

# Метрические алгоритмы

## лекция 1

Алексей Ярошенко



Проверить,  
включена ли  
запись лекции



# Что сегодня будет

Договоримся о терминах и задачах

Ближайшие соседи

Как измерить расстояние

Алгоритм KNN

Проклятие размерности

Оценка качества

Развитие метрических алгоритмов

KNN в ноутбуке над текстами

# Термины ML



# Машинное обучение — ?

# Словарь

**Объект / точка** — то, для чего делаем предсказание. Стока в табличке данных. Картинка. Текст. И т.д.

**Признаки / факторы / фичи** (X) — вектор характеристик, описывающих объект.

**Целевая переменная / таргет / лейбл / ответы** (y) — то, что предсказываем

```
N = 1000
d = 5
df = pd.DataFrame(np.random.randn(N, d), columns=[f'feat_{i}' for i in range(d)])
df['label'] = np.random.randint(0, 2, N)
df
```

	feat_0	feat_1	feat_2	feat_3	feat_4	label
0	1.501396	1.188850	0.654410	0.513332	-0.187409	0
1	-0.002910	-0.289089	0.098120	0.499454	0.382641	0
2	0.190024	0.511595	-1.475563	-0.834804	1.517679	1
3	0.096616	-0.713458	-0.781668	1.899929	-0.547762	0
4	1.763597	-0.133891	0.049155	-0.409800	-2.108633	1

# Словарь

**Модель** — алгоритм, который распознает закономерности определенного типа. Когда мы обучаем модели, мы моделируем что-то. Результат моделирования — модель.

**Инференс** — запуск обученной модели для получения предсказаний

**Глубокое обучение** — раздел машинного обучения, где модели — нейронные сети

# Задачи ML



Разделение условно и одни задачи могут  
сводиться к другим

## Базовые типы задач

- **Обучение с учителем (supervised learning)** — есть целевая переменная
- **Обучение без учителя (unsupervised learning)** — нет целевой переменной
- **Частичное обучение (semi-supervised learning)** — целевая переменная есть не везде
- **Обучение с подкреплением (reinforcement learning)** — агент в среде учится максимизировать награду

# Регрессия

Предсказываем обычное вещественное число (12.954)

- Цена квартиры
- Координата на картинке
- Прогноз будущей выручки
- Предсказание возраста по фото

# Классификация

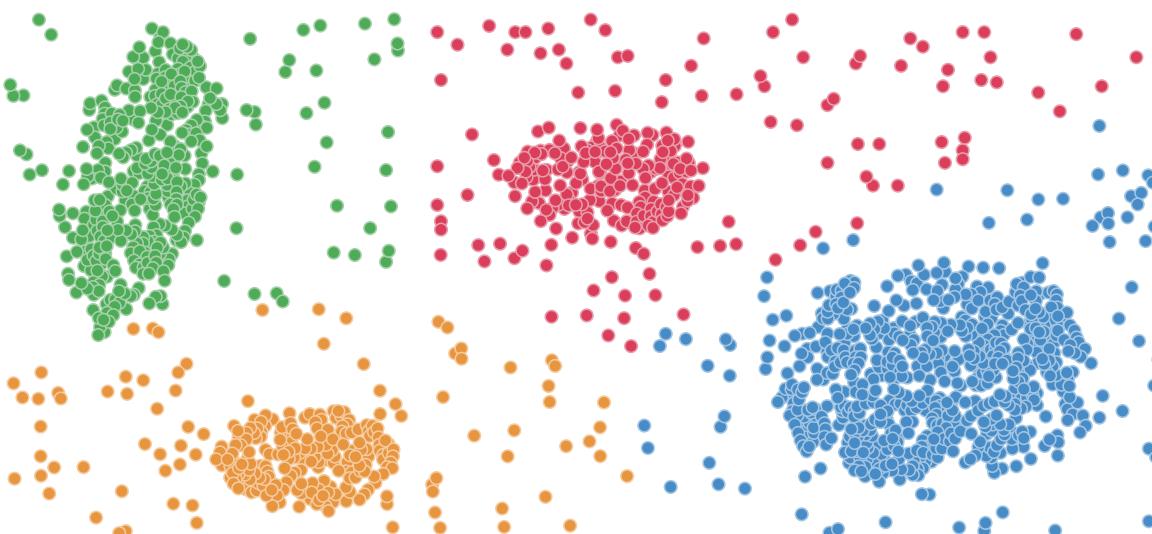
Предсказываем метку/метки класса для объекта

- Бинарная классификация  $Y = \{0,1\}$ 
  - *спам / не спам*
- Мноклассовая классификация  $Y = \{0,1,\dots,K\}$ , у объекта 1 класс
  - *апельсин / банан / груша*
- Multilabel классификация  $Y = \{0,1\}^K$ , может быть несколько классов
  - *теги отзыва: место, цены, обслуживание, кухня, меню*

