

Машинное обучение: базовые концепции машинного обучения

МТС Тета
Эмели Драль

About me



- **Co-founder & CTO** Evidently AI
- Ex **Chief Data Scientist** at Yandex Data Factory and Mechanica AI
- Co-founder of **Data Mining in Action**, largest offline data science course in Russia
- Co-author of two **Coursera** specializations in data science with > 100K students
- Lecturer at **Harbour.Space University**, GSOM MBA, MADE by Mail.ru

50+

Industrial applications of
machine learning

Программа курса

Курс состоит из **3x** блоков:

1. **Basics:** базовые концепции машинного обучения
2. **Cases:** проект по анализу данных
3. **Services:** сервис на основе данных

Basics

1. ML basics & tools
2. Валидация моделей по историческим данным
3. Тестирование моделей в production

Результат изучения: знаете стандартные виды обучения, понимаете логику работы базовых алгоритмов, можете валидировать модели

Cases

1. Жизненный цикл проекта по анализу данных
2. Предпроектное исследование
3. Работа над проектом

Результат изучения: понимаете структуру **проекта по анализу данных**, можете **работать в индустрии** под руководством старшего специалиста

Services

1. Data-based service
2. Мониторинг ML моделей
3. Check-list для проекта по анализу данных

Результат изучения: понимаете жизненный цикл сервиса на основе данных, можете разработать demo stand

Система оценки

В курсе 4 домашних задания:

1. **Basics** – 1 персональное задание (20 баллов)
2. **Cases** – 1 командное задание (35 баллов)
3. **Services** – 1 командное задание (45 баллов)

Итоговая оценка определяется суммой баллов, полученных за задания:

- от 60 до 70 - зачленено
- от 71 до 85 - хорошо
- от 86 и выше - отлично

Базовые концепции ML

1. Области применения
2. Базовые концепты
3. Виды обучения
4. Обзор алгоритмов
5. Жизненный цикл модели

Области применения машинного обучения

Области применения



В каких сферах есть место для
применения машинного обучения?

Области применения

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

Области применения

Home > Overview of CatBoost

Overview of CatBoost

CatBoost is a machine learning algorithm that uses gradient boosting on decision trees. It is available as an open source library.

Key features:



Training

[Training](#)

[Training on GPU](#)

[Python train function](#)

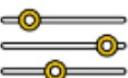
[Cross-validation](#)

[Overfitting detector](#)

[Pre-trained data](#)

[Categorical features](#)

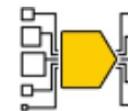
[Text features](#)



Model analysis

[Feature importances](#)

[Object importances](#)



Applying models

[Regular prediction](#)

[C and C++](#)

[Java](#)

[Rust](#)

[Calculate metrics](#)

[Staged prediction](#)

[Applying the model in ClickHouse](#)



Metrics

[Implemented metrics](#)

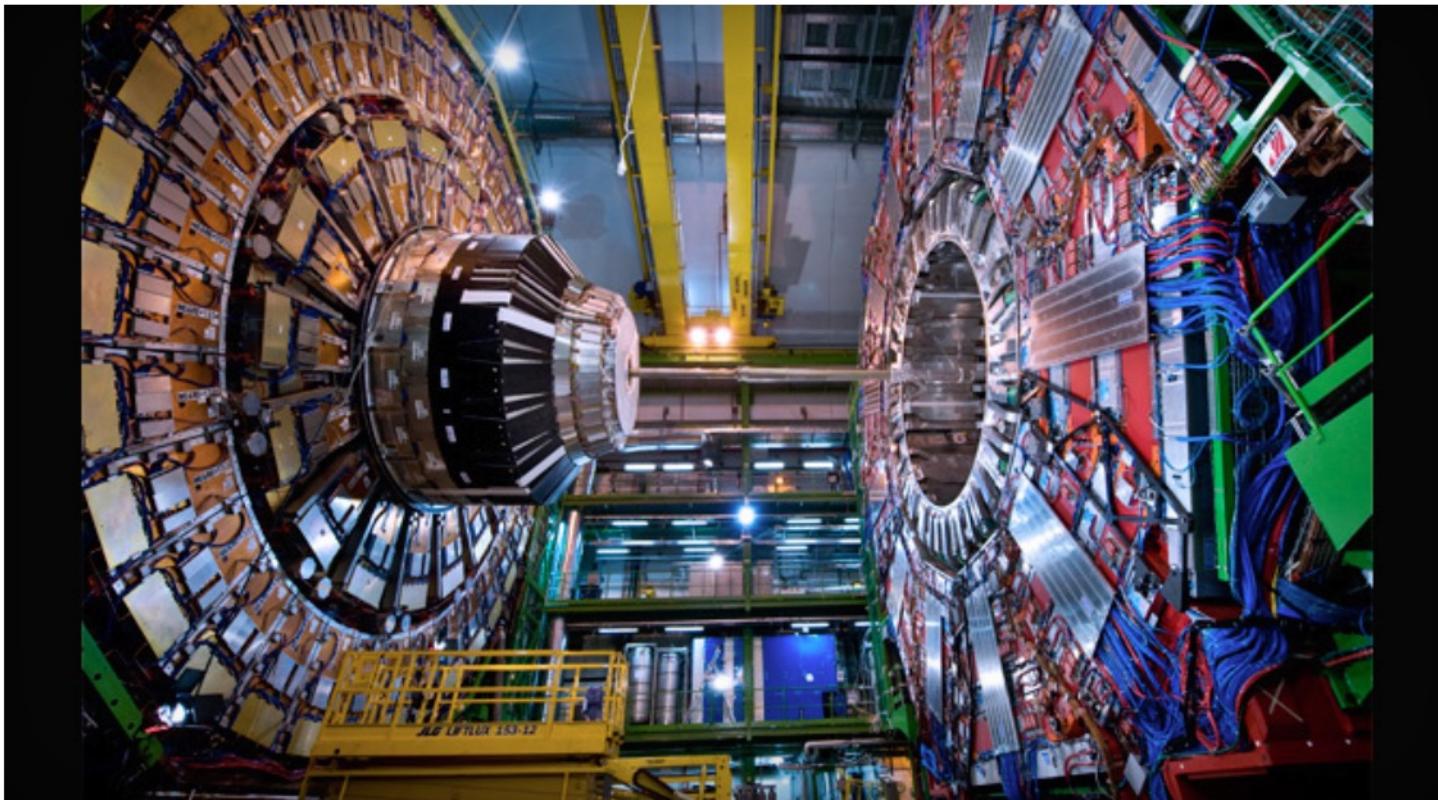
[User-defined metrics](#)

CERN boosts its search for antimatter with Yandex's MatrixNet search engine tech

By Tim Verry on February 1, 2013 at 8:36 am | [5 Comments](#)



Области применения



Области применения



FEATURES DOC

Open-source Version Control System for Machine Learning Projects

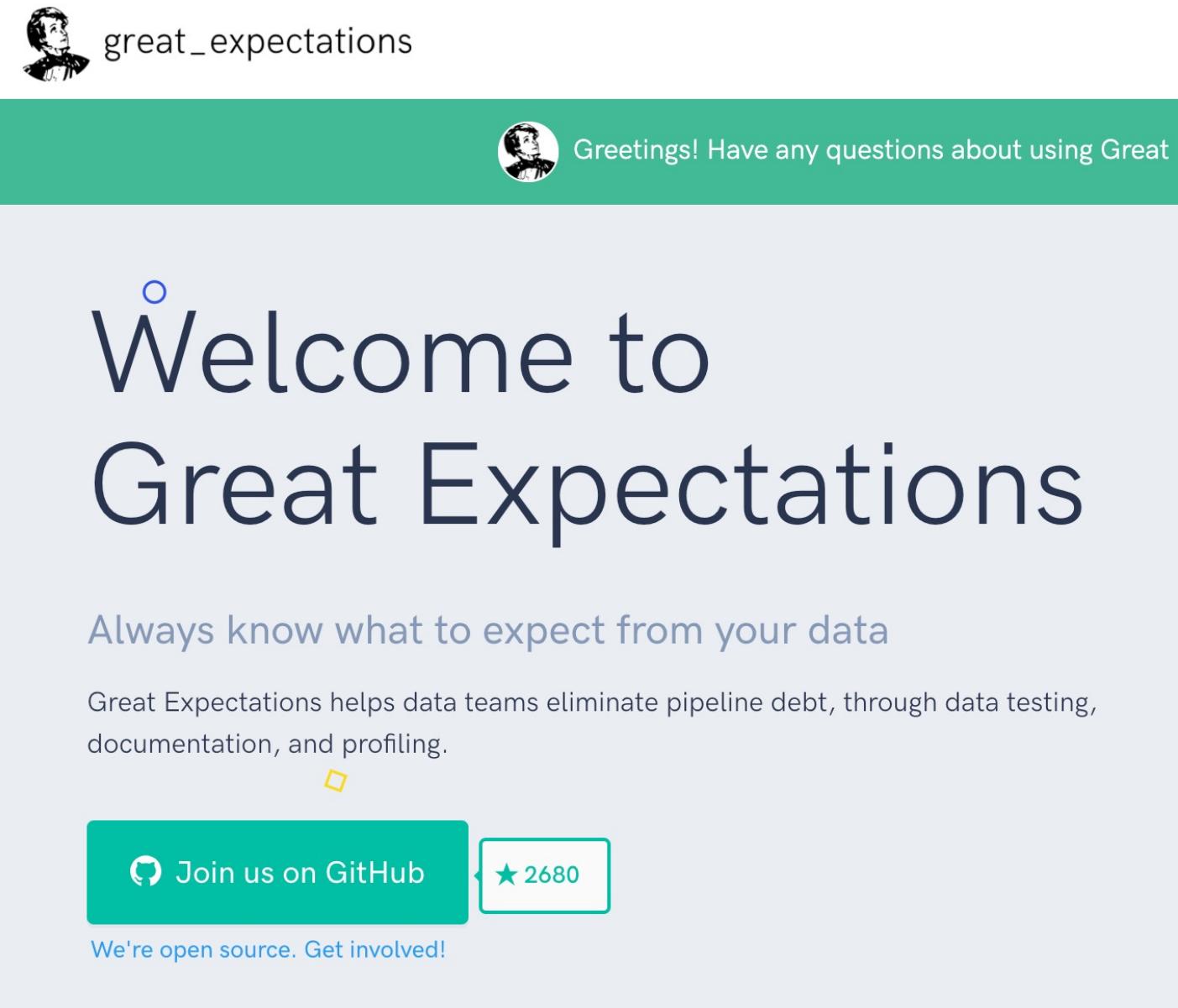


Download
(Mac OS)



Watch video
How it works

Области применения



The image shows the homepage of the Great Expectations project. At the top left is a small portrait of a man with a mustache. Next to it is the GitHub repository name "great_expectations". A green header bar contains the same portrait and the text "Greetings! Have any questions about using Great". The main title "Welcome to Great Expectations" is prominently displayed in large, dark blue serif font. Below it is a subtitle "Always know what to expect from your data" in a smaller, lighter blue font. A paragraph explains the project's purpose: "Great Expectations helps data teams eliminate pipeline debt, through data testing, documentation, and profiling." To the right of this text is a yellow diamond icon. Below the paragraph are two buttons: a teal one with the GitHub logo and the text "Join us on GitHub", and a white one with a star icon and the number "2680". At the bottom, a blue link reads "We're open source. Get involved!".

great_expectations

Greetings! Have any questions about using Great

Welcome to Great Expectations

Always know what to expect from your data

Great Expectations helps data teams eliminate pipeline debt, through data testing, documentation, and profiling.

