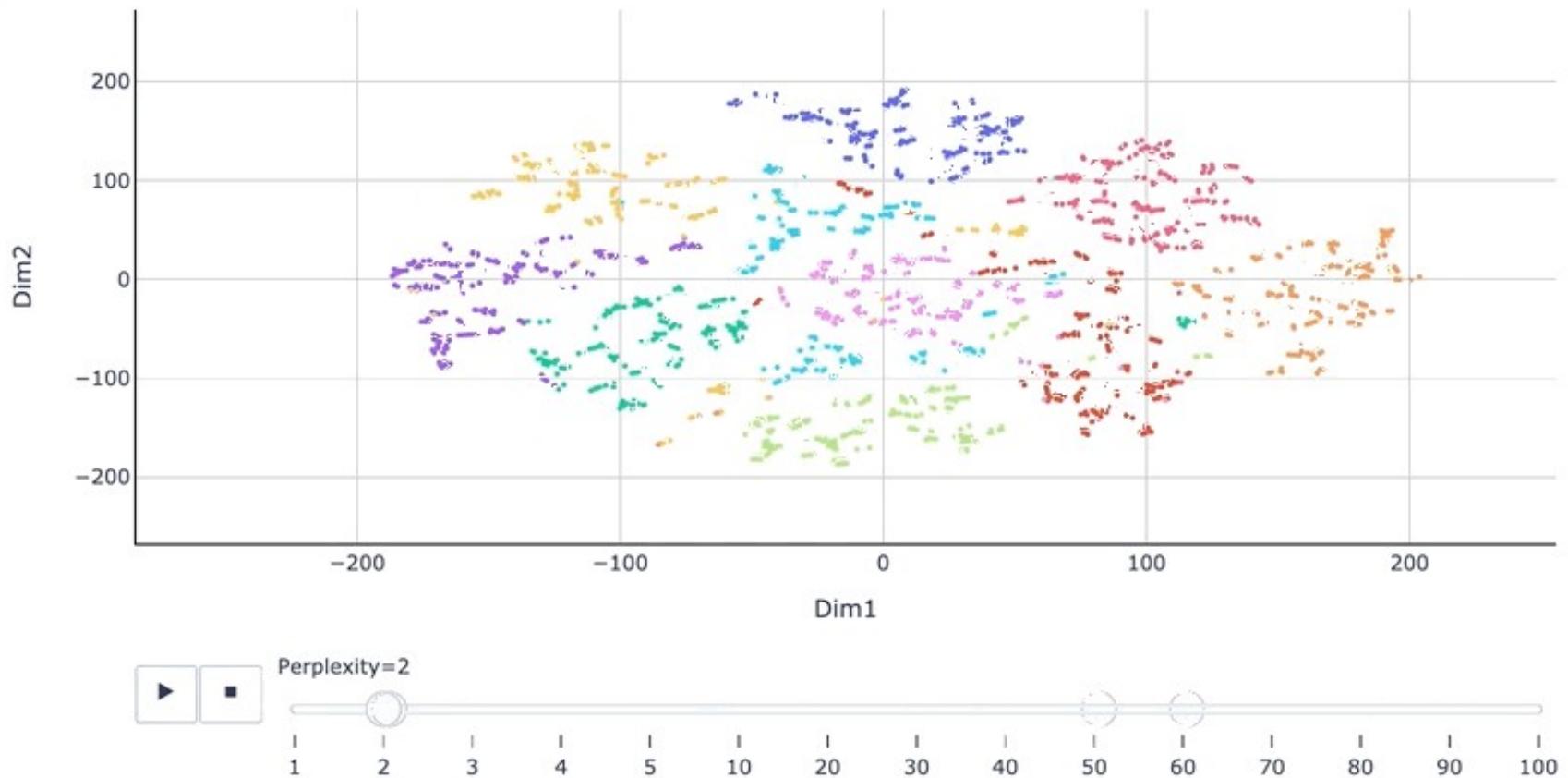
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



