

Лекция 12

Работа с текстами. Поиск аномалий.

Кантонистова Е.О.

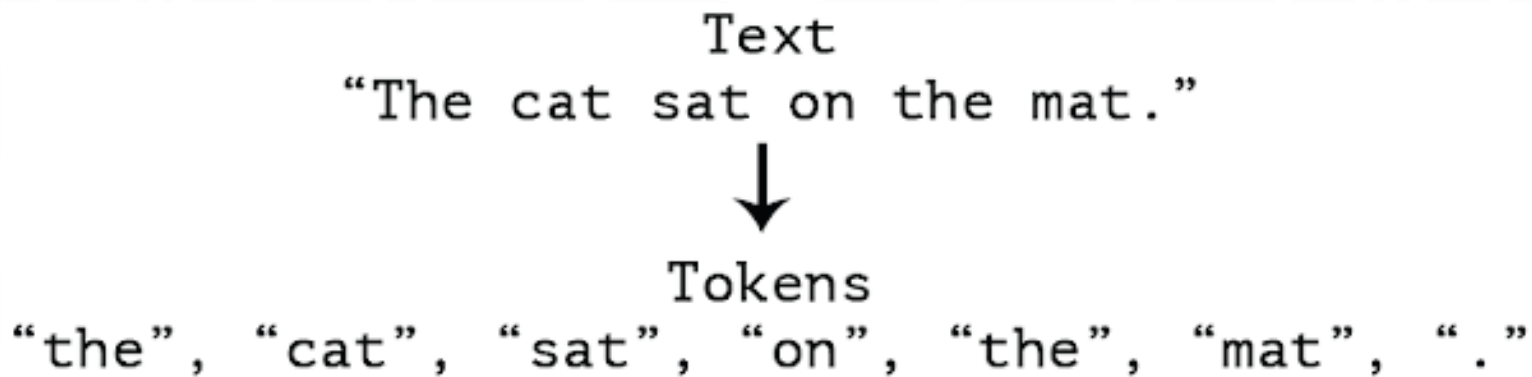
ВШЭ, 2021

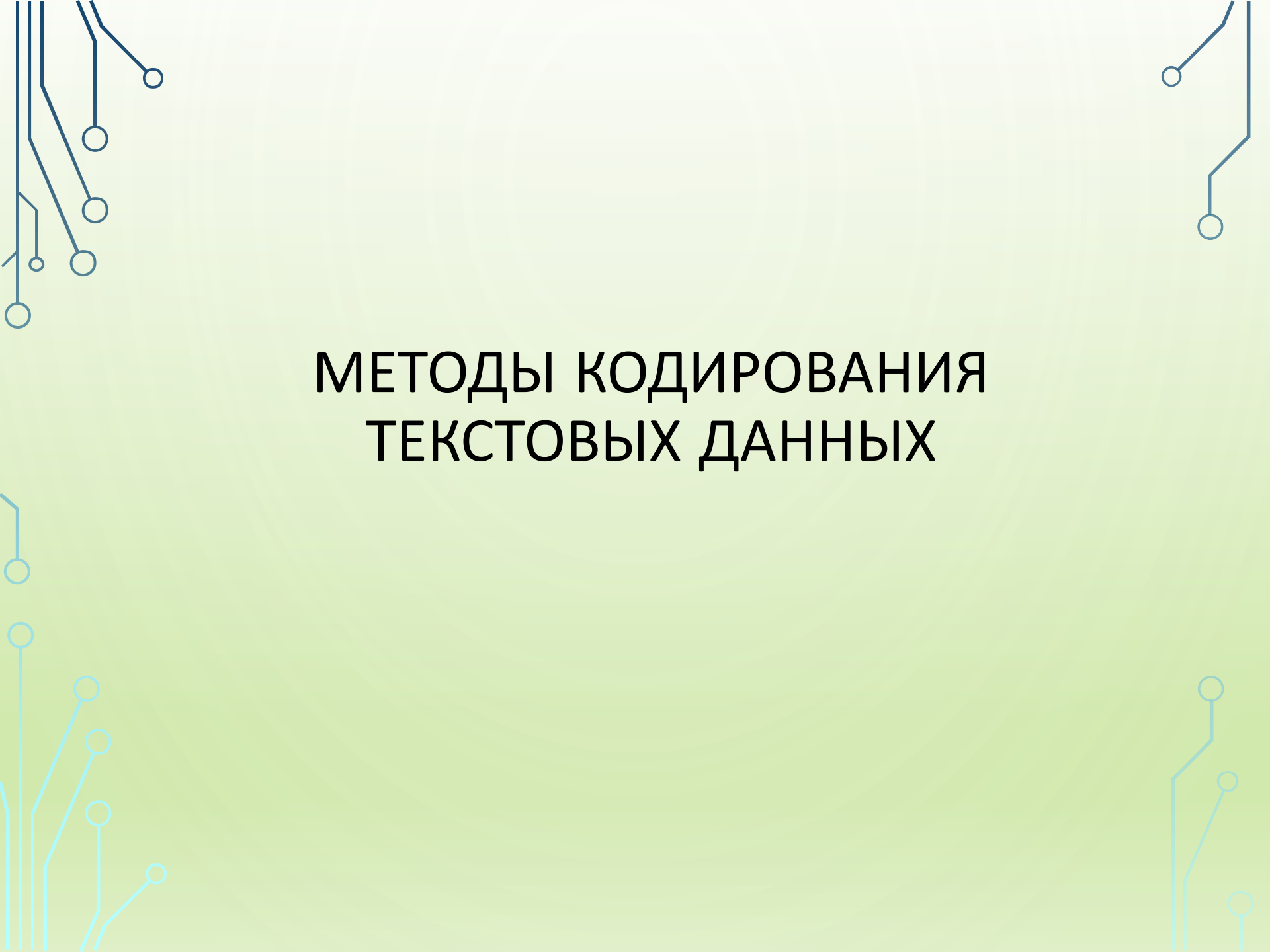
ТЕРМИНОЛОГИЯ

- 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.).



The background is a light green gradient. In the corners, there are decorative circuit-like patterns. The top-left and bottom-left corners feature dark blue lines, while the top-right and bottom-right corners feature light blue lines. These lines form various geometric shapes, including circles and rectangles, resembling a stylized circuit board.

МЕТОДЫ КОДИРОВАНИЯ ТЕКСТОВЫХ ДАННЫХ

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.