Join us on GitHub

2680

We're open source. Get involved!

Области применения

Google



Search Google or type a URL



Области применения

NETFLIX

Home Characters TV Shows Movies Latest My List

Everyone's Watching



Animated



Области применения

Competitions

Grow your data science skills by competing in our exciting competitions. Find help in the [Documentation](#) or learn about [InClass competitions](#).



New to Kaggle? Start here!

Our Titanic Competition is a great first challenge to get started.



Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics
Getting Started • Ongoing • 18123 Teams

All Competitions

[Active](#) [Completed](#) [InClass](#)



OSIC Pulmonary Fibrosis Progression

Predict lung function decline
Featured • 11d to go • Code Competition • 1939 Teams



Области применения

Boosters.pro - сила данных

Крупнейшая в России и Восточной Европе платформа для проведения контестов по анализу данных.

15
проведённых
контестов

1700
активных
пользователей

7200
зарегистрированных
пользователей

Зарегистрироваться

Области применения



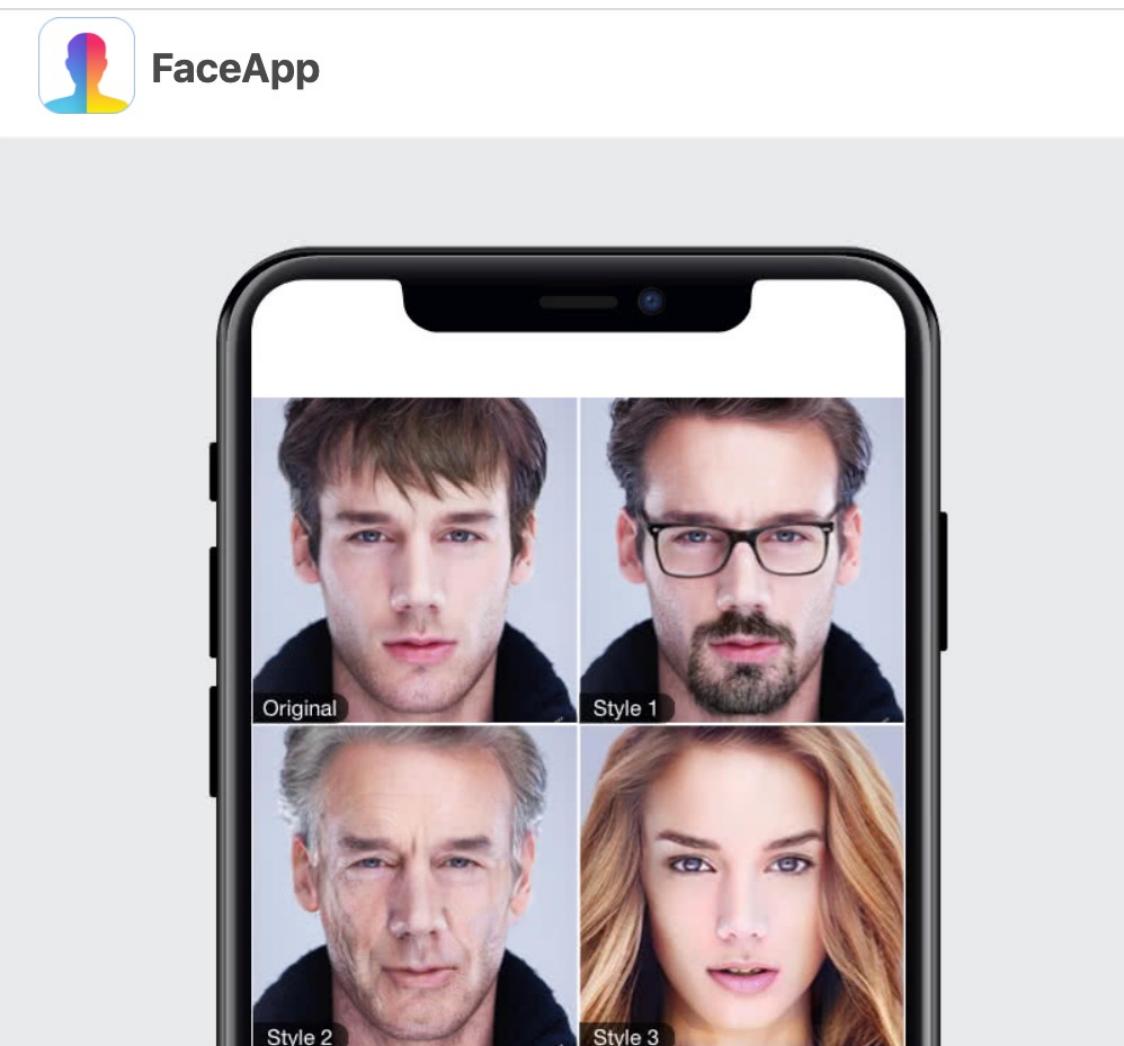
APP OF
THE YEAR 2016
APP STORE



BEST APP OF 2016
GOOGLE PLAY



Области применения



App Store
BEST OF 2017

Google Play
BEST OF 2017 AWARD

Области применения

To take away:

1. Работа специалистов по анализу данных очень сильно различаются в разных областях: задачи, навыки, инструменты разные.
2. Каждый специалист может выбрать подходящую сферу в зависимости от того, чем действительно интересно заниматься!
3. С другой стороны изучать ML можно очень по-разному: участие в соревнованиях, pet projects, исследовательские группы, стажировки.

Области применения

To take away:

1. Работа специалистов по анализу данных очень сильно различаются в разных областях: задачи, навыки, инструменты разные.
2. Каждый специалист может выбрать подходящую сферу в зависимости от того, чем действительно интересно заниматься!
3. С другой стороны изучать ML можно очень по-разному: участие в соревнованиях, pet projects, исследовательские группы, стажировки.

В нашем курсе мы фокусируемся на **индустриальном применении** машинного обучения.

Базовые концепты

Базовые концепты

Объекты и ответы

- x – объект
- y – ответ или значение

- X – множество объектов
- Y – множество ответов

Базовые
концепты

Объекты и ответы



250k \$



375k \$



179k \$

Базовые концепты

Признаки объектов (features)

- $f_1 \dots f_n$ – признаки, описывающие объект
- $x = (f_1, f_2, \dots, f_n)$
- x - вектор размера n , описывающий объект с помощью признаков

Базовые концепты

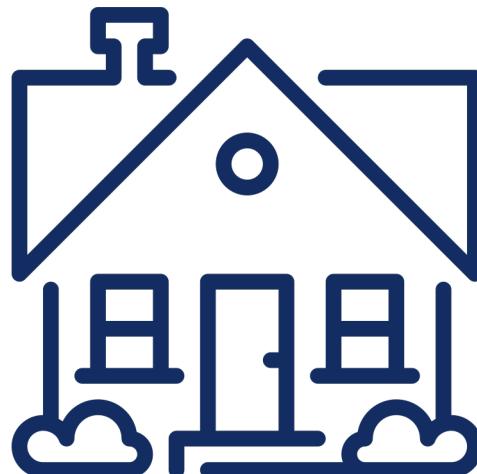
Признаки объектов (features)



250k \$

- 270m²
- 2 спальни
- 2 ванные
- 1 парковочное место

...



375k \$

- 420m²
- 4 спальни
- 3 ванные
- 2 парковочных места

...



179k \$

- 120m²
- 1 спальни
- 1 ванные
- 0 парковочных место

...

Базовые концепты

Выборка (dataset)

- $X = (x_i, y_i)_{i=1,l}$
- X – выборка

- (x_i, y_i) – элемент выборки, пара (объект, ответ)
- $x_i = (f^1_i, f^2_i, \dots, f^n_i)$
- f^k_i – значение признака k на объекте i

Базовые концепты

Выборка (dataset)

Количество ванных комнат	Количество спален	Стоимость
Площадь		
2	2	250k \$
420m ²	3	375k \$
120m ²	1	179k \$

Базовые концепты

Обучающая выборка

Количество ванных комнат	Количество спален	Стоимость
Площадь		
2	2	250k \$
420m ²	3	375k \$
120m ²	1	179k \$

Базовые концепты

Тестовая выборка

	Площадь	Количество ванных комнат	Количество спален	Стоимость
	270m ²	2	2	250k \$
	420m ²	3	4	375k \$
	120m ²	1	1	179k \$

Базовые концепты

Модель

- $a: X \rightarrow Y$
- $a(x) = y$
- A – семейство моделей

ФУНКЦИЯ ПОТЕРЬ

- $Q(a, X)$ – ошибки модели $a(x)$ на выборке X

Базовые
концепты

Базовые концепты

Базовые концепты

Объекты и признаки:

- x – объект
- y – ответ
- $(f_1, f_2 \dots f_n)$ – признаки, описывающие объекты

Модель:

- $a: X \rightarrow Y$
- $a(x) = y$
- A – семейство моделей

- X – пространство объектов
- Y – пространство ответов

Оценка качества

- $Q(a, X)$ – ошибки модели $a(x)$ на группе объектов X

Базовые концепты

Логика построения модели

1. Определяем объекты
2. Формулируем задачу:
на какой вопрос касательно объектов мы хотим
ответить?
3. Определяем признаки, собираем выборку
4. Строим модель
5. Оцениваем её качество

Виды обучения

Виды обучения

Классификация постановок задач

- По типу задач
- По виду обучения
- По алгоритмам
- По области применения
- Смешенная

Виды обучения

По типу задач

- Обучение с учителем/Supervised learning
- Обучение без учителя/Unsupervised learning
- Частичное обучение/Semi-supervised learning
- Обучение с подкреплением/ Reinforcement learning

Виды обучения

По виду обучения

- Classic learning
- Active learning
- Online learning
- Transfer learning
- ...

Виды обучения

По алгоритмам

- Ensemble learning
- Deep learning
- Bayesian learning
- ...

Виды обучения

По данным и задачам

- Tabular data
- Timeseries data
- Computer vision
- Natural language processing
- Cognitive technologies
- Recommender systems
- Learning to rank
- ...

Виды обучения

Смешенная (особенно грустно)

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning
- Deep learning
- Active learning
- Online learning
- Transfer learning
- ... you name it =(

Виды обучения

По типу задач



- Обучение с учителем/Supervised learning
- Обучение без учителя/Unsupervised learning
- Частичное обучение/Semi-supervised learning
- Обучение с подкреплением/ Reinforcement learning

В курсе мы фокусируемся на **обучении с учителем**, но затронем и другие виды обучения.