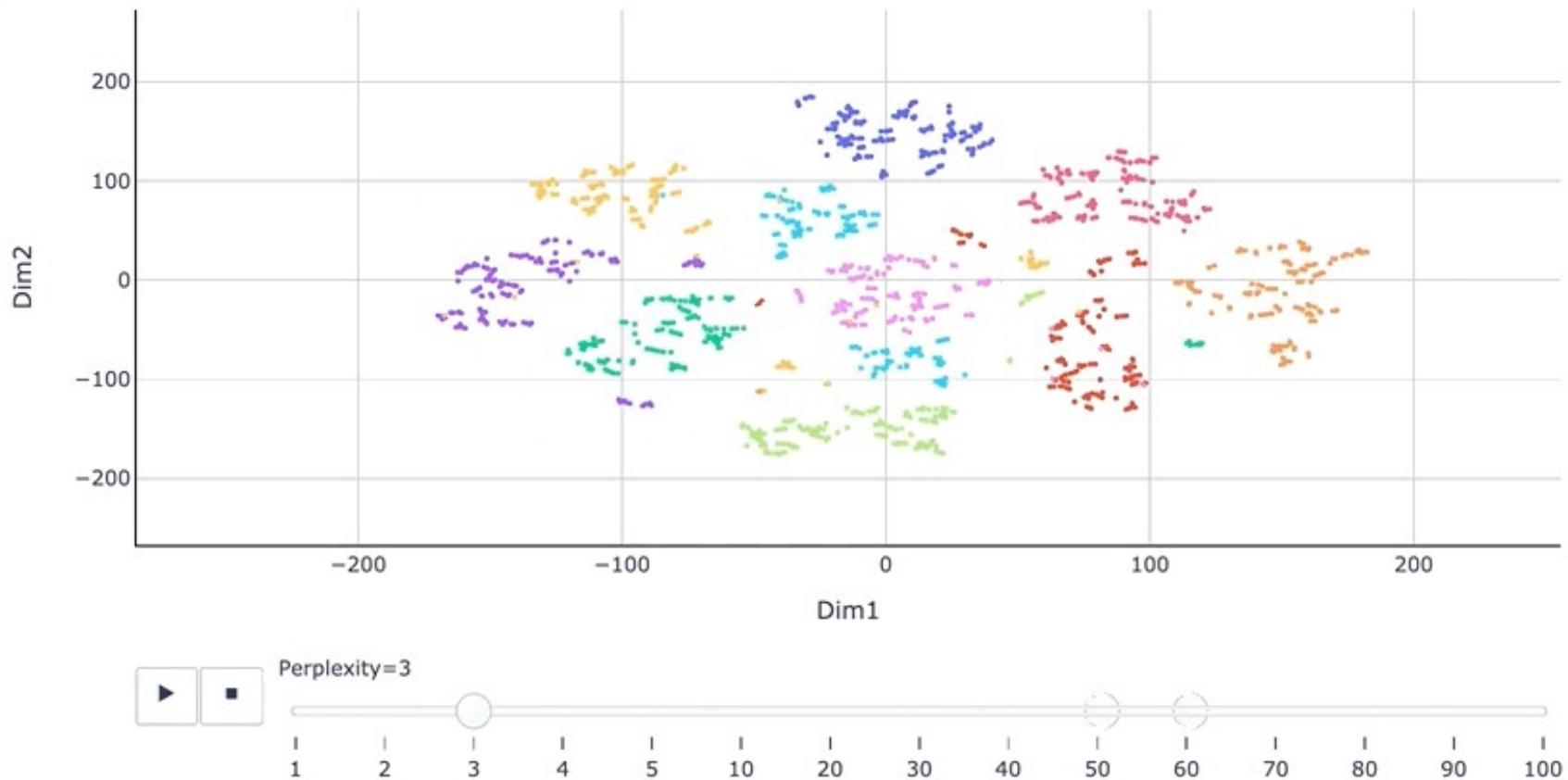
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