# Кластеризация

Группирует объекты в кластера с похожими характеристиками

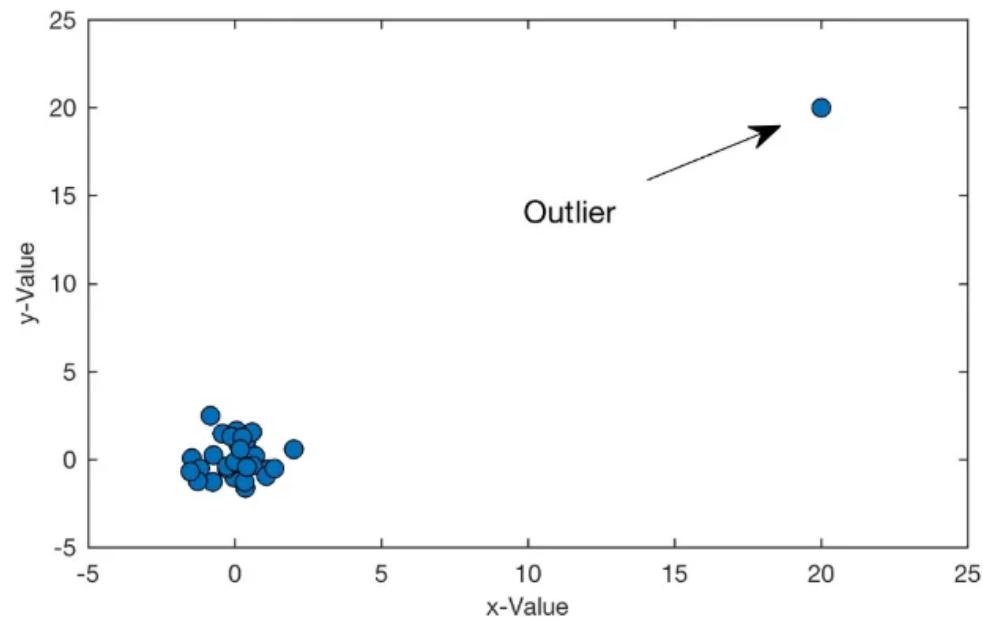
Пример: у нас есть клиенты магазина, нужно разделить их на какие-то сегменты. И мы не знаем, что за сегменты. На картинке - раскрасить.



# Обнаружение аномалий / выбросов

Формально похоже на бинарную классификацию, но для очень редких событий. Т.е. предсказываем то, чего еще не было или случалось очень редко. Нет лейблов положительного класса (1) или очень мало

- Мошеннические транзакции
- Сетевые атаки
- Обнаружения поломки систем



# Ранжирование

ранжирование что это простыми словами ×

**поиск** картинки видео карты переводчик все

 **Значение слова РАНЖИРОВАНИЕ.** Что такое...  
[kartaslov.ru](#) > значение-слова/ранжирование  
Значение слова «ранжирование». Ранжирование — сортировка сайтов в поисковой выдаче, применяемая в поисковых системах. Читать ещё

 **Ранжирование — Википедия**  
[ru.wikipedia.org](#) > Ранжирование  
Ранжирование — сортировка сайтов в поисковой выдаче, применяемая в поисковых системах. Существует множество факторов для ранжирования...

 **Ранжирование - что это простыми словами, факторы...**  
[otzyvmarketing.ru](#) > articles/chto-takoe-...  
В статье расскажем, как именно поисковые системы ранжируют сайты и как можно получить от них высокий ранг и попасть на первые строчки выдачи. Как работает ранжирование. Читать ещё

Не только сайтов, если что :)

# Рекомендации

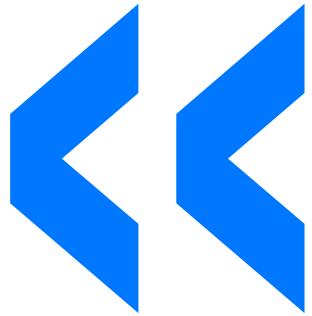
Рекомендуем пользователю наиболее релевантные объекты.

И наоборот.

- Этому пользователю будут интересны эти товары
- Этот товар можно рекомендовать этим пользователям

# Понижение размерности

Снижение числа признаков с помощью выделения главного



# Вопрос на собеседовании (забегая вперед)

А зачем снижать размерность?

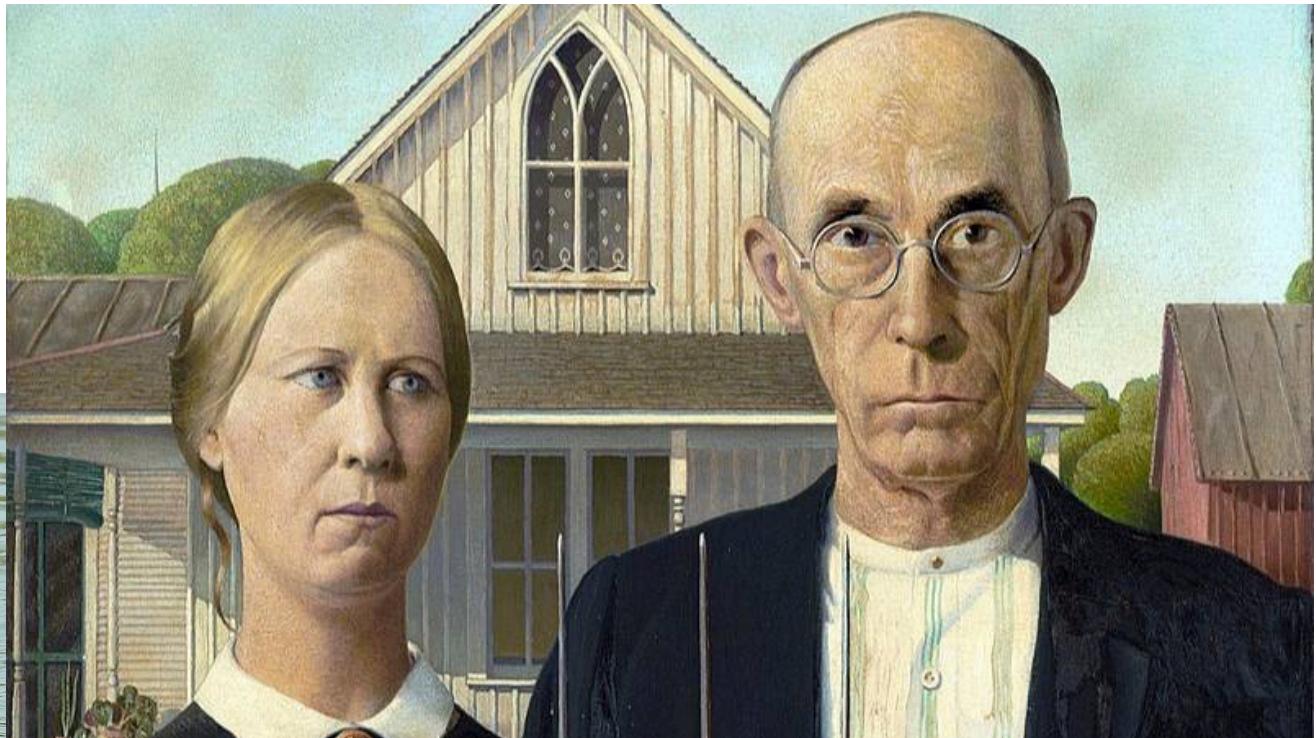
# Задач, если захочеть, можно выделить еще много

- Сегментация
- Детекция
- Поиск распределения
- Визуализация
- Детекция
- Сегментация
- NER
- OCR
- Генерация текста
- ...