Постановка задач обучения

Задачи машииного обучения

Обучение с учителем

- Задача – найти верный ответ для каждого объекта:
 - метку класса или вероятность в случае задачи классификации
 - численное значение в случае регрессии
- Нам доступны верные ответы на достаточно большой группе объектов для обучения
- Мы учим модель находить закономерности между значениями признаков и ответами

Задачи машииного обучения

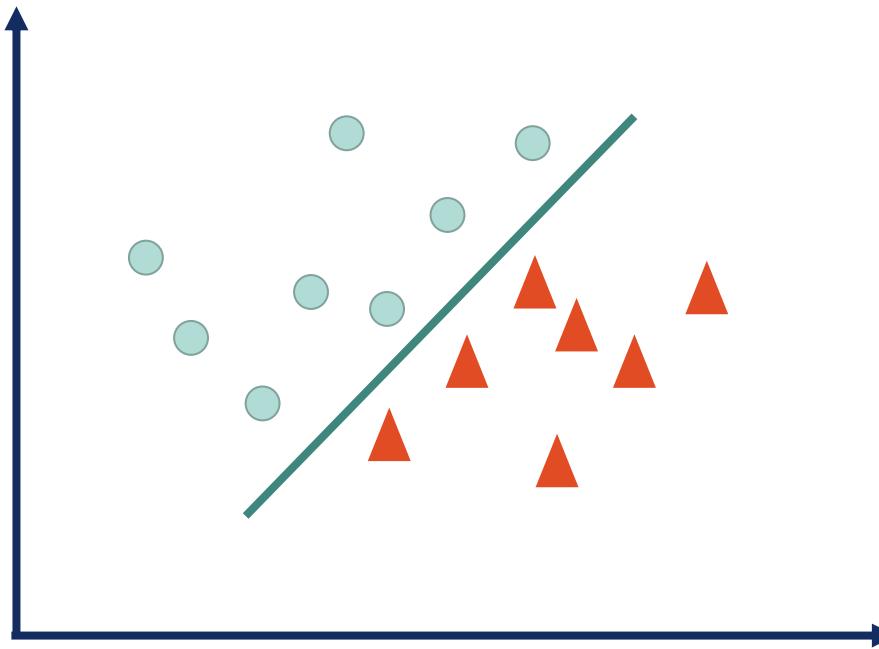
Обучение с учителем

- После обучения мы применяем обученную модель к внешним данным – данным, которые не входили в обучающую выборку
- Объекты из тестовой выборки могут существенно отличаться от объектов в обучающей выборке
- Чем больше верных ответов мы получаем (особенно в тестовой выборке) – тем лучше обучена модель

Задачи машииного обучения

Обучение с учителем

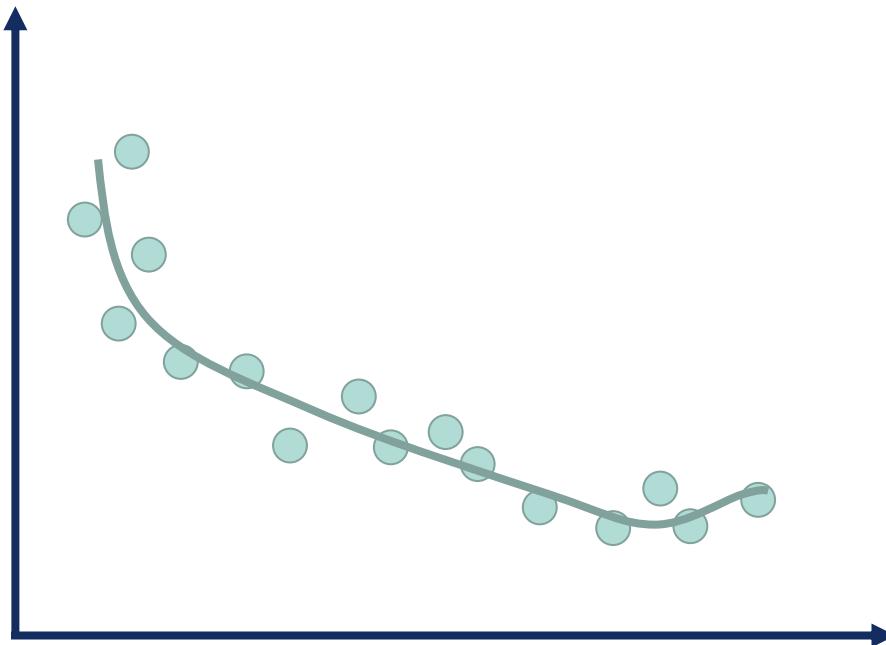
Классификация



Задачи машинного обучения

Обучение с учителем

Регрессия



Задачи машииного обучения

Обучение с учителем

Ранжирование

[Learning to rank - Wikipedia](#)

https://en.wikipedia.org/wiki/Learning_to_rank ▾

Learning to rank or machine-learned ranking (MLR) is the application of machine learning, Often a learning-to-rank problem is reformulated as an optimization problem with respect to one of these metrics. Examples of ranking quality ...

[Applications](#) · [Feature vectors](#) · [Evaluation measures](#) · [Approaches](#)

[Ranking \(information retrieval\) - Wikipedia](#)

[https://en.wikipedia.org/wiki/Ranking_\(information_retrieval\)](https://en.wikipedia.org/wiki/Ranking_(information_retrieval)) ▾

Ranking of query results is one of the fundamental **problems** in information retrieval (IR), the scientific/engineering discipline behind search engines. Given a ...

[\[PDF\] Statistical Ranking Problem](#)

<https://web.stanford.edu/group/mmds/slides/zhang-mmds.pdf> ▾

Statistical Ranking Problem. Tong Zhang. Yahoo! Inc. New York City. Joint work with. David Cossack. Yahoo! Inc. Santa Clara ...

[Problem & Preference Ranking | SSWM](#)

www.sswm.info/content/problem-preference-ranking ▾

Problem/Preference Ranking is a participatory technique that allows analysing and identifying problems or preferences stakeholder share in order to implement ...

Задачи машииного обучения

Обучение без учителя

Задача – разделить объекты на группы (кластеры) таким образом, чтобы:

- группы соответствовали исходной структуре данных
- объекты внутри одной группы были схожи
- объекты из разных группы существенно различались

Задачи машинного обучения

Обучение без учителя

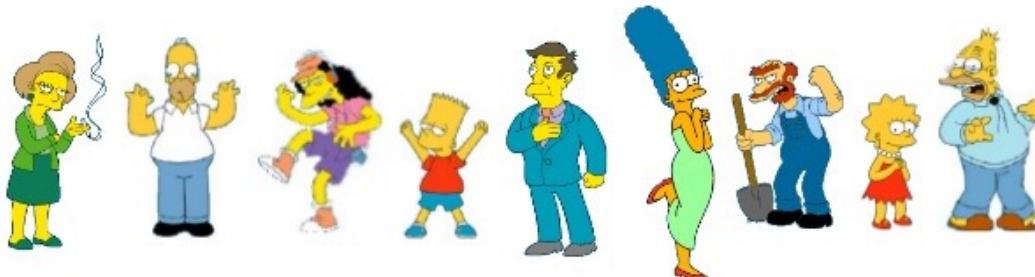
В чем сложность:

- нам не известна исходная структура данных
- ответы для обучения не доступны
- не известно даже количество групп

Кластеризация

Задачи
машинного
обучения

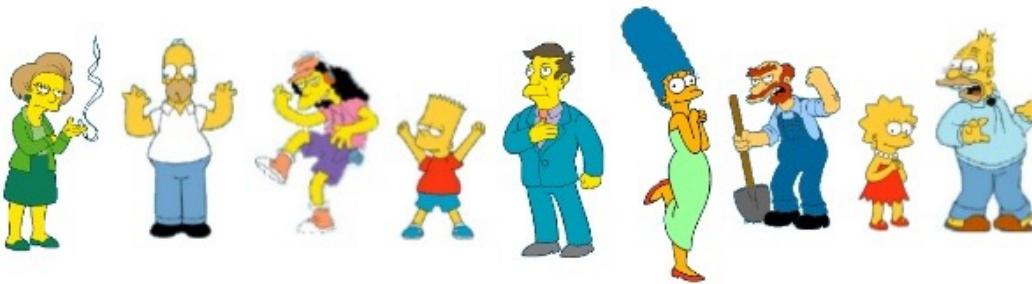
What is a natural grouping among these objects?



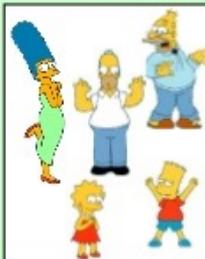
Задачи машинного обучения

Кластеризация

What is a natural grouping among these objects?



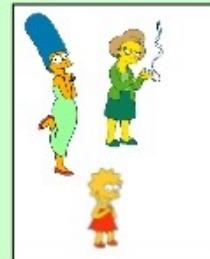
Clustering is subjective



Simpson's Family



School Employees



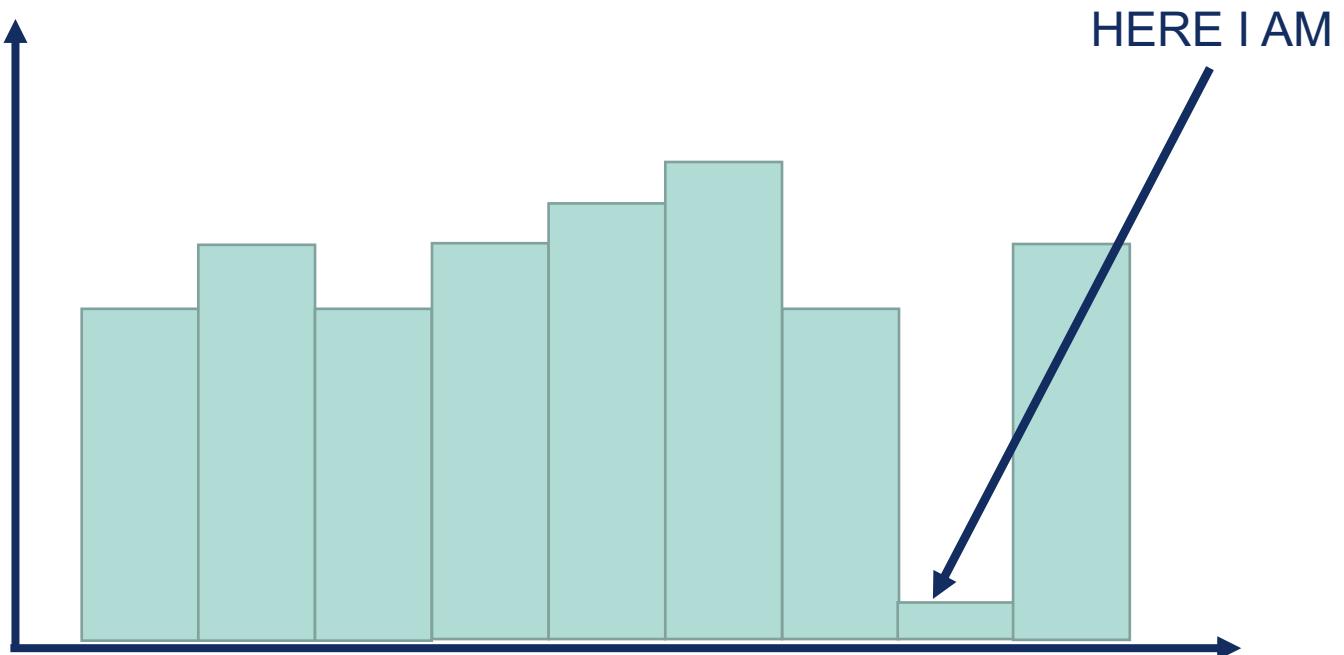
Females



Males

Задачи машииного обучения

Детектирование аномалий



Задачи машинного обучения

Частичное обучение

- На границе между обучением с учителем и обучением без учителя
- Нам доступны ответы на небольшой группе объектов
- Для большинства объектов ответы не доступны

