

# Занятие 8

## Работа с текстами. Поиск аномалий.

Кантонистова Е.О.

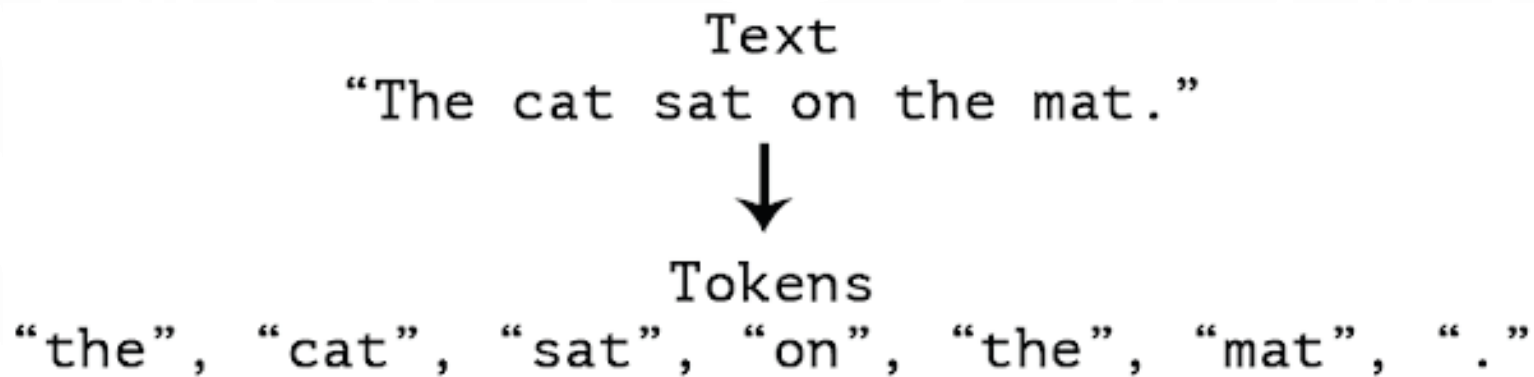
ВШЭ, 2021

# ТЕРМИНОЛОГИЯ

- документ = текст
- корпус – набор документов
- токен – формальное определение “слова”; токен может не иметь смыслового значения (например, “12fdh” или “авыдшл”), но обычно отделен от остальных токенов пробелами или знаками препинания

# ТОКЕНИЗАЦИЯ ТЕКСТА

Чтобы работать с текстом, необходимо разбить его на токены. В простейшем случае токены – это слова (а также наборы букв, знаки препинания и т.д.).



# BAG OF WORDS (МЕШОК СЛОВ)

- По корпусу создадим словарь из всех встречающихся в нем слов (можно убрать общеупотребительные часто встречающиеся слова и очень редкие слова).
- Каждое слово закодируем вектором, в котором стоит единица на месте, соответствующем месту этого слова в словаре, все остальные компоненты вектора – 0.
- Для кодирования документа сложим коды всех его слов.

**Raw Text**

it is a puppy and it  
is extremely cute

**Bag-of-words  
vector**

it	2
they	0
puppy	1
and	1
cat	0
aardvark	0
cute	1
extremely	1
...	...

# BAG OF WORDS (ПРИМЕР)

Пусть корпус состоит из следующих документов:

- D1 - "I am feeling very happy today"
- D2 - "I am not well today"
- D3 - "I wish I could go to play"

Кодировка этих документов будет такой:

	I	am	feeling	very	happy	today	not	well	wish	could	go	to	play
D1	1	1	1	1	1	1	0	0	0	0	0	0	0
D2	1	1	0	0	0	1	1	1	0	0	0	0	0
D3	2	0	0	0	0	0	0	0	1	1	1	1	1

