

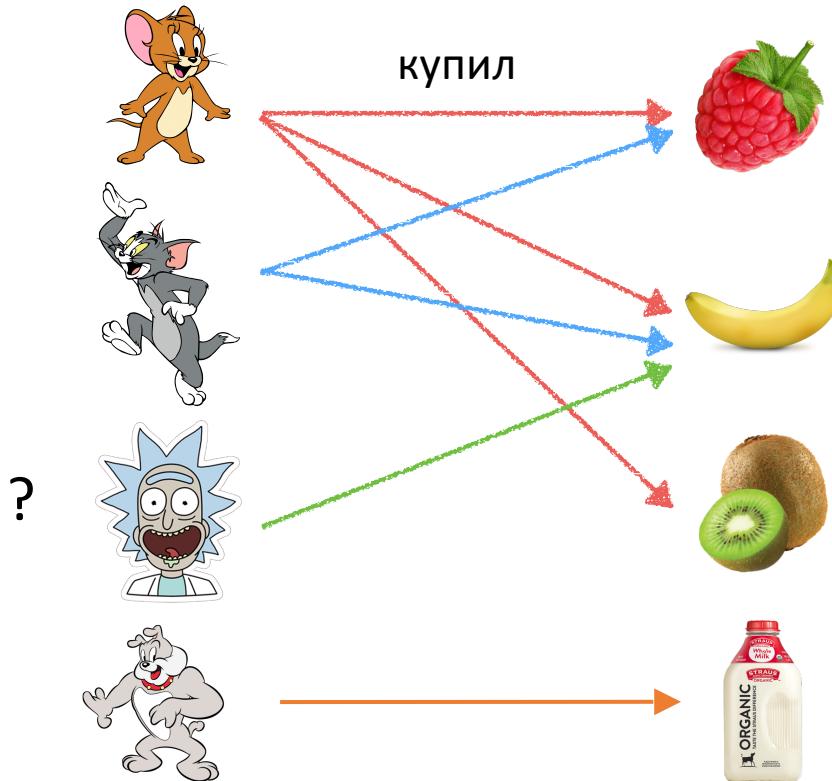
Основные алгоритмы машинного обучения

Елена Кантоностова

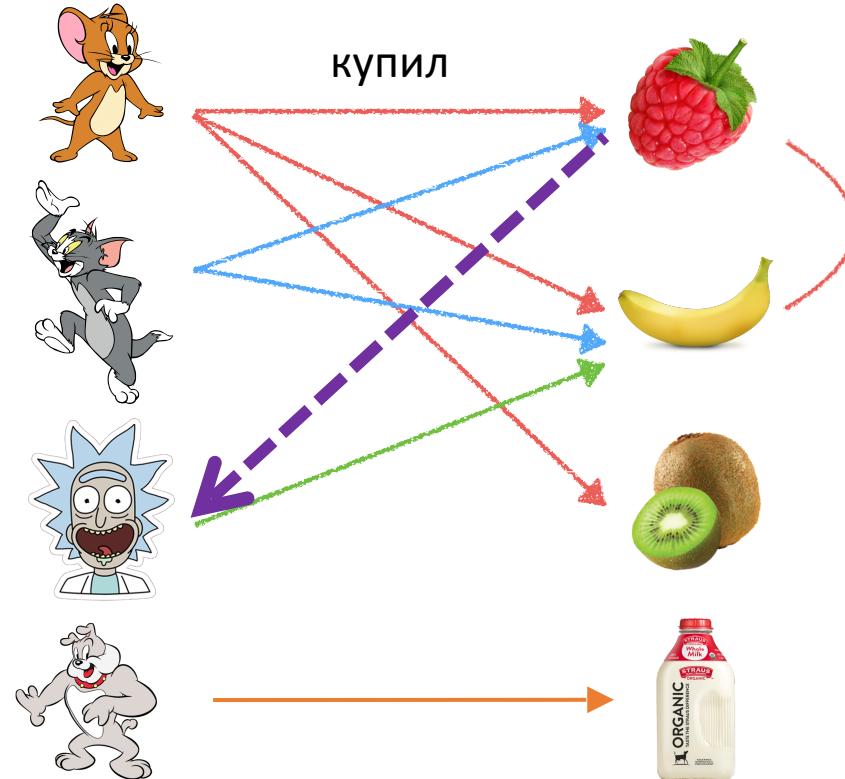
Как устроен интенсив

- Вводное занятие (сегодня)
- Три занятия по рекомендательным системам от Артема Селезнева:
 - 19, 21 и 24 апреля с 18:00 до 21:00
- Соревнование по построению рекомендательной системы на Kaggle
 - 22 - 29 апреля
- Доработка задачи до промышленного вида для участников из топ-40 соревнования

Рекомендательные системы



Рекомендации на основе похожести товаров



Предсказание оценки пользователя товару

Рассмотрим матрицу оценок "пользователь-товар"

Пользователи

Понравится?

The figure shows a matrix of user-item ratings. The columns represent items: SHERLOCK, HOUSE OF CARDS, THE AVENGERS, ARRESTED DEVELOPMENT, Breaking Bad, and THE WALKING DEAD. The rows represent users, indicated by icons: a man, a woman, a person in a suit, a person in a headset, a person in a blue shirt, and a person in a green shirt. A red magnifying glass highlights the cell at row 6, column 5, which contains the value '2'. To the right of the matrix, there is a legend for 'Оценка' (Rating) showing five stars: two yellow stars and three blue stars.

	SHERLOCK	HOUSE OF CARDS	THE AVENGERS	ARRESTED DEVELOPMENT	Breaking Bad	THE WALKING DEAD
Пользователи	2		2	4	5	
	5		4			1
			5		2	
			1		5	
			4			4
	4	5		1		

Товары

Оценка

★★ ★★ ★☆ ★☆ ★☆

Рекомендательные системы

- Какого типа задача?

Рекомендательные системы

- Классификация
- Регрессия
- Кластеризация
- Ранжирование

Какие алгоритмы пригодятся?

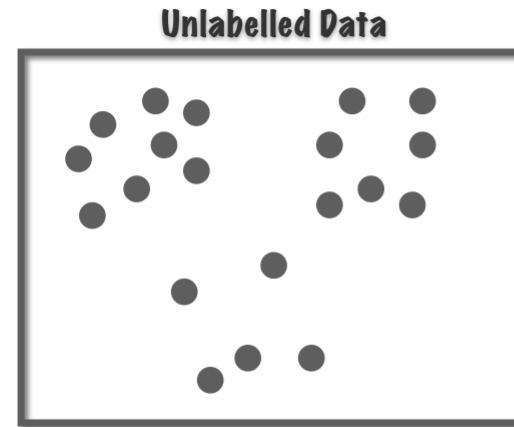
- Алгоритмы классификации и регрессии - особенно бустинг!
- Алгоритмы кластеризации
- Матричные разложения

Кластеризация

Кластеризация

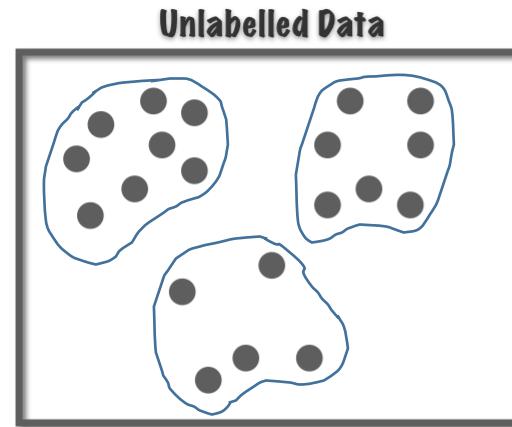
- Делим товары на кластеры по похожести
- Если пользователь купил что-то из кластера, рекомендуем другие товары из этого кластера