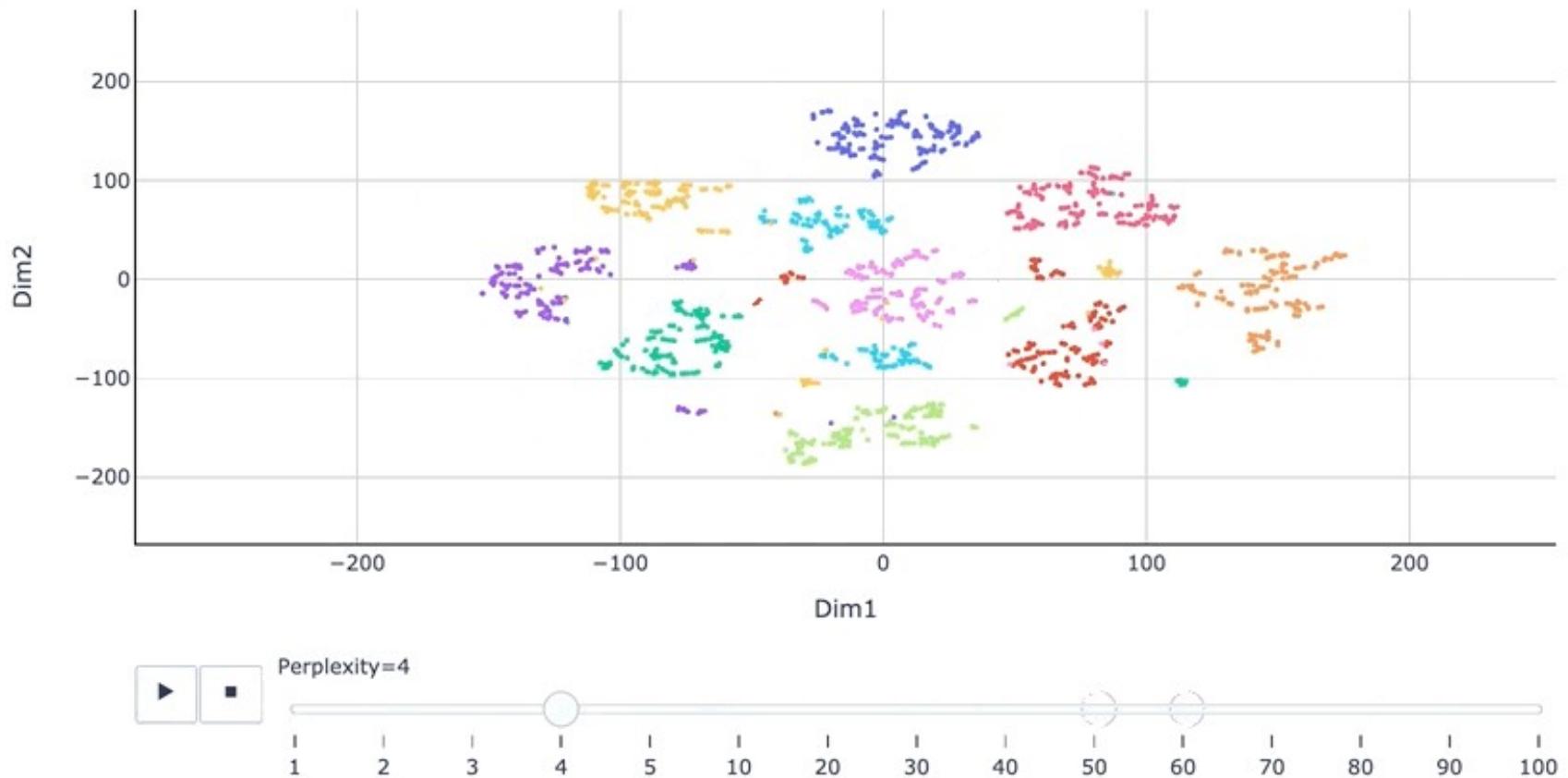
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



