



# Генерация и отбор признаков, методы снижения размерности

МАКСИМОВСКАЯ  
АНАСТАСИЯ

# Снижение размерности

---

# Снижение размерности

---

- Методы отбора признаков отбирают из исходных признаков некоторое подмножество признаков
- Теперь мы хотим придумать новые признаки, каким-то образом выражающиеся через старые, причем новых признаков хочется меньше, чем старых
- Будем рассматривать только случай, когда новые признаки линейно выражаются через старые.

# Метод главных компонент

---

- $f_1(x), \dots, f_n(x)$  – исходные признаки
- $g_1(x), \dots, g_m(x)$  – новые числовые признаки,  $m \leq n$

Мы хотим, чтобы новые числовые признаки  $g_i(x)$  линейно выражались через исходные признаки  $f_i(x)$  при этом чтобы исходные признаки также линейно восстанавливались по новым признакам. При этом мы хотим, чтобы при переходе к новым признакам было потеряно наименьшее количество исходной информации.

# Метод главных компонент

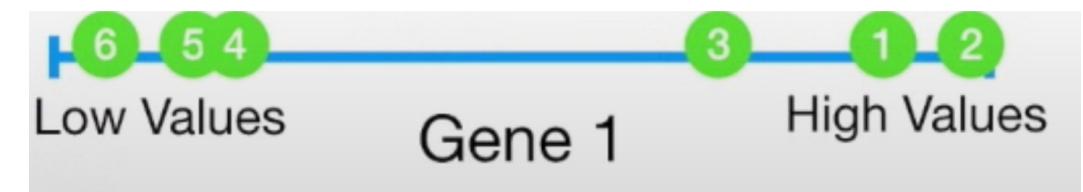
---

- Метод главных компонент работает только с признаками
- Для него не важна целевая переменная (если она есть)
- Таким образом, метод главных компонент - это обучение без учителя

# Метод главных компонент

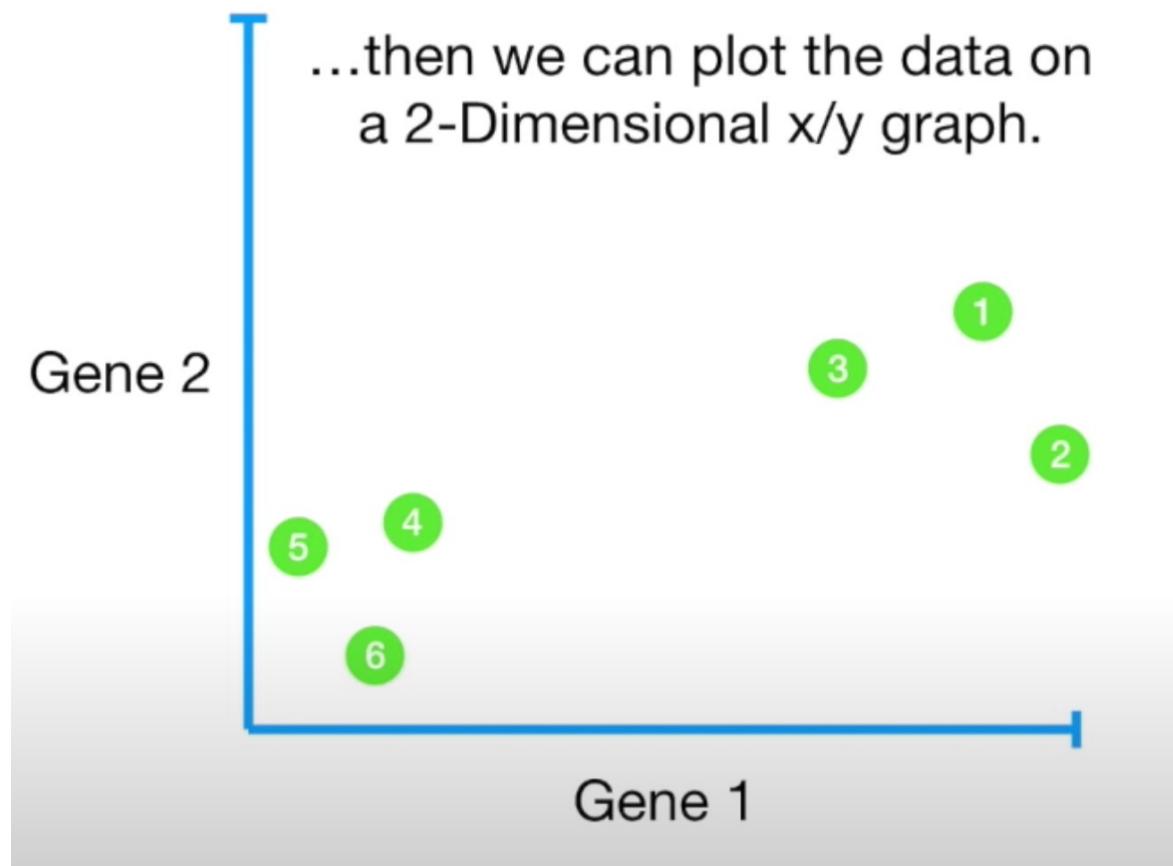
---

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



# Метод главных компонент

---



# Метод главных компонент

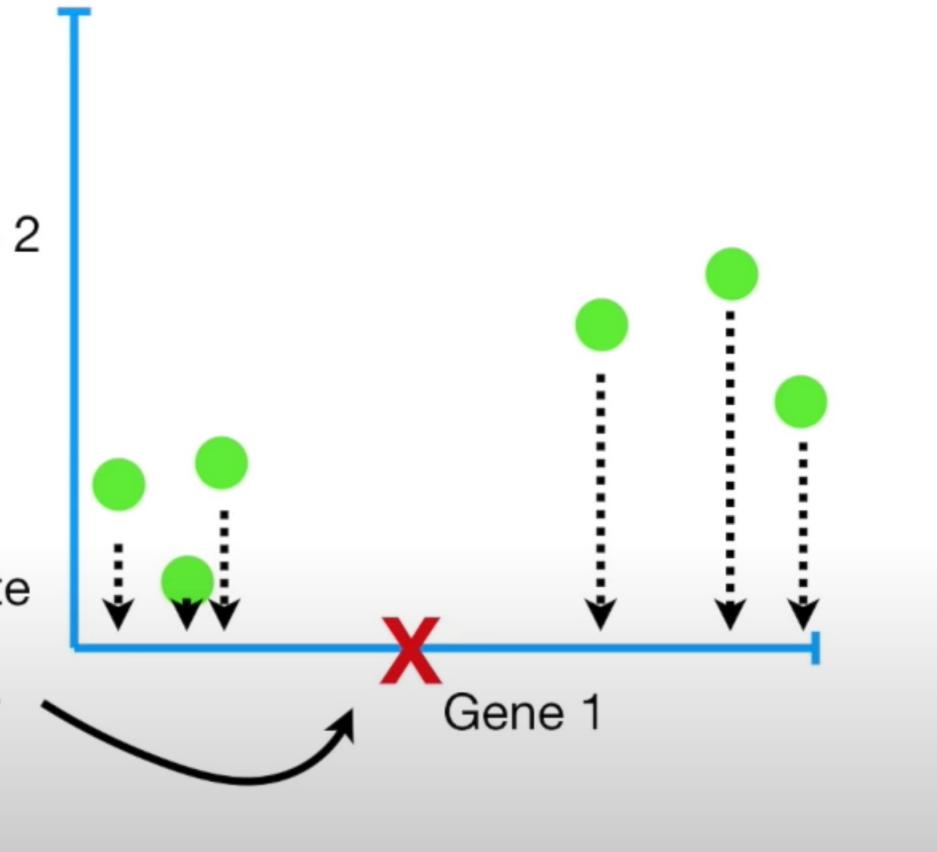
---

- А что, если измерим 4 и более генов?
- С помощью РСА можем изобразить на двухмерном графике и оценить насколько он точен
- А также сможем узнать какая переменная самая важная для разбиения данных на кластеры

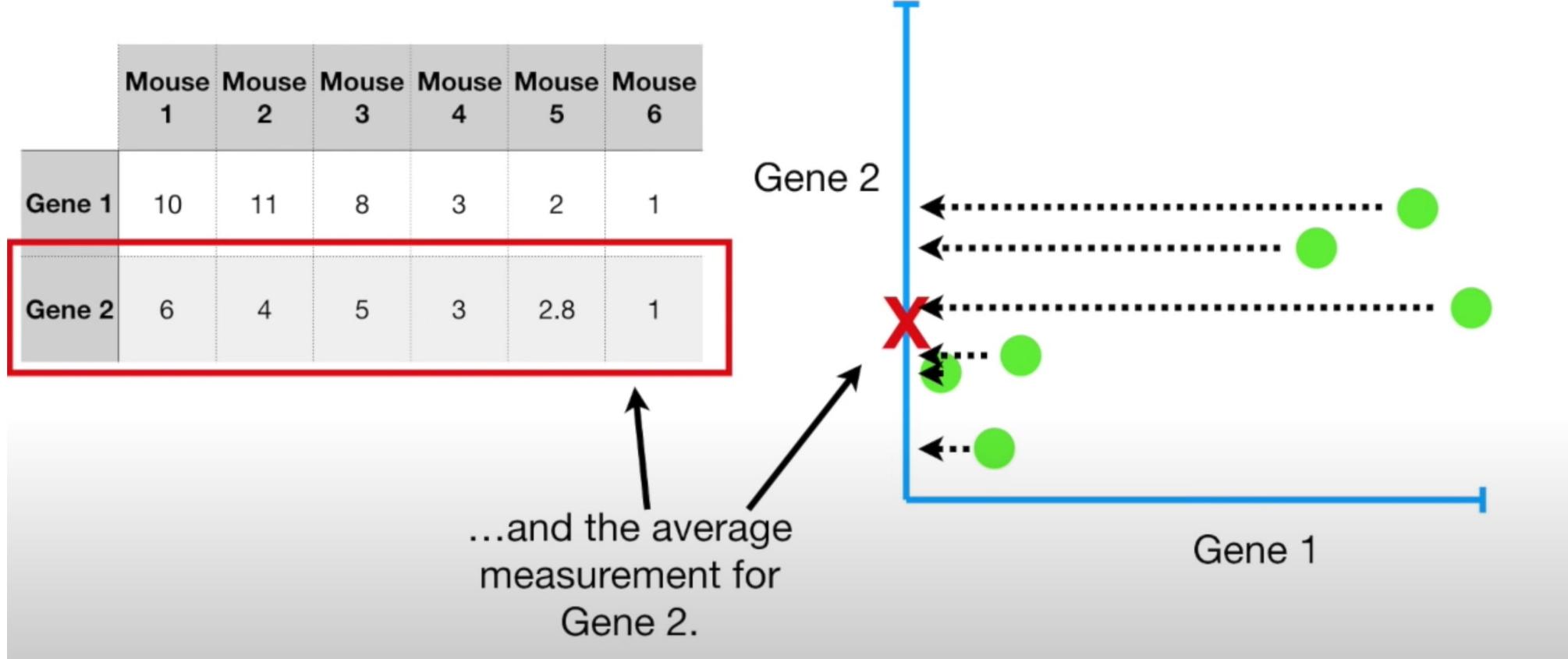
# Метод главных компонент

	Mouse					
	1	2	3	4	5	6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

Then we'll calculate  
the average  
measurement for  
Gene 1...