K-Means



K-Means

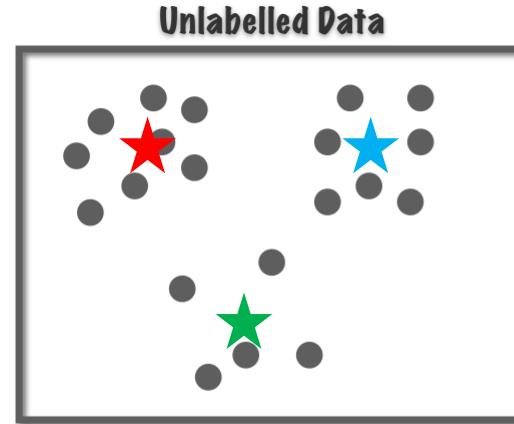
$k=3$



Вы видите 3 сгустка

K-Means

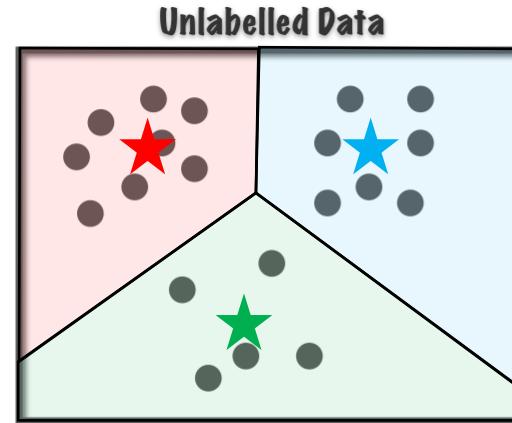
$k=3$



Опишем их центрами

K-Means

$k=3$

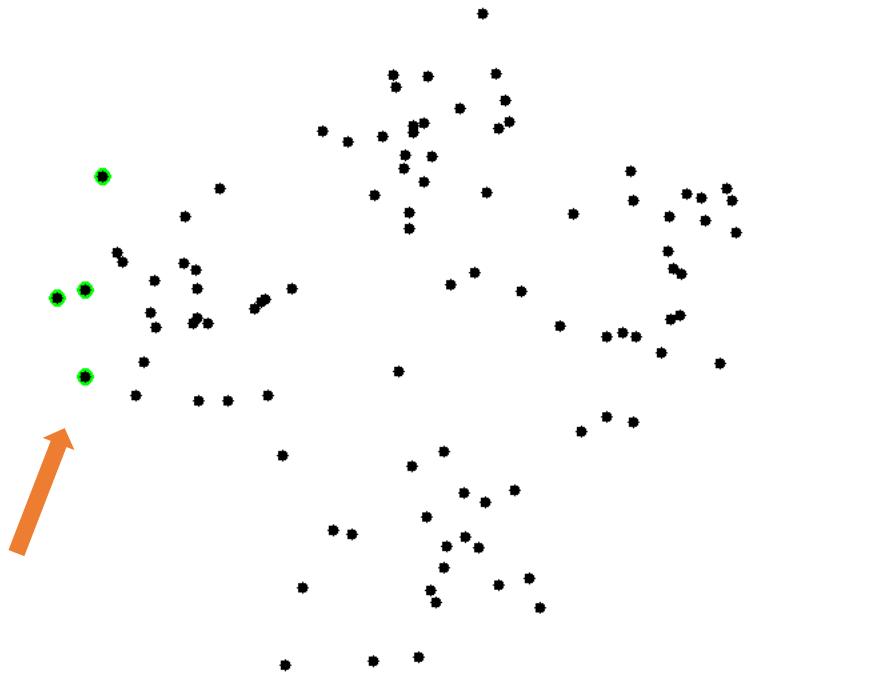


Каждая точка относится к ближайшему центру

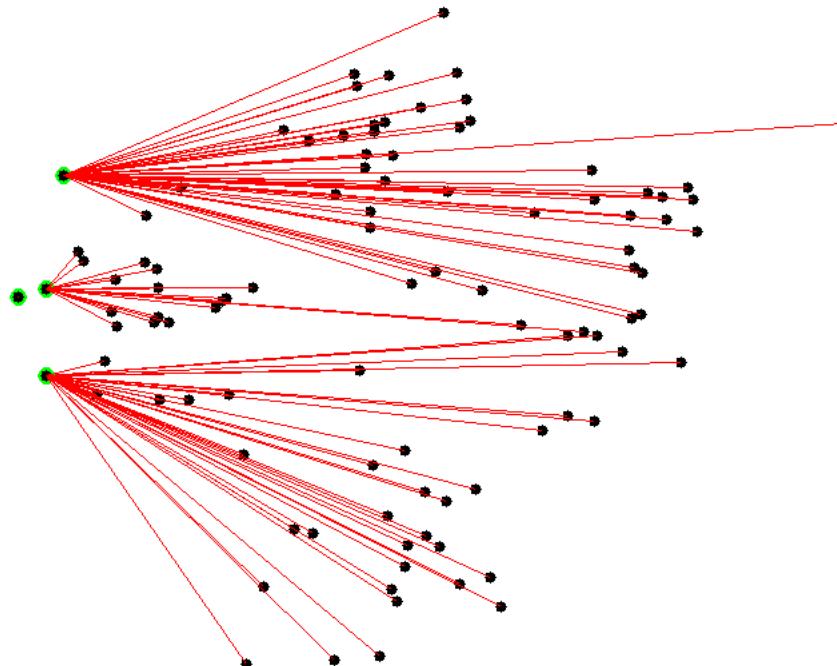
<https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c>