Задачи машииного обучения

Обучение с подкреплением

- Модели доступен ограниченный набор действий
- Модель взаимодействует с динамической средой для получения обратной связи в ответ на выбранное действие
- Обратная связь: штраф или награда
- Чаще всего получение обратной связи осложненно: долго, дорого, вычислительно затратно
- Модель ограничена в получении обратной связи (в единицу времени)

Обучение с подкреплением

- Обучение игре в шахматы, ГО
- Симуляторы движения

Задачи
машинного
обучения

Обзор классических алгоритмов

ML
алгоритмы

Семейства алгоритмов

- Метод ближайших соседей: похожие объекты относятся к одному классу
- Линейные модели
- Деревья решений
- Ансамбли
- Нейронные сети
- и др.

ML
алгоритмы

Метод ближайших соседей



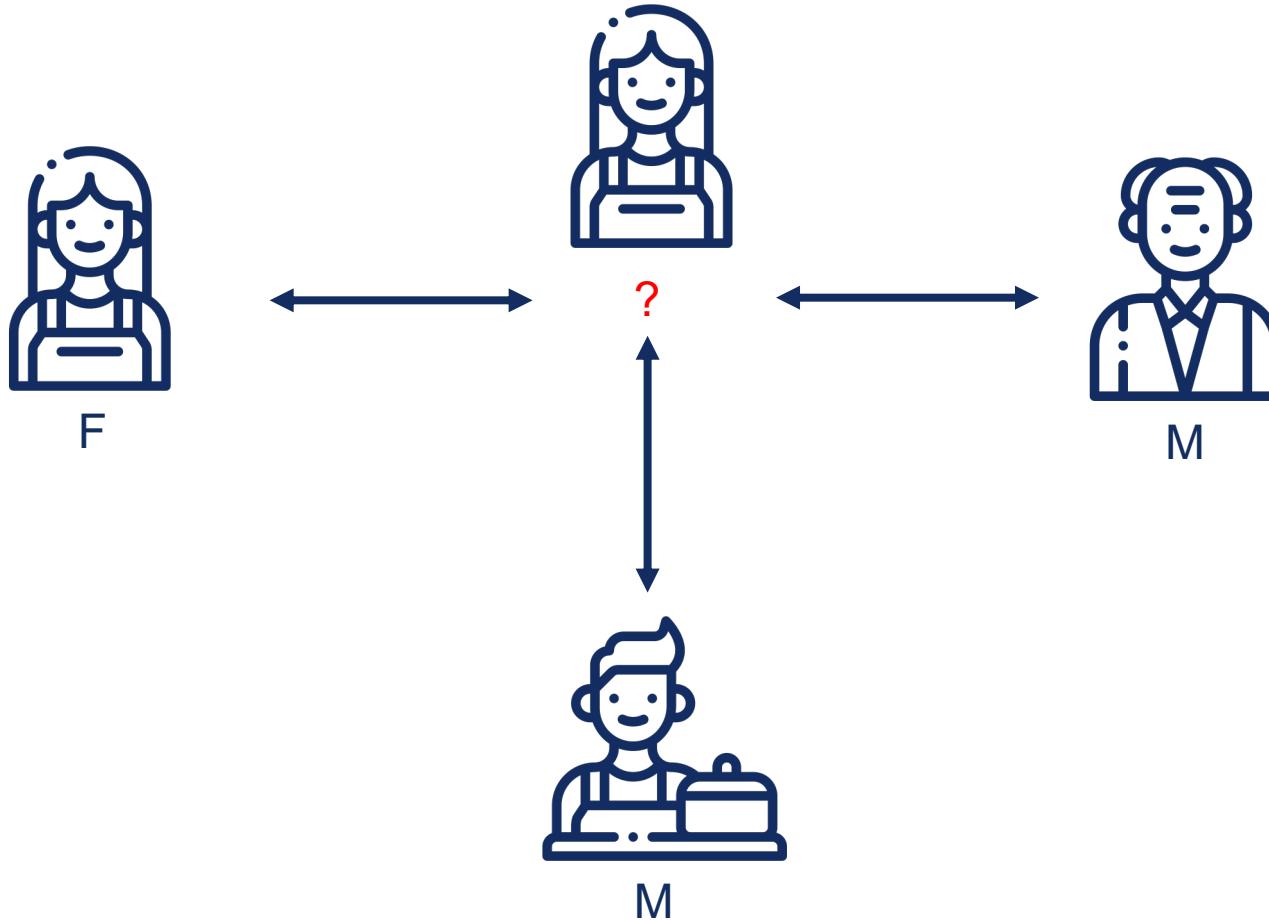
ML
алгоритмы

Метод ближайших соседей



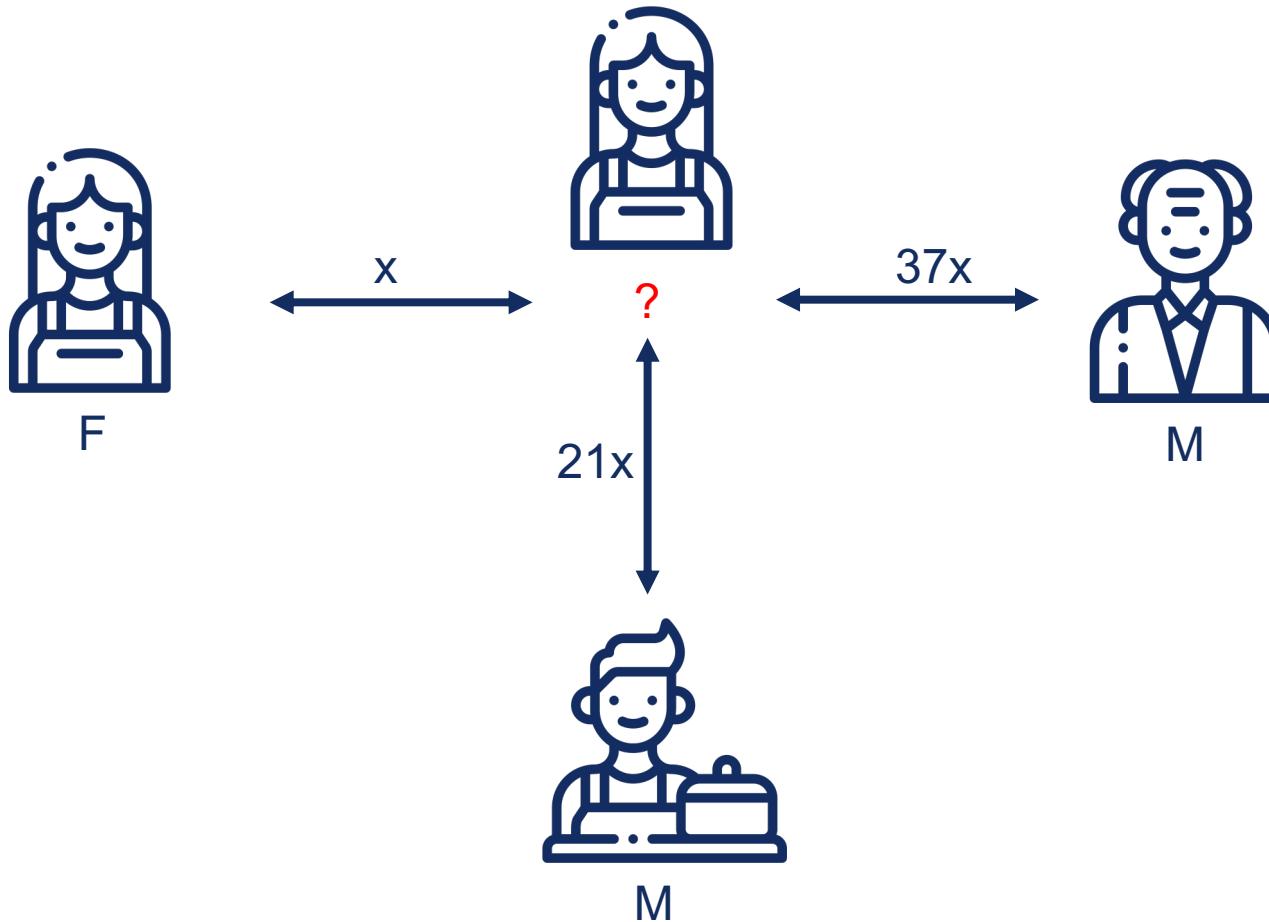
ML
алгоритмы

Метод ближайших соседей



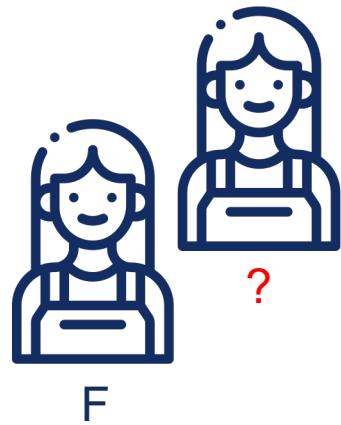
ML алгоритмы

Метод ближайших соседей



ML
алгоритмы

Метод ближайших соседей



F



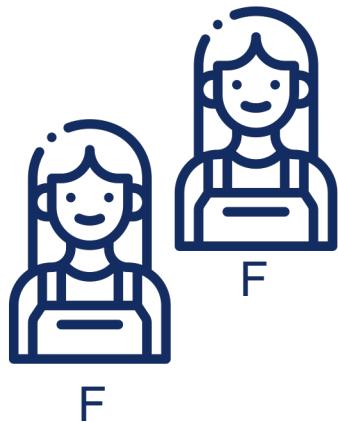
M



M

ML
алгоритмы

Метод ближайших соседей



ML алгоритмы

Метод kNN

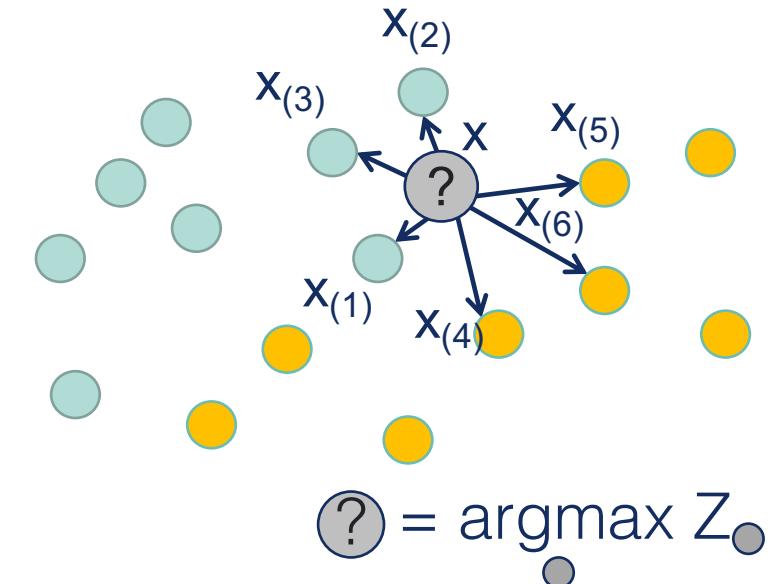
Пример классификации ($k = 6$):