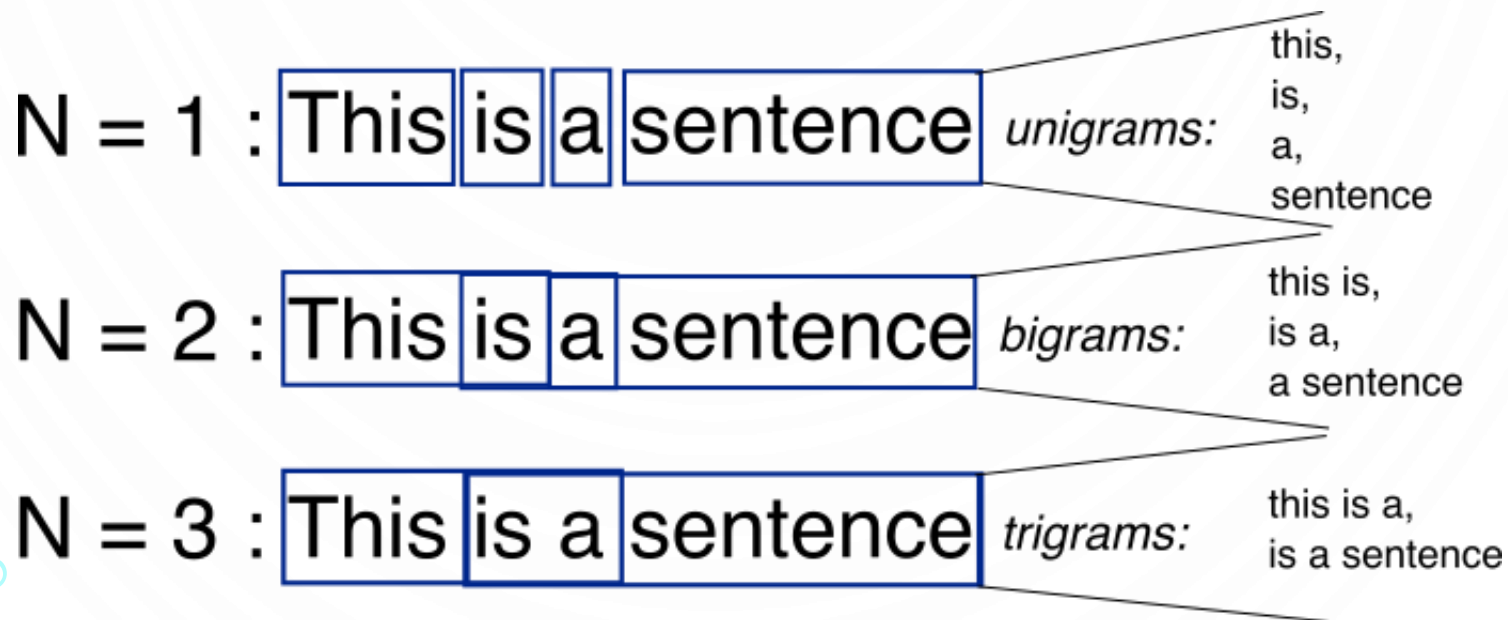
Пример: векторы документов “I have no cats” и “No, I have cats” будут идентичны.

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

WORD2VEC

- Word2Vec – векторизация слов (и текстов), полученная при помощи определенной архитектуры нейронной сети.
- Word2Vec опирается на предположение о том, что похожие слова находятся в похожих контекстах.

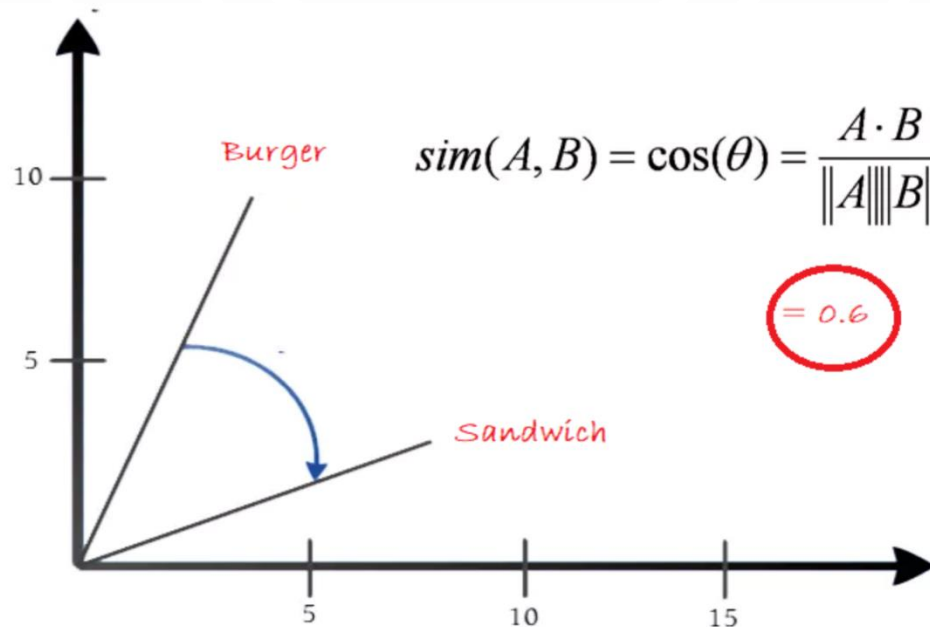
КОСИНУСНОЕ РАССТОЯНИЕ

- Скалярное произведение векторов x и y :

$$(x, y) = \|x\| \cdot \|y\| \cdot \cos(x, y)$$

В качестве расстояния между словами используется косинусное расстояние:

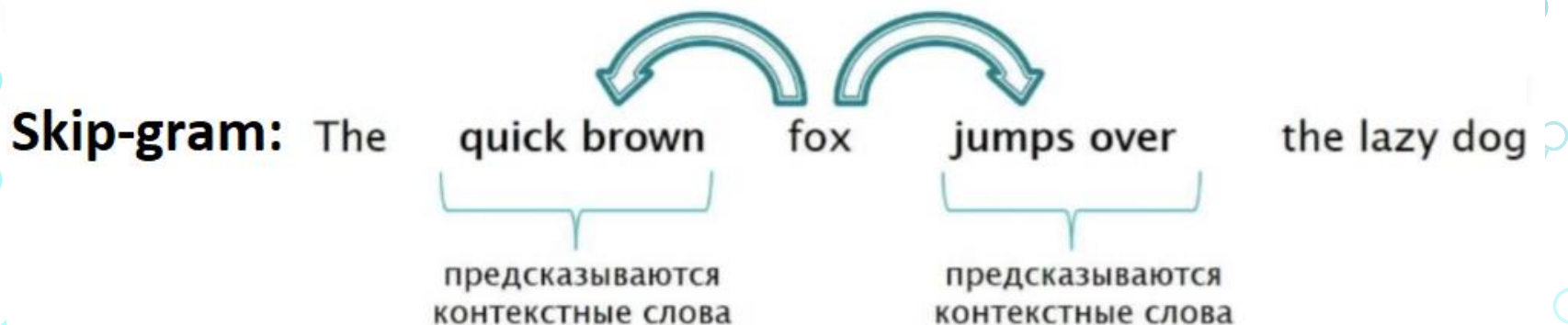
$$\rho(w_i, w_j) = \frac{(w_i, w_j)}{\|w_i\| \cdot \|w_j\|}$$