Кластеризация при помощи K-means

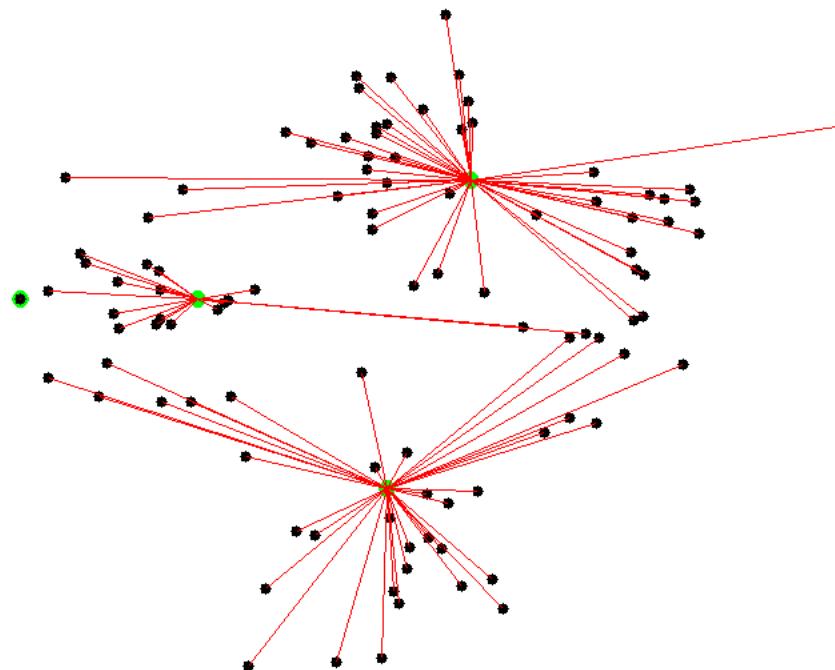
Возьмем
 $K=4$
случайных
центра



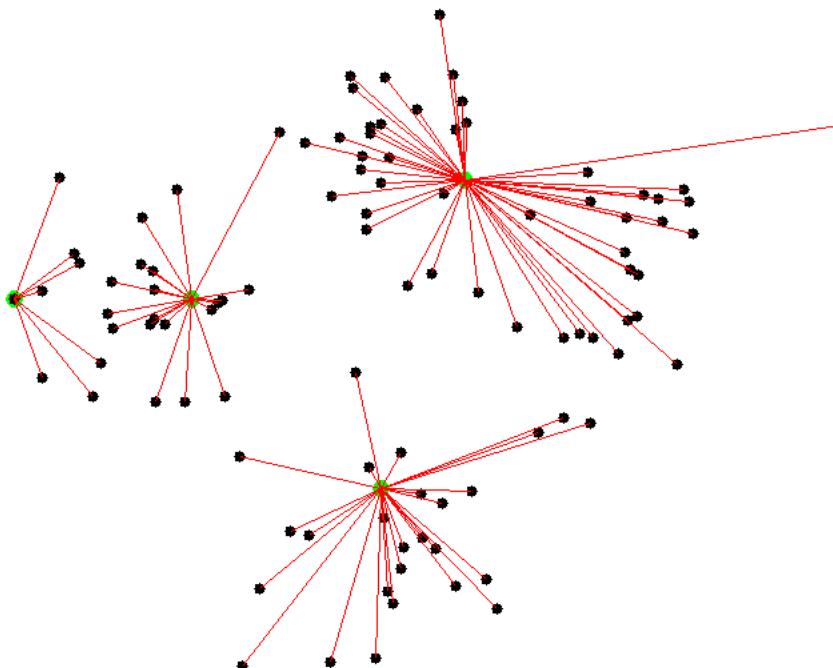
Для каждой точки находим ближайший центр



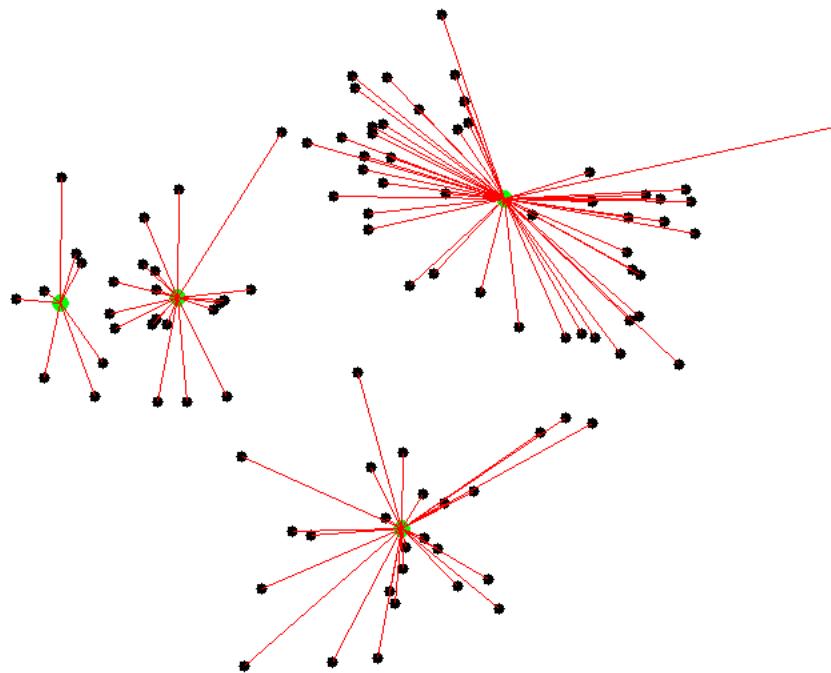
Пересчитываем центры



Опять ищем для каждой точки ближайший центр



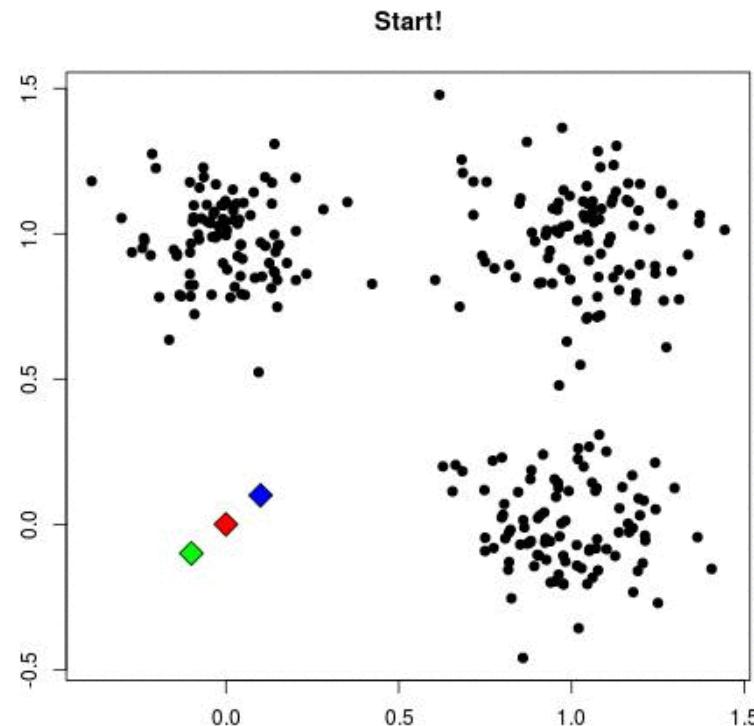
Опять пересчитываем центры



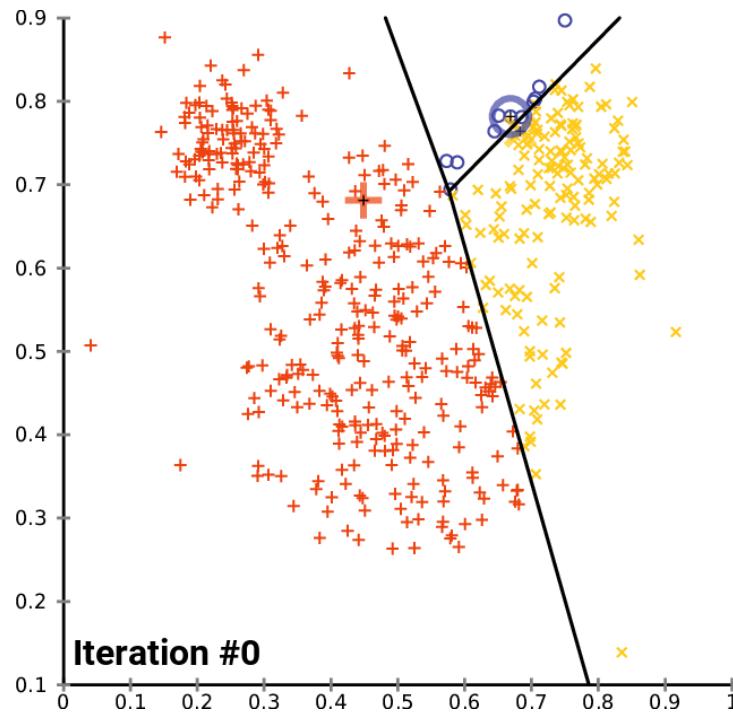
Анимация процесса K-means (K=4)



