

# Поиск аномалий

Елена Кантонистова

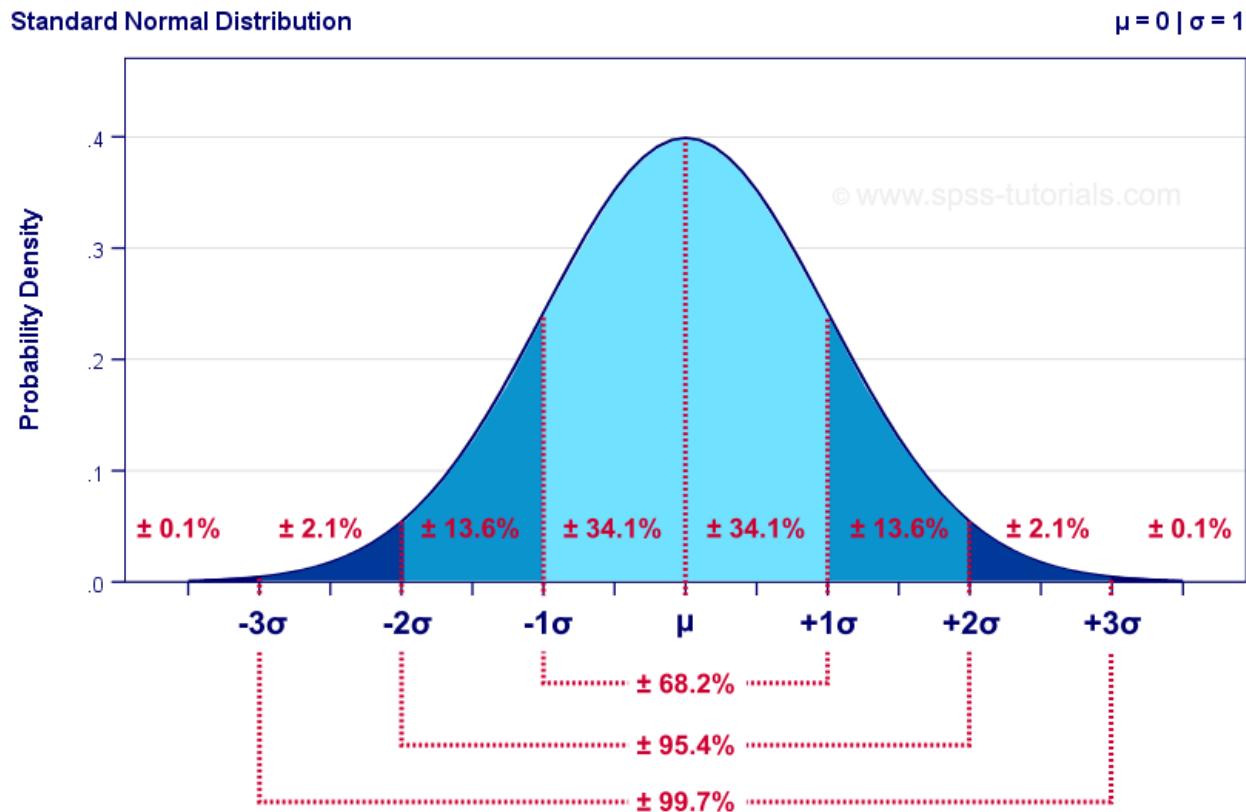
# ПЛАН

- Простые (одномерные) подходы
- Статистические подходы
- Машинное обучение

# Z-SCORE

Если данные распределены нормально, то большинство измерений находится в диапазоне  $(m - 3\sigma; m + 3\sigma)$ .

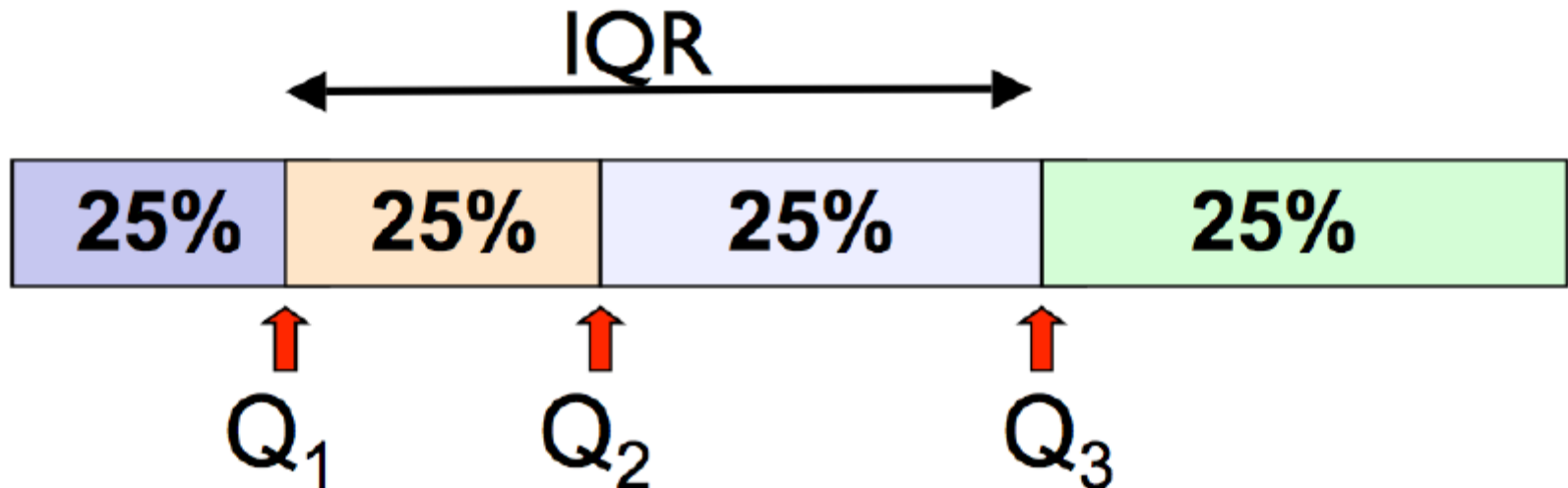
*Точки, не попавшие в этот диапазон, можно считать выбросами.*