WORD2VEC

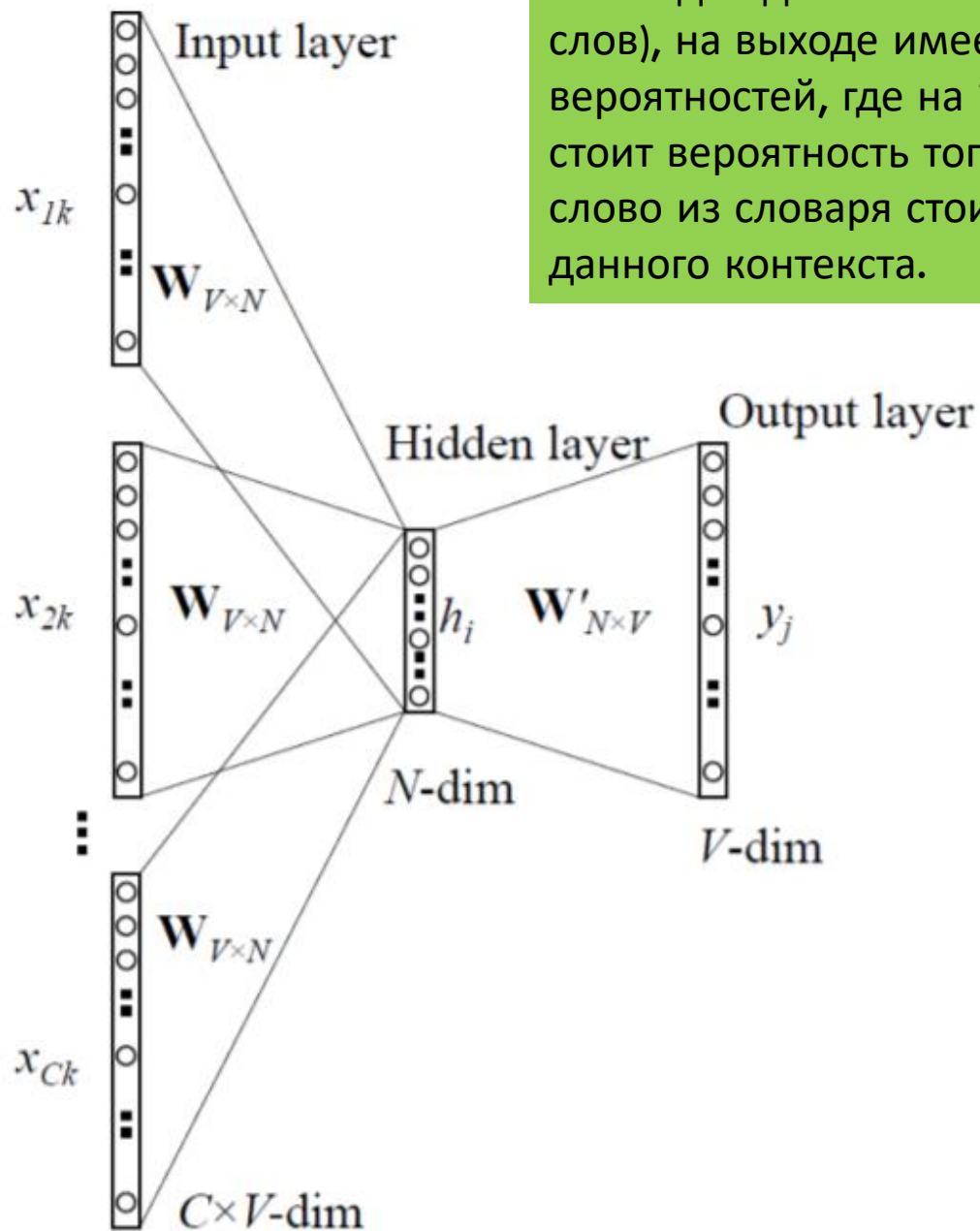
Есть две разные модели word2vec – CBOW и Skip-gram.

- CBOW (Continuous Bag of Words) предсказывает вероятность слова по данному контексту
- Skip-gram (“словосочетание с пропуском”) предсказывает по данному слову вектор контекста

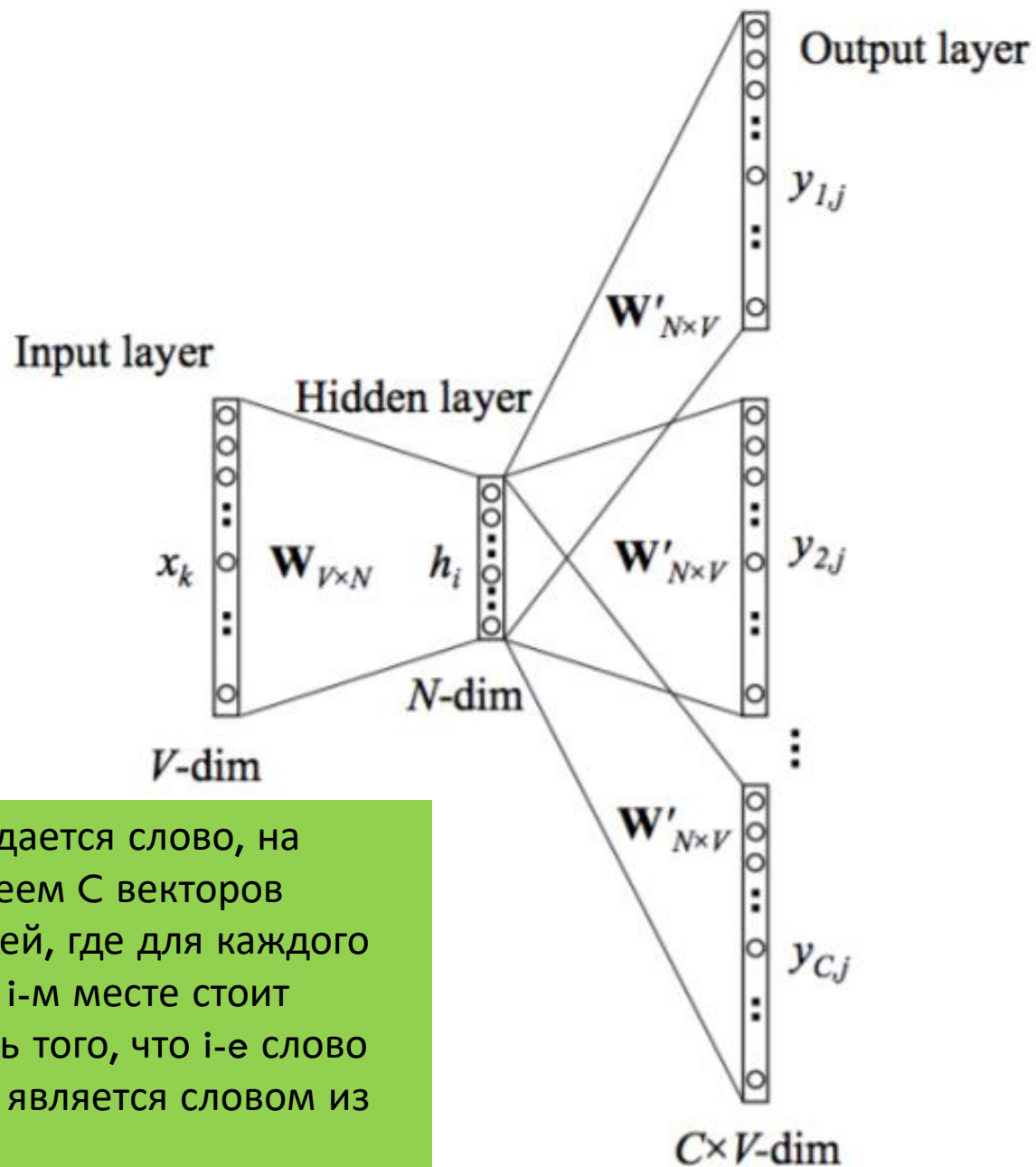


МОДЕЛЬ СВОВ

На вход подается контекст (С слов), на выходе имеем вектор вероятностей, где на i -м месте стоит вероятность того, что i -е слово из словаря стоит внутри данного контекста.