Анимация процесса K-means (K=3)

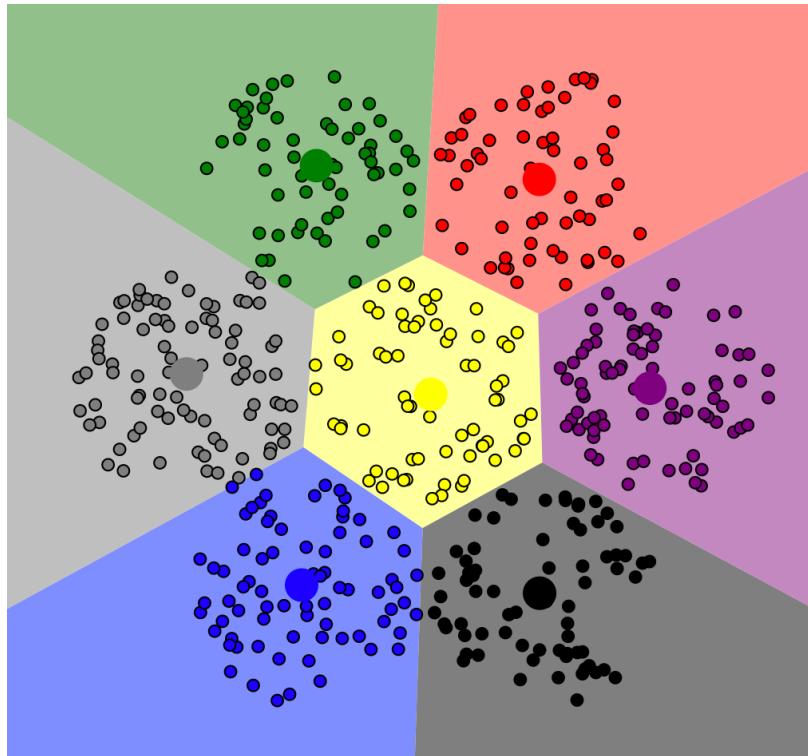


Неудачный пример



Демо K-Means

<https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>



Плюсы и минусы K-means

- Плюсы:
 - Очень простой алгоритм
 - Работает даже на больших данных
- Минусы:
 - Надо задавать число К руками
 - Не всегда находит кластеры правильно

Кластеризация музыки по прослушиваниям

Тяжелый рок

Attribute	cluster_0 ↓
niИ	0.031
metallica	0.015
iron maiden	0.011
tool	0.010
dir en grey	0.009
nightwish[0.009
opeth	0.009
judas priest	0.009
in flames	0.009
dream theater	0.008
megadeth	0.008
black sabbath	0.008
rammstein	0.008
the misfits	0.007
johnny clash	0.007
ルートヴィヒ · ...	0.007
marilyn manson	0.007

Рок

Attribute	cluster_1 ↓
the beatles	0.073
dylan. bob	0.017
pink fluid	0.015
the rolling stones	0.011
led zeppelin.	0.011
divid bowie	0.011
radiohead	0.009
queen	0.007
who	0.007
the grateful dead	0.007
young, neil	0.006
u2	0.006
the beach boys	0.006
the kinks	0.006
red hot chili pe...	0.006
phish	0.006
iohnny clash	0.006

Рэп

Attribute	cluster_4 ↓
lil' wayne	0.035
kanye west	0.031
jay-z	0.020
nas	0.019
common	0.016
atmosphere	0.014
t.i.	0.013
lupe the gorilla	0.013
outkast	0.012
the notorious b...	0.011
a tribe called q...	0.011
the roots featur...	0.011
eminem	0.010
j dilla	0.010
50 cent	0.009
tupak shakur	0.009
ghostface killah	0.009

Поп

Attribute	cluster_5 ↓
coldplay	0.019
britney spears	0.018
madonna	0.012
johnson jack	0.012
linkin park	0.011
jason mraz	0.010
john mayer	0.010
보아	0.009
muse	0.009
enya	0.009
u2	0.009
evanescence	0.009
reliant k	0.007
tori amos	0.007
depeche Mode	0.007
micheal bublé	0.006
kelly clarkson	0.006

Классификация и регрессия

Классификация и регрессия: решающее дерево

- Решающие деревья - это деревья (как математический объект), то есть ориентированные графы с *корнем* и несколькими концевыми вершинами (*листьями*).

Как строить дерево

Предикаты, то есть условия, которые мы проверяем в вершинах дерева, могут быть разными. Но в классических решающих деревьях предикаты очень простые:

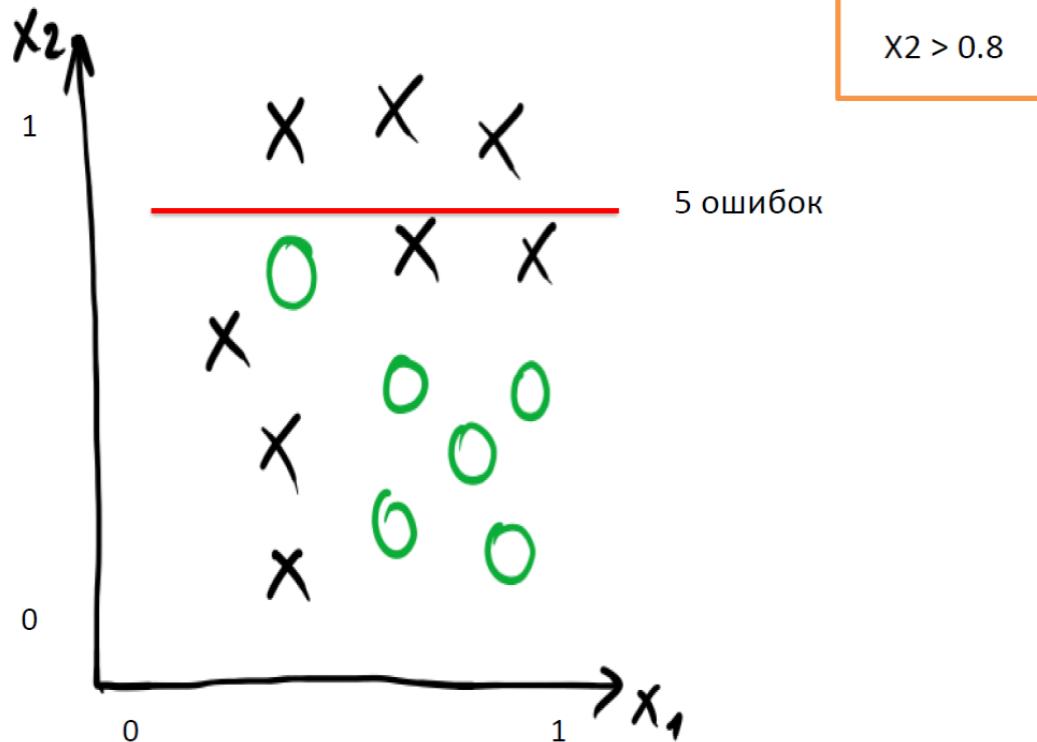


Предикат имеет вид "**признак > порога**", то есть в каждой вершине используется ровно один признак и он сравнивается с некоторым пороговым значением. Например, в задаче скоринга предикаты могут иметь вид:

- возраст клиента > 40 лет
- доход клиента > 100000 рублей

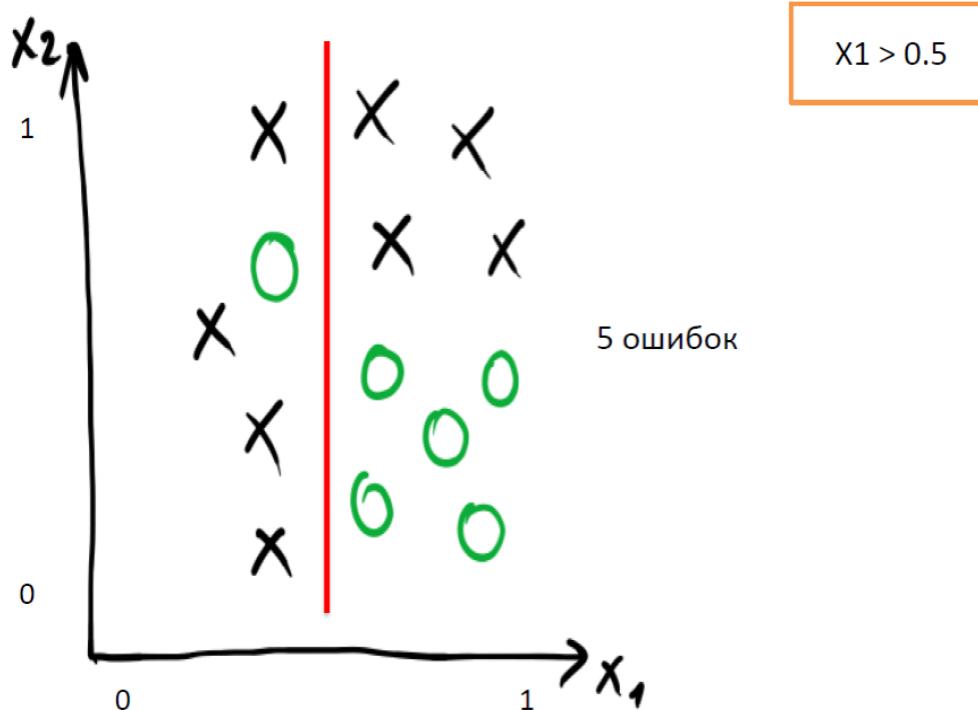
Пример

- Жадно найдем наилучший предикат



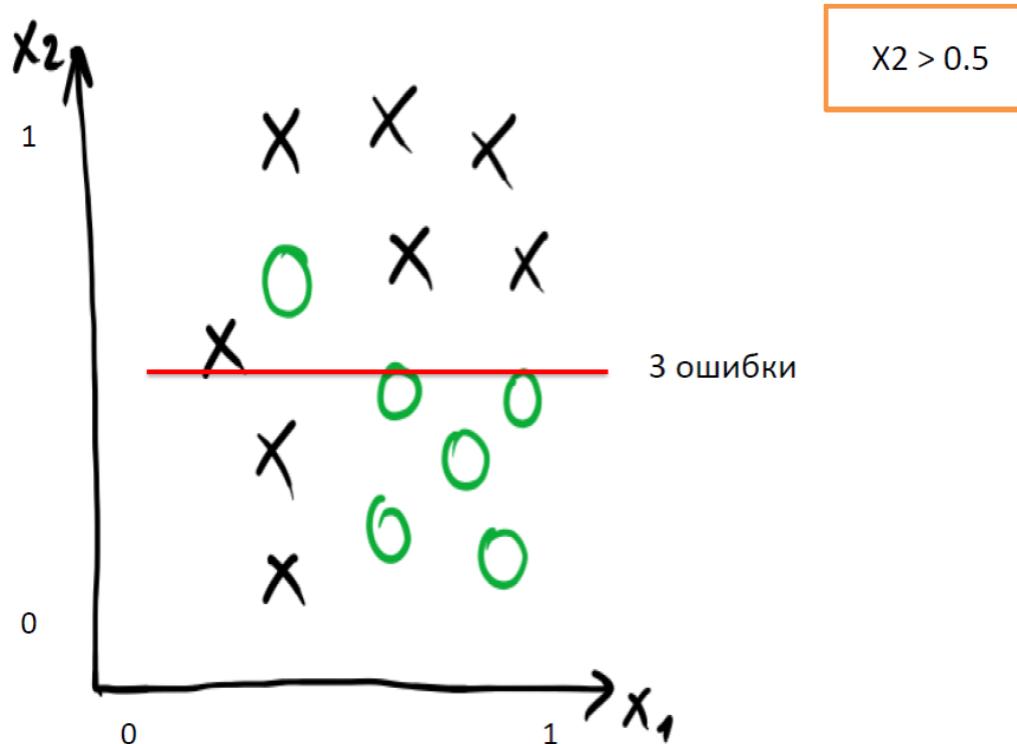
Пример

- Жадно найдем наилучший предикат



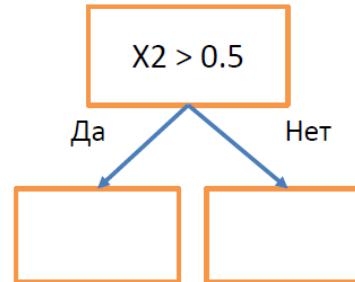
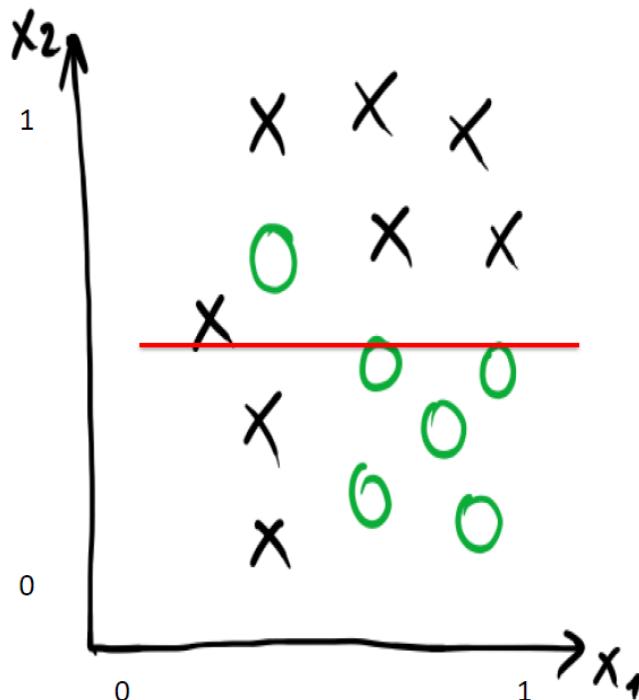
Пример