# Ближайшие соседи. KNN



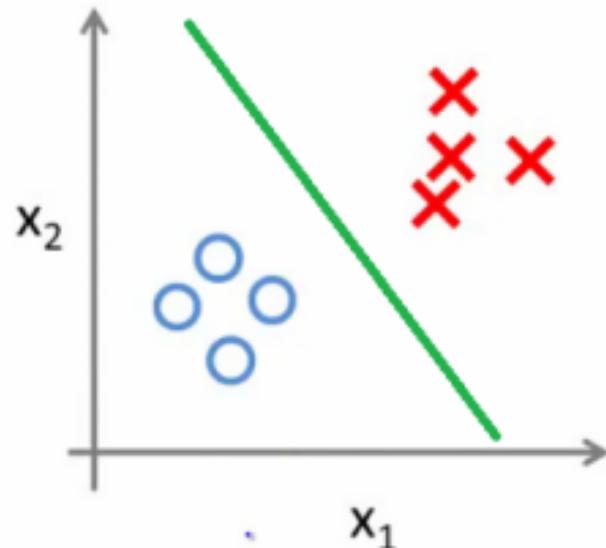
Идея: что близко, то  
похоже



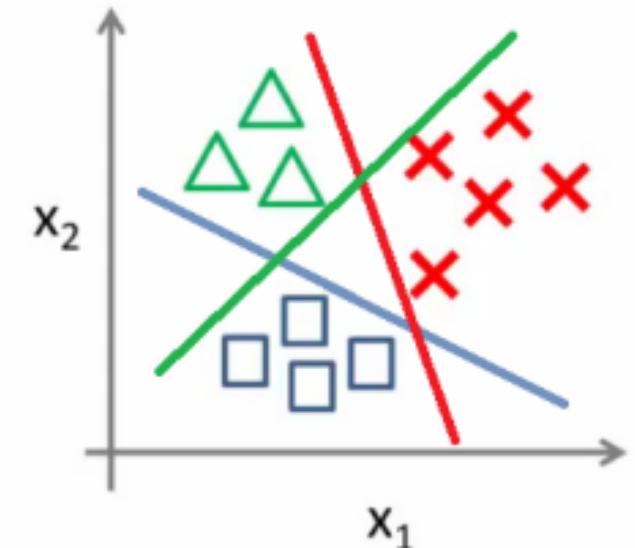
# Гипотеза компактности: похожие объекты лежат в одном классе

- Кадровики — в отделе кадров, бухгалтера в бухгалтерии и т.д.
- Следствие: если человек пришел на мероприятие, и вокруг него одни микробиологи, скорее всего он тоже микробиолог

Binary classification:



Multi-class classification:

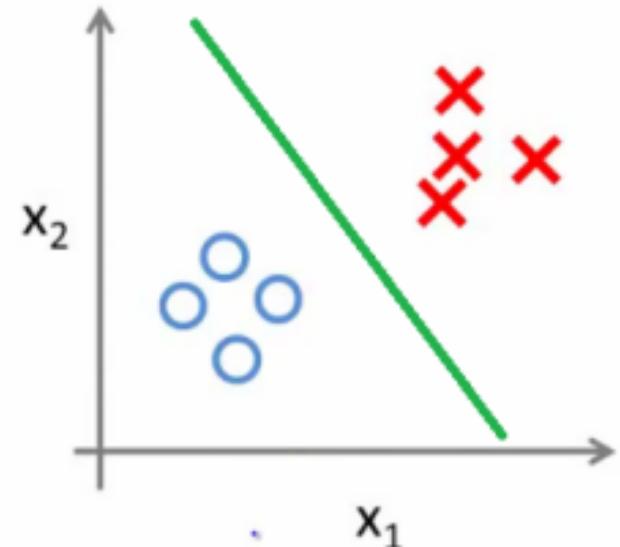


# Классификация

$$a(x, X_{train}) = \operatorname{argmax}_c \sum_{i=1}^N w(x, x_i) I[y_i = c], x_i \in X_{train}$$

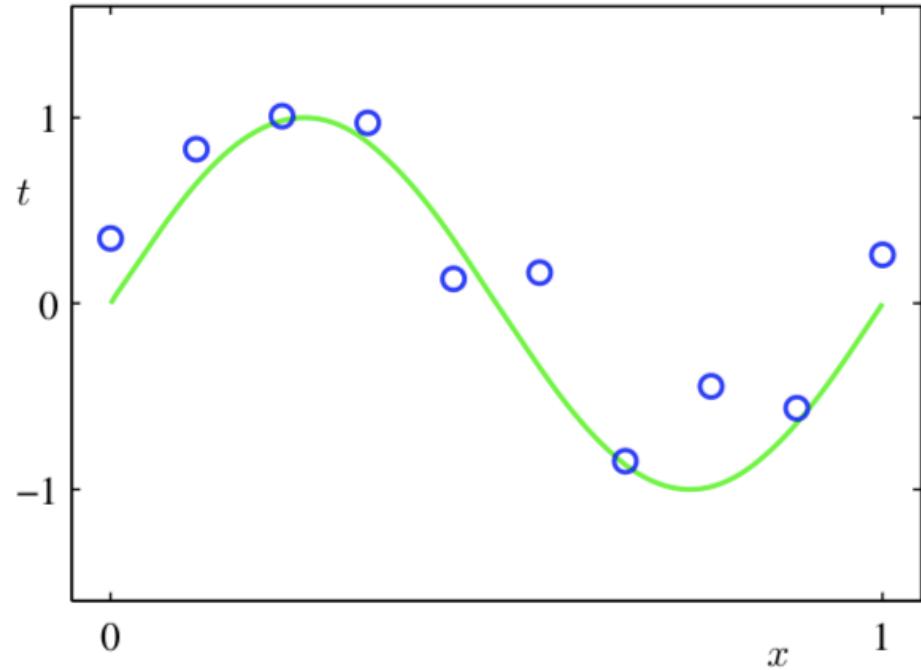
- $w(x, x_i)$  – какая-то функция веса между объектами. Какая, разберем дальше. Но пока, давайте представим, что это функция близости