# НАХОЖДЕНИЕ ВЫБРОСОВ В ДАННЫХ

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

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



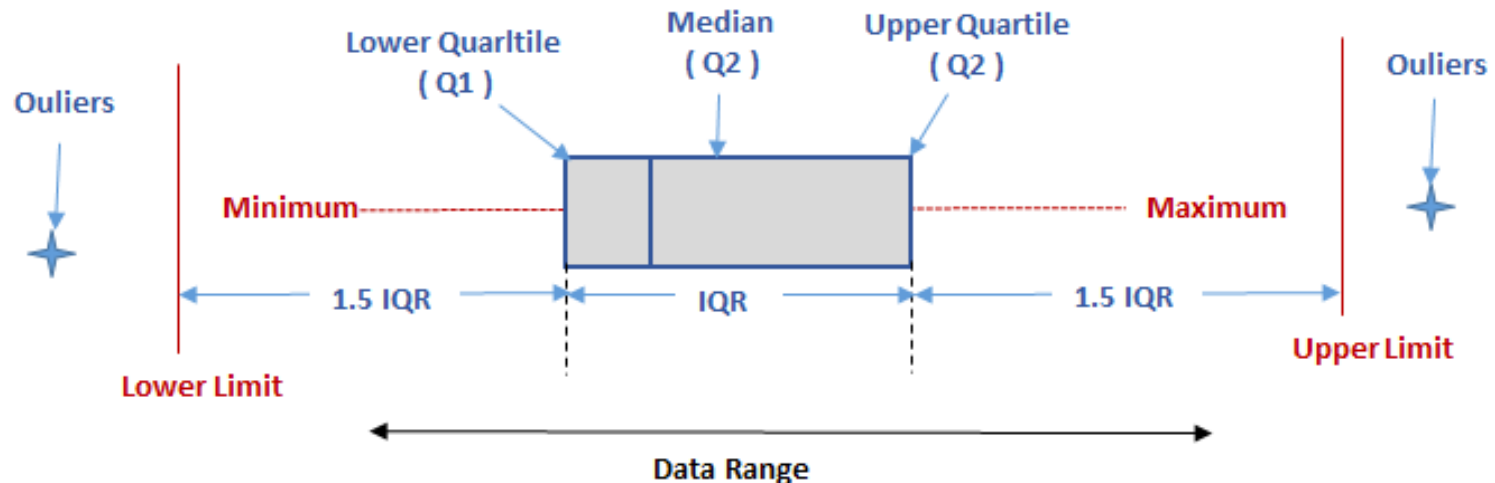
# НАХОЖДЕНИЕ ВЫБРОСОВ В ДАННЫХ

- **Слабые выбросы** – это значения, которые меньше 25%-квартили минус  $1,5 \cdot IQR$  или больше 75%-квартили плюс  $1,5 \cdot IQR$ :

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

- **Сильные выбросы** – это значения, которые меньше 25%-квартили минус  $3 \cdot IQR$  или больше 75%-квартили плюс  $3 \cdot IQR$ :

$$x < Q1 - 3 \cdot IQR \text{ или } x > Q3 + 3 \cdot IQR$$



## ДИАГРАММА ЯЩИК УСАМИ (BOX-AND- WHISKER PLOT)

- Ящик с усами — это диаграмма, визуализирующая основные характеристики данных.



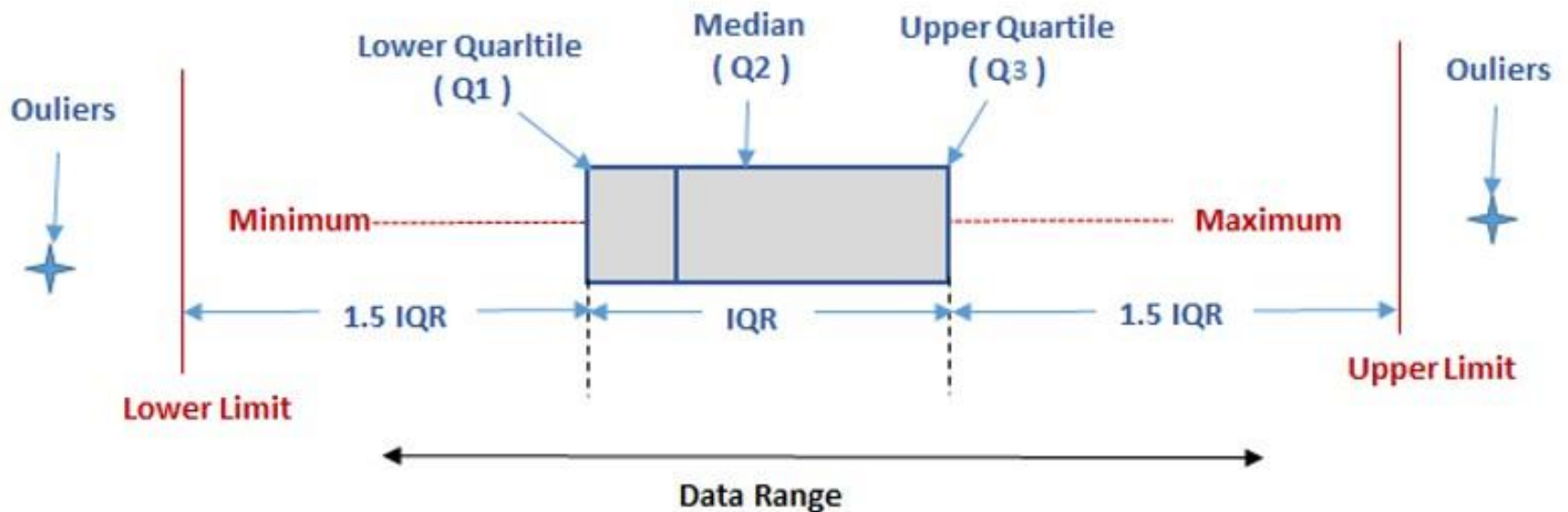
# ДИАГРАММА ЯЩИК С УСАМИ (BOX-AND-WHISKER PLOT)

Ящик с усами — это диаграмма, визуализирующая основные характеристики данных.