- Жадно найдем наилучший предикат



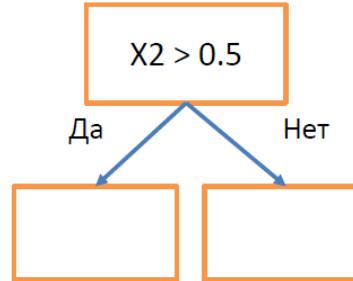
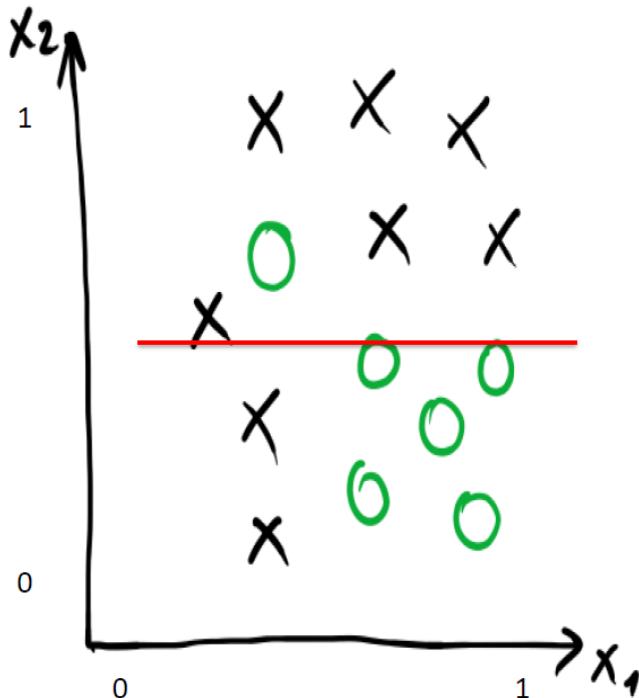
Пример

- Нашли лучшее первое ветвление



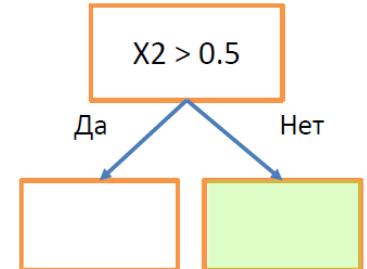
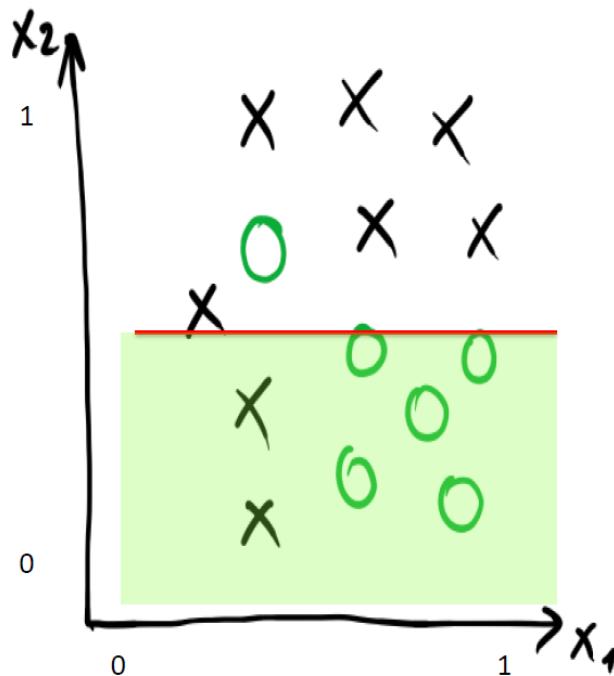
Пример

- Нашли лучшее первое ветвление



Пример

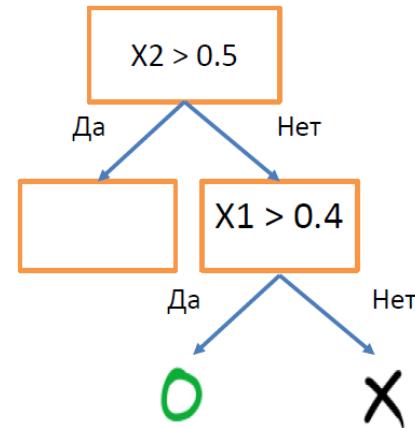
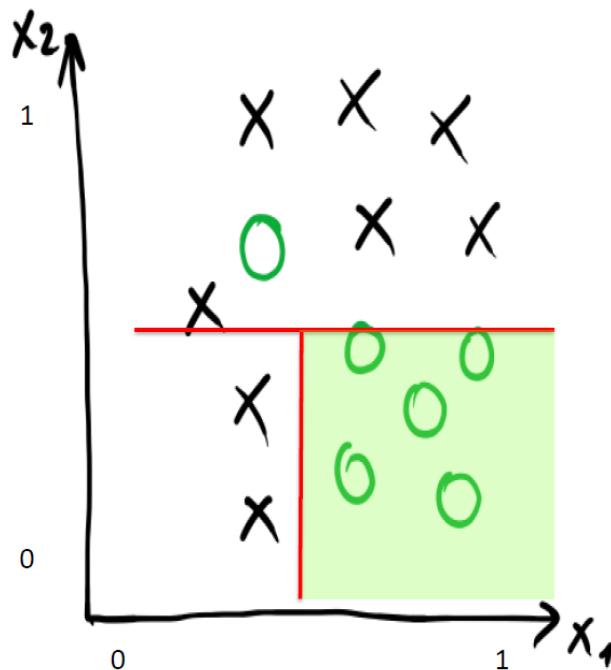
- Нашли лучшее первое ветвление