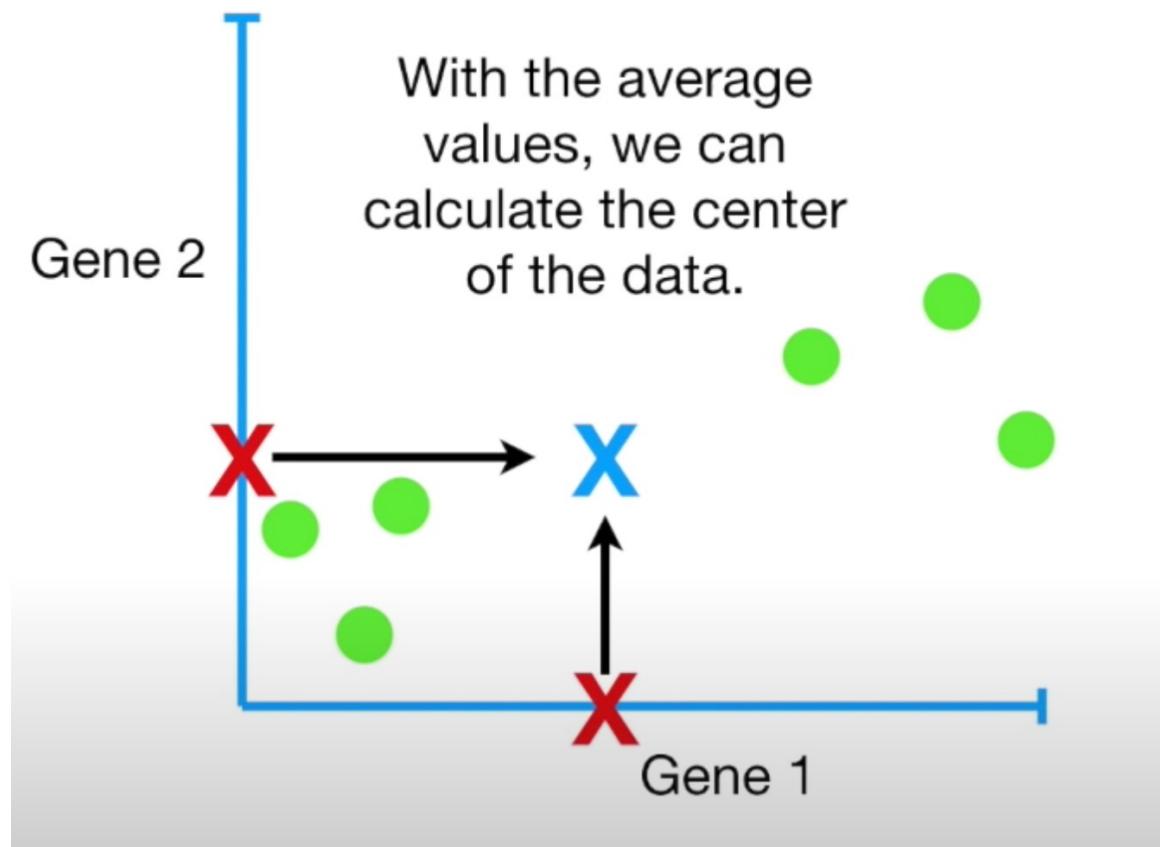


# Метод главных компонент



# Метод главных компонент

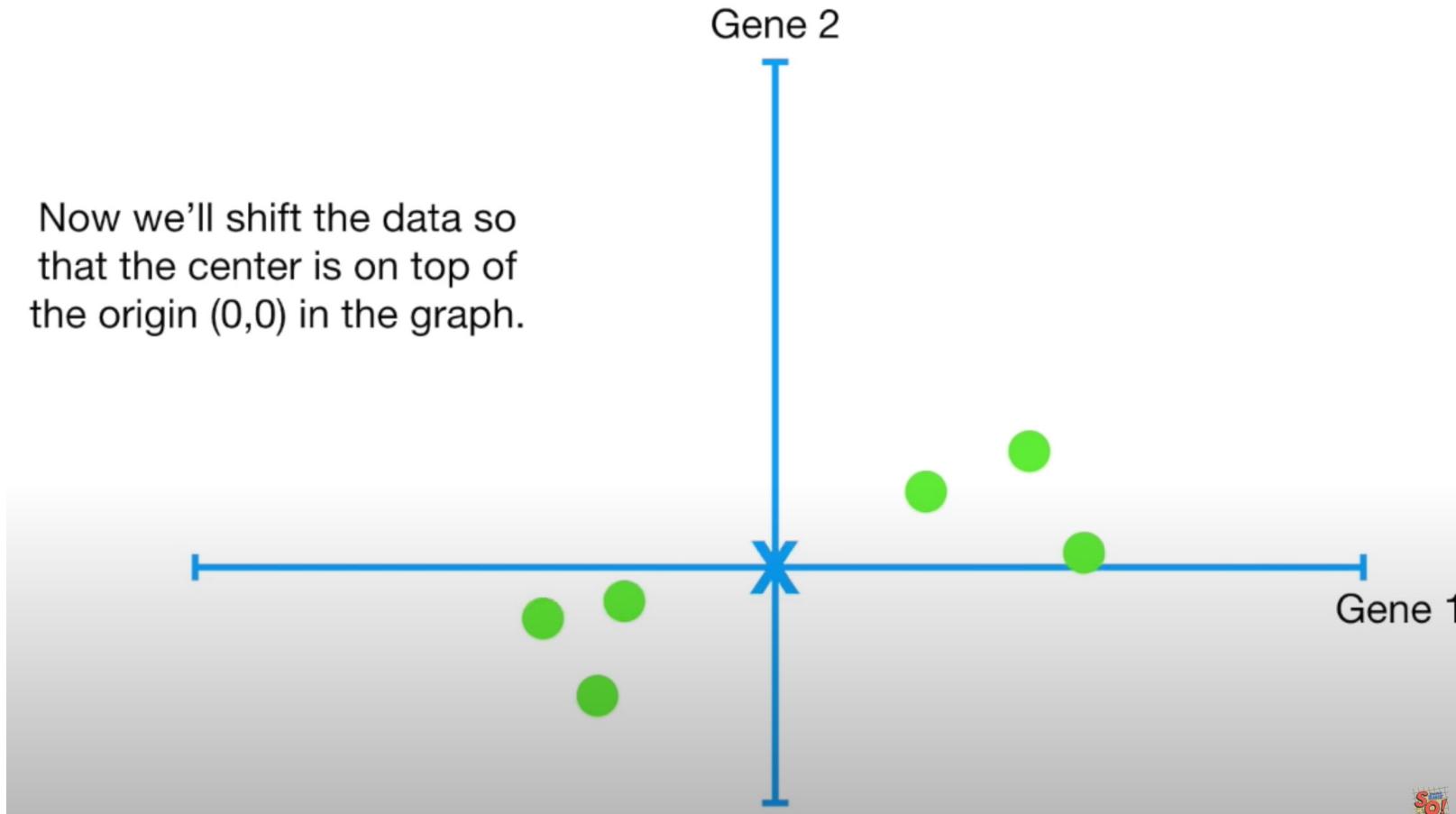
---



# Метод главных компонент

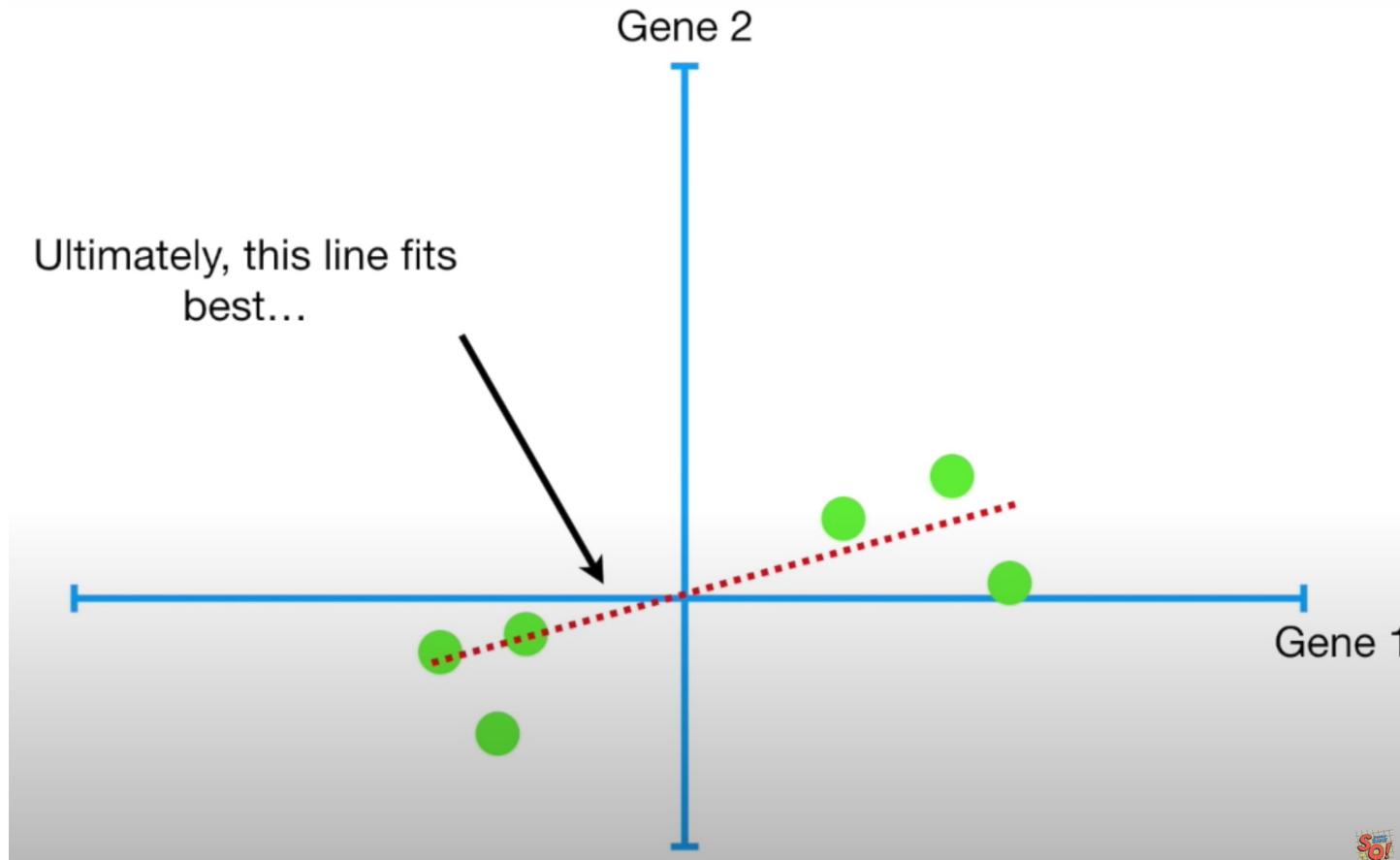
---

Now we'll shift the data so that the center is on top of the origin (0,0) in the graph.



# Метод главных компонент

---



# Метод главных компонент

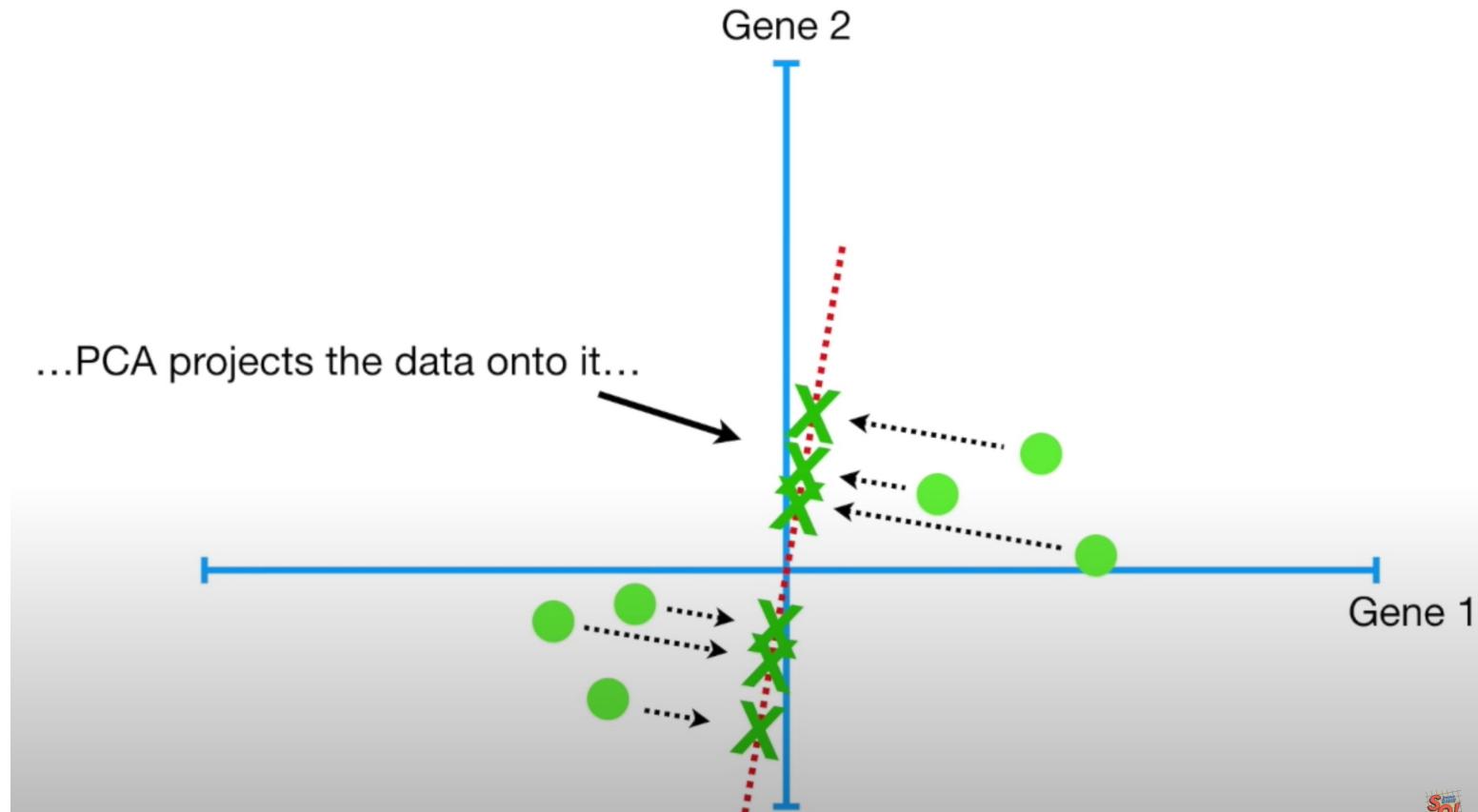
---

Будем искать главные компоненты  $u_1, \dots, u_D$ , которые удовлетворяют следующим требованиям:

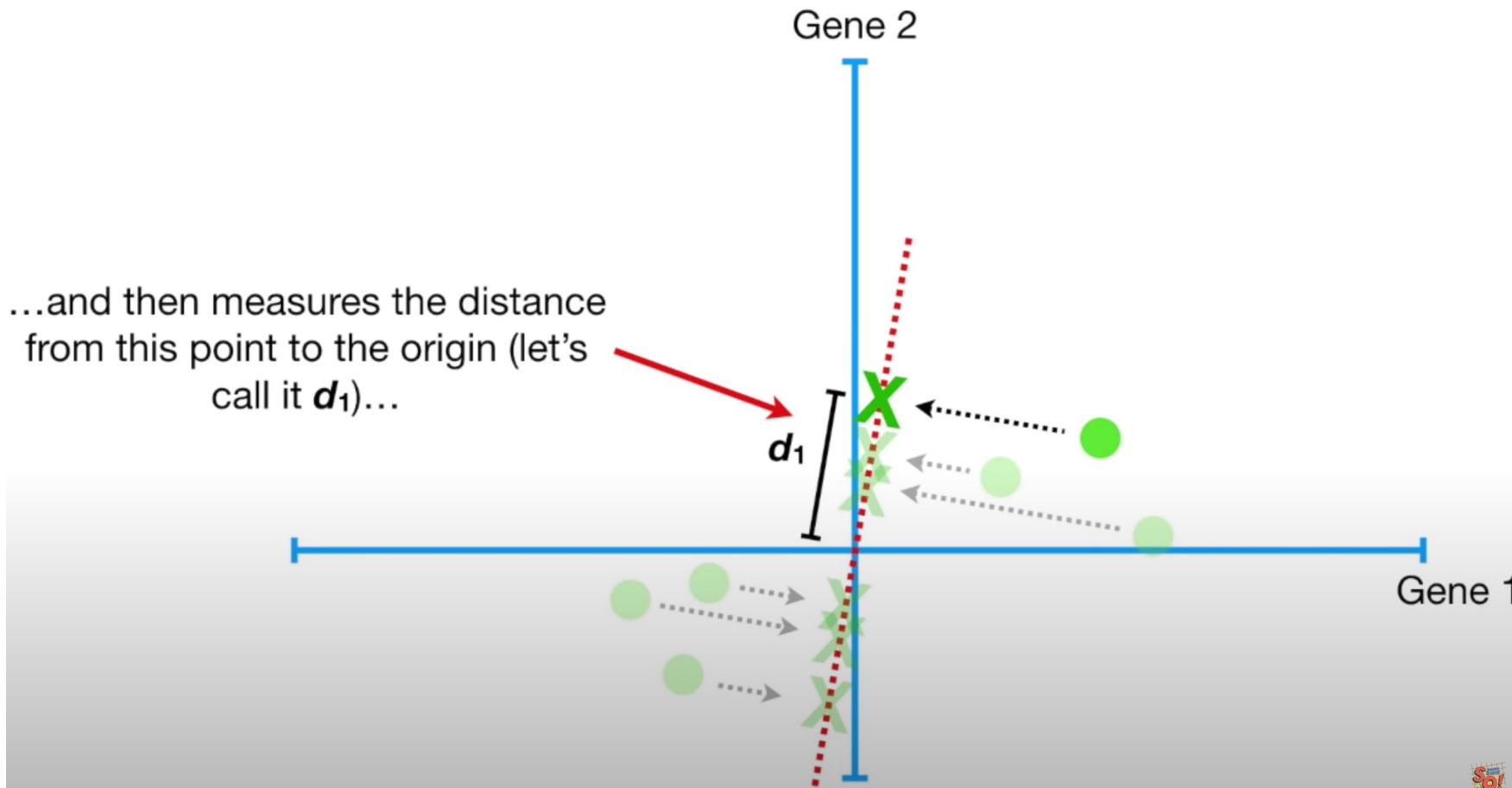
- Они ортогональны:  $\langle u_i, u_j \rangle = 0, i \neq j;$
- Они нормированы:  $\| u_i \| ^2 = 1;$
- При проецировании выборки на компоненты  $u_1, \dots, u_D$  получается максимальная дисперсия среди всех возможных способов выбрать  $d$  компонент.