# BAG OF WORDS

*Используя bag of words (BOW), мы теряем информацию о порядке слов в документе.*

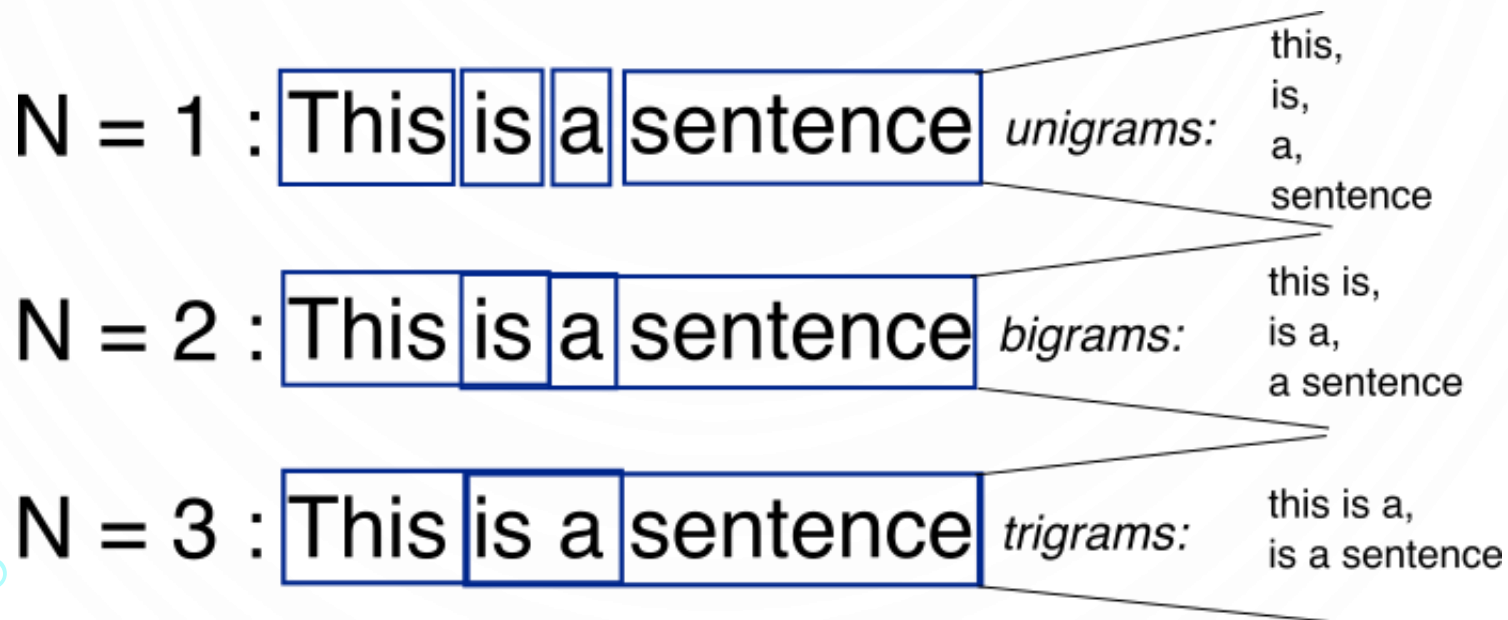
Пример: векторы документов “I have no cats” и “No, I have cats” будут идентичны.

# N-GRAM BAG OF WORDS

В качестве слов в словаре можно использовать:

- N-граммы из букв (наборы букв длины N в слове)
- N-граммы из слов (наборы фраз длины N в документе)

*Такой подход поможет учесть сходственные слова и опечатки.*



# TF-IDF

- слова, которые редко встречаются в корпусе, но присутствуют в документе, могут оказаться важными для характеристики документа.
- слова, которые встречаются во всех документах, наоборот, не важны.



## TF-IDF

***Tf-Idf (term frequency – inverse document frequency):***

- *$tf(t, d)$  - частота вхождения слова  $t$  в документ  $d$ :*

$$tf(t, d) = \frac{n_t}{\sum_k n_k} = \frac{\text{число вхождений слова } t \text{ в документ}}{\text{общее число слов в документе}}$$

*$tf(t, d)$  показывает важность слова  $t$  в документе  $d$ .*

# TF-IDF

- $tf(t, d)$  - частота вхождения слова  $t$  в документ  $d$ :

$$tf(t, d) = \frac{n_t}{\sum_k n_k} = \frac{\text{число вхождений слова } t \text{ в документ}}{\text{общее число слов в документе}}$$

$tf(t, d)$  показывает важность слова  $t$  в документе  $d$ .

- $idf(t, D)$  - величина, обратная частоте, с которой слово  $t$  встречается в документах корпуса  $D$ .

$$idf(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|},$$

$|D|$  — число документов в корпусе,

$|\{d_i \in D \mid t \in d_i\}|$  - число документов, в которых встречается слово  $t$

Учёт  $idf$  уменьшает вес часто используемых в корпусе слов.

# TF-IDF

Tf-idf слова  $t$  в документе  $d$  из корпуса  $D$ :

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D),$$

Пример:

Дана коллекция  $D$  из  $10000000 = 10^7$  документов, в 1000 из них встречается слово “заяц”. В данном документе  $d$  из коллекции 100 слов, и слово “заяц” встречается 3 раза.

$$tf(\text{заяц}, d) = \frac{3}{100} = 0,03$$

$$idf(\text{заяц}, D) = \log\left(\frac{10^7}{10^3}\right) = 4$$

Поэтому  $tfidf(\text{заяц}, d, D) = 0,03 \cdot 4 = 0,12$ .