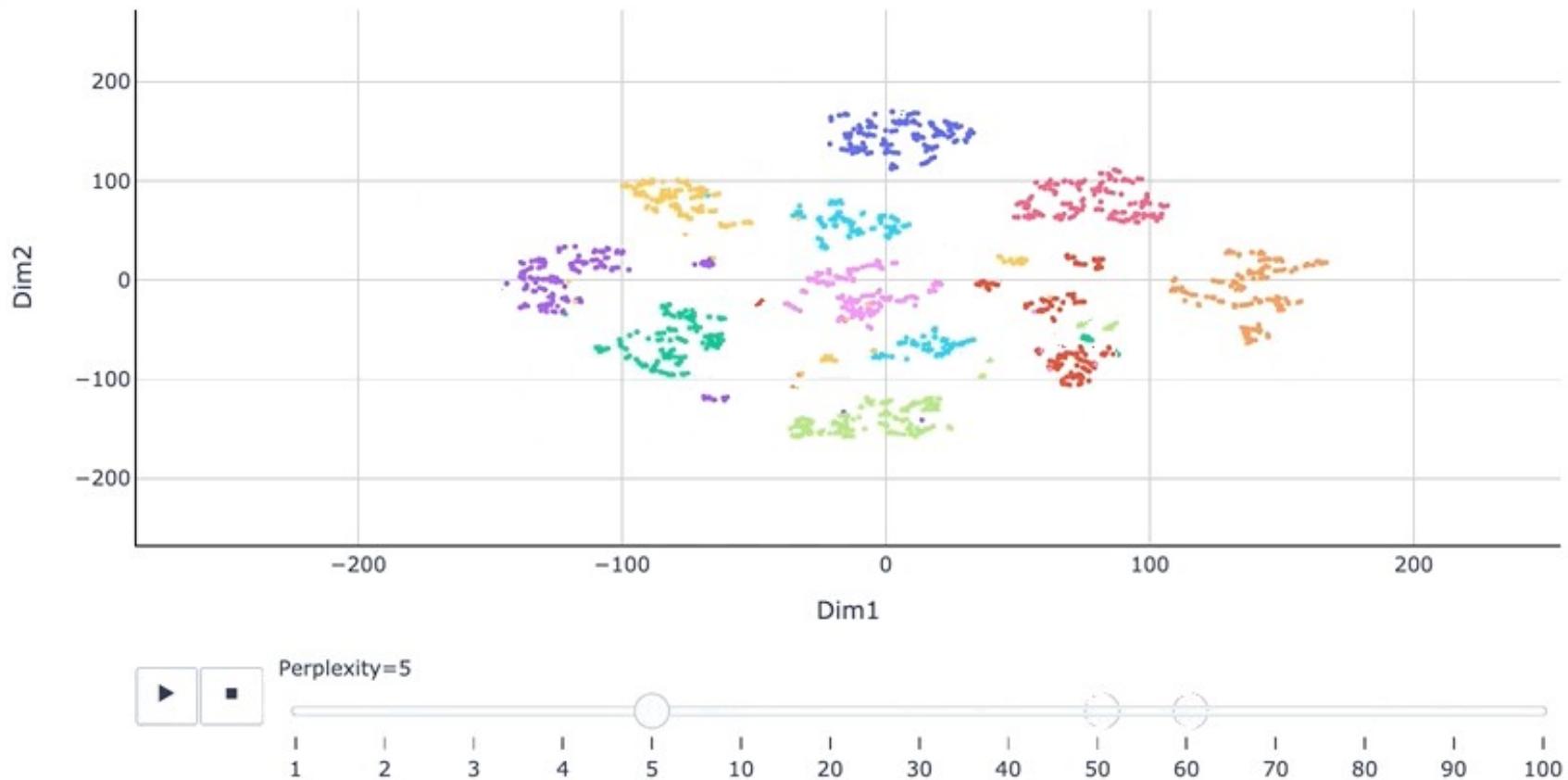
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