# Метод главных компонент

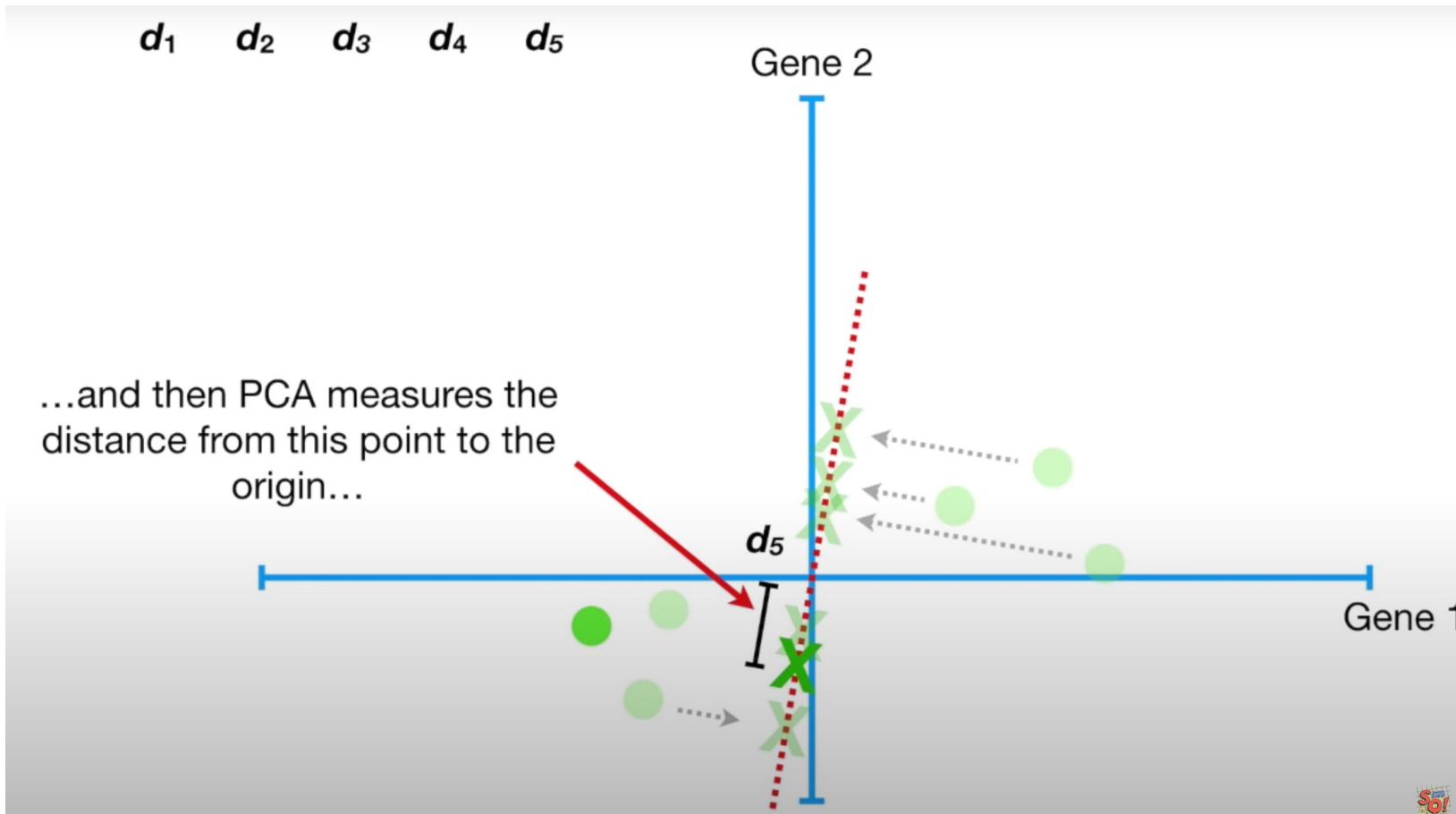
---



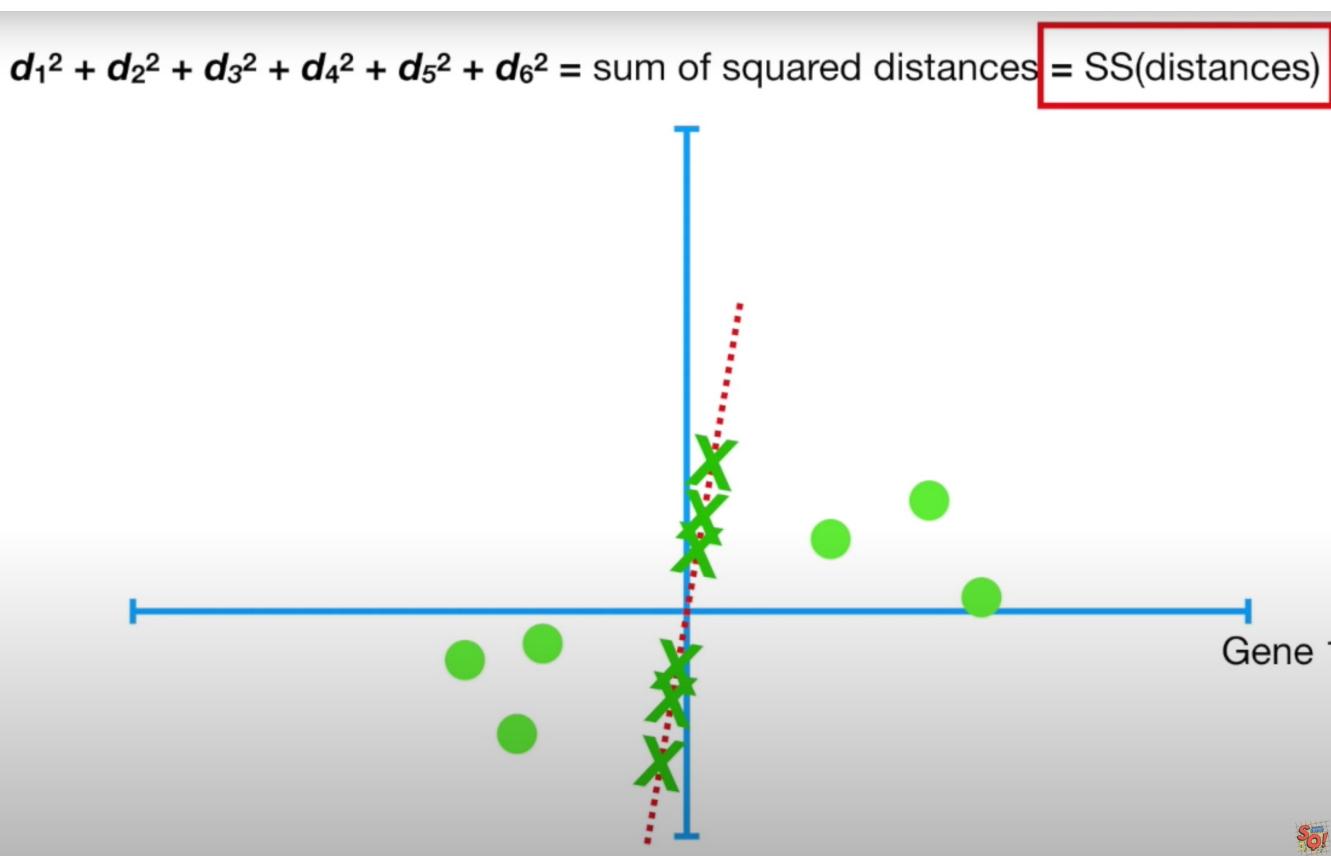
# Метод главных компонент



# Метод главных компонент



# Метод главных компонент

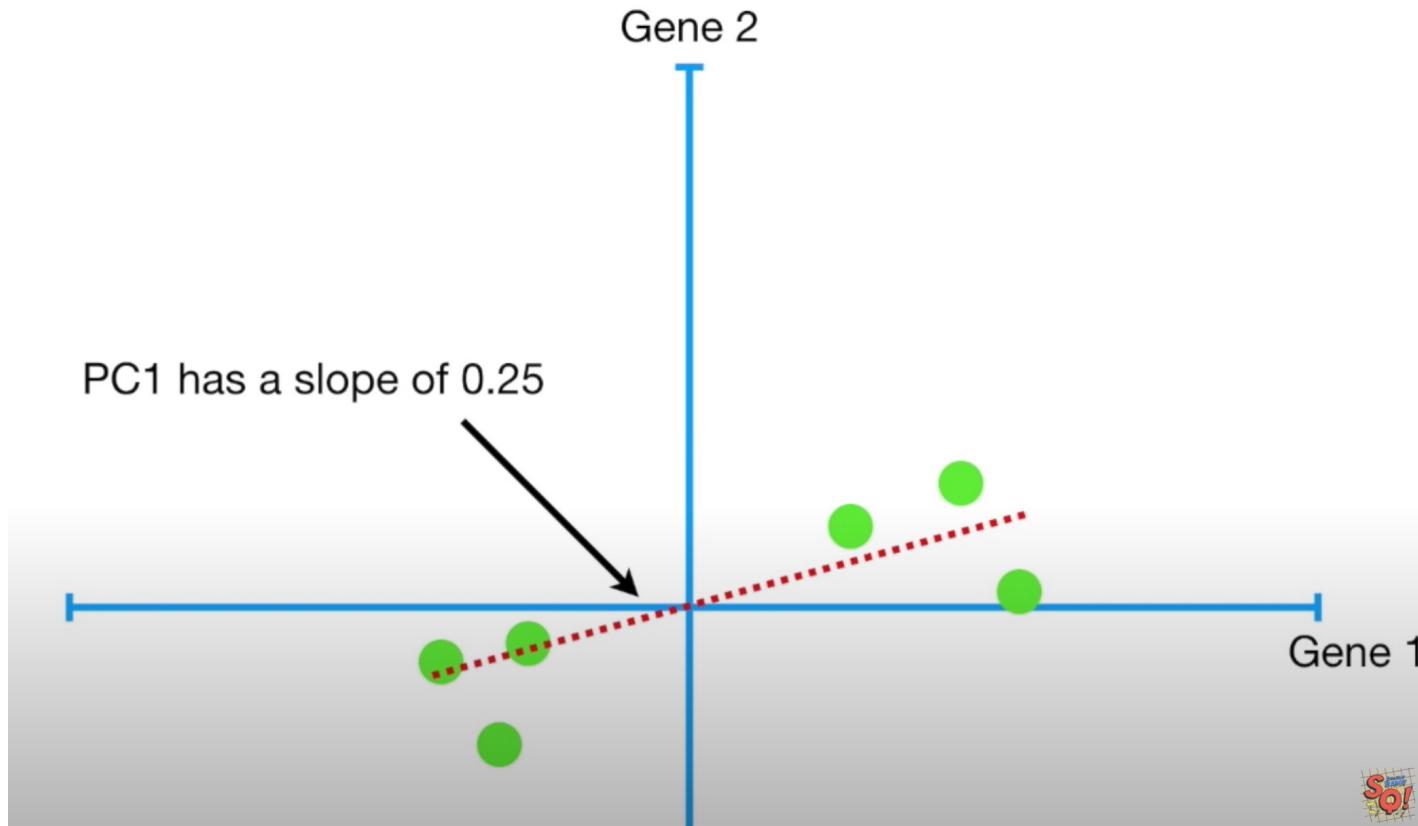


Задача – найти  
максимальную SS

- SS для PC1 –  
собственное  
значение для PC1

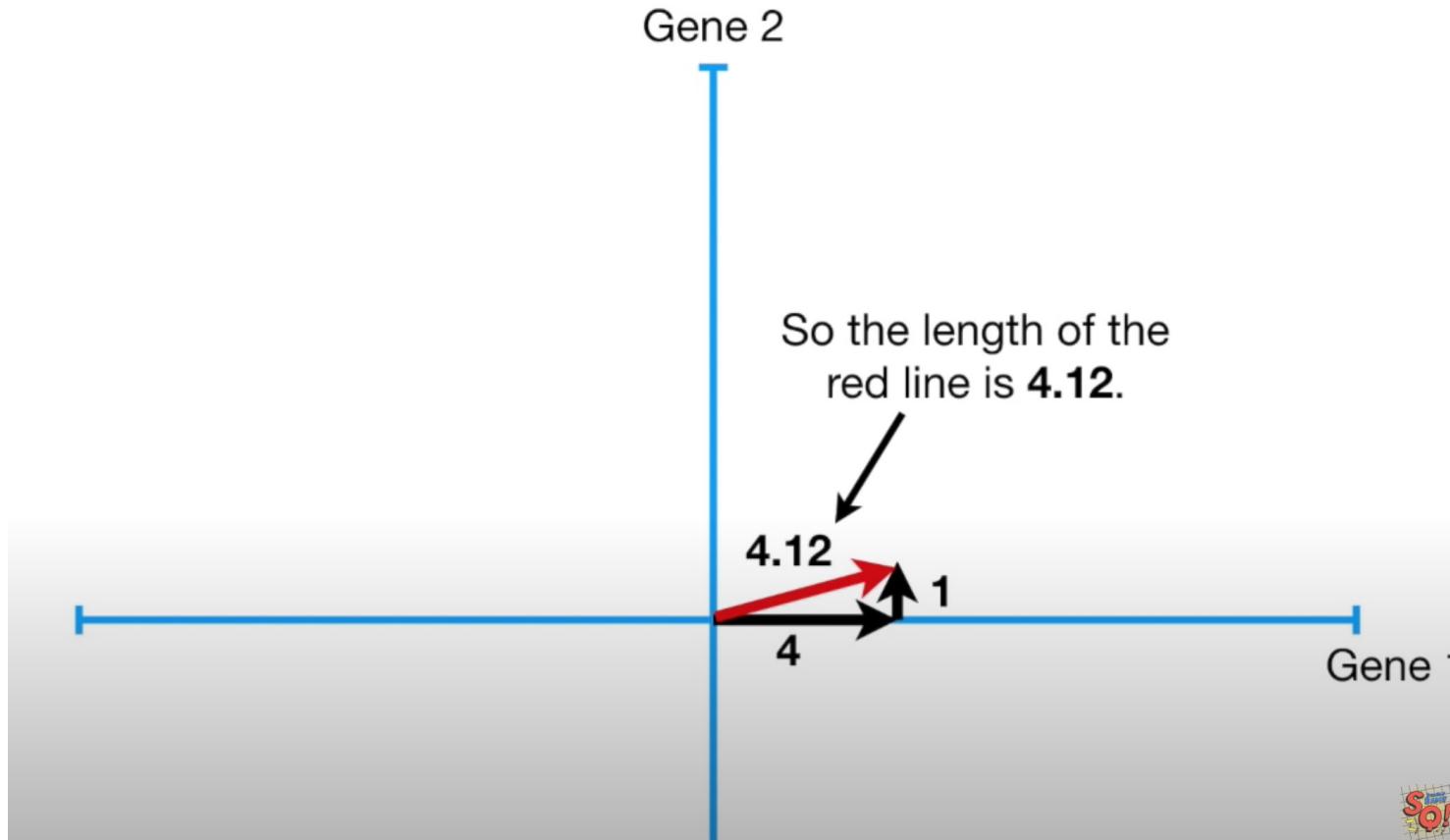
# Метод главных компонент

---



# Метод главных компонент

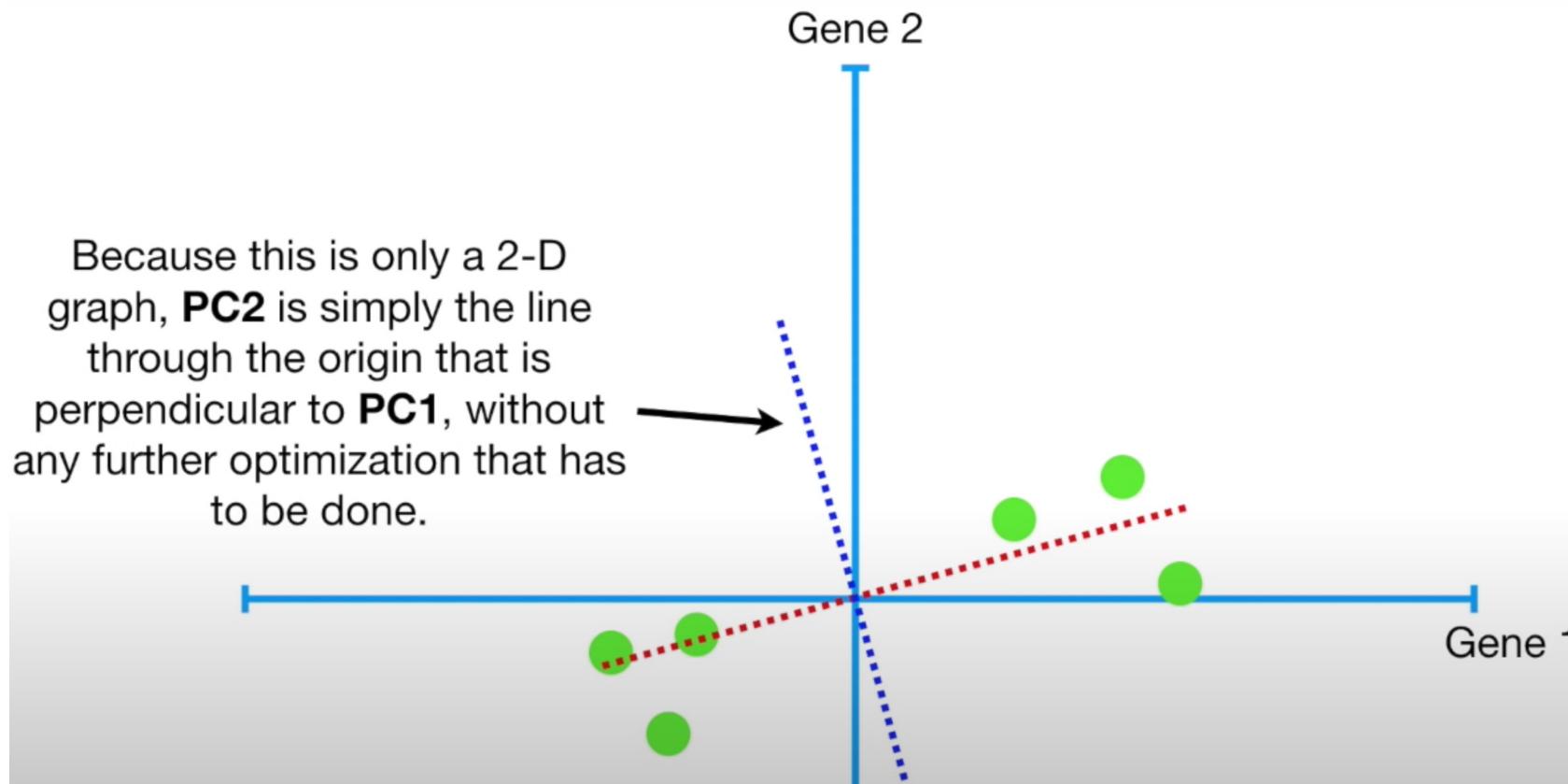
---



Для SVD необходимо, чтобы длина собственного вектора (красной линии) была 1, соответственно, масштабируем остальные стороны, деля на 4.12