# ИНТЕРПРЕТАЦИЯ ЛИНЕЙНОЙ МОДЕЛИ

text	label
отвратительное обслуживание был у меня вклад в...	0
мнение о банке изменилось в худшую сторону это...	0
банк поступил красиво у меня дебетовая карта б...	1
прошу принять меры по исправлению ситуации бан...	0
спокойно и качественно пользуюсь услугами альф...	1

# ИНТЕРПРЕТАЦИЯ ЛИНЕЙНОЙ МОДЕЛИ

- 0.99 accuracy на обучении
- 0.93 accuracy на валидации

спасибо 15.3812631501  
приятно 10.195153067  
благодарность 8.75099611487  
оперативность 7.9119980712  
быстро 7.20768729913  
всегда 6.49503091778  
оперативно 6.36190679808  
большое 6.02762583473  
доволен 5.86536526776  
отзыв 5.64047141286  
помощь 5.43980835894  
поблагодарить 5.19673514028

## Примеры весов

претензию -3.84736026948  
не работает -3.89934654597  
два -3.9180675684  
звонков -3.99518600488  
готовности -4.00435284458  
говорят -4.10305804728  
дозвониться -4.10647379932  
пусть -4.20500663563  
видимо -4.32809243057  
не -4.59523464931  
звонки -4.63261991797  
отказ -4.90228031373

# ПРИМЕР

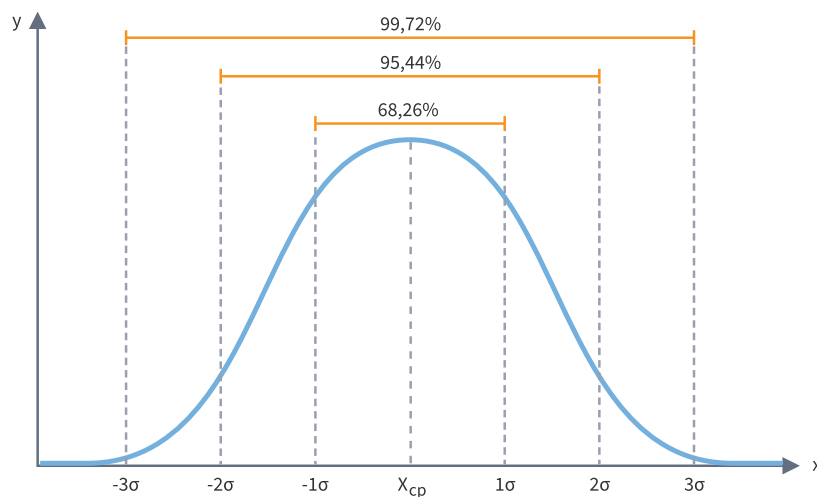
- <https://colab.research.google.com/drive/1s9fJkYoli89m236zLTSjlyCz1uUXAbjU?usp=sharing>
- [https://colab.research.google.com/drive/1QvS1mzqja7n-pqvzmw8NKmg38l9l\\_Cub?usp=sharing](https://colab.research.google.com/drive/1QvS1mzqja7n-pqvzmw8NKmg38l9l_Cub?usp=sharing)



# РАБОТА С ВЫБРОСАМИ

# 1. ПРАВИЛО ТРЕХ СИГМ

- Для случайных величин, распределенных по нормальному закону, вероятность того, что случайная величина отклонится от своего математического ожидания более чем на три стандартных отклонения, практически равна нулю.