Binary classification:



# Гипотеза непрерывности: у похожих объектов похожий ответ

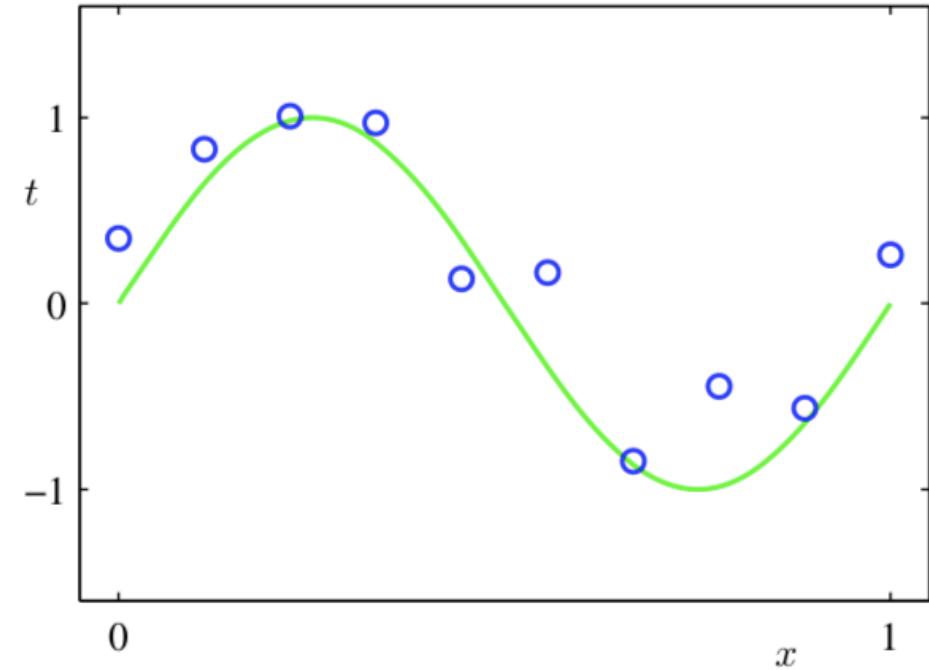
- В Москве 17 градусов, в Минске 17 градусов. В Смоленске, скорее всего, тоже сейчас 17 градусов
- Позавчера продали 147 плюшевых мишек, вчера 153 плюшевых мишки, сегодня, скорее всего продадим  $\sim 150$  мишек



# Регрессия

$$a(x, X_{train}) = \frac{\sum_{i=1}^N w(x, x_i) y_i}{\sum_{i=1}^N w(x, x_i)}, x_i \in X_{train}$$

- По сути, взвешенное среднее



## А что за функция веса $w(x, x_i)$ ?

- Если ненулевой вес только у ближайшего объекта, то алгоритм называют алгоритмом ближайшего соседа
- Если ненулевые веса для  $k$  ближайших объектов, то алгоритм называют алгоритмом  $k$  ближайших соседей ( **$k$ -nearest neighbors, knn**).

Пусть  $x_i$  —  $i$ -тый ближайший сосед объекта  $x$

- $w(x, x_i) = 1/k$
- $w(x, x_i) = \frac{k+1-i}{k}$
- $w(x, x_i) = \alpha^i, \alpha \in (0,1)$

## Добавим расстояния между объектами

Проблема: в прошлом варианте никак не учитываем величину расстояния.

$$w(x, x_i) = K(\rho(x, x_i)),$$

где  $K(x)$  – любая монотонно убывающая функция. Близость.

Чем меньше расстояние, тем значение функции больше.

Примеры:

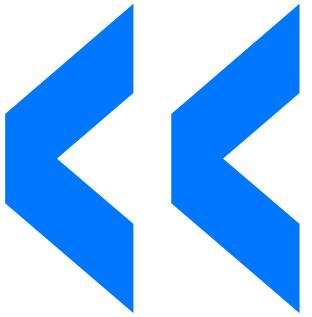
- $K(x) = \frac{1}{x + \beta}$
- $K(x) = \exp(-x)$
- $K(x) = \alpha^x, \alpha \in (0,1)$

# Модель непараметрическая

- В ней нет обучаемых параметров
- Ей даже не нужны признаки объектов, достаточно расстояний между объектами

```
m = torch.jit.load('bert_distil/roberta_success_v1_registry.pt')

list(m.parameters())
tensor([[-0.0003, -0.0362, -0.0601, ..., -0.0852,  0.0190, -0.0467],
       [-0.0455,  0.0244, -0.0173, ..., -0.0271, -0.0267,  0.0173],
       [-0.0471,  0.1600, -0.0999, ..., -0.0685, -0.0381,  0.0086],
       ...,
       [-0.0945,  0.1559,  0.0098, ..., -0.0165,  0.0969, -0.1915],
       [-0.1043, -0.0716, -0.0965, ...,  0.0511, -0.0386, -0.0526],
       [-0.0600,  0.0379,  0.0511, ..., -0.0447,  0.0817, -0.0166]],
      device='cuda:0', dtype=torch.float16, requires_grad=True),
tensor([[ -8.4305e-03,   6.0720e-03,   1.6332e-05, ...,  1.5364e-03,
          -1.4183e-02,   1.4261e-03],
       [ 3.3493e-03,   9.6798e-04,   8.8959e-03, ..., -4.8714e-03,
          -1.591e-03,  -7.7782e-03],
       [ -5.0720e-02,  -4.6722e-02,   3.1235e-02, ..., -2.7740e-02,
          -1.0828e-01,  -3.6865e-01],
```



# Вопрос на собеседовании

Какая асимптотическая сложность по памяти у алгоритма KNN?