Продолжим
этую ветку

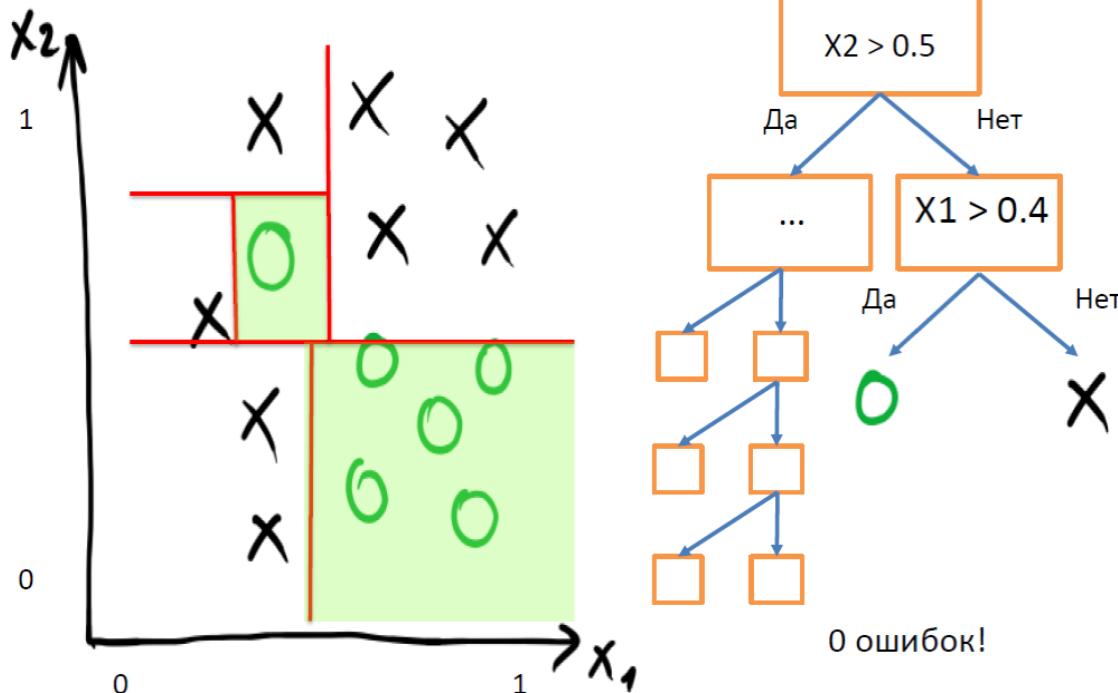
Пример

- Нашли лучшее второе ветвление



Пример

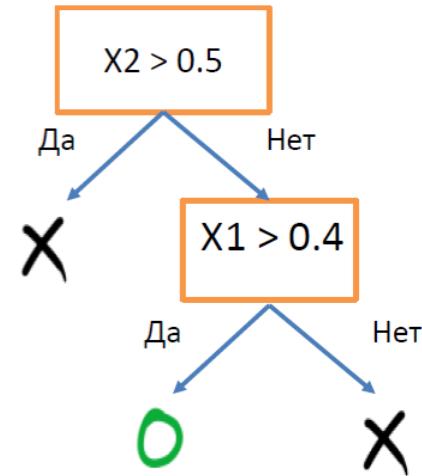
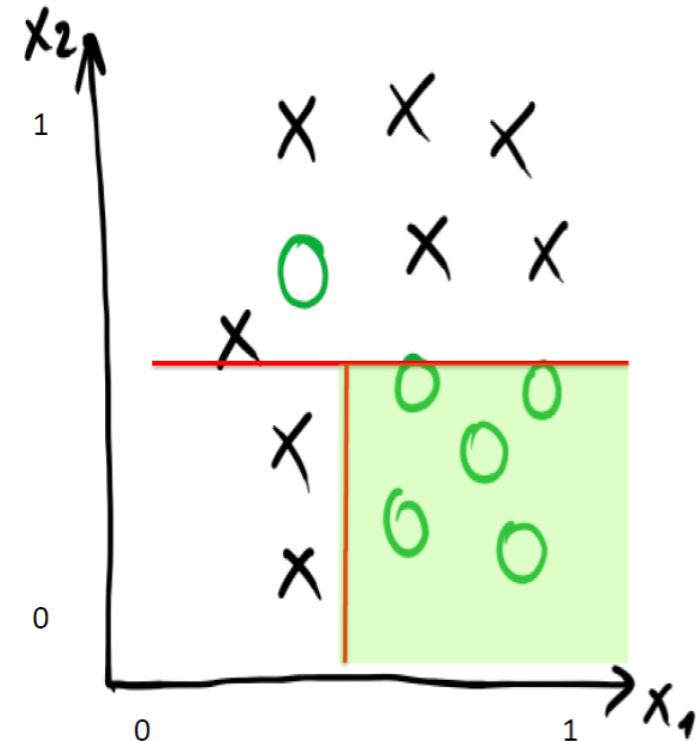
- Построили всё дерево



Переобучение

Для (почти) любой выборки можно построить решающее дерево, не допускающее на ней ни одной ошибки. Такое дерево скорее всего будет переобученным.

Пример



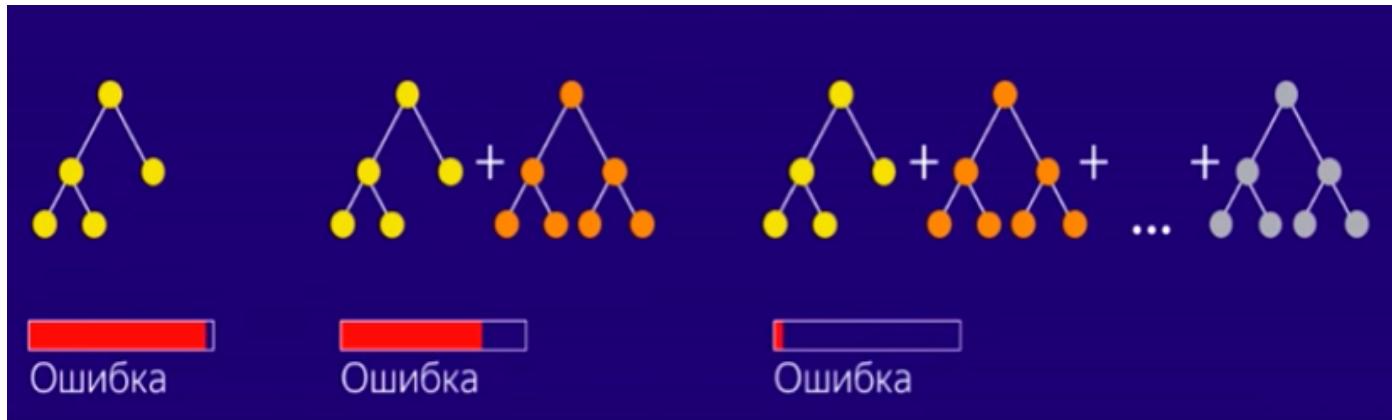
1 ошибка, но
скорее всего будет
лучше на тесте!

Бустинг

Идея: строим набор алгоритмов, каждый из которых исправляет ошибку предыдущих.

Бустинг

Идея: строим набор алгоритмов, каждый из которых исправляет ошибку предыдущих.



Бустинг в задаче регрессии

Решаем задачу регрессии с минимизацией квадратичной ошибки:

$$\frac{1}{2} \sum_{i=1}^l \left(a(x_i) - y_i \right)^2 \rightarrow \min_a$$

Ищем алгоритм $a(x)$ в виде суммы N базовых алгоритмов:

$$a(x) = \sum_{n=1}^N b_n(x),$$

где базовые алгоритмы $b_n(x)$ принадлежат некоторому семейству A .

Бустинг в задаче регрессии

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l (b(x_i) - \textcolor{blue}{y}_i)^2$$

- Ошибка на объекте x :

$$s = y - b_1(x)$$

Бустинг в задаче регрессии

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l (b(x_i) - y_i)^2$$

- Ошибка на объекте x :

$$\textcolor{blue}{s} = y - b_1(x)$$

Следующий алгоритм должен настраиваться на эту ошибку, т.е. **целевая переменная для следующего алгоритма – это вектор ошибок s** (а не исходный вектор y)

Бустинг в задаче регрессии

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l (b(x_i) - y_i)^2$$

Шаг 2: Ищем алгоритм $b_2(x)$, настраивающийся на ошибки s первого алгоритма:

$$b_2(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l (b(x_i) - s_i)^2$$

Бустинг в задаче регрессии

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l \left(b(x_i) - y_i \right)^2$$

Шаг 2: Ищем алгоритм $b_2(x)$, настраивающийся на ошибки s первого алгоритма:

$$b_2(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l \left(b(x_i) - s_i \right)^2$$

Следующий алгоритм $b_3(x)$ будем выбирать так, чтобы он минимизировал ошибку предыдущей композиции (т.е. $b_1(x) + b_2(x)$) и т.д.