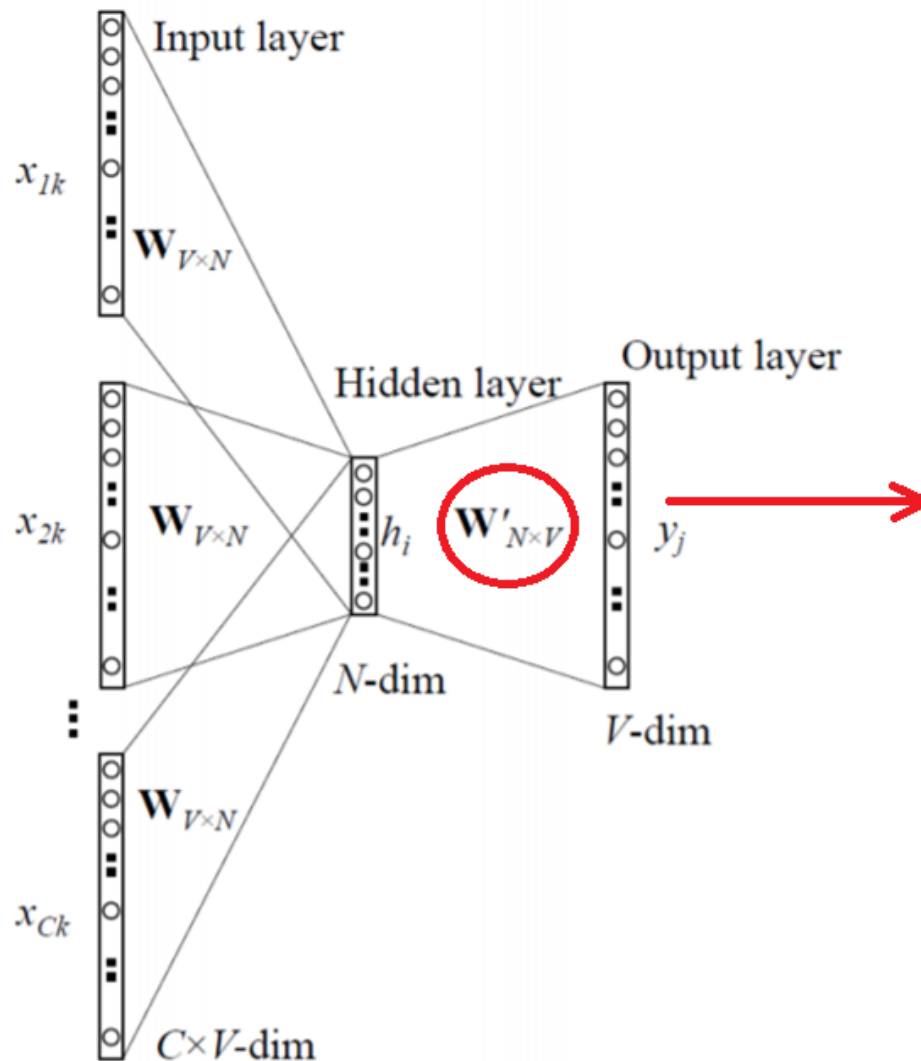


МОДЕЛЬ SKIP-GRAM



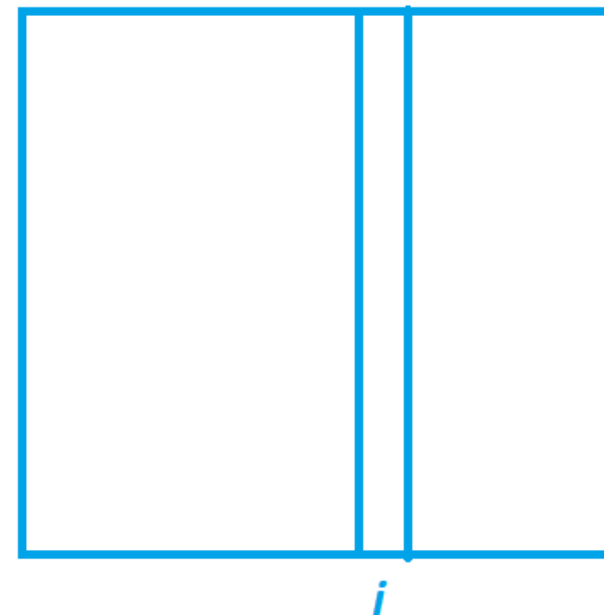
На вход подается слово, на выходе имеем C векторов вероятностей, где для каждого вектора на i -м месте стоит вероятность того, что i -е слово из словаря является словом из контекста.

ВЕКТОРЫ СЛОВ



N - количество нейронов на скрытом слое

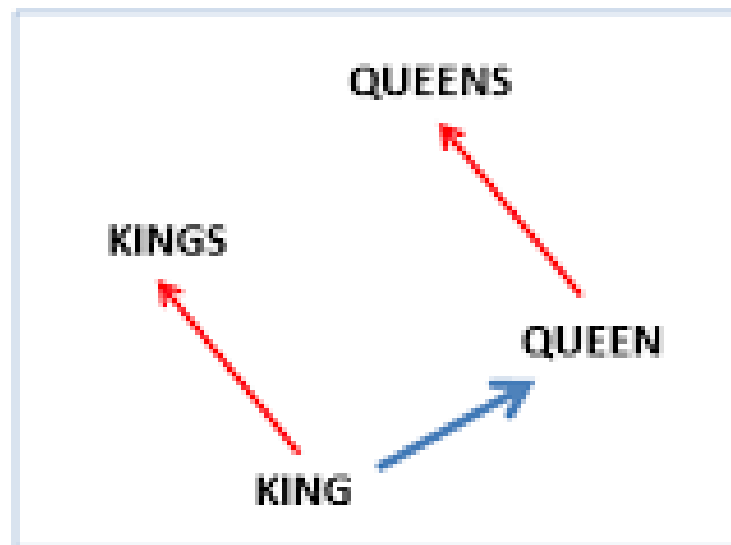
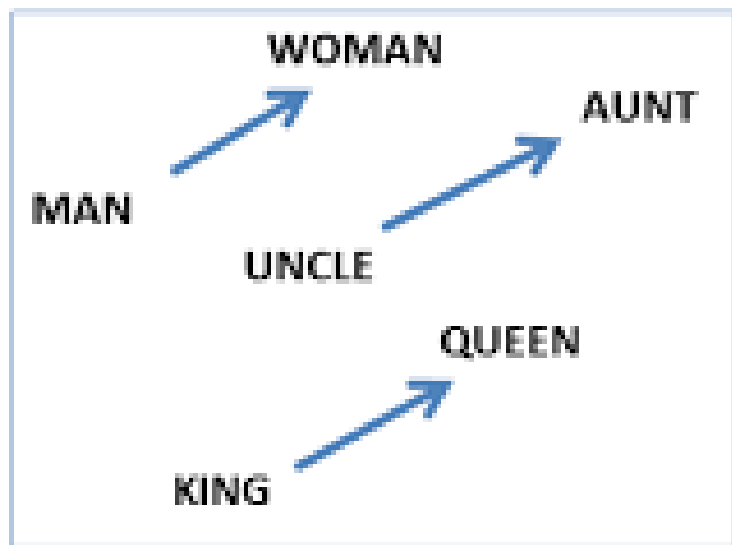
V - количество слов в словаре



Числа, стоящие в i -м столбце в полученной матрице весов – это word2vec-вектор длины N , представляющий i -е слово из словаря.