Она состоит из:

- 1) Медианы** (“центрального” значения распределения)
- 2) Первой и третьей квартилей:**  $Q1$  и  $Q3$ .
- 3) Минимума и максимума**
- 4)левой и правой границ, выйдя за которые точки считаются выбросами.**

# Диаграмма ящик с усами (Box-and-Whisker plot)



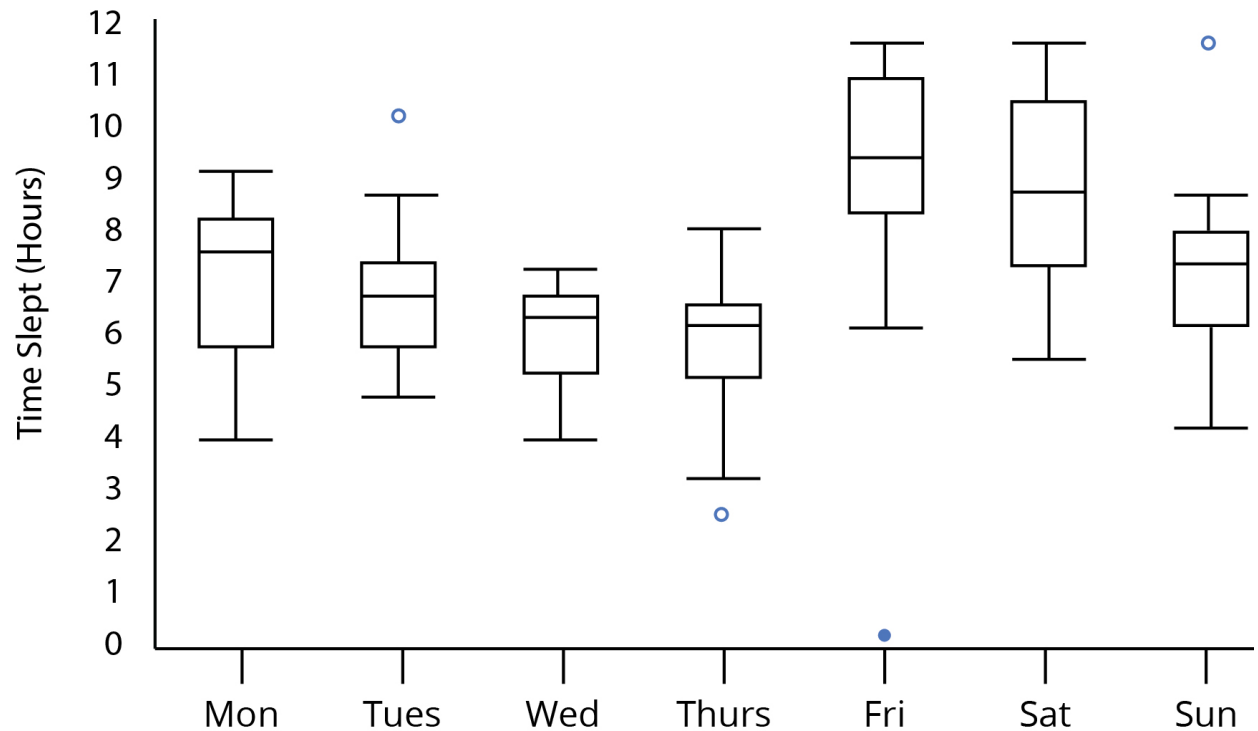
<http://www.whatissixsigma.net/box-plot-diagram-to-identify-outliers/>



# ПРИМЕР: ДИАГРАММА ЯЩИК С УСАМИ

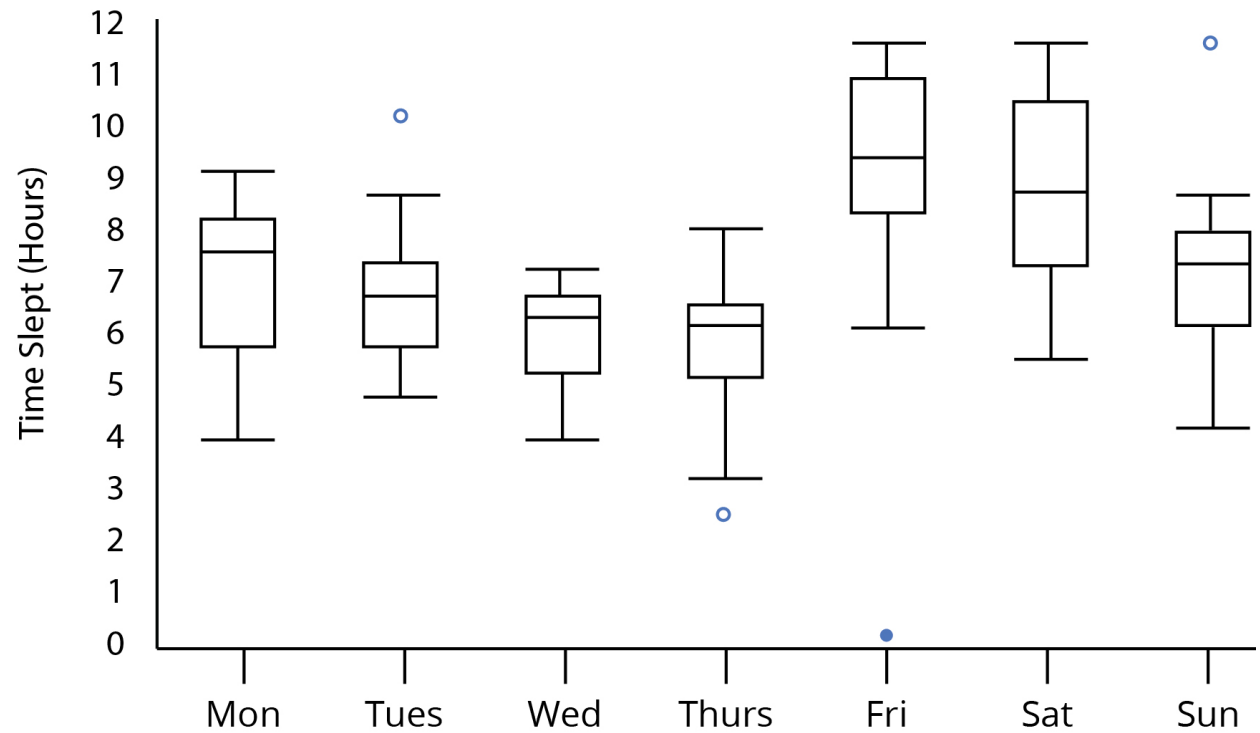
Очень легко видеть, как  
распределяется  
продолжительность сна в  
зависимости от дня недели.