Бустинг в задаче регрессии

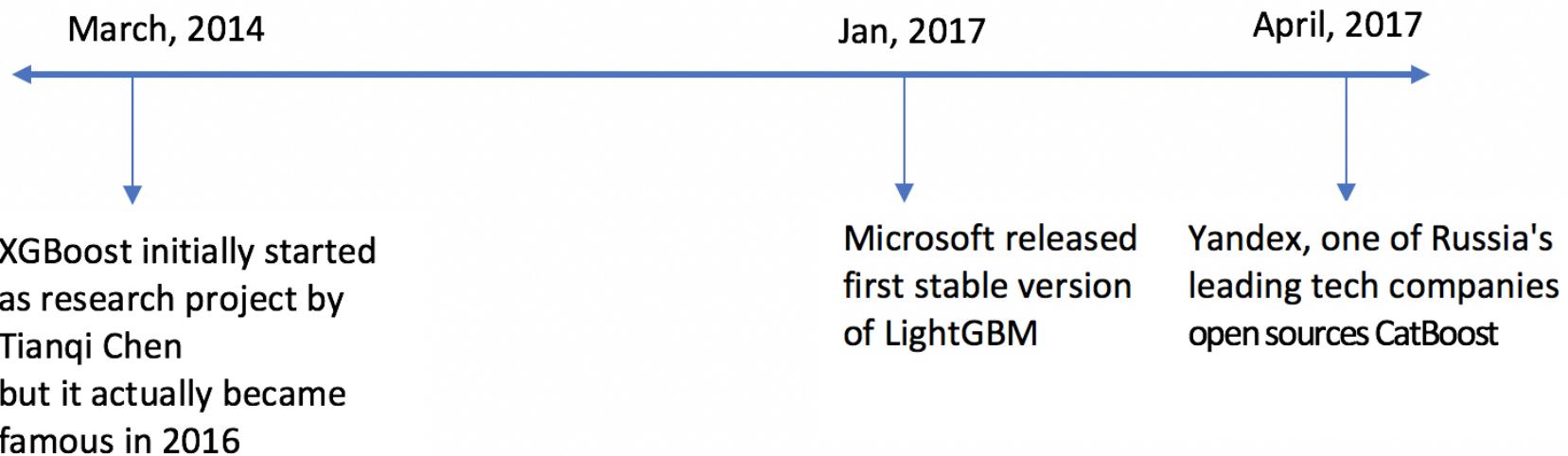
Каждый следующий алгоритм настраиваем на ошибку предыдущих.

Шаг N: Ошибка: $s_i^{(N)} = y_i - \sum_{n=1}^{N-1} b_n(x_i) = y_i - a_{N-1}(x_i)$

Ищем алгоритм $b_N(x)$:

$$b_N(x) = \operatorname{argmin}_{b \in A} \frac{1}{2} \sum_{i=1}^l \left(b(x_i) - s_i^{(N)} \right)^2$$

XGBoost, LightGBM, CatBoost



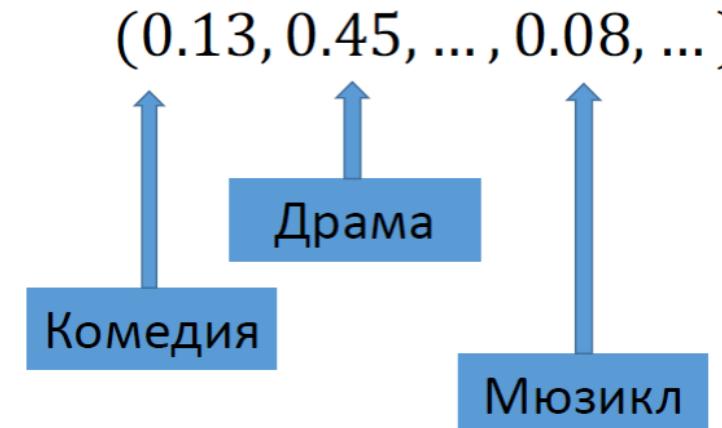
Матричные разложения

Матричные факторизации

Решаем задачу рекомендации пользователям различных фильмов.

Можно описать пользователя и фильм векторами интересов:

- для пользователя – насколько он интересуется каждым жанром
- для фильма – насколько он относится к каждому жанру



Рейтинг

Будем определять *заинтересованность* как *скалярное произведение* вектора пользователя и вектора фильма:

$$(0.1, 0.5, 0.01, 0.92) \times (0, 0, 0.1, 0.95) = 0.875$$

$$(0.1, 0.5, 0.01, 0.92) \times (0.9, 0, 0, 0.1) = 0.182$$

Пользователь

Фильм

Предсказание оценки пользователя товару

Рассмотрим матрицу оценок "пользователь-товар"

Пользователи

Понравится?

	SHERLOCK	HOUSE OF CARDS	THE AVENGERS	ARMED DEVELOPMENT	Breaking Bad	WALKING DEAD	Товары
Пользователи	2		2	4	5		
	5		4			1	Оценка
			5		2		
				1	5		
					4		
						2	
	4	5		1			

Модели со скрытыми переменными

У нас есть матрица рейтингов для задачи пользователь-фильм:

Будем определять *заинтересованность* как *скалярное произведение* вектора пользователя и вектора фильма:

2	5	
5		4
	1	
	2	5

Цель: найти такие векторы пользователей и векторы фильмов, скалярное произведение которых максимально близко к рейтингам из таблицы.

Модели со скрытыми переменными

У нас есть матрица рейтингов для задачи пользователь-фильм:

Будем определять **заинтересованность** как **скалярное произведение** вектора пользователя и вектора фильма:

The diagram illustrates the calculation of user interest. A red arrow points from the vector $(2.1, 5)$ to the first row of the matrix. Another red arrow points from the vector $(0.9, 0.05)$ to the first column of the matrix. The matrix itself is a 4x3 grid of numbers.

Matrix data:

	(0.9, 0.05)	(0.02, 1.1)	(1.05, 0.01)
(2.1, 5)	2	5	
(4.6, 0)	5		4
(0, 1)		1	
(4.9, 0.9)		1	5

Цель: найти такие векторы пользователей и векторы фильмов, скалярное произведение которых максимально близко к рейтингам из таблицы.

Матричные факторизации

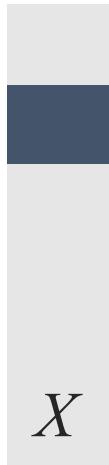
R

A matrix R representing user ratings for TV shows. The columns represent TV shows: SHERLOCK, HOUSE OF CARDS, AVENGERS, ARRESTED DEVELOPMENT, Breaking Bad, and THE WALKING DEAD. The rows represent users, shown as icons of people. The matrix has dimensions $m \times n$. A red box highlights the rating for the second user (female icon) watching AVENGERS, which is a value of 1.

	SHERLOCK	HOUSE OF CARDS	AVENGERS	ARRESTED DEVELOPMENT	Breaking Bad	THE WALKING DEAD
0			0	1	1	
1			1			0
			1		0	
	0		1			1
			1			0
1	1		0			

$m \times n$

$m \times k$



← профиль пользователя

\times Y $k \times n$

X
↑
профиль карточки

Сингулярное разложение

Сингулярное разложение - это разложение матрицы в произведение трех матриц:

$$M = U \cdot D \cdot V^T,$$

columns are orthonormal

diagonal matrix

rows are orthonormal

M

$n \times m$

U

$n \times k$

D

$K \times K,$
 $K = \text{rank } M$

V^T

$k \times m$

где

- матрицы U и V - ортогональны
- а матрица D - диагональная.

Спасибо за внимание!