ВЕКТОРНЫЕ СООТНОШЕНИЯ МЕЖДУ СЛОВАМИ

За счёт использования косинусного расстояния между векторами слов, к векторам слов, полученных в результате применения word2vec, можно применять векторные операции сложения и вычитания, которые будут иметь смысл:



ТРАНСФОРМЕРЫ

- В октябре **2018** года Google выпустила модель кодирования текстовых данных под названием BERT – *Bidirectional Encoder Representations from Transformers*.
- Такой способ кодировать тексты даёт state-of-the-art результаты во многих задачах машинного обучения, связанных с обработкой естественного языка.

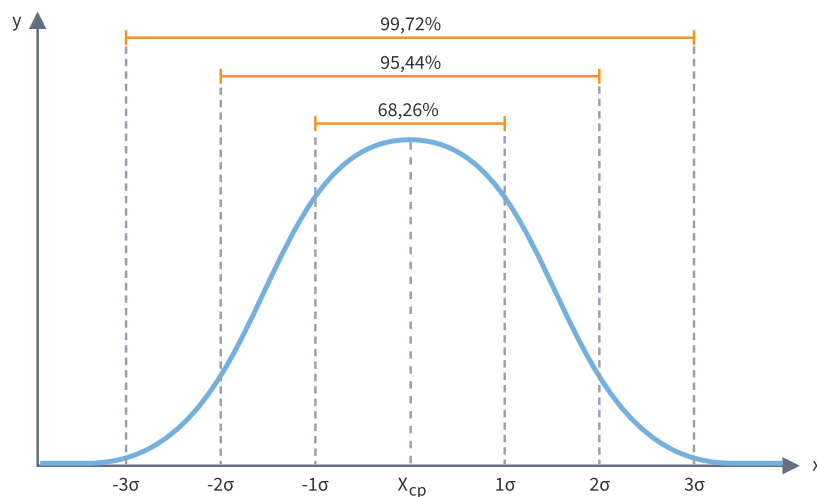
Почитать про трансформеры и механизм attention можно [здесь](#).

РАБОТА С ВЫБРОСАМИ

1. Статистические методы (правило трех сигм, интерквартильный размах).
2. Методы машинного обучения.

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.

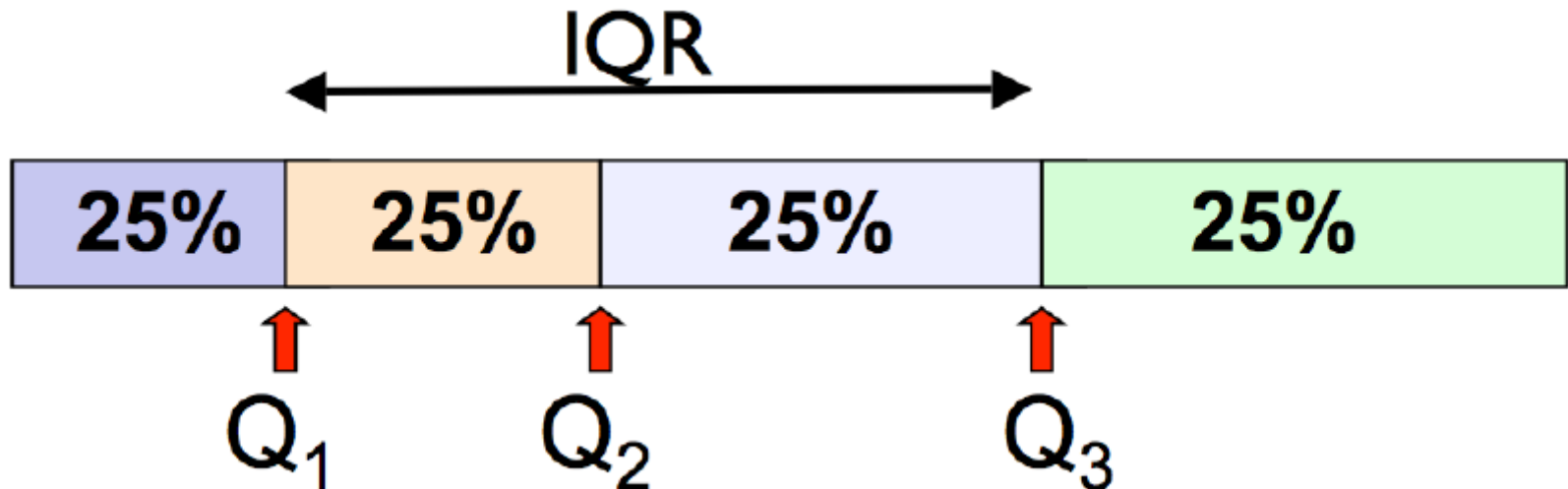


- Выбросами объявляются объекты, имеющие стандартное отклонение $\geq 3\sigma$ от математического ожидания.

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*.