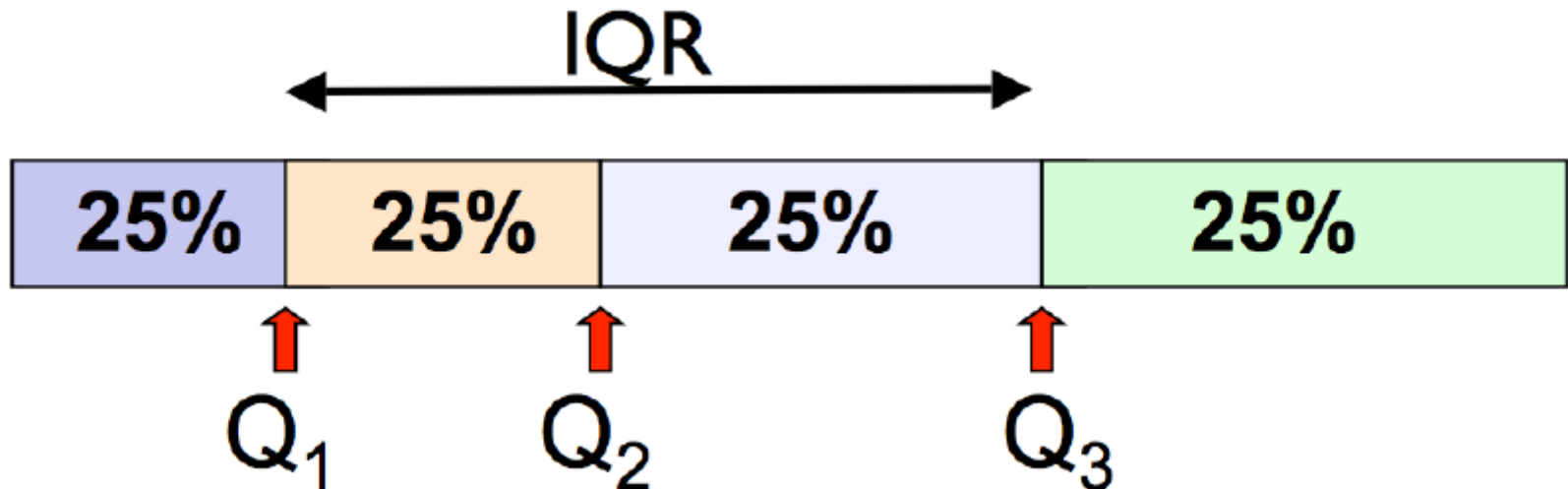
- Выбросами объявляются объекты, имеющие стандартное отклонение  $\geq 3\sigma$  от математического ожидания.



## 2. ИНТЕРКВАРТИЛЬНЫЙ РАЗМАХ

Пусть  $Q_1$  – первая (25%) квартиль распределения,  
 $Q_3$  – третья (75%) квартиль распределения.

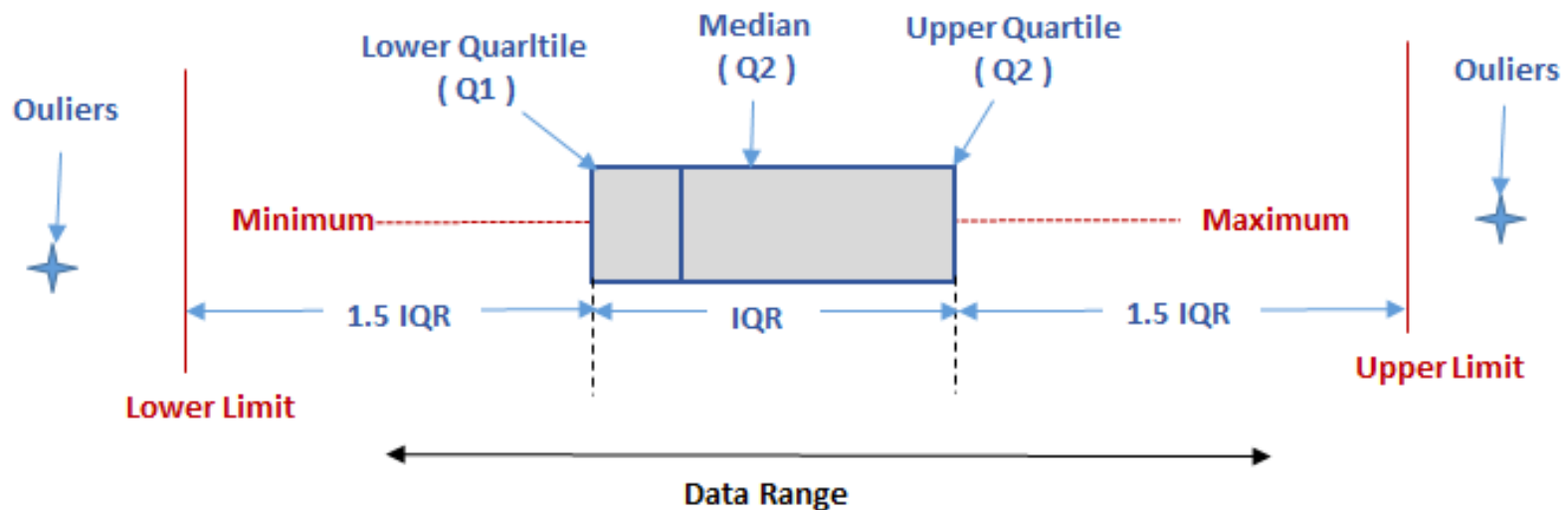
- Величина  $IQR = Q_3 - Q_1$  называется **интерквартильным размахом**.