# Как измерять расстояние



# Метрика Минковского

Аксиомы:

$$1. \rho(x, y) = 0, \text{ т.и.т.д } x = y$$

$$2. \rho(x, y) = \rho(y, x)$$

$$3. \rho(x, z) \leq \rho(x, y) + \rho(y, z)$$

Пусть  $D$  – число признаков. Метрика Минковского (при  $p \in (0, 1)$  не метрика):

$$\rho(x, y) = \left( \sum_{j=1}^D |x_j - y_j|^p \right)^{\frac{1}{p}}$$

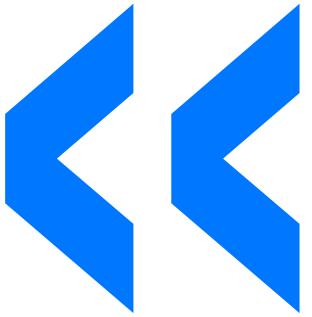
## Метрика Минковского, частные случаи

$p = 2$  – Евклидово расстояние

$$\rho(x, y) = \sqrt{\sum_{j=1}^D (x_j - y_j)^2}$$

$p = 1$  – Манхэттенское расстояние

$$\rho(x, y) = \sum_{j=1}^D |x_j - y_j|$$



## Вопрос на собеседовании

В данных есть значительные выбросы, и вы учите KNN.  
Какое расстояние вы будете использовать, L1 или L2

# Манхэттенское и евклидово наглядно

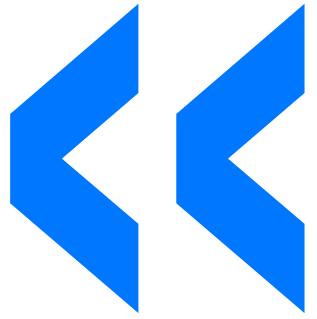


# Косинусное расстояние

По определению скалярного произведения считаем угол между векторами:

$$\text{sim}(x, y) = \cos\alpha = \frac{x \cdot y}{|x| |y|}$$

$$\text{rho}(x, y) = 1 - \text{sim}(x, y)$$



## Вопрос на собеседовании

Косинусное расстояние обычно используют для текстов. Почему именно оно?

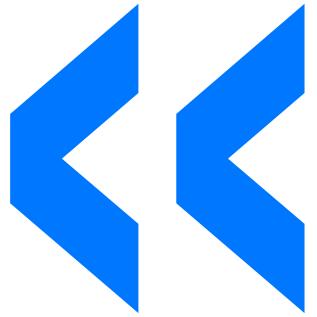
# Нормировка признаков

Нельзя так считать расстояния с признаками разных масштабов!

Два вида нормировки:

1. Стандартизация —  $x^j = \frac{x^j - \text{mean}(x^j)}{\text{std}(x^j)}$

2. Нормализация —  $x^j = \frac{x^j - \text{min}(x^j)}{\text{max}(x^j) - \text{min}(x^j)}$



## Вопрос на собеседовании

В каком диапазоне будет лежать признак при нормализации и при стандартизации?

# Расстояние Хэмминга

Число категориальных признаков, которые имеют разные значения

$A$	1	0	1	1	0	0	1	0	0	1
			‡				‡		‡	
$B$	1	0	0	1	0	0	0	0	1	1

**Расстояние Хэмминга = 3**

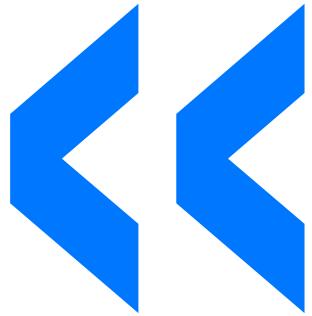
# Расстояние Джаккарда

Как измерить расстояние между множествами?

Например, предложение — мешок (множество) слов.

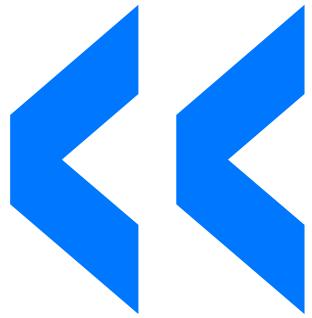
$$IoU = \frac{|X \cap Y|}{|X \cup Y|} - \text{Intersection over Union, Jaccard index}$$

$$\rho(X, Y) = 1 - IoU - \text{Jaccard distance}$$



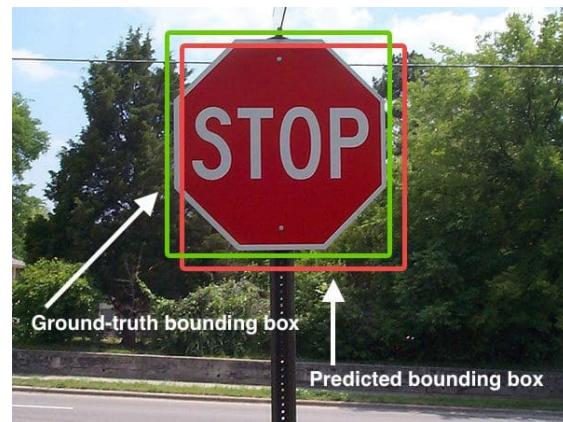
## Вопрос на собеседовании

Где в задачах машинного обучения принято использовать расстояние IoU?



# Вопрос на собеседовании

Где в задачах машинного обучения принято использовать расстояние Джаккарда?