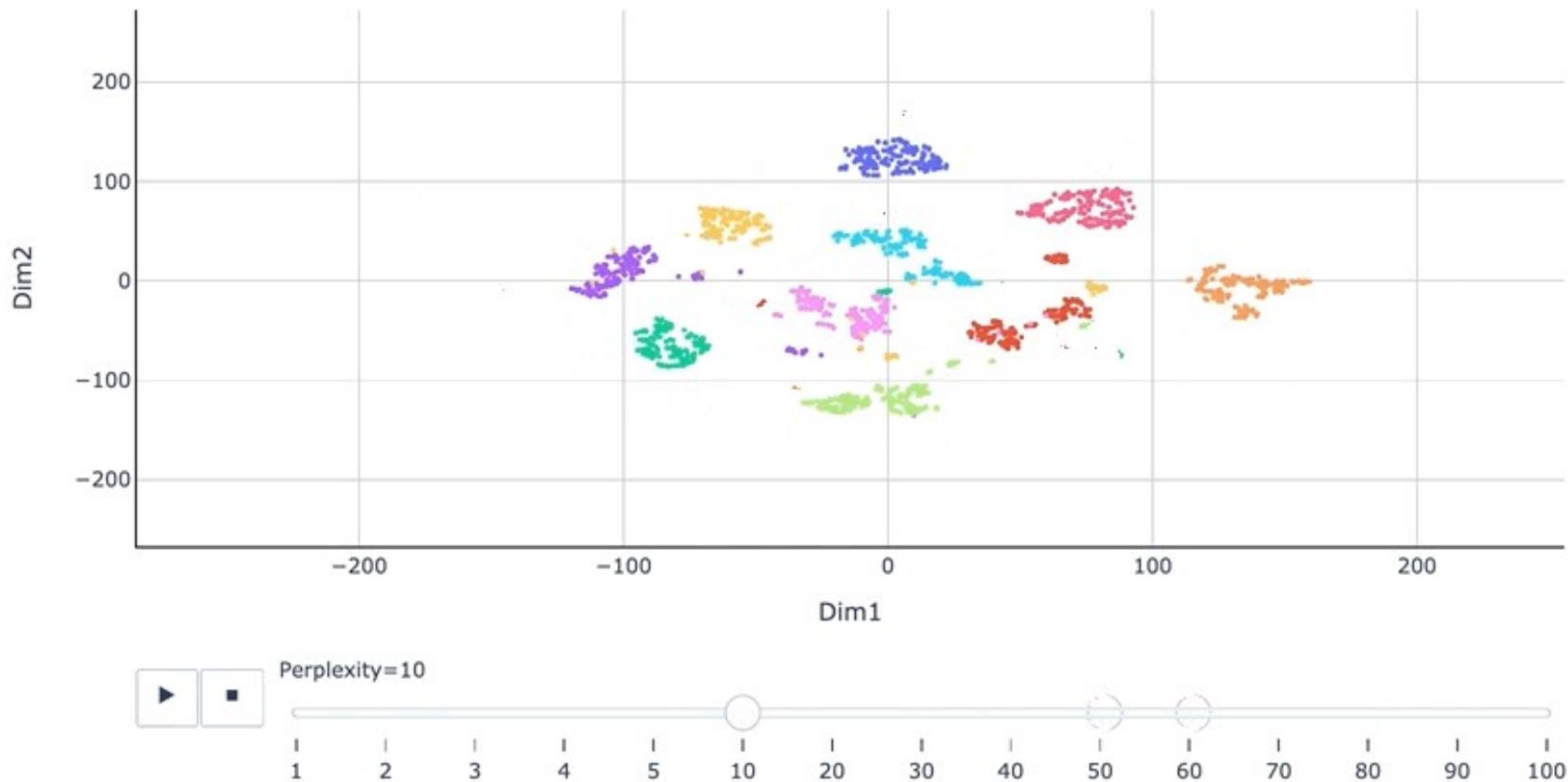
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



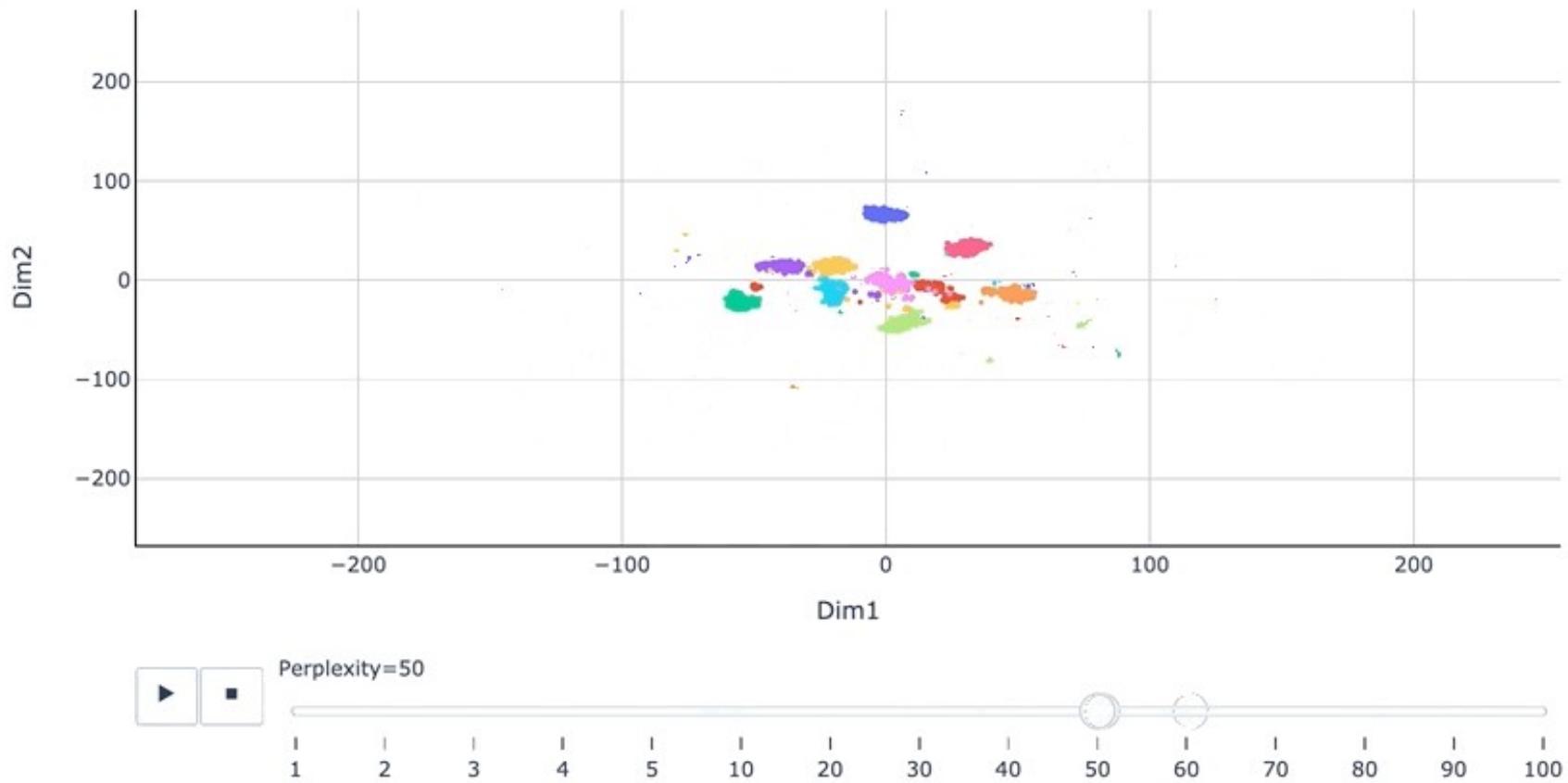
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