Хорошо видны выбросы и  
общие тенденции.



# ВОПРОС

*Почему усы у  
каждого ящика  
разной длины?*



# СТАТИСТИЧЕСКИЕ МЕТОДЫ: НЕПАРАМЕТРИЧЕСКИЙ ПОДХОД

Будем восстанавливать плотность по выборке.

*Аномалии здесь – объекты в области низкой плотности.*

Запишем определение плотности:

$$p(x) = \lim_{h \rightarrow 0} \frac{1}{2h} \mathbb{P}(\xi \in [x - h, x + h])$$

# СТАТИСТИЧЕСКИЕ МЕТОДЫ: НЕПАРАМЕТРИЧЕСКИЙ ПОДХОД

Будем восстанавливать плотность по выборке.

*Аномалии здесь – объекты в области низкой плотности.*

Запишем определение плотности:

$$p(x) = \lim_{h \rightarrow 0} \frac{1}{2h} \mathbb{P}(\xi \in [x - h, x + h])$$

Пользуясь тем, что  $P(X) = E I[X]$ , а также **ЗБЧ** (сумма н.о.р.с.в. стремится к математическому ожиданию одной из этих с.в.), получаем:

$$\hat{p}(x) = \frac{1}{2h} \frac{1}{\ell} \sum_{i=1}^{\ell} [ |x - x_i| < h ] = \frac{1}{\ell h} \sum_{i=1}^{\ell} \frac{1}{2} \left[ \frac{|x - x_i|}{h} < 1 \right]$$

# СТАТИСТИЧЕСКИЕ МЕТОДЫ: НЕПАРАМЕТРИЧЕСКИЙ ПОДХОД

Вместо индикатора используем некоторую гладкую функцию  $K$  (ядро):

$$\hat{p}(x) = \frac{1}{\ell h} \sum_{i=1}^{\ell} K \left( \frac{x - x_i}{h} \right)$$

Ядро должно удовлетворять свойствам:

- чётность:  $K(-z) = K(z)$ ;
- нормированность:  $\int K(z) dz = 1$ ;
- неотрицательность:  $K(z) \geq 0$ ;

Примером может быть Гауссово ядро:

$$K(z) = (2\pi)^{-1/2} \exp(-0.5z^2)$$

# СТАТИСТИЧЕСКИЕ МЕТОДЫ: ПАРАМЕТРИЧЕСКИЙ ПОДХОД

Параметрический подход состоит в том, что мы задаем некоторое распределение (фиксируем формулу плотности), и подбираем параметры этого распределения по выборке.

Для подбора используется метод максимума правдоподобия (ММП):

$$\sum_{i=1}^{\ell} \log p(x_i | \theta) \rightarrow \max_{\theta}$$

# ПОИСК АНОМАЛИЙ С ПОМОЩЬЮ ML-МОДЕЛЕЙ

Идея: можно настроить модель машинного обучения так, чтобы на нормальных объектах она принимала значения, близкие к нулю (или, например, положительные значения). Тогда если прогноз на объекте сильно отличается от прогноза на обучающей выборке, то такой объект можно считать аномальным.