Веса:

- $w(x_{(i)}) = w(i)$
- $w(x(i)) = w(d(x, x_{(i)}))$

$$Z_{\text{teal}} = \frac{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

$$Z_{\text{yellow}} = \frac{w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

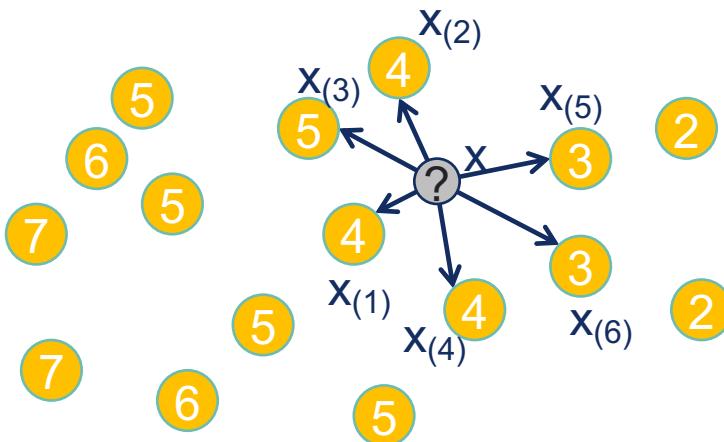


$$\text{?} = \underset{\bullet}{\operatorname{argmax}} Z_{\bullet}$$

ML алгоритмы

Метод kNN в задаче регрессии

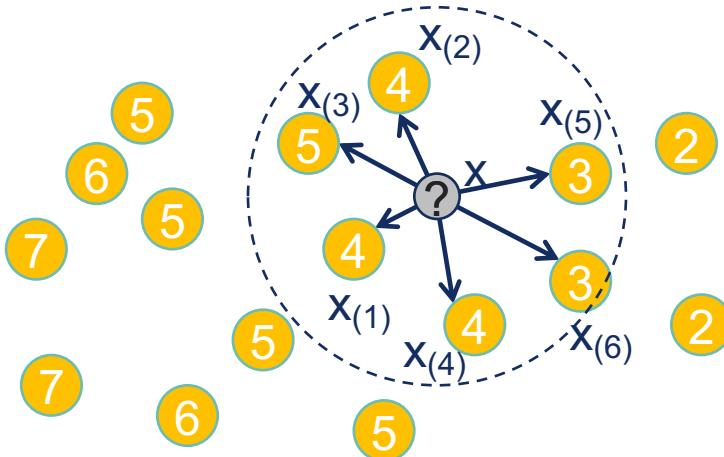
Пример взвешенного kNN ($k = 6$) в задаче регрессии:



ML алгоритмы

Метод kNN в задаче регрессии

Пример взвешенного kNN ($k = 6$) в задаче регрессии:



$$\text{(?)} = \frac{4 \cdot w(x_{(1)}) + 4 \cdot w(x_{(2)}) + 5 \cdot w(x_{(3)}) + 4 \cdot w(x_{(4)}) + 3 \cdot w(x_{(5)}) + 3 \cdot w(x_{(6)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

ML алгоритмы

Семейства алгоритмов

- Метод ближайших соседей: похожие объекты относятся к одному классу
- Деревья решений: класс получается в результате последовательных ответов на простые вопросы
- Ансамбли
- Линейные модели
- Нейронные сети
- и др.

Дерево решений



Уйдет ли этот клиент к конкуренту?

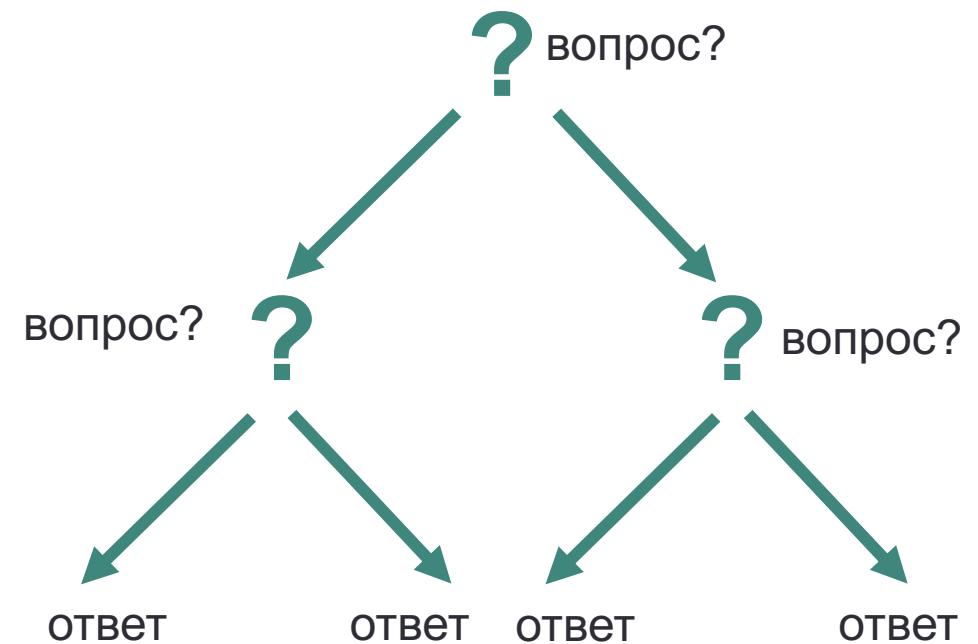
ML
алгоритмы

Дерево решений



Уйдет ли этот клиент к конкуренту?

ML
алгоритмы

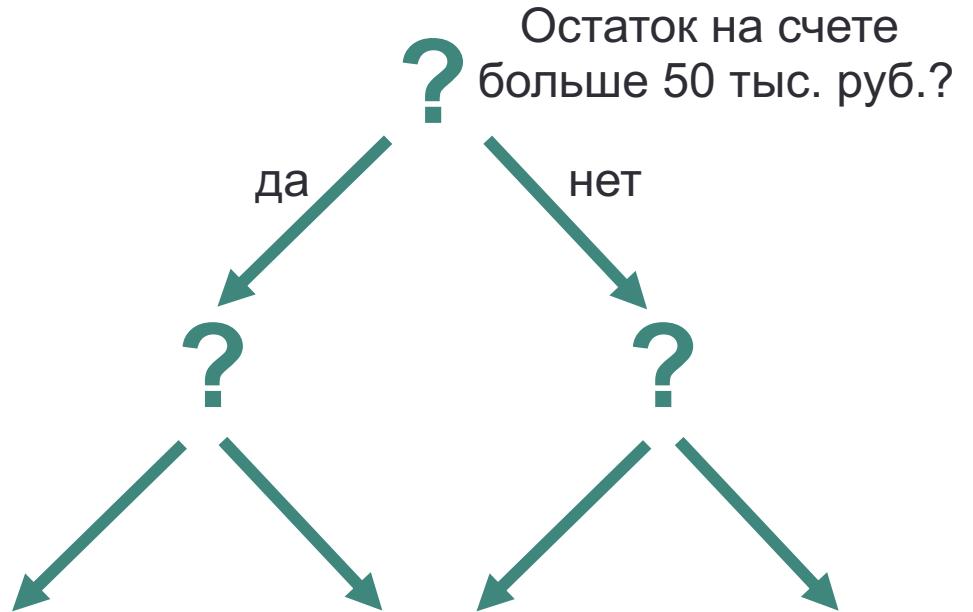


ML алгоритмы

Дерево решений



Уйдет ли этот клиент к конкуренту?

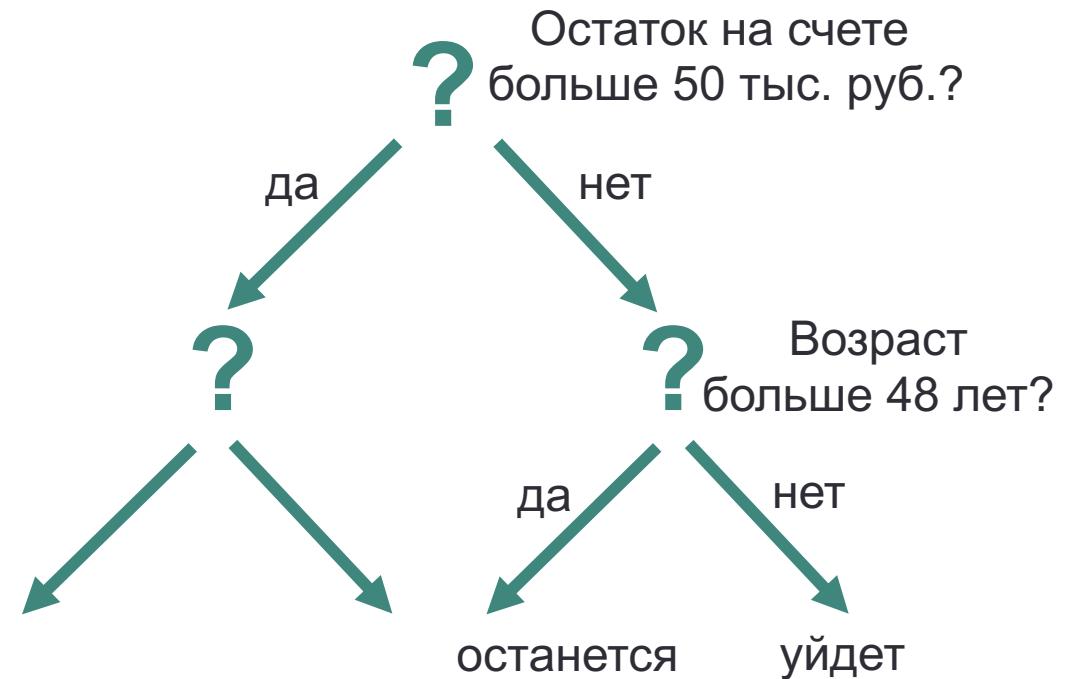


ML алгоритмы

Дерево решений



Уйдет ли этот клиент к конкуренту?