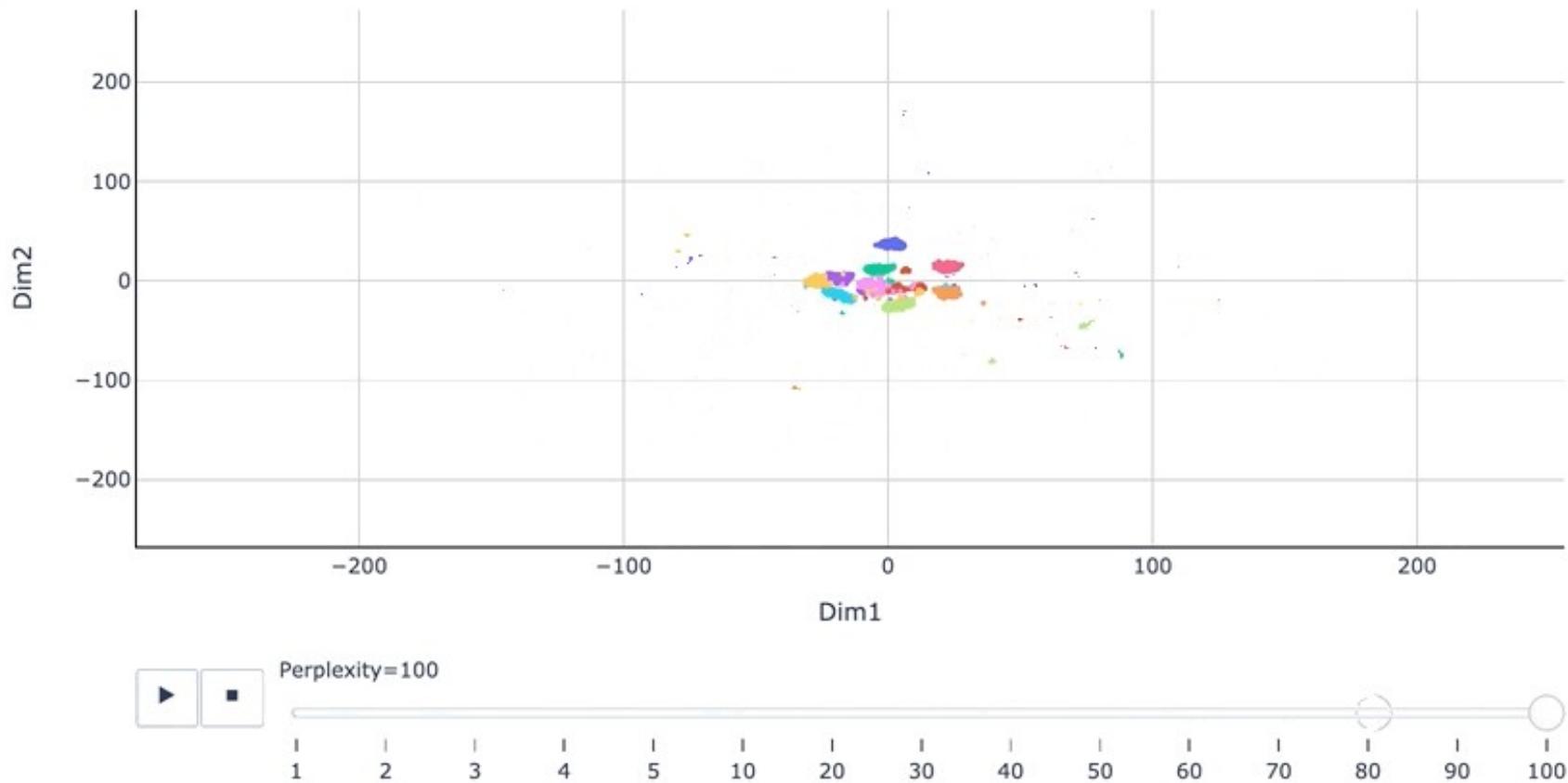
Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