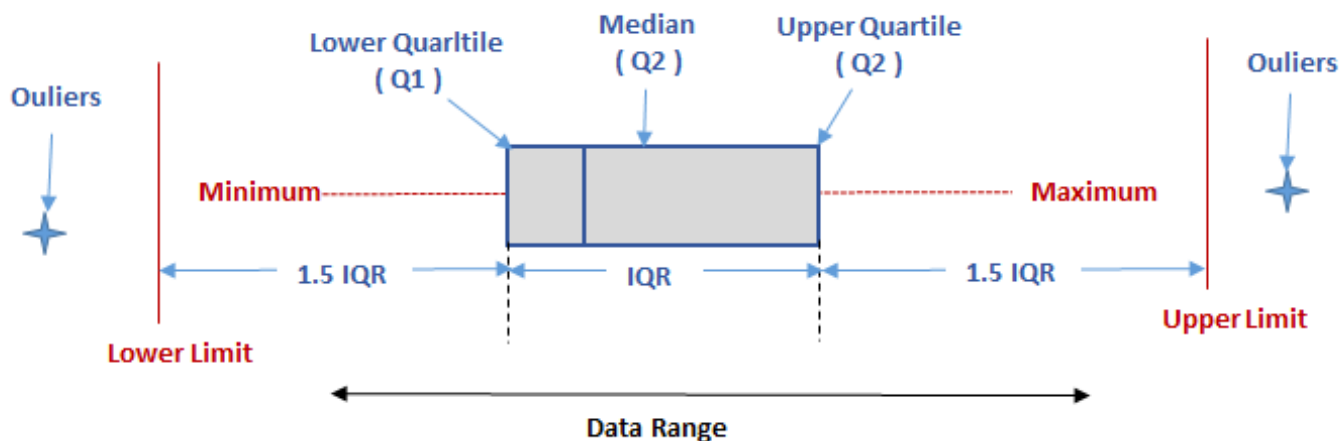
2. ИНТЕРКВАРТИЛЬНЫЙ РАЗМАХ

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

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

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

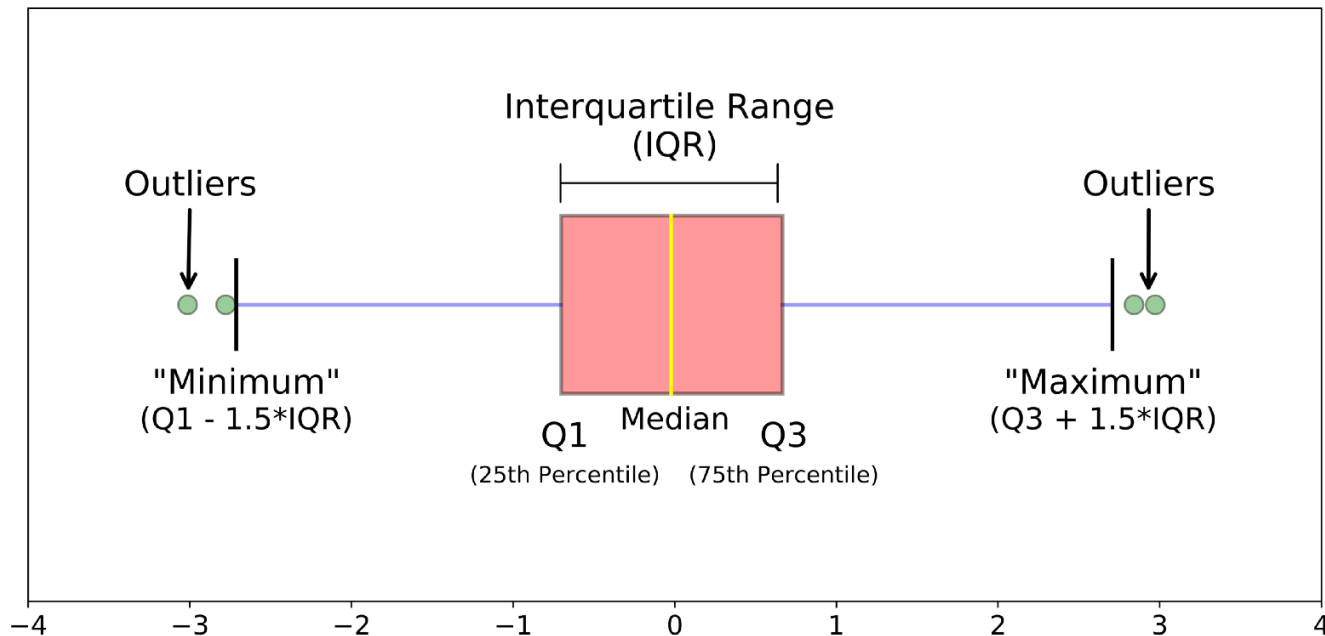
$$x < Q1 - 3 \cdot IQR \text{ или } x > Q3 + 3 \cdot IQR$$



ЯЩИК С УСАМИ

Ящик с усами – это диаграмма, которая показывает:

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



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

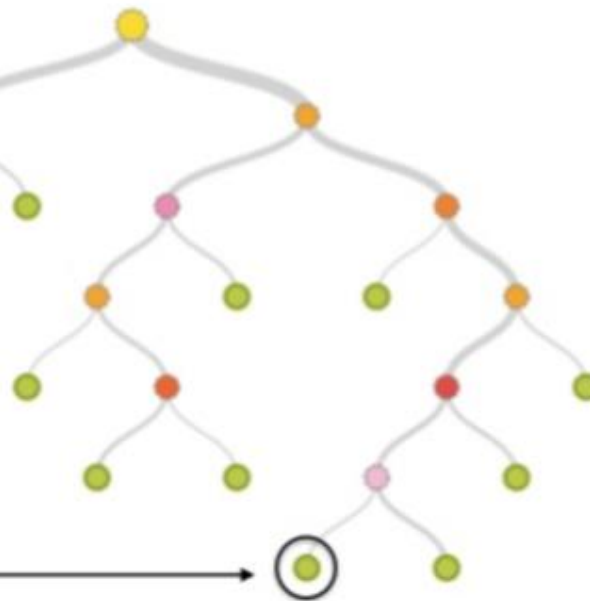
“easy” to isolate →



Depth

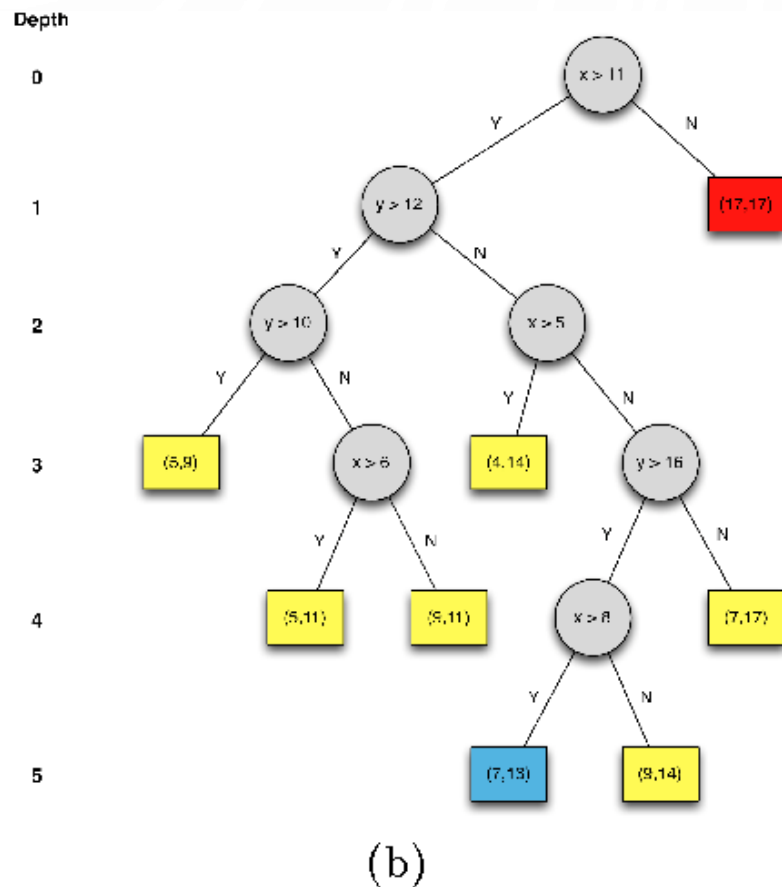
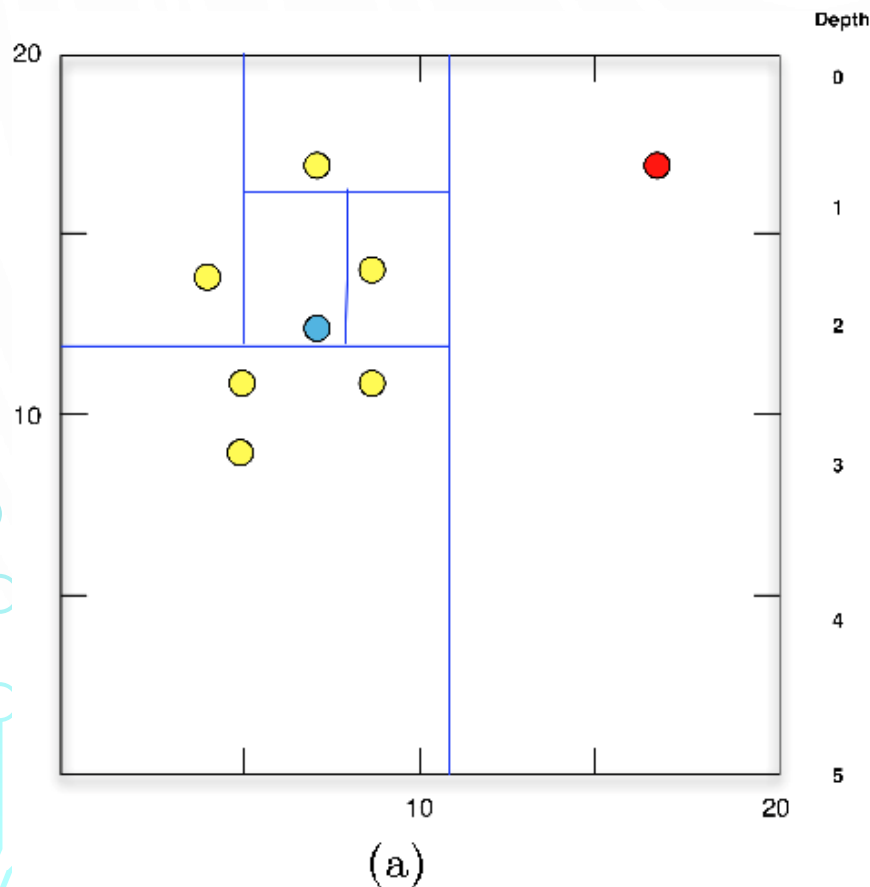
“hard” to isolate →