## 2. ИНТЕРКВАРТИЛЬНЫЙ РАЗМАХ

- **Выбросы** – это значения, которые меньше 25%-квантили минус  $1,5IQR$  или больше 75%-квантили плюс  $1,5IQR$ :

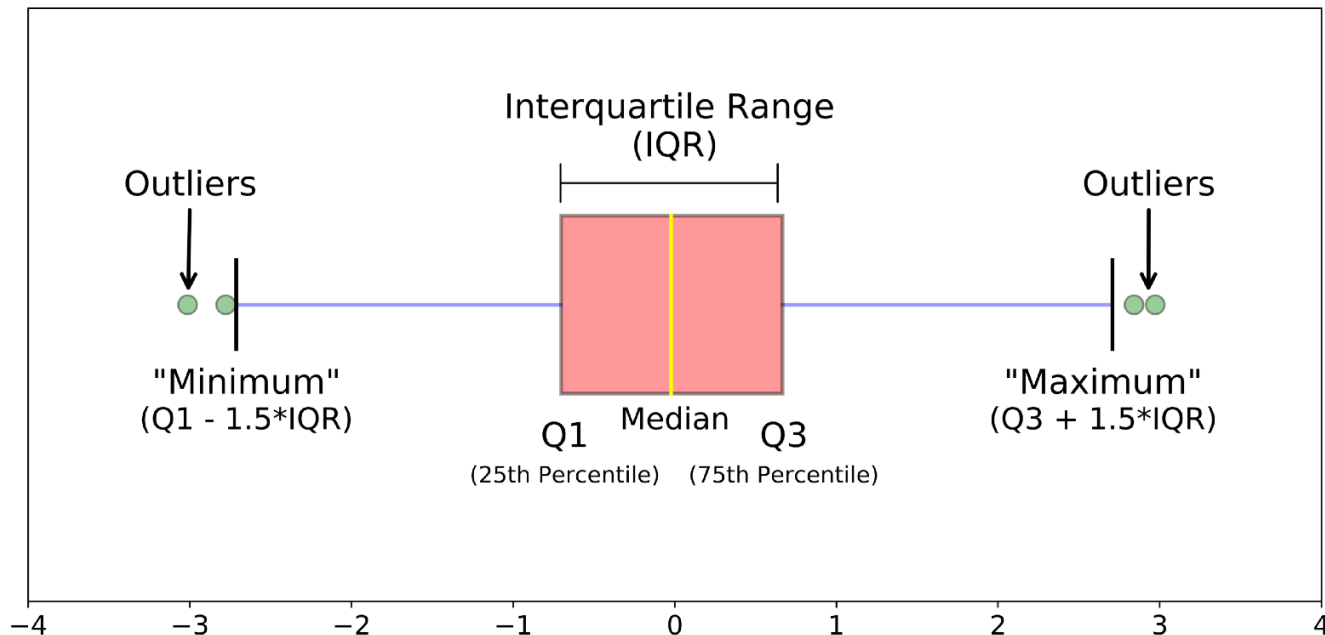
$$x < Q1 - 1,5 \cdot IQR \text{ или } x > Q3 + 1,5 \cdot IQR$$