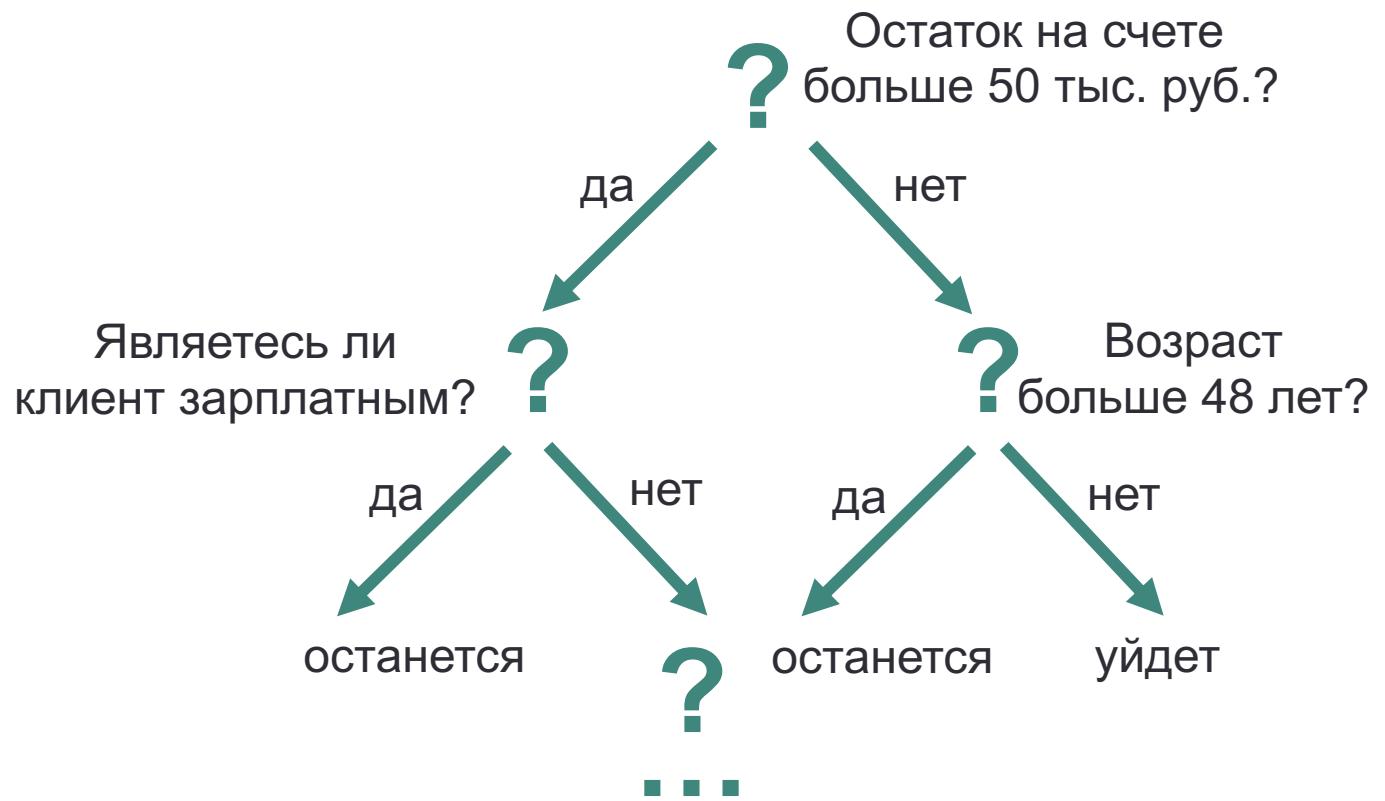


ML алгоритмы

Дерево решений

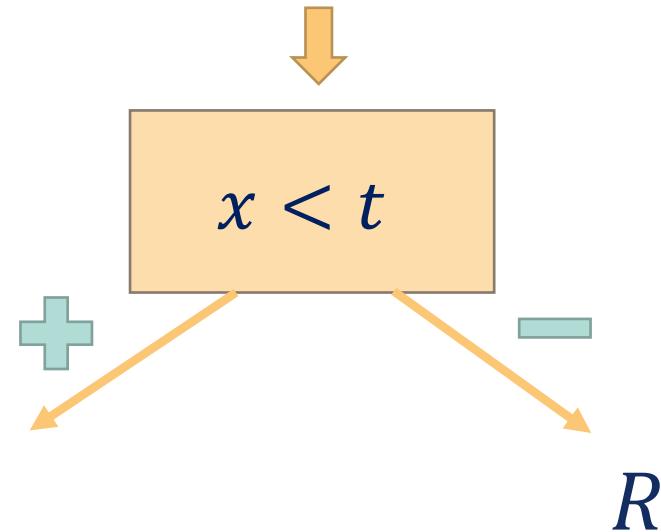


Уйдет ли этот клиент к конкуренту?



Оптимизация разбиения

Вся выборка (n объектов)



ML
алгоритмы

$$G(t) = H(L) + H(R) \rightarrow \min_t$$

$H(R)$ - мера «неоднородности»
(impurity) множества R

Оптимизация разбиения

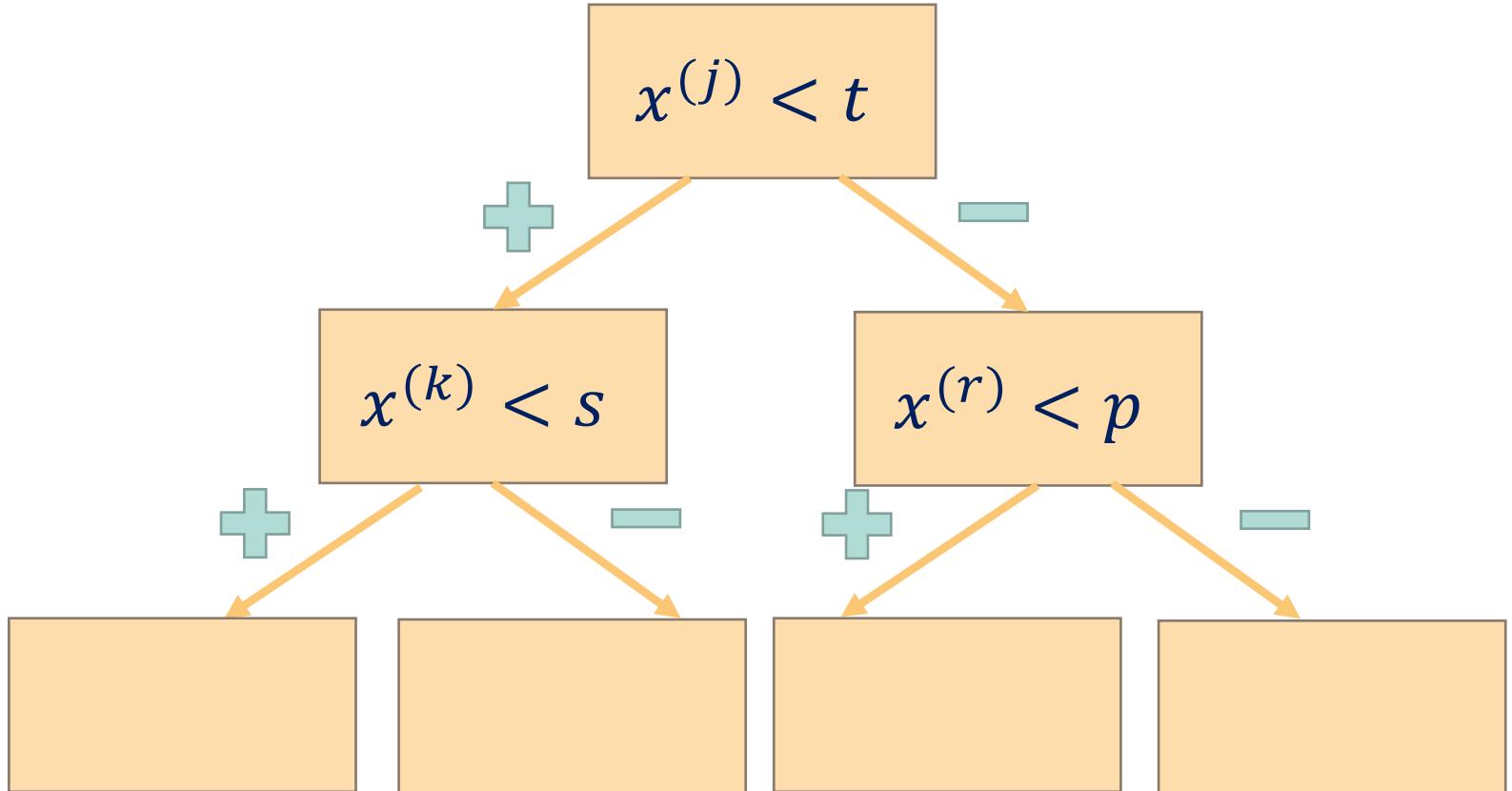
$H(R)$ – мера «неоднородности» множества R

Варианты этой функции:

- 1) Misclassification criteria: $H(R) = 1 - \max\{p_0, p_1\}$
- 2) Entropy criteria: $H(R) = -p_0 \ln p_0 - p_1 \ln p_1$
- 3) Gini criteria: $H(R) = 1 - p_0^2 - p_1^2 = 2p_0p_1$

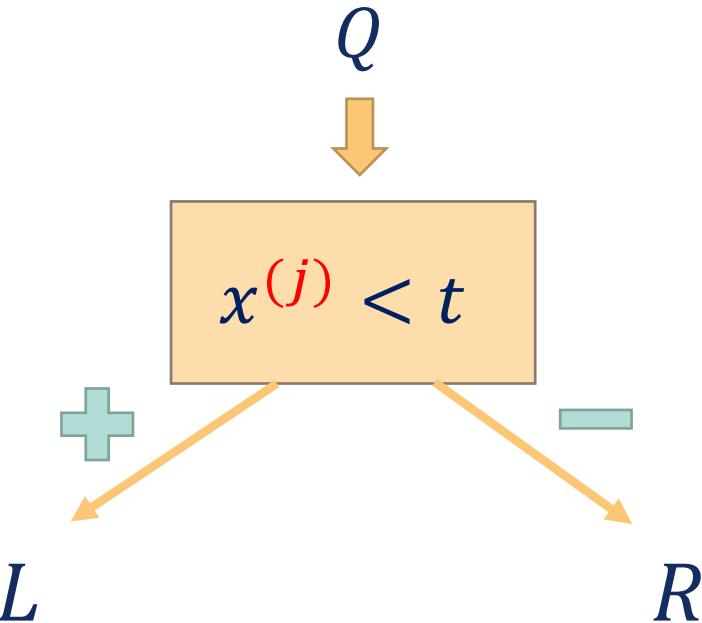
ML алгоритмы

Рекурсивное построение



Процесс можно продолжать в тех узлах, в
которые попадает достаточно много объектов

Рекурсивное построение



ML
алгоритмы

$$G(j, t) = \frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R) \rightarrow \min_{j,t}$$

ML алгоритмы

Семейства алгоритмов

- Метод ближайших соседей: похожие объекты относятся к одному классу
- Деревья решений: класс получается в результате последовательных ответов на простые вопросы
- Ансамбли: решение принимается на основе ответов нескольких моделей
- Линейные модели
- Нейронные сети
- и др.