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.



ISOLATION FOREST

- Если объект единственный в листе, то его оценка аномальности в дереве – это глубина листа $h_n(x) = k$.

- Оценка аномальности объекта в Isolation Forest:

$$a(x) = 2^{-\frac{a}{b}},$$

где $a = \frac{1}{N} \sum h_n(x)$ – средняя глубина, где N – число деревьев в лесе,

$b = c(l)$ – средняя длина пути, посчитанная по всем объектам и всем деревьям в лесе, построенном по выборке размера l .

ПОИСК АНОМАЛИЙ С ПОМОЩЬЮ МОДЕЛЕЙ ML

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.

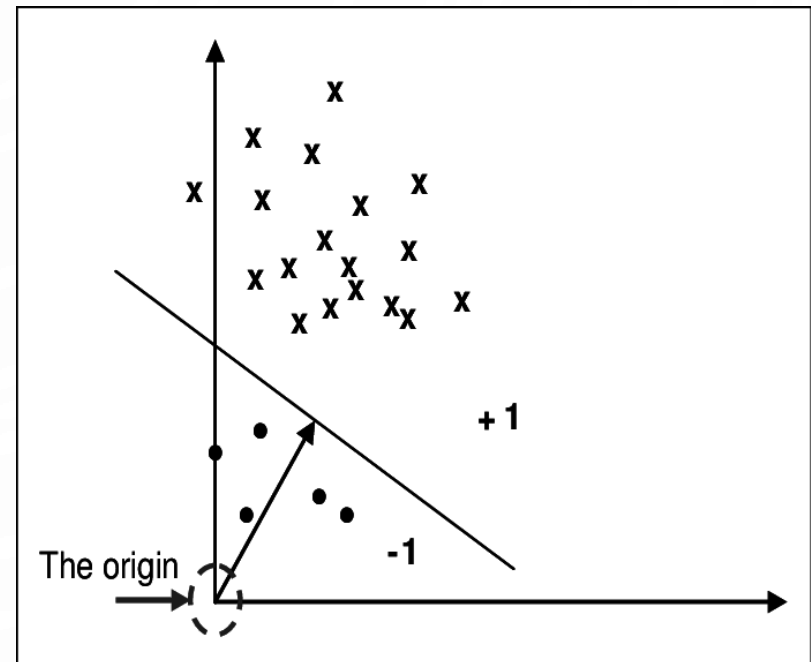
ONE-CLASS SVM

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: $a(x) = +1$ на области как можно меньшего объема, содержащей как можно больше объектов выборки
- имеет большой отступ от 0.

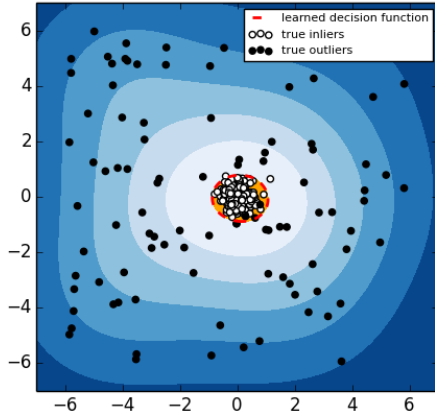
Тогда объекты с $a(x) = -1$

– это аномалии.



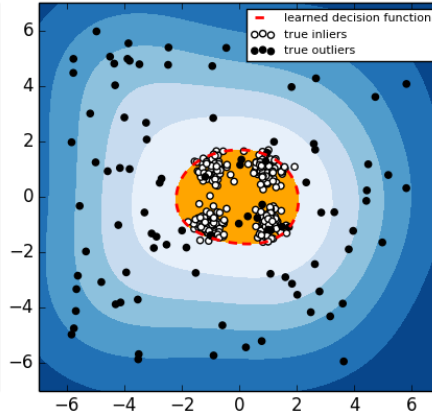
ONE-CLASS SVM С RBF-ЯДРОМ

Outlier detection



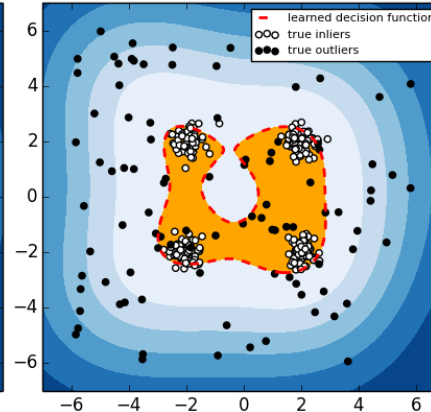
1. one class SVM (errors: 6)

Outlier detection



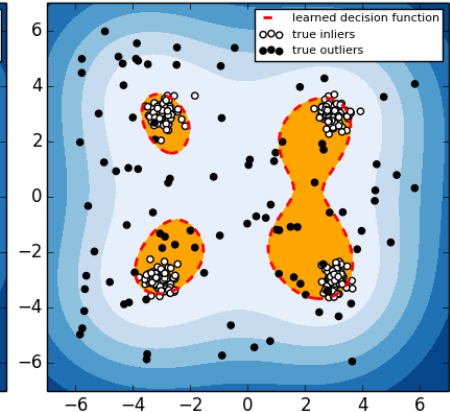
2. one class SVM (errors: 26)

Outlier detection



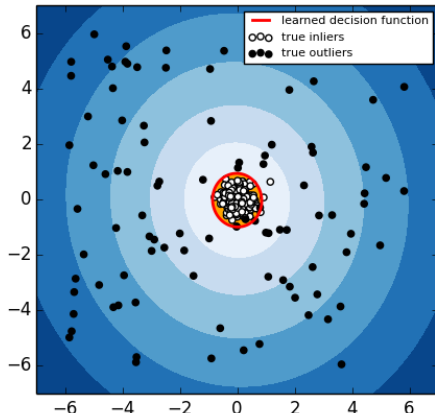
3. one class SVM (errors: 40)