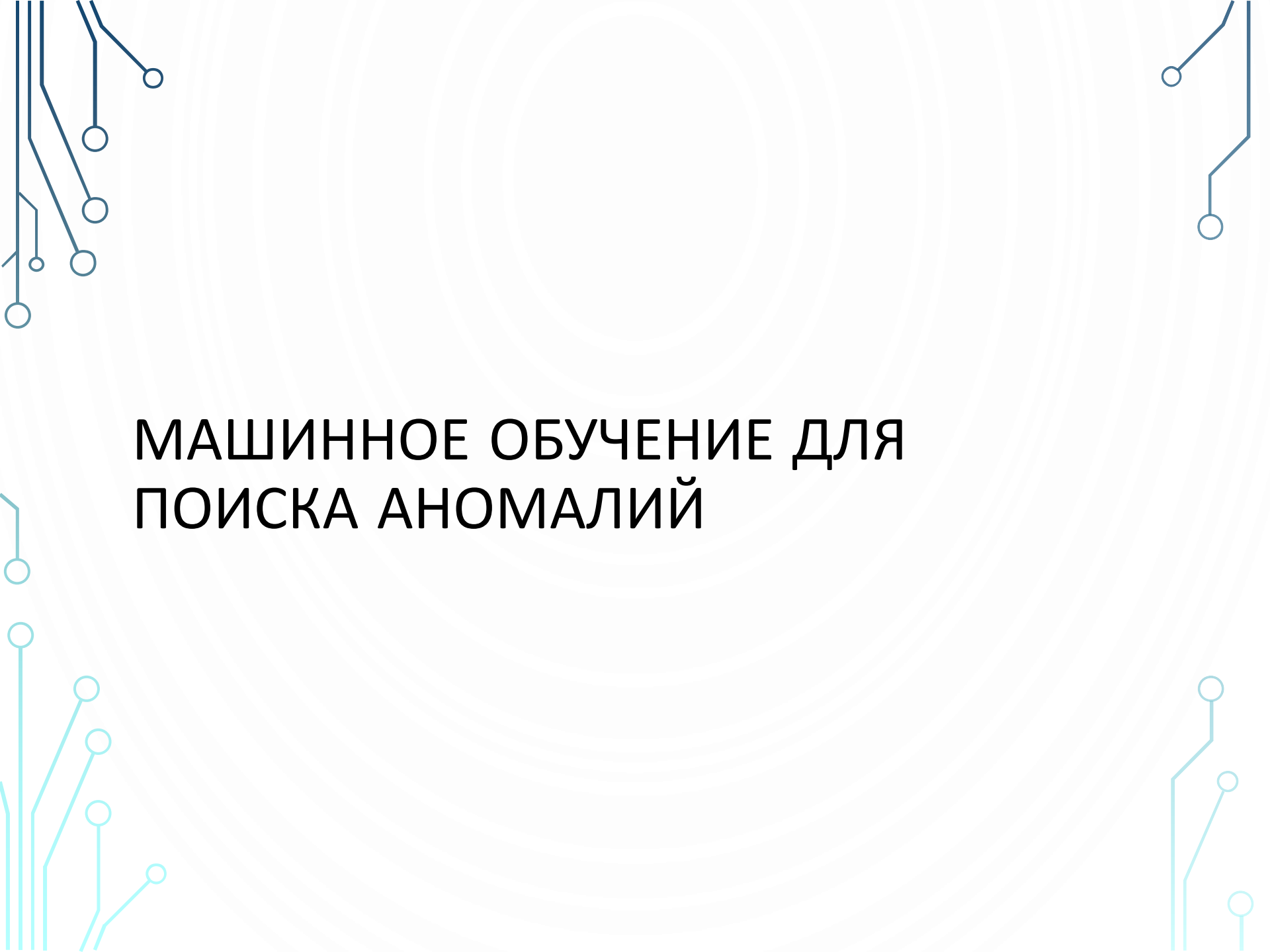


# ЯЩИК С УСАМИ

Ящик с усами – это диаграмма, которая показывает:

- одномерное распределение вероятностей (квартили)
- границы попадания “нормальных” точек
- выбросы



The background features a light gray circular pattern. The corners are decorated with stylized circuit lines: dark blue in the top-left and top-right, and light blue in the bottom-left and bottom-right. These lines include small circles at various points, resembling nodes or components of a circuit.

# МАШИННОЕ ОБУЧЕНИЕ ДЛЯ ПОИСКА АНОМАЛИЙ

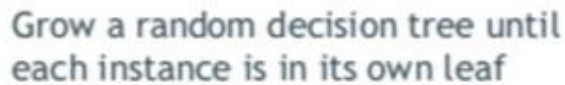
# 1. ISOLATION FOREST

- Строим лес, состоящий из  $N$  деревьев. Каждый признак и порог выбираем случайно. Останавливаемся, когда в вершине 1 объект или когда построили дерево максимальной глубины.

Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.