Композиция моделей



Уйдет ли этот клиент к конкуренту?

ML
алгоритмы

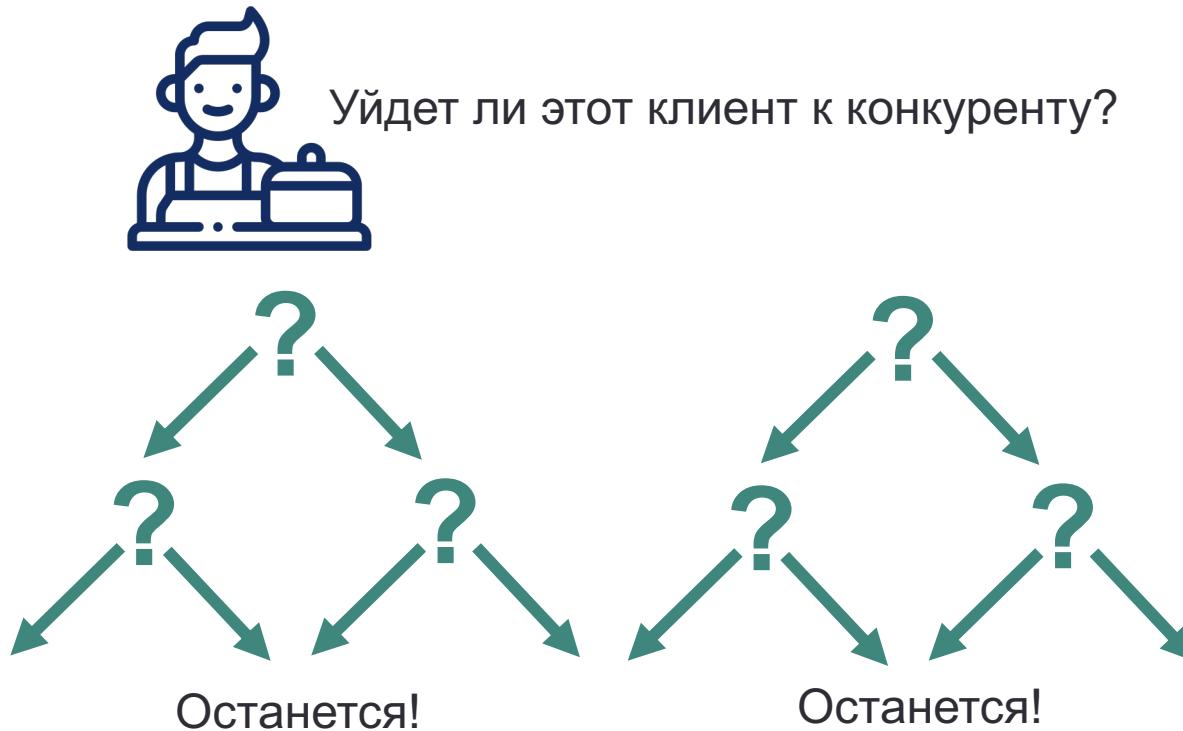
Композиция моделей

ML
алгоритмы



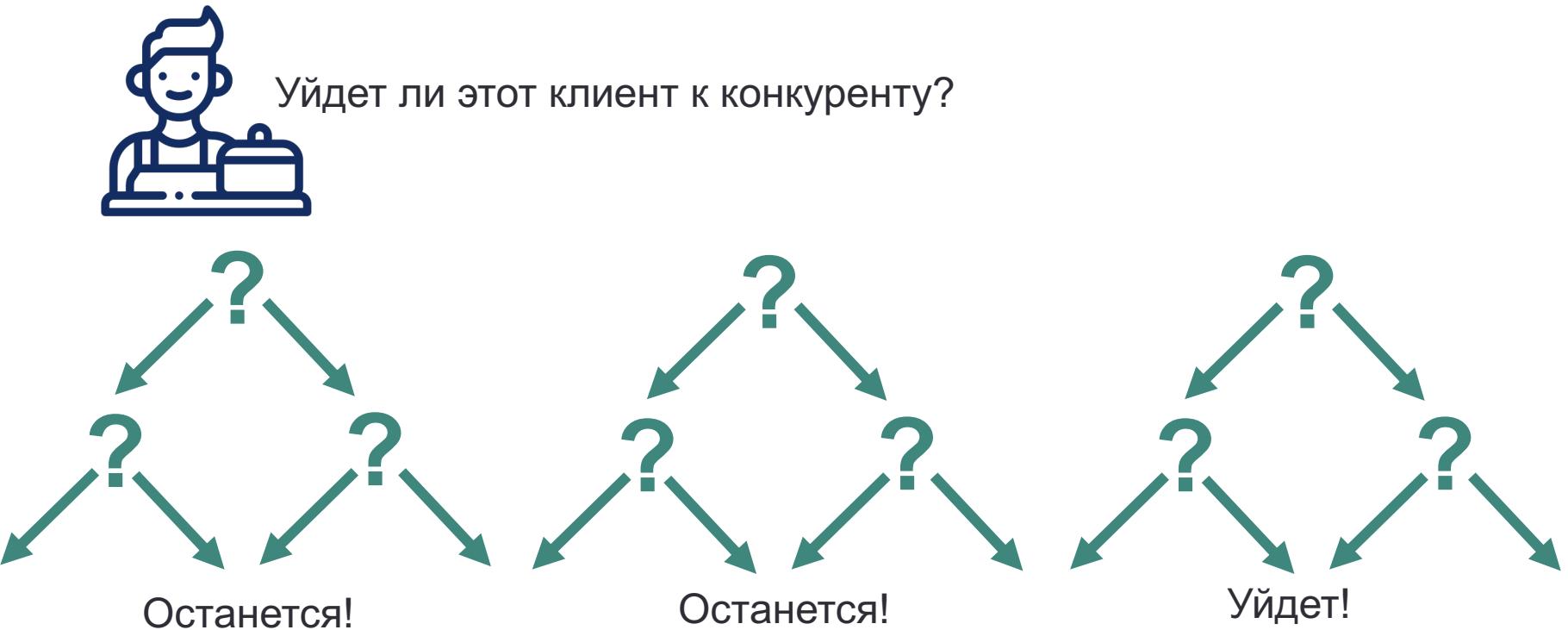
Композиция моделей

ML
алгоритмы



Композиция моделей

ML
алгоритмы

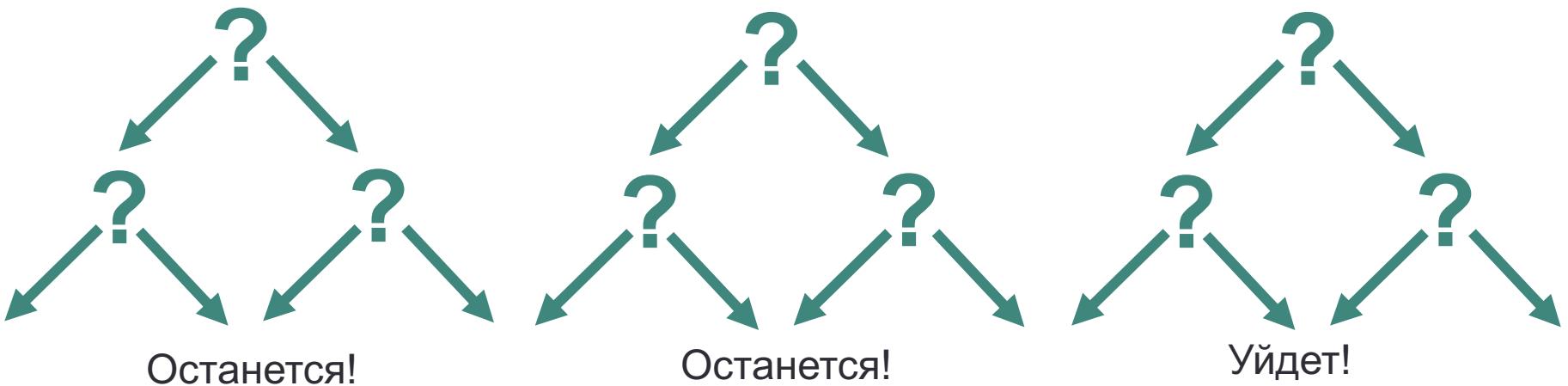


Композиция моделей

ML
алгоритмы



Уйдет ли этот клиент к конкуренту?



2 vs 1

Композиция моделей

Для чего строить композицию?

Идея: за счет использования комбинации из нескольких моделей вместо одной снизить ошибку прогноза

ML
алгоритмы

Проблема: если просто взять выборку X и построить N моделей, они все будут одинаковыми

ML алгоритмы

Композиция моделей

$X^l = (x_i, y_i)_{i=1}^l \subset X \times Y$ – обучающая выборка

$a(x) = C(b(x))$ – алгоритм, где

$b: X \rightarrow R$ – базовый алгоритм

$C: R \rightarrow Y$ – решающее правило

R – пространство оценок

Композиция базовых алгоритмов b_1, \dots, b_N :

$$a(x) = C(F(b_1(x), \dots, b_N(x))),$$

где $F: R^N \rightarrow R$ – корректирующая операция.

Решающее правило вводится для удобства работы с задачами классификации, где часто для получения метки класса требуется преобразование ответа модели.