Outlier detection



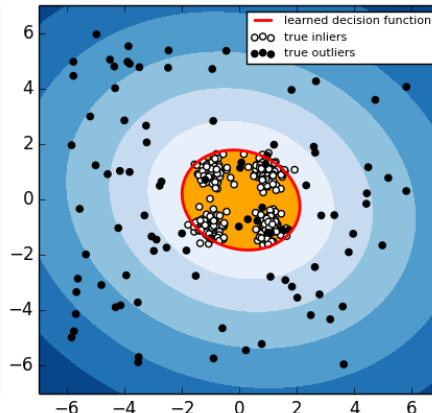
4. one class SVM (errors: 46)

Outlier detection



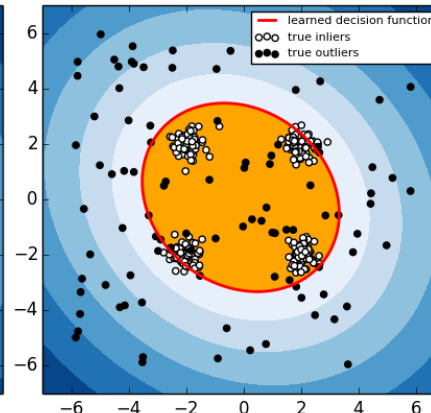
1. covariance estimation (errors: 6)

Outlier detection



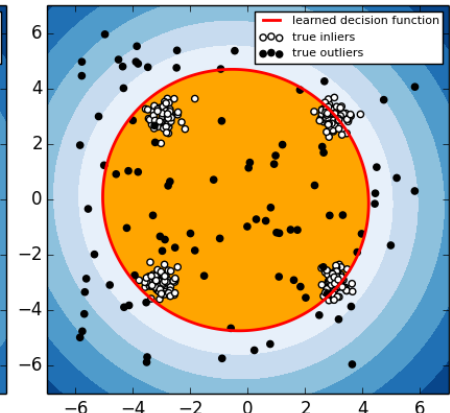
2. covariance estimation (errors: 26)

Outlier detection



3. covariance estimation (errors: 54)

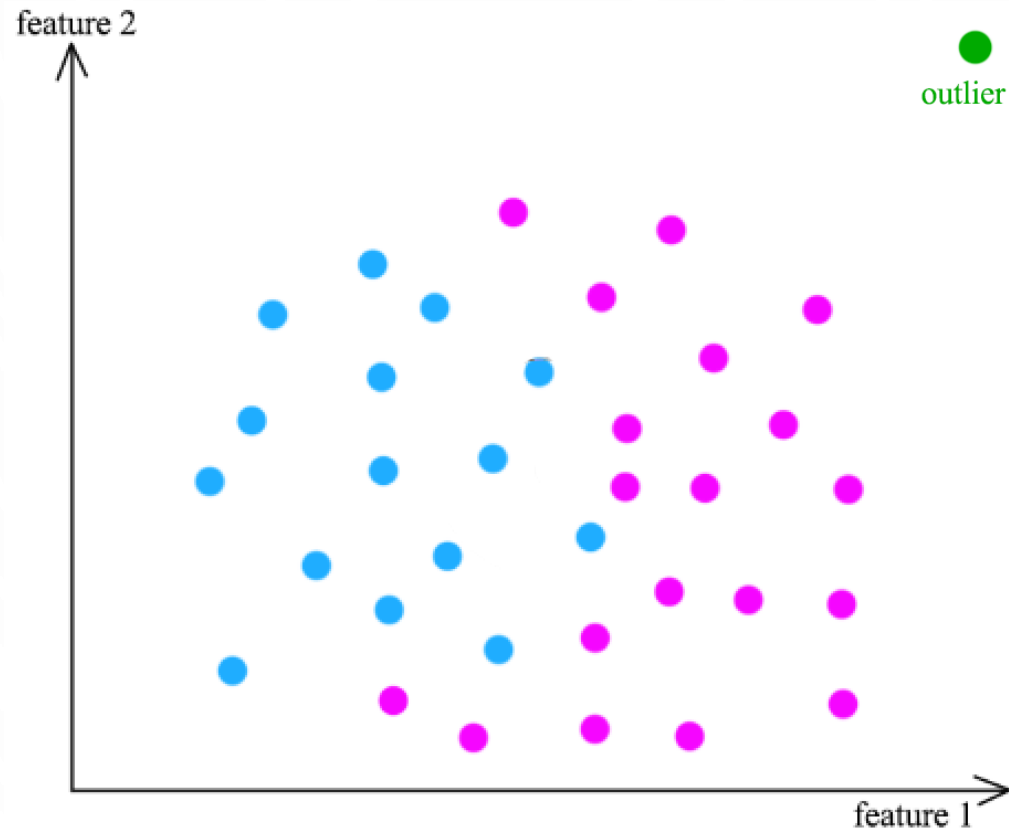
Outlier detection



4. covariance estimation (errors: 98)

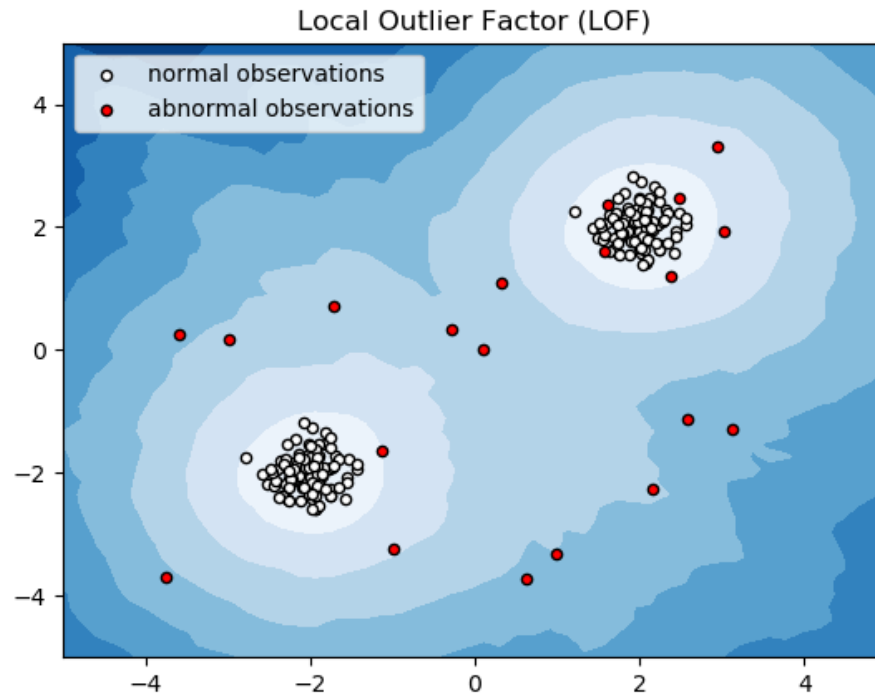
ПОИСК ВЫБРОСОВ С ПОМОЩЬЮ KNN

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



LOCAL OUTLIER FACTOR

- 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://github.com/yzhao062/pyod>