# Метод главных компонент

---

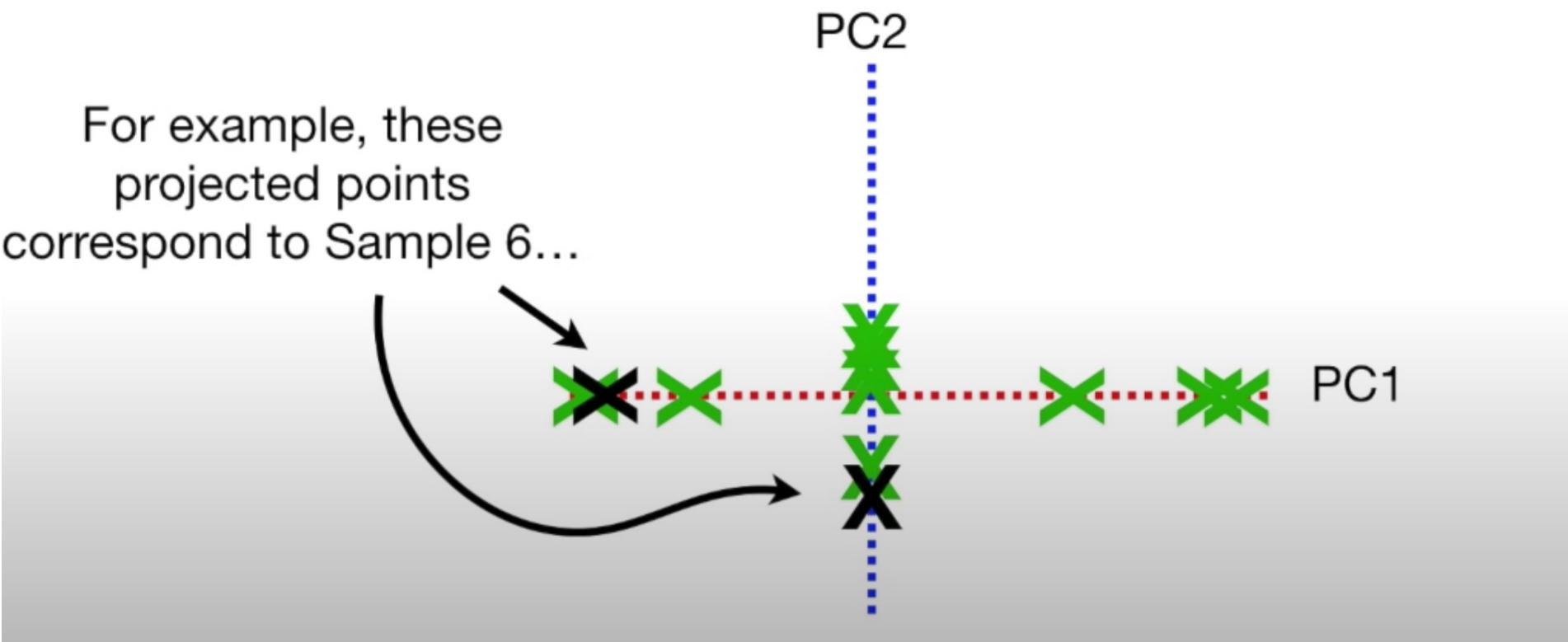


PC3 мы бы искали как линию, перпендикулярную PC1 и PC2

# Метод главных компонент

---

For example, these projected points correspond to Sample 6...



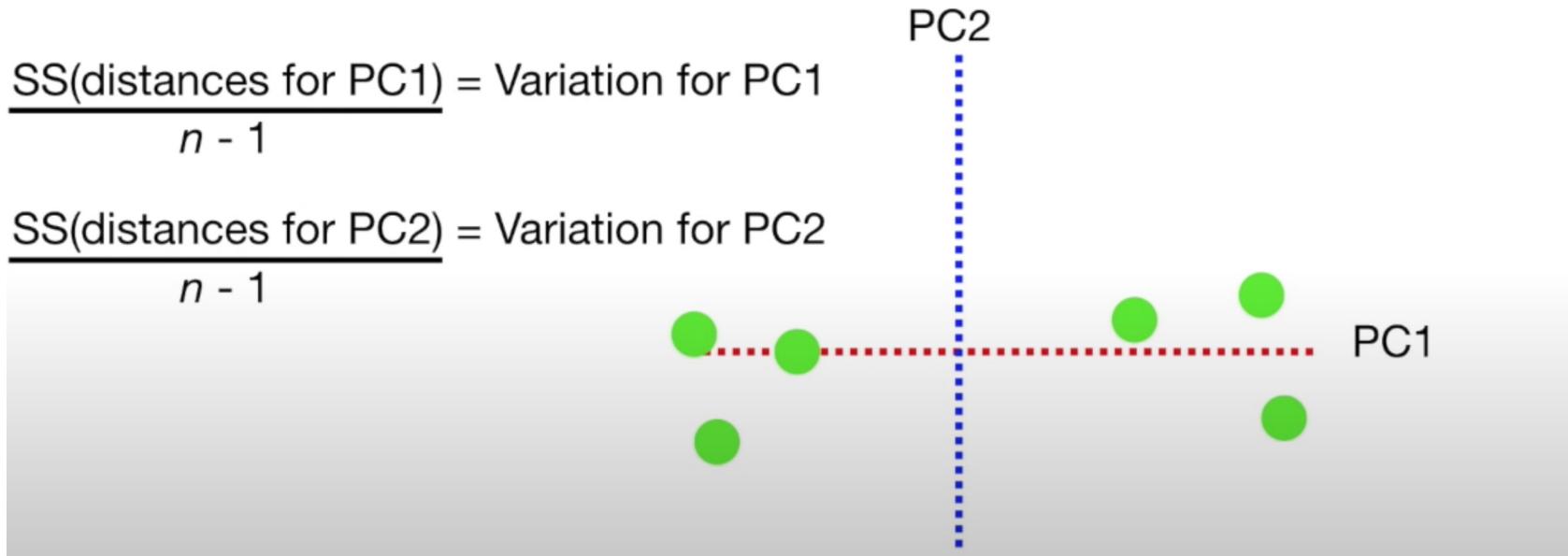
# Метод главных компонент

---

We can convert them into variation around the origin  $(0, 0)$  by dividing by the sample size minus 1 (i.e.  $n - 1$ ).

$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Variation for PC1}$$

$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Variation for PC2}$$



# Метод главных компонент

---

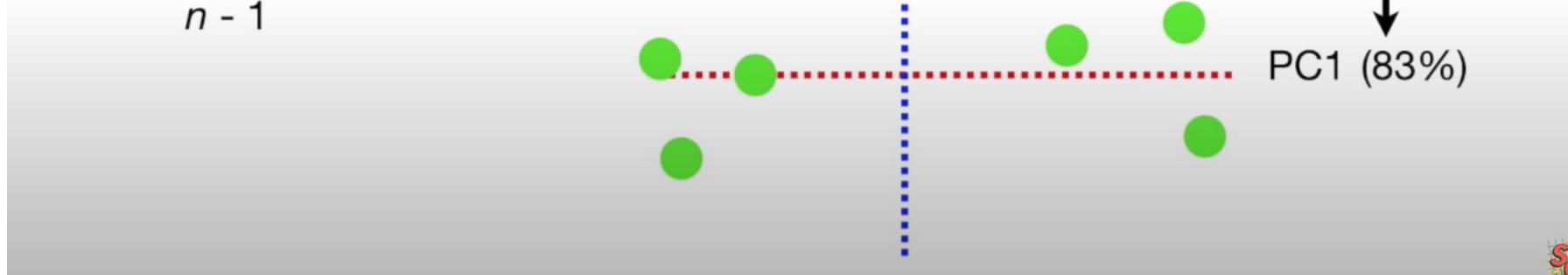
For the sake of the example, imagine that the Variation for **PC1 = 15**, and the variation for **PC2 = 3**.

$$\frac{\text{SS}(\text{distances for PC1})}{n - 1} = \text{Variation for PC1}$$

$$\frac{\text{SS}(\text{distances for PC2})}{n - 1} = \text{Variation for PC2}$$

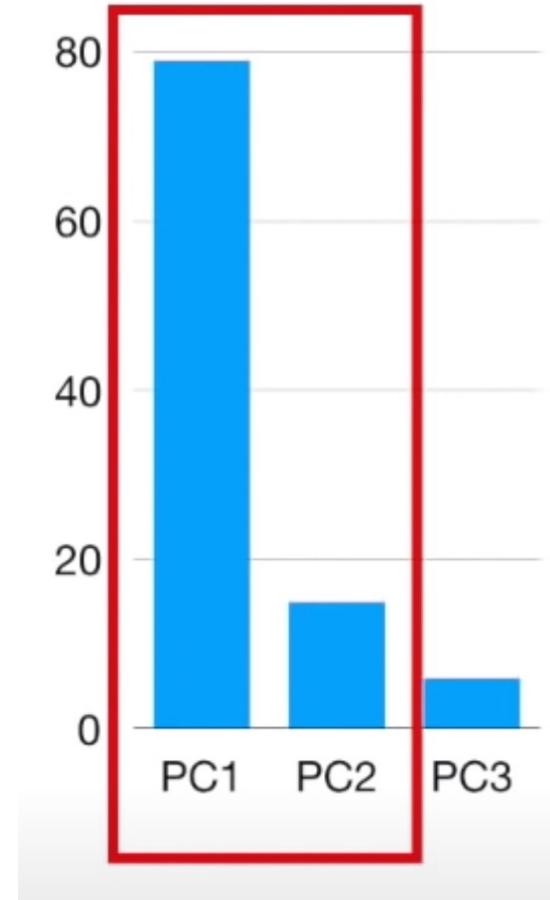
That means that the total variation around both PCs is  **$15 + 3 = 18$** ...

PC2 ...and that means PC1 accounts for  **$15 / 18 = 0.83 = 83\%$**  of the total variation around the PCs.



# Метод главных компонент

---



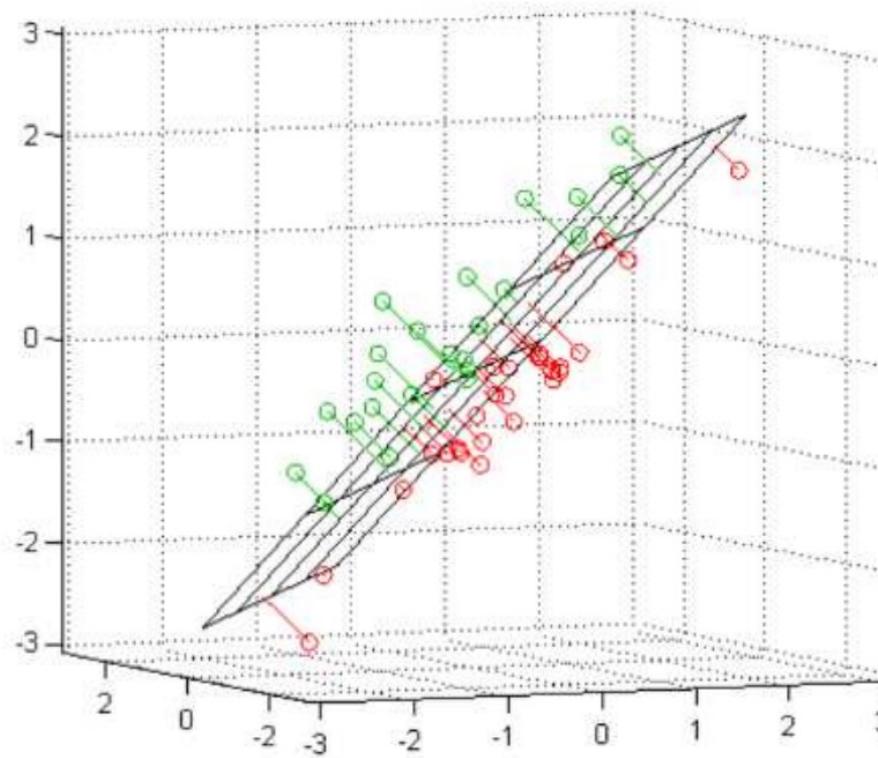
# Геометрическая интерпретация

---

Геометрически метод главных компонент ищет гиперплоскость заданной размерности, при проекции на которую сумма квадратов расстояний от исходных точек будет минимальной.