Детекция



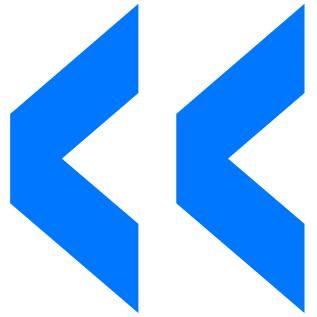
Сегментация

# Расстояние Левенштейна

Минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

- $\rho(\text{огурец}, \text{агурец}) = 1$
- $\rho(\text{мама}, \text{папа}) = 2$
- $\rho(\text{длинный}, \text{длиныЙ}) = 1$

*В каких задачах часто применяется расстояние Левенштейна?*



# Вопрос на собеседовании

Где применяется расстояние Левенштейна?

# Резюмируем алгоритм KNN



# Алгоритм KNN

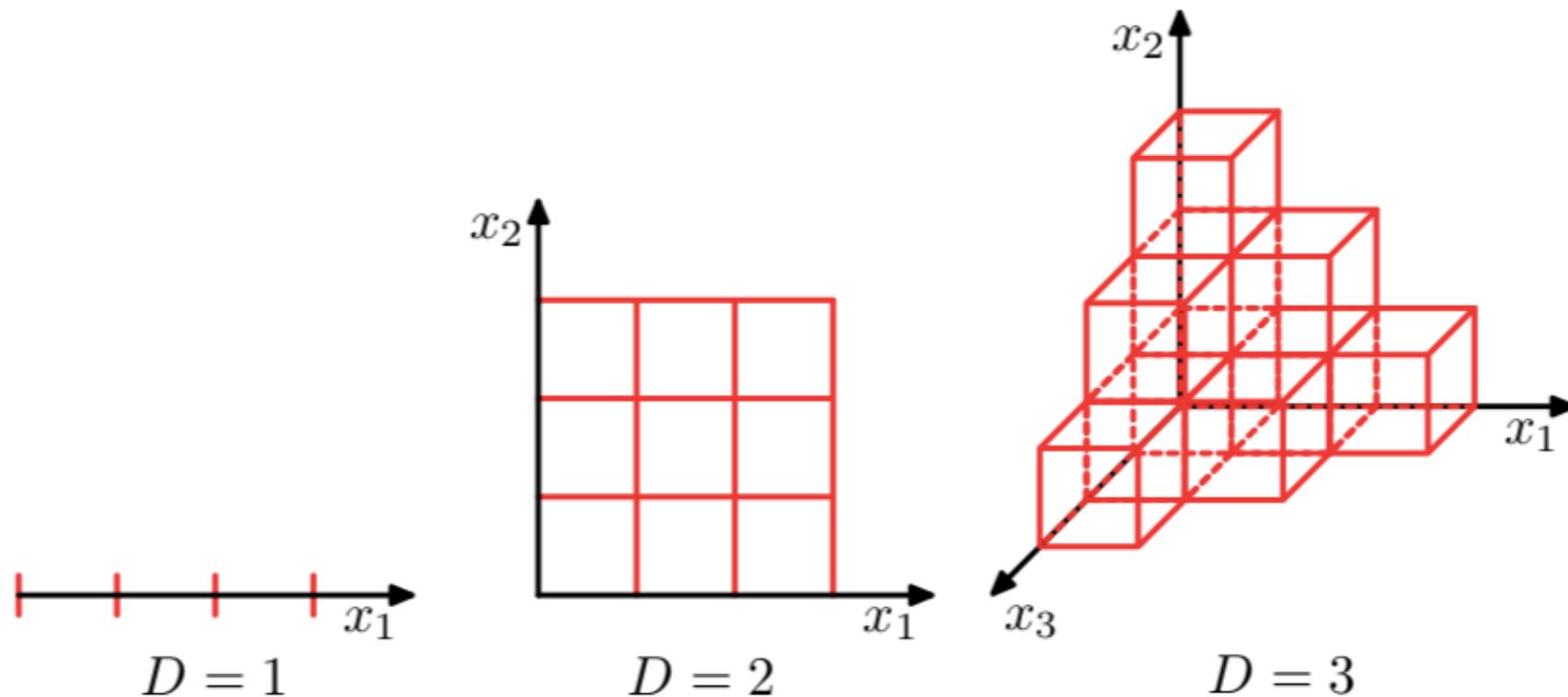
1. Посчитать расстояния между пришедшим на вход объектом и всеми объектами обучающей выборки
2. Отобрать К ближайших соседей (**K Nearest Neighbor**)
3. Посчитать ответ алгоритма на основе этих соседей:
  - Для классификации — самый частый класс
  - Для регрессии — среднее значение соседей

# Проклятье размерности



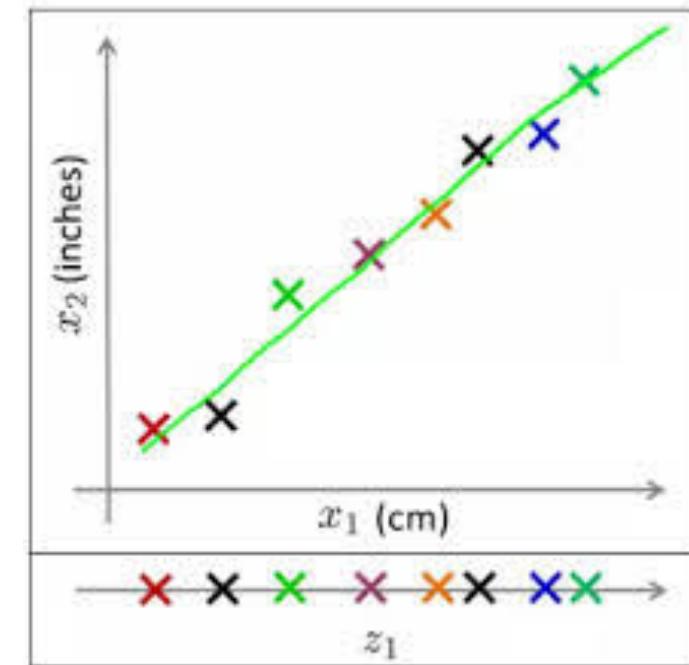
# Проклятие размерности. Что пожет пойти не так?