# 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.

Grow a random decision tree until each instance is in its own leaf

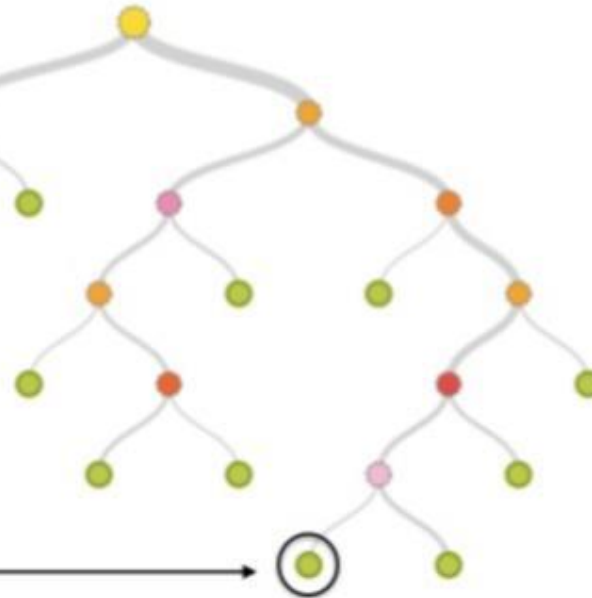
“easy” to isolate →



Depth

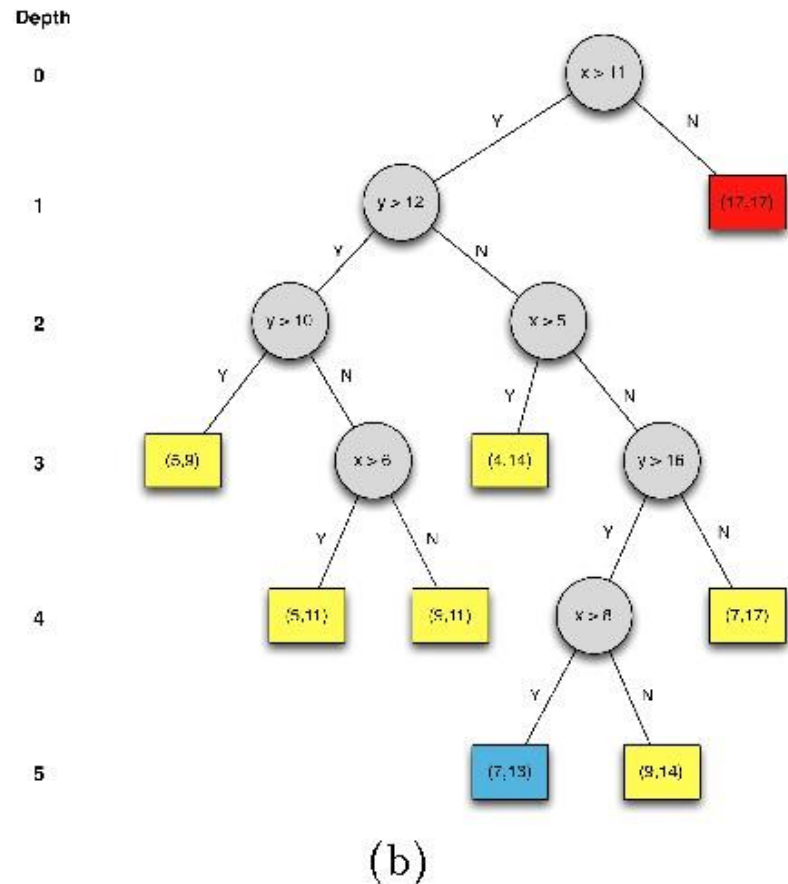
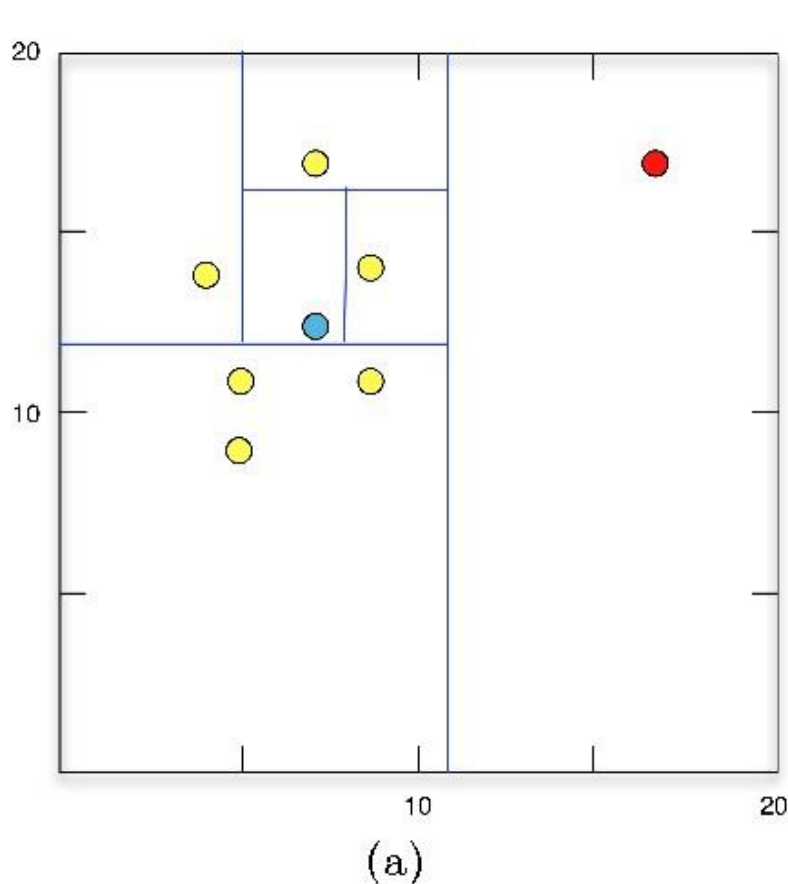
“hard” to isolate →

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.



# ONE-CLASS SVM

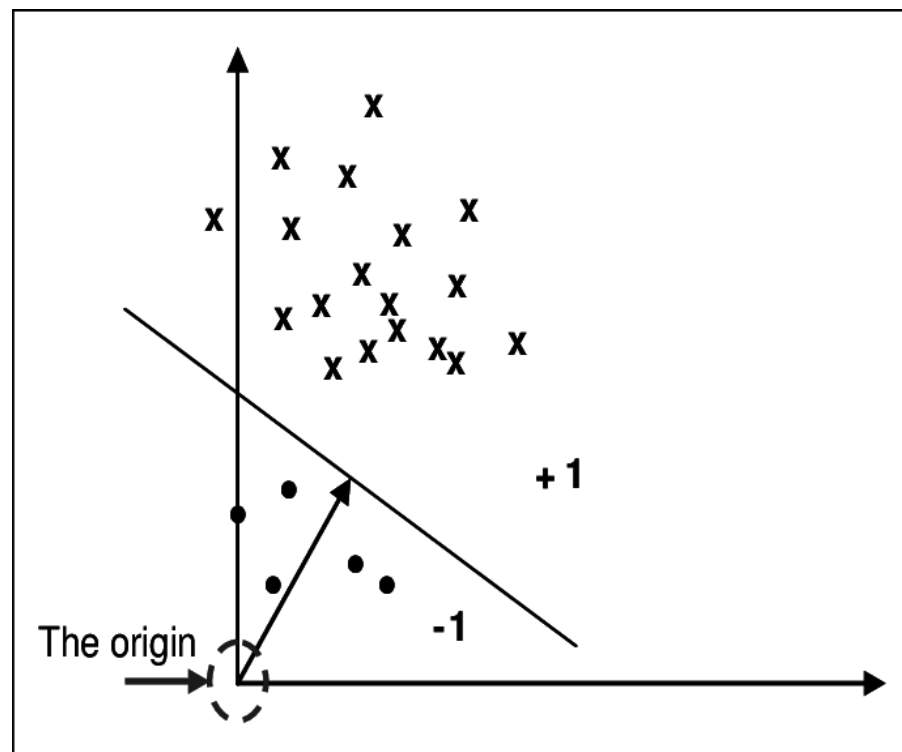
Метод строит линейную функцию  $a(x) = \text{sign}(w, x)$  так, чтобы она отделяла выборку от начала координат с максимальным отступом, а именно:

- $a(x)$  отделяет как можно больше объектов выборки от нуля
- имеет большой отступ

Тогда объекты с  $a(x) = -1$

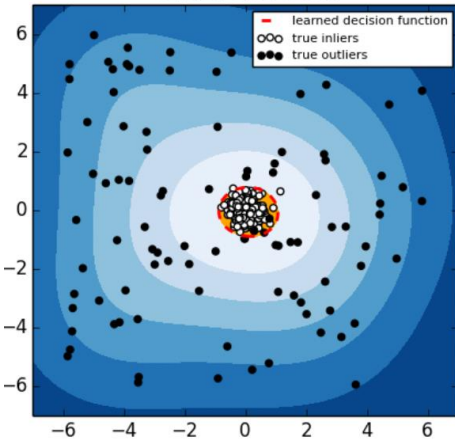
— это аномалии.

Начало координат здесь играет роль “другого” класса.



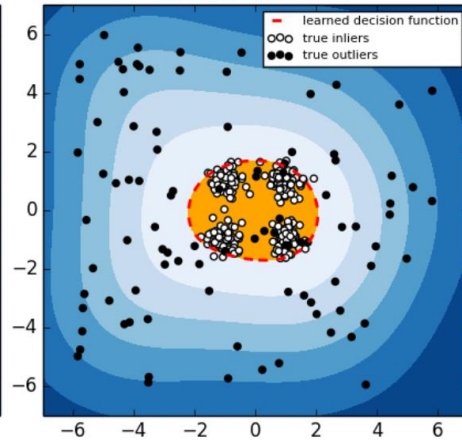
# ONE-CLASS SVM с RBF-ЯДРОМ

Outlier detection



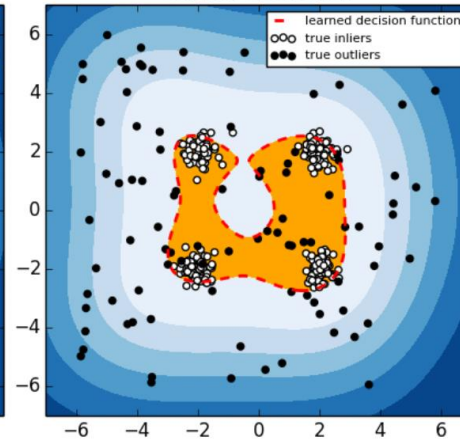
1. one class SVM (errors: 6)

Outlier detection



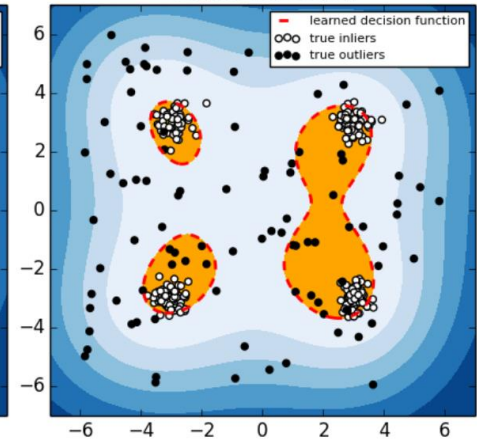
2. one class SVM (errors: 26)

Outlier detection



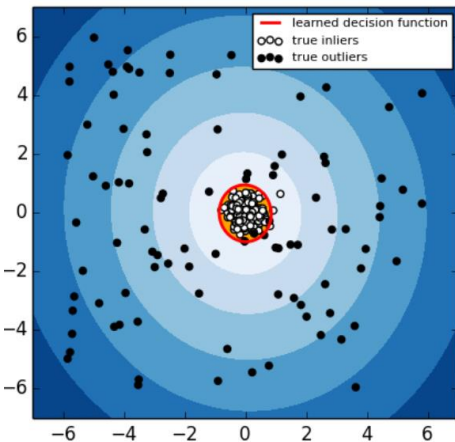
3. one class SVM (errors: 40)

Outlier detection



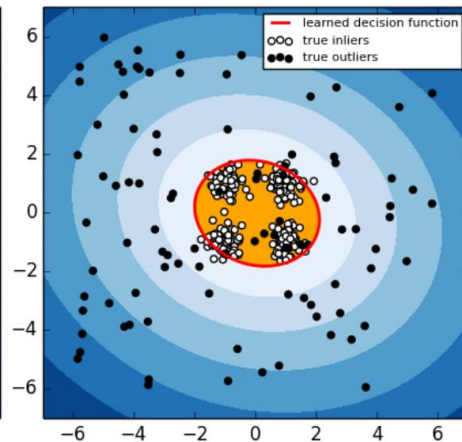
4. one class SVM (errors: 46)

Outlier detection



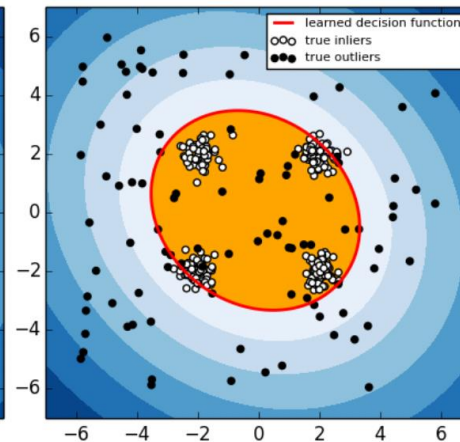
1. covariance estimation (errors: 6)