# Геометрическая интерпретация

---



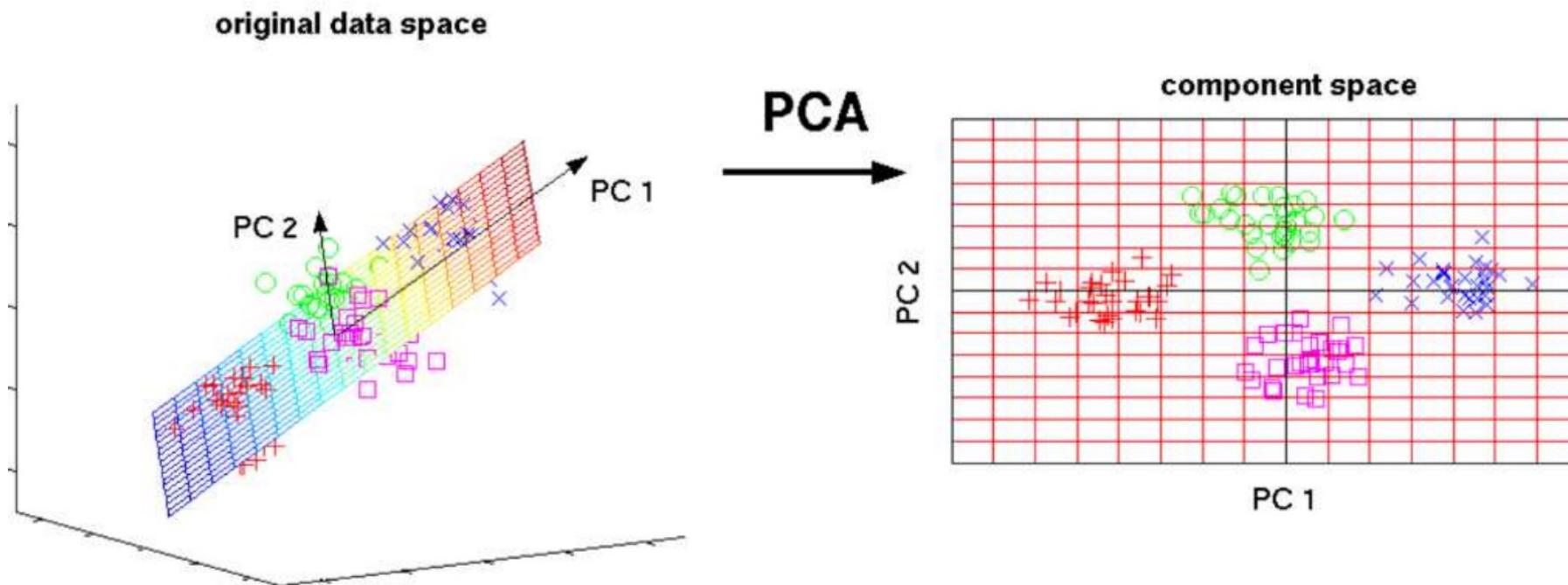
# Визуализация проекции на гиперплоскость

---

- Точки, плохо разделимые в исходном пространстве, могут быть лучше разделимы при проекции на некоторую гиперплоскость

# Визуализация проекции на гиперплоскость

---



# Linear Discriminant Analisys, LDA

---

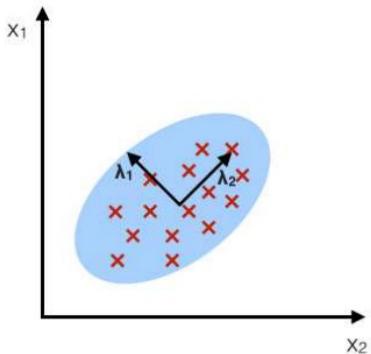
- LDA - это обучение с учителем
- При помощи метода линейного дискриминантного анализа выбирается проекция исходного пространства признаков на новое пространство признаков таким образом, чтобы минимизировать внутриклассовый разброс точек и максимизировать межклассовое расстояние в пространстве признаков
- Главное отличие от PCA - LDA фокусируется на максимальном разделении по категориям

# LDA vs PCA

---

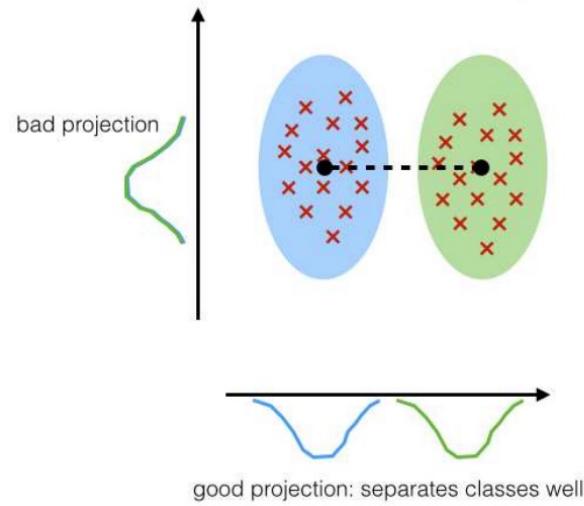
## PCA:

component axes that  
maximize the variance



## LDA:

maximizing the component  
axes for class-separation



# LDA

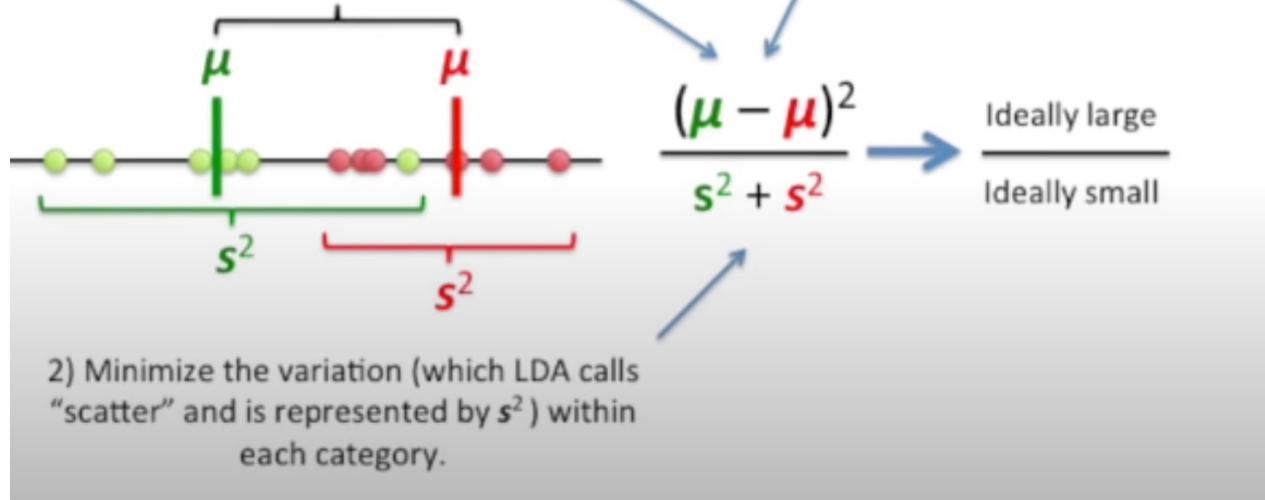
---

## How LDA creates a new axis...

The new axis is created according to two criteria (considered simultaneously):

- 1) Maximize the distance between means.

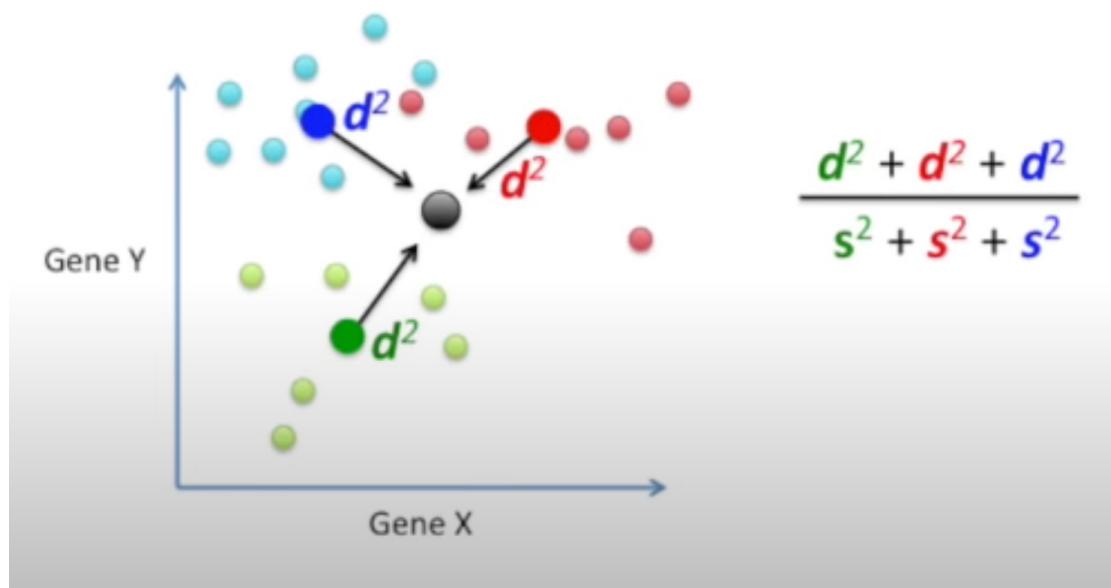
Let's call  $(\mu - \mu)$  *d* for *distance*.



# LDA

---

LDA for 3 categories



# LDA

---

**The second difference** is LDA creates 2 axes to separate the data.

This is because the 3 central points for each category define a plane.

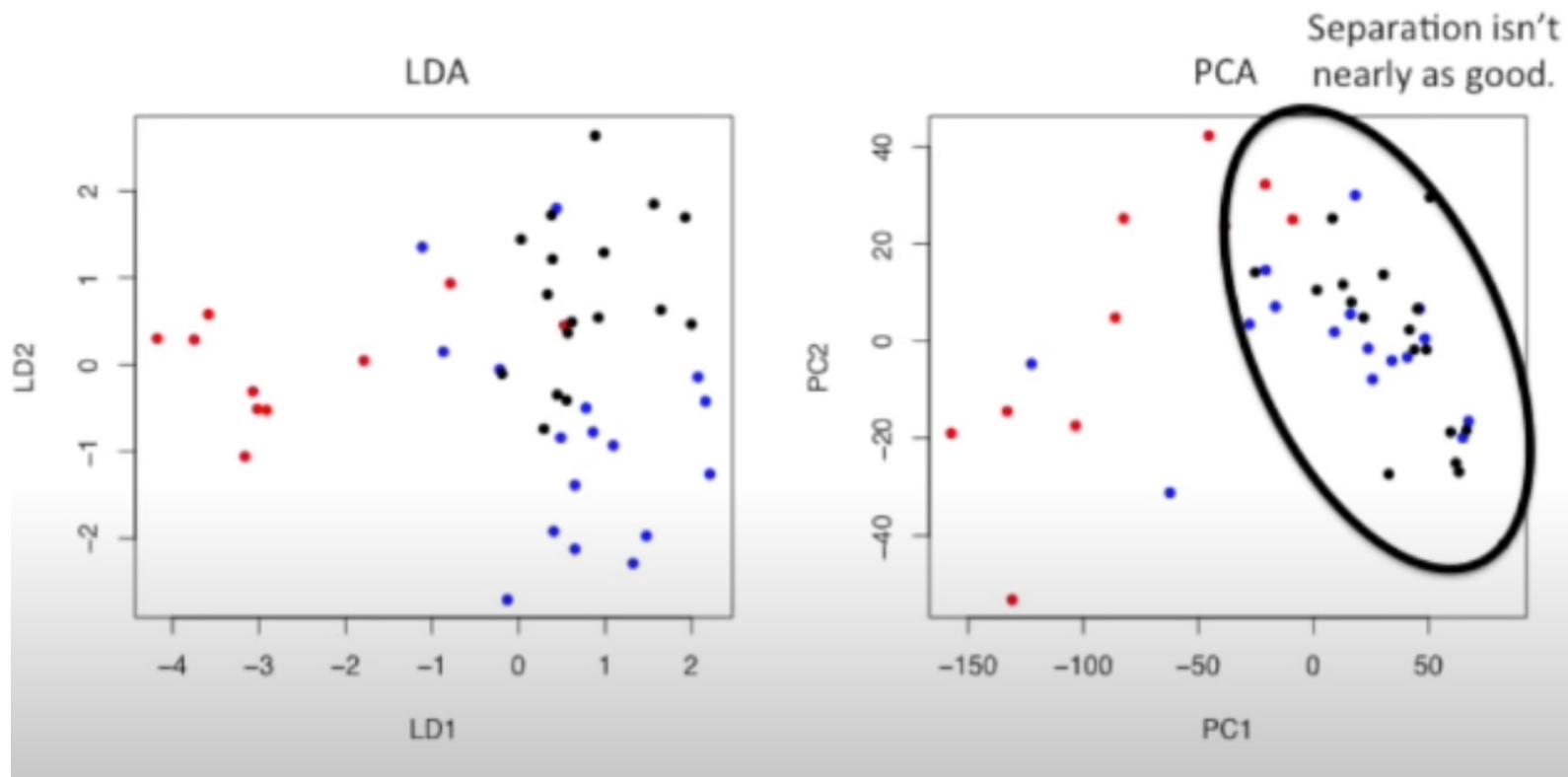
(Remember from high school: 2 points define a line, 3 points define a plane...)