# ISOLATION FOREST

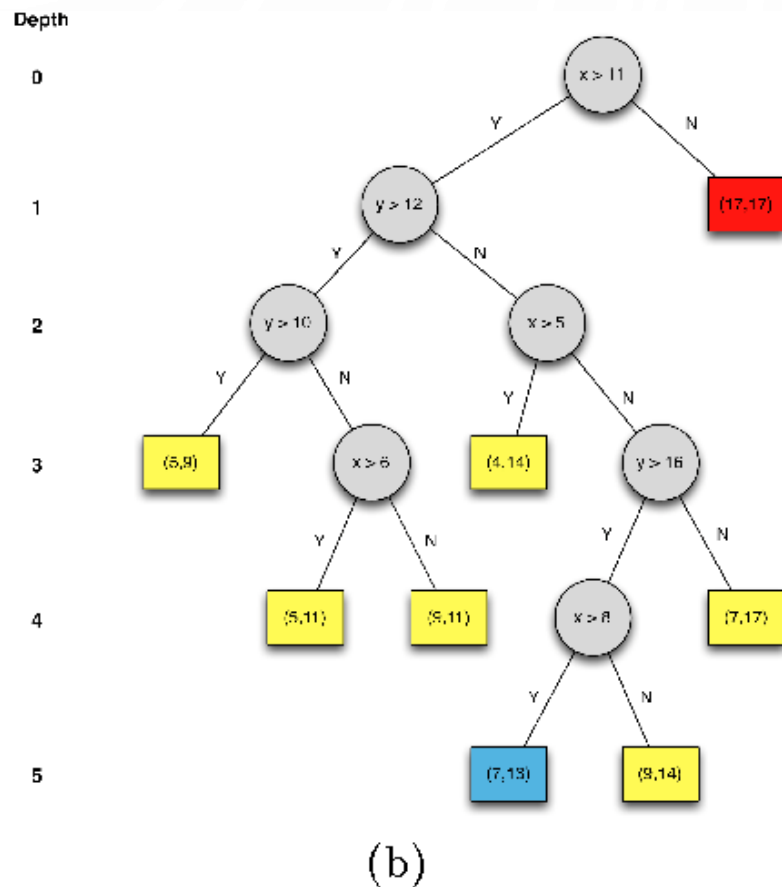
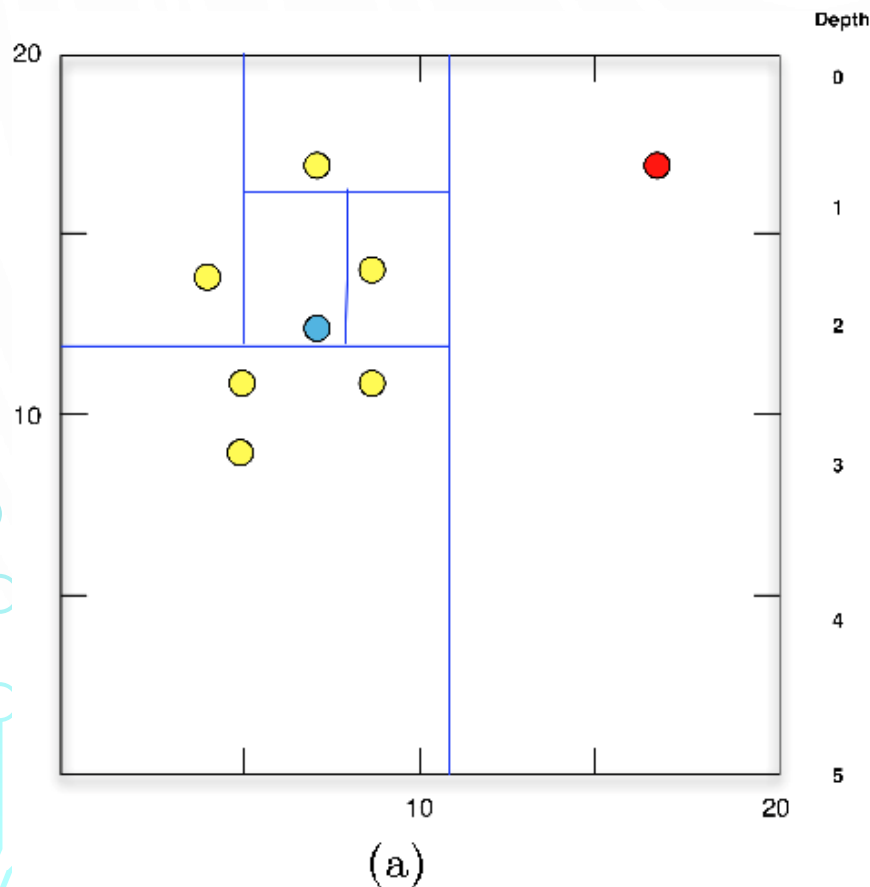
Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.



Now repeat the process several times and use average Depth to compute anomaly score: 0 (similar) -> 1 (dissimilar)