Outlier detection



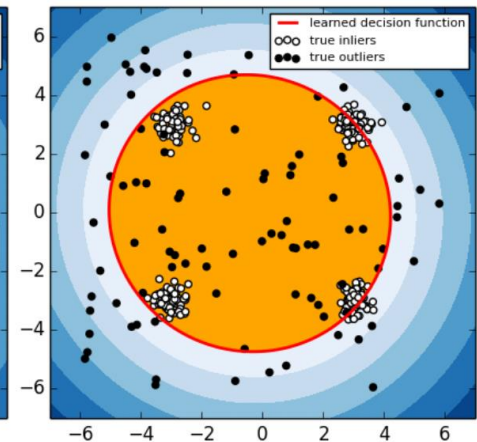
2. covariance estimation (errors: 26)

Outlier detection



3. covariance estimation (errors: 54)

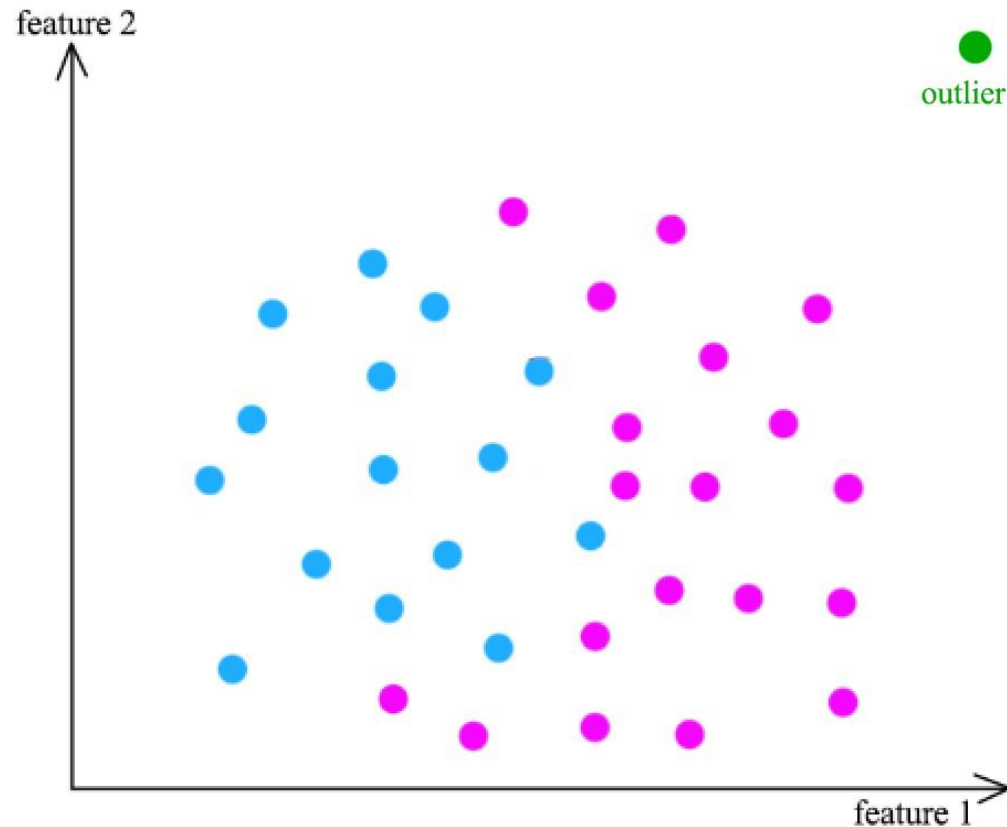
Outlier detection



4. covariance estimation (errors: 98)

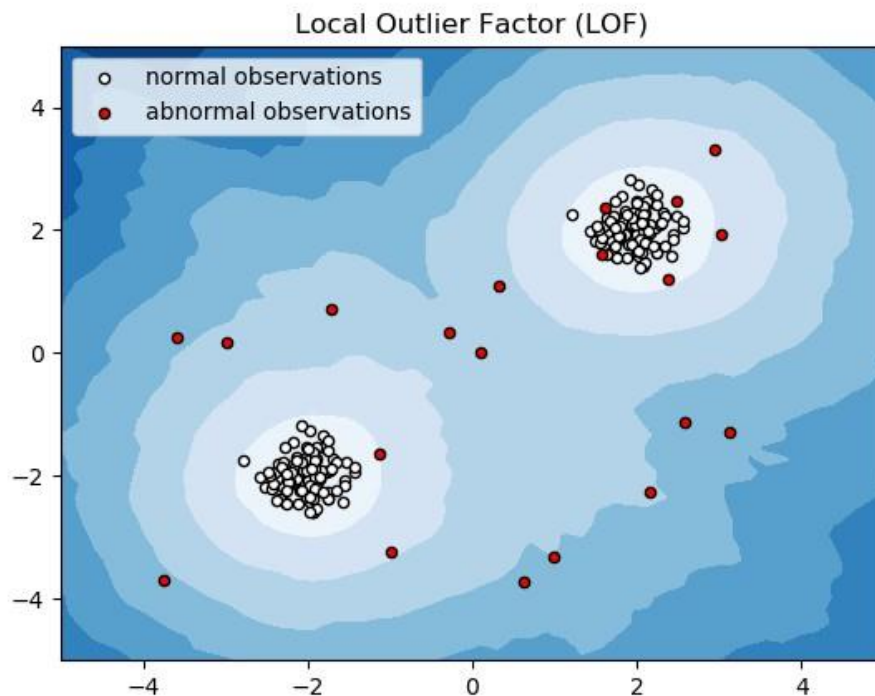
# ПОИСК ВЫБРОСОВ С ПОМОЩЬЮ KNN

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



# LOCAL OUTLIER FACTOR (LOF)

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



## Алгоритм работы local-outlier factor

### 1 Находим $k$ ближайших соседей

- Для каждой точки  $p$  ищем  $k$  ближайших соседей (обычно по евклидову расстоянию).
- $k$  — гиперпараметр (обычно  $k = 20$ ).