# LDA vs PCA

---

Comparing LDA to PCA with 10,000 genes.



# Визуализация

---

- Задача визуализации состоит в отображении объектов в 2х- или 3хмерное пространство с сохранением отношений между ними.

# MULTIDIMENSIONAL SCALING (MDS)

---

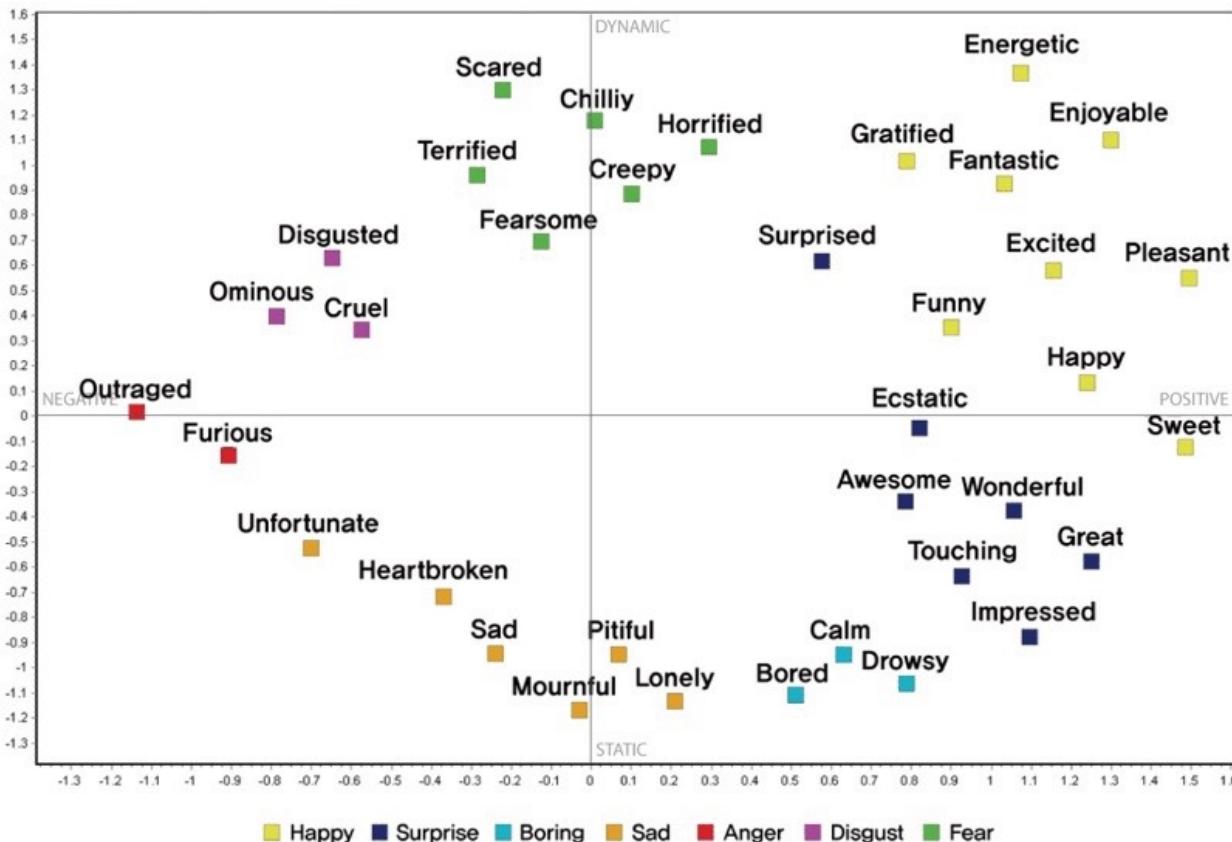
Идея метода – минимизация квадратов отклонений между исходными и новыми попарными расстояниями:

$$\sum_{i \neq j}^l (\rho(x_i, x_j) - \rho(z_i, z_j))^2 \rightarrow \min_{z_1, \dots, Z_l}$$

Снижаем размерность, сохраняя похожесть между наблюдениями.

# MDS

---



# t-SNE

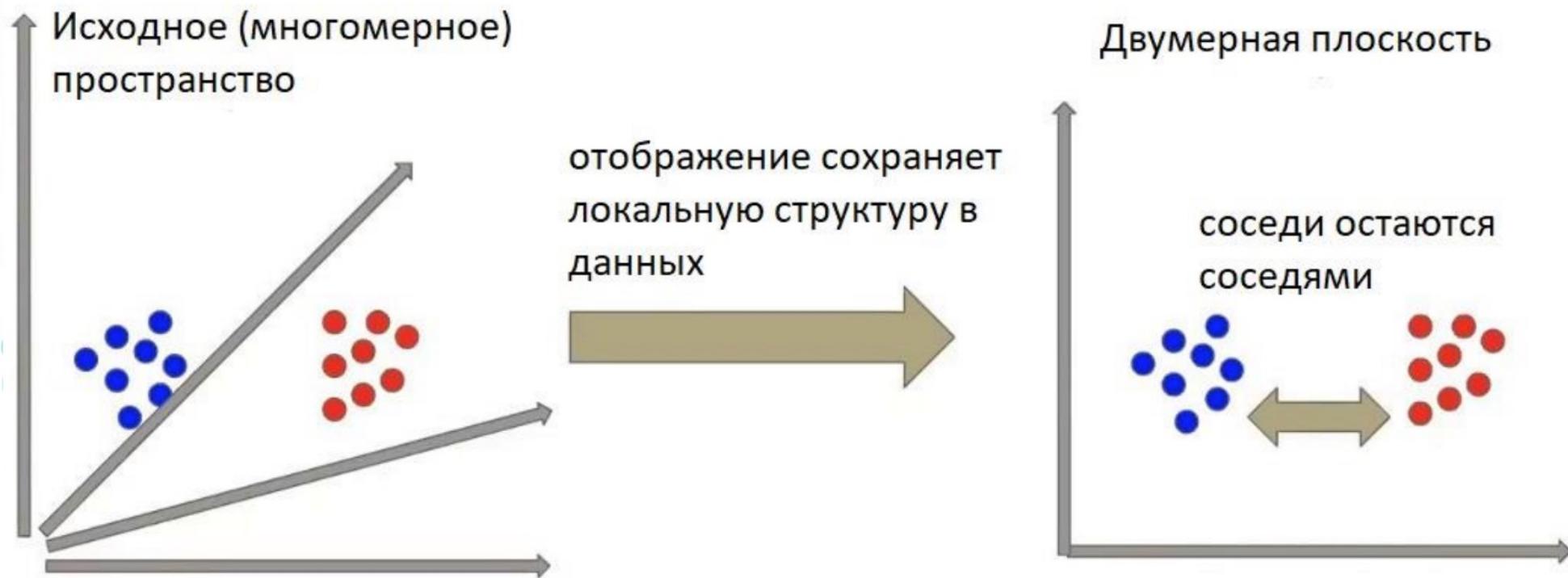
---

- t-SNE – t-distributed stochastic neighbor embedding
- При проекции нам важно не сохранение расстояний между объектами, а сохранение пропорций:

$$\rho(x_1, x_2) = \alpha \rho(x_1, x_3) \Rightarrow \rho(z_1, z_2) = \alpha \rho(z_1, z_3)$$

# t-SNE

---



# t-SNE

---

- Не метод многомерного шкалирования – полученные расстояния не будут соотноситься с исходными
- Пытаемся перенести «окрестность» точек из исходного пространства в пространство меньшей размерности

# t-SNE

---

Будем использовать нормальную плотность для измерения сходства объектов в исходном пространстве:

$$\rho(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Отнормируем эти близости так, чтобы получить вектор распределений расстояний от объекта  $x_j$  до всех остальных объектов:

$$p(i | j) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_j^2)}{\sum_{k \neq j} \exp(-\|x_k - x_j\|^2/2\sigma_j^2)}$$

# t-SNE

---

$$p(i | j) = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_j^2)}{\sum_{k \neq j} \exp(-\|x_k - x_j\|^2 / 2\sigma_j^2)}$$

Вероятность встретить объект  $x_i$  при гауссовом распределении с центром в  $x_j$  и дисперсией  $\sigma_j^2$

# t-SNE

---

Данные величины не являются симметричными, что может добавить нам дополнительных сложностей при дальнейшей работе. Симметризуем их:

$$p_{ij} = \frac{p(i | j) + p(j | i)}{2\ell}.$$

# t-SNE

---

Схожесть в целевом пространстве:

$$q_{ij} = \frac{(1 + \|z_i - z_j\|^2)^{-1}}{\sum_{k \neq m} (1 + \|z_k - z_m\|^2)^{-1}}$$

# t-SNE

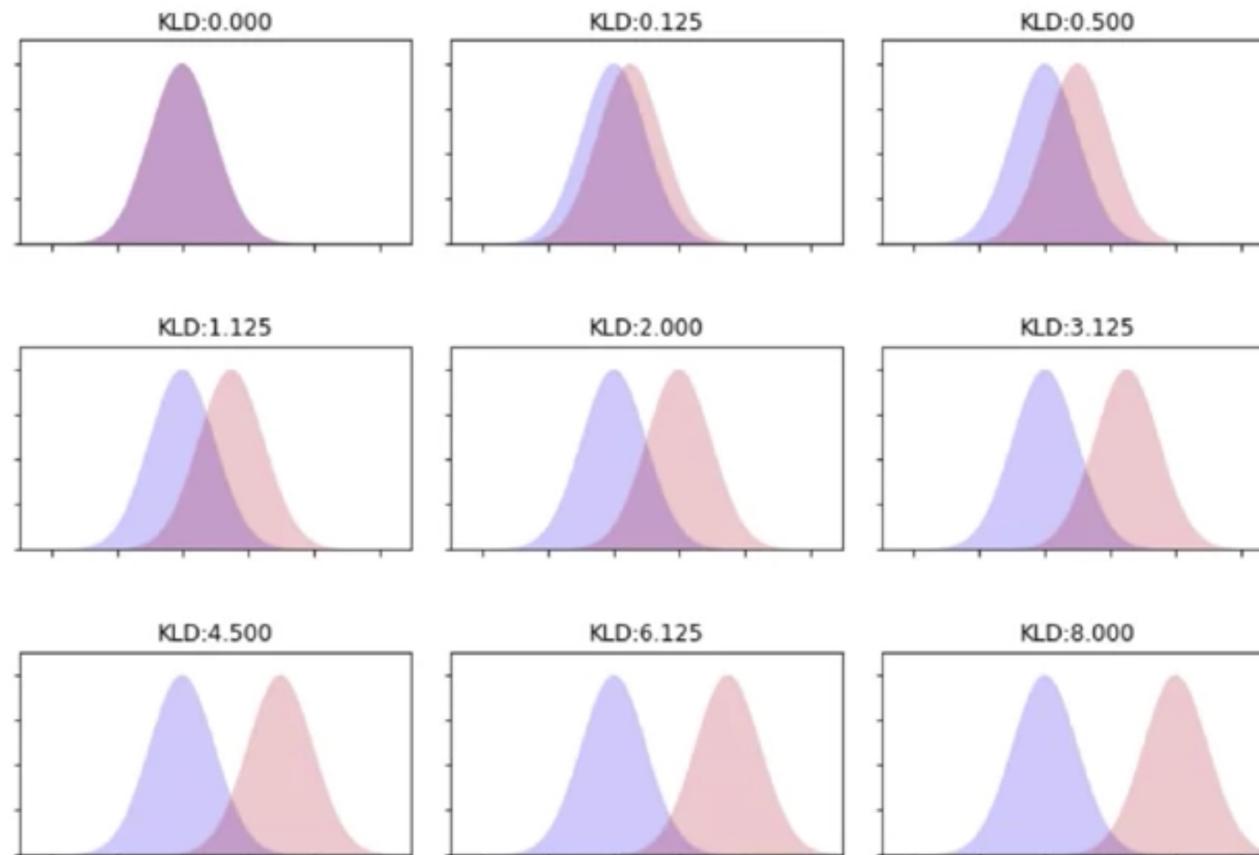
---

Будем измерять ошибку с помощью дивергенции Кульбака-Лейблера, которая часто используется для измерения расстояний между распределениями:

$$\text{KL}(p \parallel q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \rightarrow \min_{z_1, \dots, z_\ell}$$

# Дивергенция Кульбака-Лейблера

---



# Недостатки t-SNE

---

- Может быть нестабильным
- Размеры полученных кластеров могут ничего не значить
- Расстояния между кластерами могут ничего не значить
- Полностью шумовые данные могут выдать структуру

# Методы отбора признаков

---

# Простые методы отбора признаков

---

## ➤ Variance Threshold

Можем удалить признаки, которые имеют очень маленькую дисперсию, т.е. практически константы.