В пространстве большой размерности объекты почти равномерно удалены друг от друга. И метрические алгоритмы не видят разницы между объектами



# Уменьшаем размерность

- Методы снижения размерности
- Методы отбора признаков



# Отбираем признаки

Задача: найти и удалить ненужные признаки.

Какие признаки для нас ненужные?

- Перебрать все варианты и посмотреть качество (лучший, если признаков мало)
- Посчитать корреляцию с целевой функцией и удалить шумные
- Посчитать корреляцию всех пар признаков и удалить скоррелированные
- Последовательно удалять худшие
- Последовательно добавлять лучшие
- ....

# Оценка качества



# Оценка качества в задачах регрессии

$$Q(a, X, Y) = \frac{1}{N} \sum_{i=1}^N L(a, x_i, y_i), x_i \in X, y_i \in Y$$

Самые распространенные метрики:

- **MSE** – Квадратичная

$$L(a, x, y) = (a(x) - y)^2$$

- **MAE** – Абсолютная

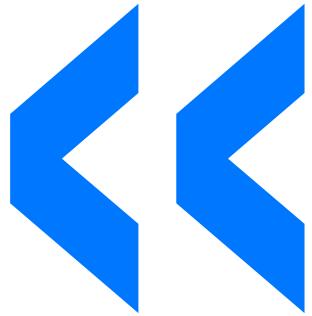
$$L(a, x, y) = |a(x) - y|$$

- **MAPE** – Абсолютная в процентах

$$L(a, x, y) = \frac{|a(x) - y|}{y}$$

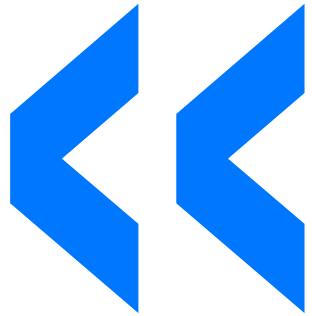
- **MSLE** – Логарифмическая

$$L(a, x, y) = (\log(a(x) + 1) - \log(y + 1))^2$$



## Вопрос на собеседовании

MSLE — это относительная или абсолютная метрика?



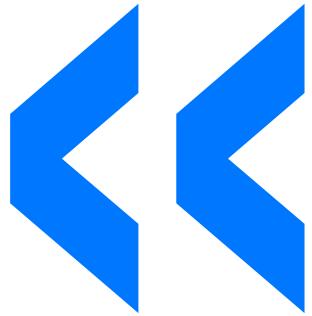
# Вопрос на собеседовании

Когда лучше использовать MAE, а когда MSE и почему?

# Оценка качества в задачах классификации

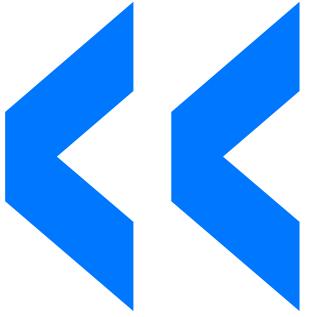
- **Accuracy** (точность) – процент правильно классифицированных объектов  $L(a, x, y) = [a(x) = y]$
- **Precision** (аккуратность??) – процент правильно классифицированных объектов класса 1 среди всех объектов, которым алгоритм присвоил метку 1
- **Recall** (полнота) – процент правильно классифицированных объектов класса 1 среди всех объектов класса 1
- **F1-score** – среднее гармоническое Precision и Recall

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



## Вопрос на собеседовании

Почему в F1-score используется именно среднее гармоническое?



## Вопрос на собеседовании

Тестовая выборка содержит 10 объектов класса 1 и 990 объектов класса 0. Какая точность у константного алгоритма?

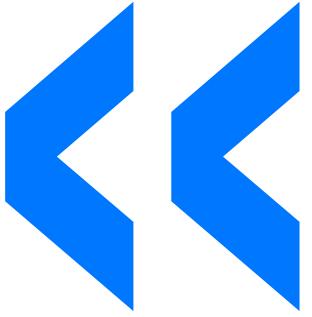
# Переобучение (overfit) и гиперпараметры

Параметры метрических алгоритмов настраивать на обучающей выборке? На тестовой? Почему?

*Из-за излишней сложности модели может появиться **эффект переобучения**: модель слишком хорошо выучивает обучающую выборку, но хуже работает на новых объектах*

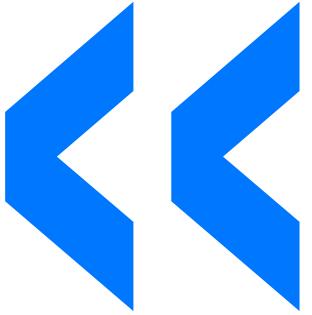
Параметры, которые нельзя настраивать на обучающей выборке, будем называть **гиперпараметрами**.

Всю выборку нужно разделить на **обучающую и валидационную**. На валидационной настраиваем гиперпараметры



# Вопрос на собеседовании

Какие есть подходы в борьбе с переобучением?



# Вопрос на собеседовании

Какие есть подходы в борьбе с переобучением?

Если очень базово:

- **упрощать модель** (меньше параметров, остановить обучение раньше, выше lr и т.д.)
- **сложнять задачу** (больше данных, L1, L2, dropout, batchnorm, weight decay, шум и т.д.)

Т.е. сложность модели сделать соразмерной сложности задачи.