### 2 Вычисляем расстояние до ближайших соседей (reachability distance)

Для каждого соседа  $o$  точки  $p$  вычисляем дистанцию достижимости:

$$\text{reach-dist}_k(p, o) = \max(k\text{-distance}(o), \text{dist}(p, o))$$

Где:

- $k\text{-distance}(o)$  — расстояние до  $k$ -го соседа точки  $o$ .
- $\text{dist}(p, o)$  — обычное расстояние между точками.
- Почему берём максимум? → Чтобы избежать ситуации, когда сосед слишком близко (учёт плотности региона).

### 3 Считаем локальную плотность (local reachability density, LRD)

Локальная плотность точки  $p$  определяется как:

$$LRD(p) = \frac{k}{\sum_{o \in kNN(p)} \text{reach-dist}_k(p, o)}$$

Где:

- Чем больше LRD, тем плотнее окружение точки.
- Чем меньше LRD, тем больше точка выделяется из окружения (потенциальная аномалия).

### 4 Вычисляем Local Outlier Factor (LOF) для точки $p$

$$LOF(p) = \frac{\sum_{o \in kNN(p)} \frac{LRD(o)}{LRD(p)}}{k}$$

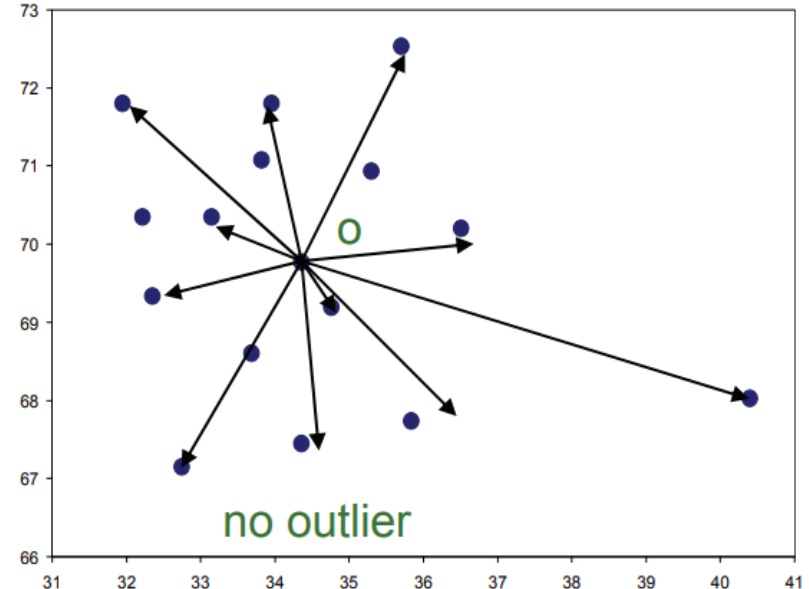
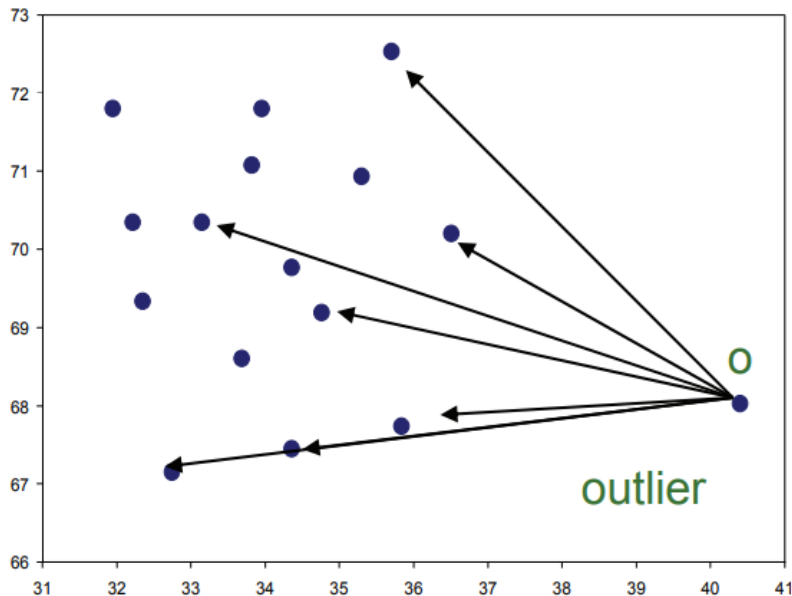
Где:

- Если  $LOF \approx 1$  → точка нормально вписывается в локальную плотность.
- Если  $LOF \gg 1$  → точка значительно реже встречается в локальном пространстве (аномалия!).



# ABOD (Angle-Based Outlier Detection)

ABOD – это алгоритм обнаружения аномалий, который анализирует углы между векторами точек в многомерном пространстве. В отличие от других методов (например, LOF, основанного на плотности), ABOD эффективен при высоких размерностях данных, так как не требует вычисления плотности вокруг точки.





# ABOD (Angle-Based Outlier Detection)

Как работает ABOD?

## 1. Рассчитывает углы между векторами точек

- Для каждой точки  $p$  рассматриваются векторы, соединяющие её с другими точками в наборе
- Углы между этими векторами используются для анализа расположения точки в пространстве

## 2. Оценивает дисперсию углов

- Если точка окружена равномерно распределёнными соседями, её углы имеют низкую дисперсию → точка считается нормальной
- Если точка находится вдалеке от большинства данных (аномалия), углы её векторов становятся более хаотичными → дисперсия углов высокая

## 3. Присваивает объекту anomaly score

- Чем выше дисперсия углов, тем более вероятно, что точка является выбросом

- [https://scikit-learn.org/stable/modules/outlier\\_detection.html](https://scikit-learn.org/stable/modules/outlier_detection.html)
- <https://github.com/yzhao062/pyod>