# Простые методы отбора признаков

---

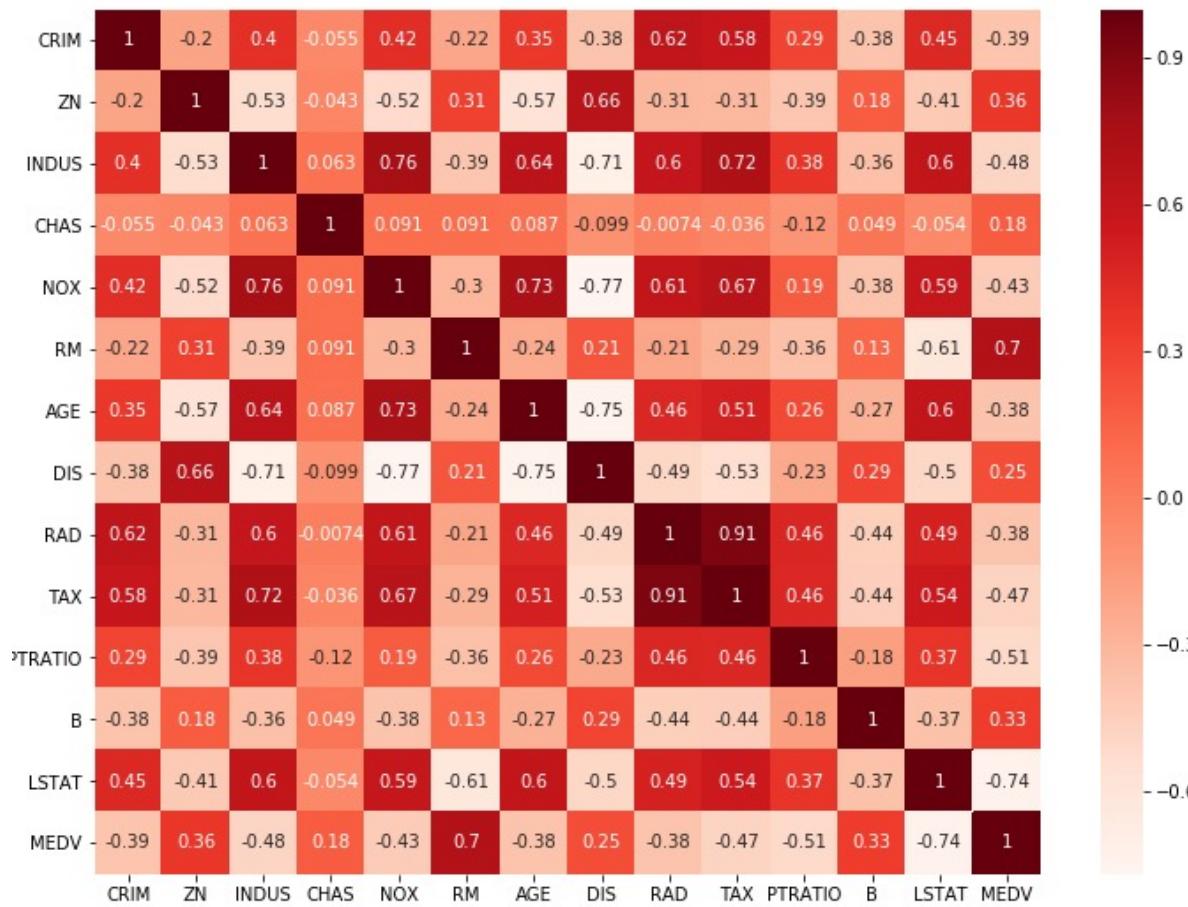
- Variance Threshold

Можем удалить признаки, которые имеют очень маленькую дисперсию, т.е. практически константы.

- Отбор по корреляции с целевой переменной

Для каждого признака вычислим его корреляцию с целевой переменной. Будем выкидывать признаки, имеющие маленькую корреляцию.

# Отбор по корреляции с целевой переменной



# Более сложные методы отбора признаков

---

- фильтрация (отбор признаков без учета модели)
- методы-обертки (wrapper methods, выбор признаков, дающих лучшее качество для модели)
- отбор с помощью моделей (embedded methods, использование свойств моделей для оценки важностей признаков).

# Фильтрационные методы

---

- Фильтрационные методы - это отбор признаков по различным статистическим тестам.
- Идея метода состоит в вычислении влияния каждого признака в отдельности на целевую переменную (с помощью вычисления некоторой статистики)
- Очевидный плюс метода: скорость, так как мы вычисляем значения N статистик, где N - количество признаков

# Фильтрационные методы

---

В sklearn есть сразу несколько методов, использующих отбор по статистическим критериям:

- SelectKBest - оставляет k признаков с наибольшим значением выбранной статистики
- SelectPercentile - оставляет признаки со значениями выбранной статистики, попавшими в заданную пользователем квантиль
- и другие (см.sklearn)

# Фильтрационные методы

---

## ➤ Корреляция

$$R_j = \frac{\sum_{i=1}^{\ell} (x_{ij} - \bar{x}_j)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{\ell} (x_{ij} - \bar{x}_j)^2 \sum_{i=1}^{\ell} (y_i - \bar{y})^2}}$$

Учитывает только линейную связь между величинами!

# Фильтрационные методы

---

## ➤ T-score

Для задачи бинарной классификации:

$$R_j = \frac{|\mu_0 - \mu_1|}{\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}}$$

$\mu$  – среднее,  $\sigma^2$  – дисперсия,  $n$  – число объектов, 0 и 1 – классы

# Фильтрационные методы

---

## ➤ F-score

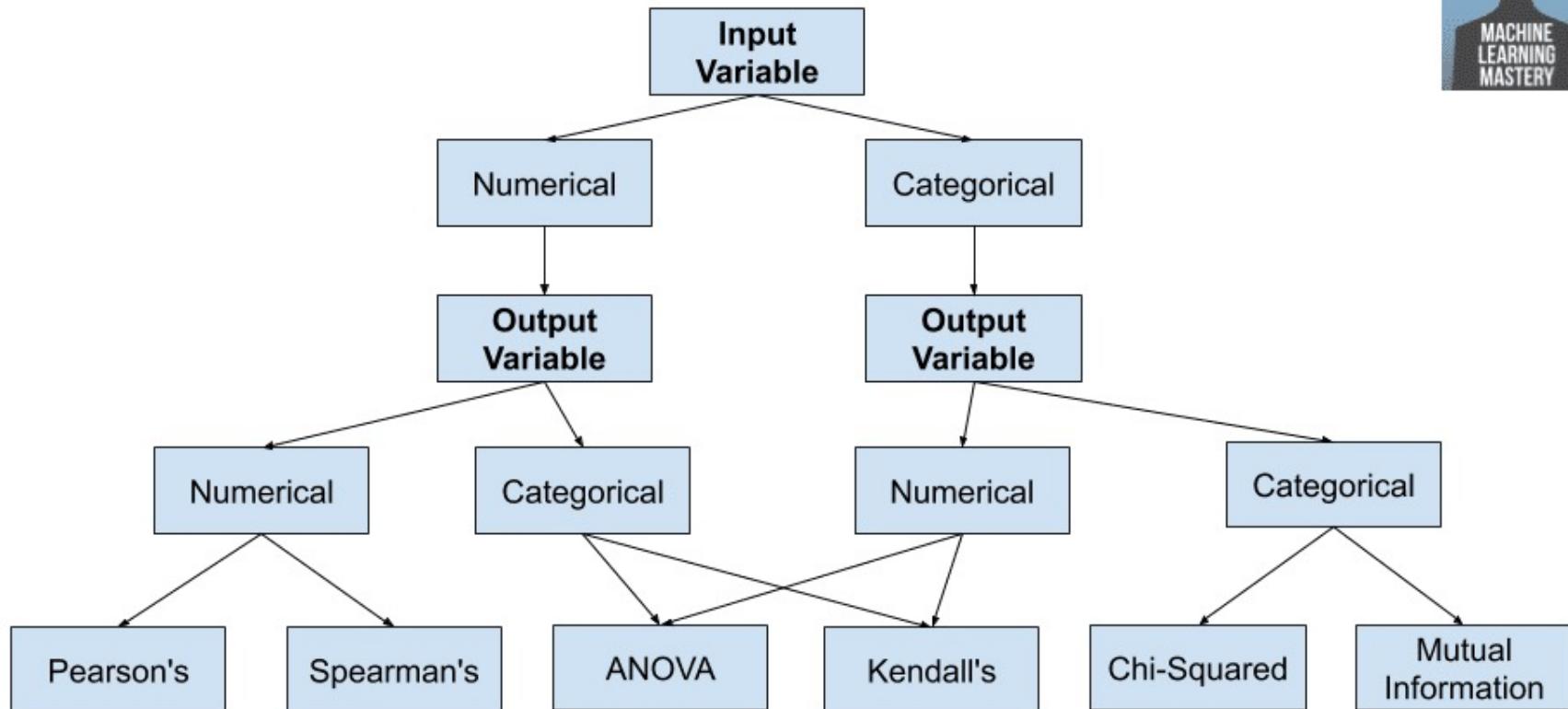
Обобщение для задачи K-классовой классификации:

$$R_j = \frac{\sum_{k=1}^K \frac{n_k}{K-1} (\mu_k - \mu)^2}{\frac{1}{\ell-K} \sum_{k=1}^K (n_k - 1) \sigma_k^2},$$

$\mu$  – среднее по всей выборке, остальные обозначения аналогичны предыдущему пункту

# Фильтрационные методы

How to Choose a Feature Selection Method



Источник (англ.): [URL](#)

Copyright © MachineLearningMastery.com

# Оберточные методы

---

- Отбор признаков производится с помощью обучения алгоритма на разных признаковых подпространствах и оценке его качества на контрольных данных
- Оберточные методы используют жадный отбор признаков, т.е. последовательно выкидывают наименее подходящие по мнению методов признаки

# Оберточные методы

---

- В sklearn есть оберточный метод - Recursive Feature Elimination (RFE).
- Параметры метода:
  - a) алгоритм, используемый для отбора признаков (например, RandomForest)
  - b) число признаков, которое мы хотим оставить

# Отбор с помощью моделей – 1

---

- При добавлении регуляризатора в модель мы уменьшаем влияние различных признаков на финальную модель
- В случае, если мы добавляем l1-регуляризатор, то в силу вида регуляризатора (сумма модулей весов), модель автоматически обнуляет веса некоторых признаков, то есть выкидывает их
- Линейная модель с l1-регуляризатором в sklearn: LASSO

# Отбор с помощью моделей – 2

---

$$Q(j, t) = H(X) - \frac{|X_\ell|}{|X|}H(X_\ell) - \frac{|X_r|}{|X|}H(X_r)$$

- Для подсчета важности  $R_j$  признака  $j$  нужно просуммировать  $Q(j, t)$  по всем вершинам, в которых разбиение выполнялось по признаку  $j$
- Для композиции деревьев суммирование нужно проводить по всем соответствующим вершинам всех деревьев
- Такой алгоритм реализован в sklearn (атрибут `feature_importances_` решающего дерева или случайного леса).

# Отбор с помощью моделей – 3

---

- Если признак важный, то при замене его на случайно генерируемый признак качество решения задачи сильно упадет
- Чтобы максимально сохранить признаковое пространство и распределение данных, признак не удаляют и не заменяют константой
- Вместо этого можно просто перемешать значения признака во всех объектах обучающей выборки

# Out-of-bag оценка

---

$$\text{OOB} = \sum_{i=1}^{\ell} L\left(y_i, \frac{1}{\sum_{n=1}^N [x_i \notin X_n]} \sum_{n=1}^N [x_i \notin X_n] b_n(x_i)\right)$$

# Отбор с помощью моделей – 3

---

Итоговый алгоритм подсчета важностей для случного леса выглядит так:

1. Вычислить ОВ для случного леса, обученного по исходной обучающей выборке
2. Для каждого признака  $j$ :
  - a) Перемешать значения признака по всем объектам обучающей выборки
  - b) Вычислить  $OOB_j$  случного леса, обученного по измененной обучающей выборке
  - c) Оценить важности:  $R_j = \max(0, OOB_j - OOB)$