Перплексия в tSNE

- Регулирует ожидаемую плотность вокруг каждой точки или, другими словами, устанавливает соотношение целевого количества ближайших соседей к интересующей точке.



Взаимная информация

Взаимная информация описывает количество информации, содержащееся в одной случайной величине относительно другой.

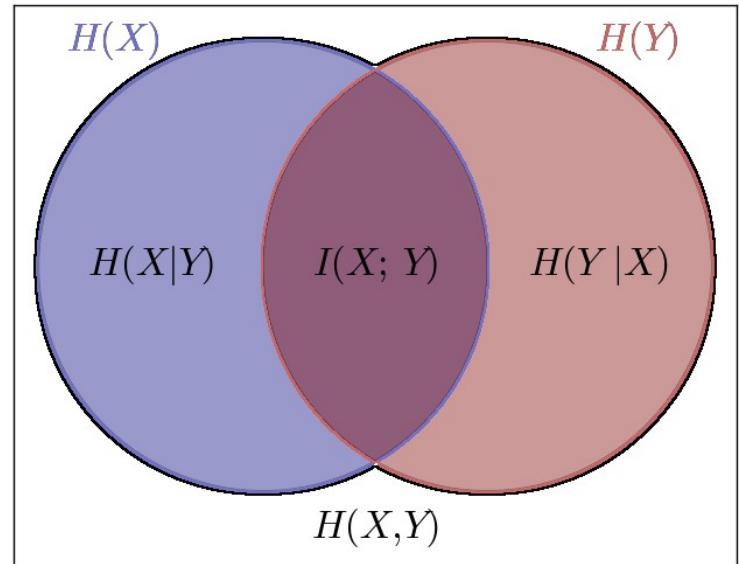
Она определяется через энтропию и условную энтропию двух случайных величин

$$I(X;Y) = H(X) - H(X|Y)$$

$$I(X;Y) = H(Y) - H(Y|X)$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

$$I(X;Y) = \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)} = \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log \frac{p(y_j | x_i)}{p(y_j)}$$



Задачи на экзамен

- Придумайте алгоритм извлечения признаков из геоданных (широта, долгота)
- Извлеките хорошие признаки по наборам взаимодействий пользователей с сайтом мехмата (множество действий и их временных меток)
- По данным из системы БРС в течении семестра предскажите оценки студентов на сессии. Объектом является пара (студент, дисциплина). Данные содержат баллы и временные метки, заносимые преподавателем в БРС в течении семестра
- Придумайте хорошие признаки, описывающие посетителя сайта по данным из HTTP-заголовков User Agent, Referer и IP-адреса

Задачи на экзамен

- Сгенерируйте несколько множеств разного размера из вещественных чисел на интервале $[0;1]$. Будем считать одно множество – характеристикой одного объекта. Придумайте алгоритм извлечения признаков и примените его к сгенерированным множествам