# Кросс-валидация

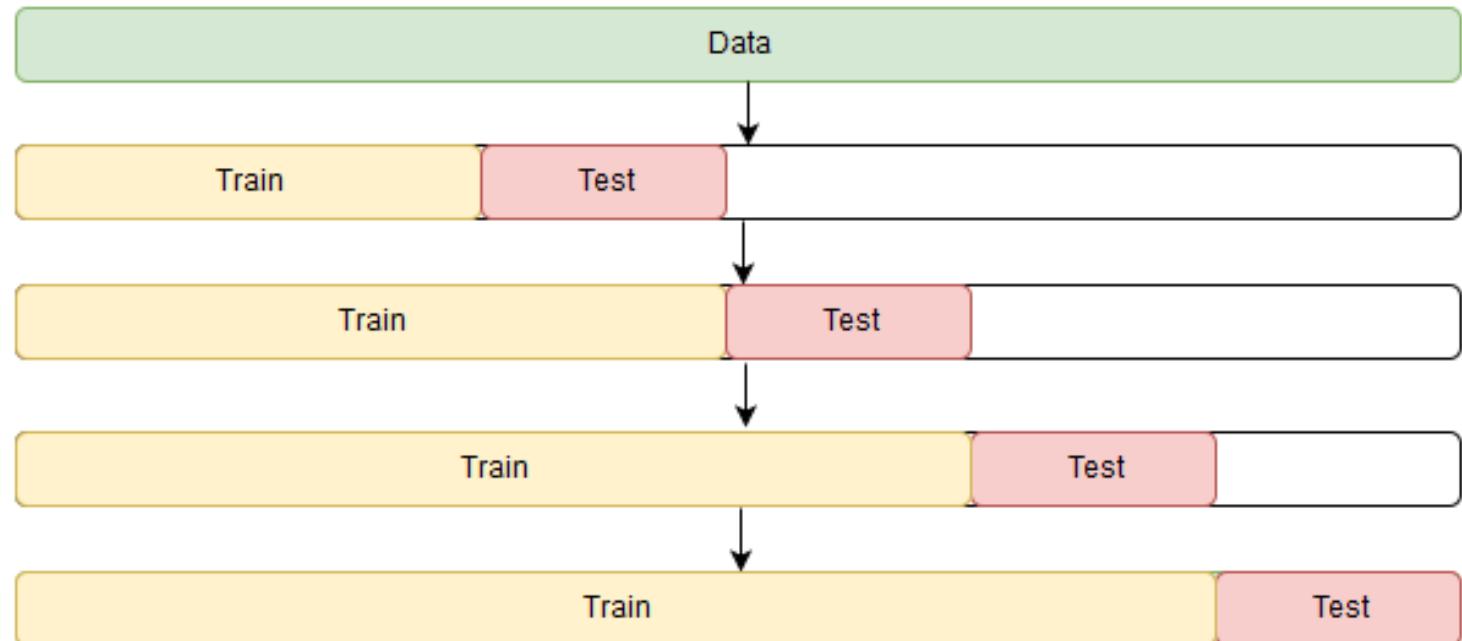
Если данных мало или хотим проверить качество на всех данных

*Что изменяется в схеме, если мы будем предсказывать продажи?*



# Кросс-валидация

Если предсказываем время ряд, не заглядываем в буд



# Развитие метрических алгоритмов



# Сложность алгоритма

Сложность обучения –  $O(ND)$  (запоминаем выборку)

Сложность предсказания –  $O(ND)$ (считаем все расстояния)

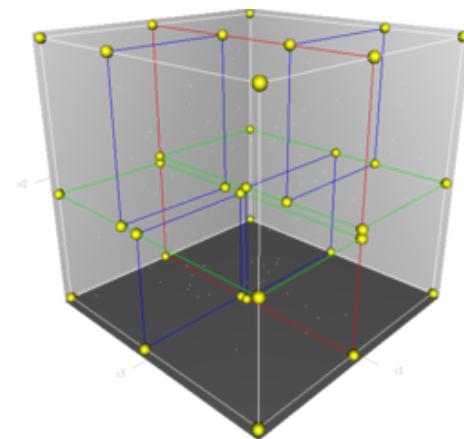
В таком виде это в real time системах это работать не будет!

Зачем мы тогда все это учим? Может, можно как-то ускорить?

# Ускоряем алгоритм

Структурируем признаковое пространство, чтобы по нему быстрее искать.

- KD-tree
- Ball tree



Если признаков мало (несколько десятков), то сложность по числу объектов логарифмическая. Если много — линейная (проклятие размерности), внедрять неэффективно

# Приближенный поиск ближайших соседей

В среднем имеют логарифмическую сложность даже для больших признаковых пространств. Лекция 6

Примеры методов:

- **ANNOY** — делим пространство случайными плоскостями, строим дерево
- **Navigable Small World** — гуляем по графу маленького мира
- **FAISS** — кластеризуем пространство и ищем расстояния до центров кластеров
- **LSH** (Locality-sensitive hashing) — делаем хэш функцию, которая близким объектам присваивает близкие значения хэша

# Применение в реальных системах

Все большие поисковые/рекомендательные системы состоят из двух компонент:

- **Грубый отбор** кандидатов быстрой моделью
- Использование медленной финальной модели на кандидатах

Быстрый приближенный поиск ближайших соседей идеально подходит под задачу **грубого отбора** кандидатов.

Лайфхак: расстояние  $\rho(x, y)$  можно подавать в финальную модель как фичу

# KNeighborsClassifier B

## sklearn



# Параметры

```
KNeighborsClassifier(  
    n_neighbors=5,  
    *,  
    weights='uniform',  
    algorithm='auto',  
    leaf_size=30,  
    p=2,  
    metric='minkowski',  
    metric_params=None,  
    n_jobs=None,  
)
```

```
KNeighborsRegressor(  
    n_neighbors=5,  
    *,  
    weights='uniform',  
    algorithm='auto',  
    leaf_size=30,  
    p=2,  
    metric='minkowski',  
    metric_params=None,  
    n_jobs=None,  
)
```

# Использование (sklearn)

```
from sklearn.neighbors import KNeighborsClassifier  
  
knn = KNeighborsClassifier(n_neighbors=5, weights='uniform')  
knn.fit(X, y)  
knn.predict(X_val)
```

```
array([0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,  
     0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0,  
     1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
     0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0,  
     1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1])
```

# KNN в ноутбуке



Спасибо!  
Задавайте  
ваши вопросы,  
ну  
пожалуйста :)

Алексей Ярошенко  
[t.me/yaroshenko](https://t.me/yaroshenko)