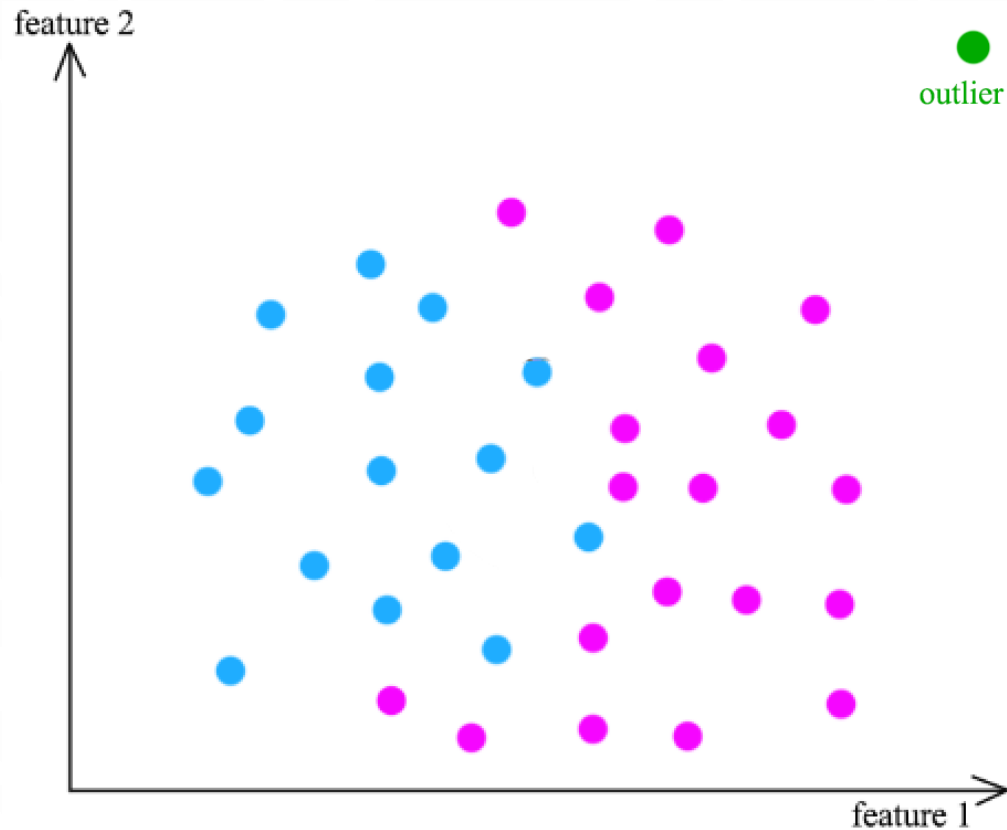
# ISOLATION FOREST

Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.



## 2. ПОИСК ВЫБРОСОВ С ПОМОЩЬЮ KNN

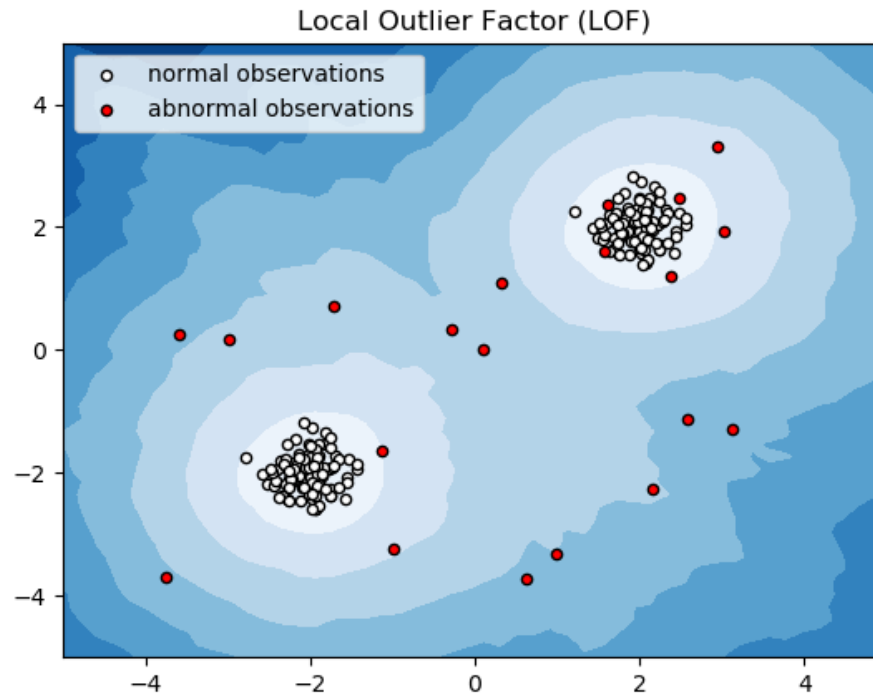
- Вычисляем среднее расстояние от каждой точки до её ближайших  $k$  соседей
- Точки с наибольшим средним расстоянием – выбросы





### 3. LOCAL OUTLIER FACTOR

- Задаем плотность распределения в точке, используя  $k$  ближайших соседей
- Точки, плотность распределения в которых значительно меньше, чем у соседей – выбросы.



# ССЫЛКИ

- <https://dyakonov.org/2017/04/19/поиск-аномалий-anomaly-detection/>
- [https://scikit-learn.org/stable/modules/outlier\\_detection.html](https://scikit-learn.org/stable/modules/outlier_detection.html)
- <https://github.com/yzhao062/pyod>