ML алгоритмы

Bootstrap

Проблема: если просто взять выборку X и построить N моделей, они все будут одинаковыми

Идея: с помощью технологии bootstrap сгенерируем подвыборки из исходной выборки с возвращением

Тогда:

X_1, \dots, X_n - подвыборки, полученные методом bootstrap

b_1, \dots, b_n - базовые модели

$y(x)$ - верные ответы

Bootstrap

X_1, \dots, X_n - подвыборки, полученные методом bootstrap

b_1, \dots, b_n - базовые модели

$y(x)$ - верные ответы

ML
алгоритмы

Определим $E_x(b_j(x) - y(x))^2$ - матожидание ошибки базового алгоритма по всем x . Обозначим ошибку $\varepsilon_j(x)$

Предположим, что:

$E_x \varepsilon_j(x) = 0$ – ошибки несмещенные

$E_x \varepsilon_j(x) \varepsilon_k(x) = 0, j \neq k$ – ошибки независимы (не коррелированы)

ML алгоритмы

Bootstrap

Определим $E_x(b_j(x) - y(x))^2$ - матожидание ошибки базового алгоритма по всем x . Обозначим ошибку $\varepsilon_j(x)$

Предположим, что:

$$E_x \varepsilon_j(x) = 0$$

$$E_x \varepsilon_j(x) \varepsilon_k(x) = 0, j \neq k$$

Рассмотрим $a(x) = \frac{1}{N} \sum_{j=1}^N b_j(x)$ - среднее N базовых алгоритмов

Давайте оценим ошибку композиции и сравним с ошибкой базового алгоритма

ML алгоритмы

Bootstrap

$$a(x) = \frac{1}{N} \sum_{j=1}^N b_j(x)$$

Оценим ошибку композиции

$$\mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^N b_j(x) - y(x) \right)^2$$

внесем $y(x)$ внутрь суммы

$$\mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^N (b_j(x) - y(x)) \right)^2 = \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^N \varepsilon_j(x) \right)^2$$

раскроем квадрат суммы

$$\frac{1}{N^2} \left(\mathbb{E}_x \sum_{j=1}^N \varepsilon_j^2(x) + \sum_{j \neq k} \mathbb{E}_x \varepsilon_j(x) \varepsilon_k(x) \right)$$

второе слагаемое равно 0

$$\frac{1}{N^2} \mathbb{E}_x \sum_{j=1}^N \varepsilon_j^2(x)$$

Bootstrap

$$a(x) = \frac{1}{N} \sum_{j=1}^N b_j(x)$$

Пусть ошибки одинаково распределены

$\frac{1}{N^2} E_x \sum_{j=1}^N \varepsilon_j^2(x) = \frac{1}{N} E_x \varepsilon_j^2$ - таким образом ошибка усредненной модели в n раз меньше

Значит, построение композиции для снижения ошибки имеет смысл

ML алгоритмы

Семейства алгоритмов

- Метод ближайших соседей: похожие объекты относятся к одному классу
- Деревья решений: класс получается в результате последовательных ответов на простые вопросы
- Ансамбли: решение принимается на основе ответов нескольких моделей
- Линейные модели: класс линейно зависит от характеристик объекта
- Нейронные сети и др.

Линейная модель



Стоит ли одобрить кредит данному клиенту?

ML
алгоритмы

ML алгоритмы

Линейная модель



Стоит ли одобрить кредит данному клиенту?



ML алгоритмы

Линейная модель



Стоит ли одобрить кредит данному клиенту?



15



99



35

ML алгоритмы

Линейная модель



Стоит ли одобрить кредит данному клиенту?



15



99



35

Линейная регрессия

$$a(x) = \langle w, x \rangle + w_0$$

ML
алгоритмы

Линейная классификация

$$a(x) = \langle w, x \rangle + w_0$$

А как получить ответ в задаче классификации?

- Выберем метки класса 1 и -1 для удобства
- Формализуем пороговое решающее правило

$$a(x) = \text{sign}(\langle w, x \rangle + w_0)$$

Если скалярное произведение неотрицательное – класс 1, в противном случае класс -1

Интерпретация

$$a(x) = \langle w, x \rangle + w_0$$

- Абсолютные значения весов w_1, \dots, w_n – можно интерпретировать как важность признаков
- Знак можно интерпретировать как класс, за который "голосует" признак

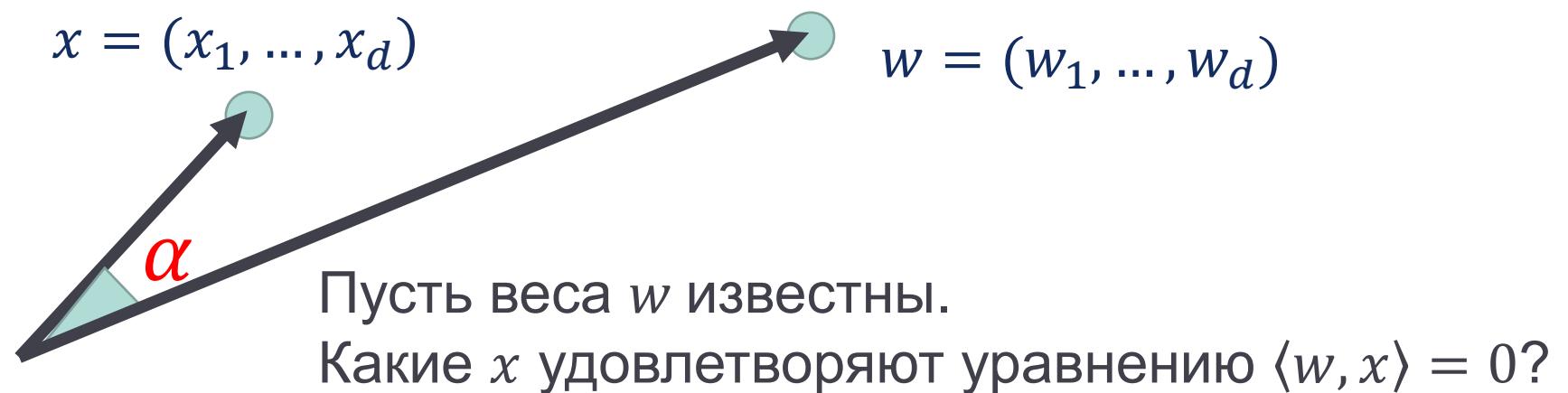
*для данной интерпретации признаки должны быть откалиброваны или бинаризованы

Геометрическая интерпретация

$$a(x) = \langle w, x \rangle + w_0$$

Веса задают гиперплоскость, разделяющую классы.

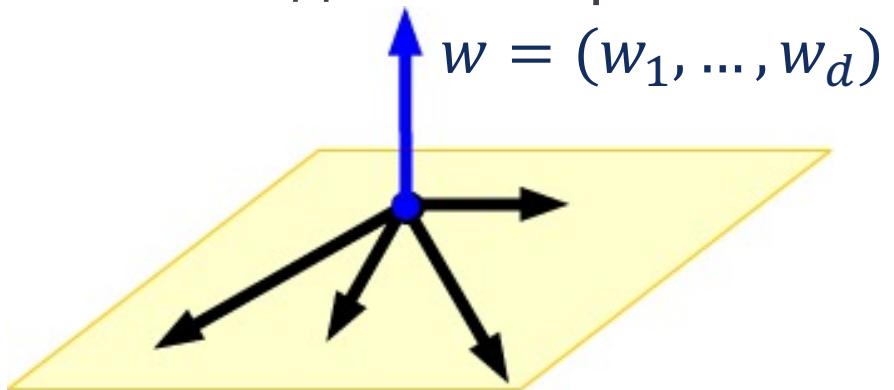
Откуда берется гиперплоскость?



Геометрическая интерпретация

$$a(x) = \langle w, x \rangle + w_0$$

Веса задают гиперплоскость, разделяющую классы.



- Если объект “над” гиперплоскостью, то его вектор и вектор w смотрят в одну сторону, скалярное произведение положительное
- Если объект с другой стороны от гиперплоскости – скалярное произведение отрицательное

Задача оптимизации

$$Q(a(w)) = \sum_{i=1}^n L(y_i, a_i) \rightarrow \min$$

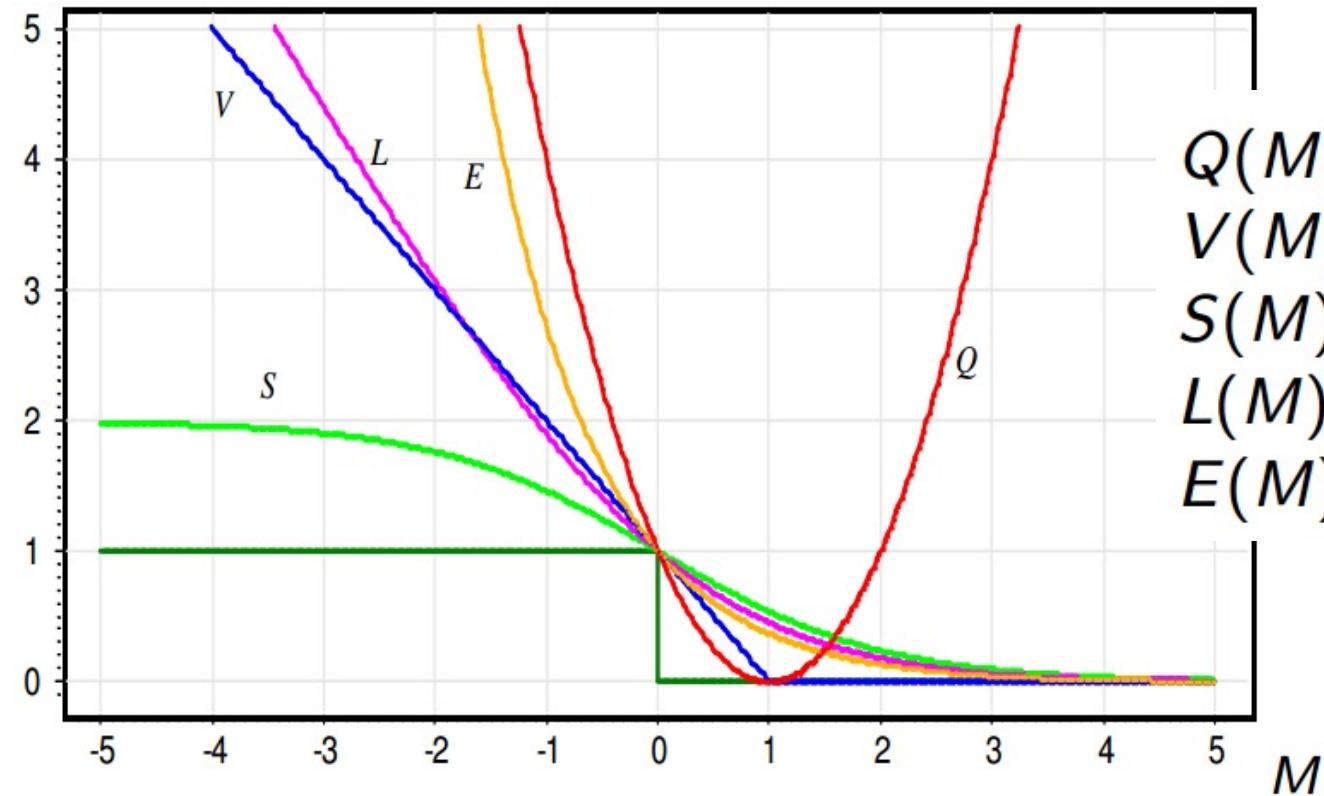
ML
алгоритмы

Общий вид задачи оптимизации задан, остается
несколько степеней свободы:

- метод оптимизации
- функция потерь (L)
- дополнительные ограничения

ML алгоритмы

ФУНКЦИЯ ПОТЕРЬ



$$Q(M) = (1 - M)^2$$

$$V(M) = (1 - M)_+$$

$$S(M) = 2(1 + e^M)^{-1}$$

$$L(M) = \log_2(1 + e^{-M})$$

$$E(M) = e^{-M}$$

Задача оптимизации

$$a(x) = \langle w, x \rangle + w_0$$

$$Q = \sum_{i=1}^l L(y_i, a(x_i)) + \gamma V(w) \rightarrow \min_w$$

Гребневая регрессия
(Ridge regression):

$$V(w) = \|w\|_{l2}^2 = \sum_{n=1}^d w_n^2$$

LASSO (least absolute
shrinkage and selection
operator):

$$V(w) = \|w\|_{l1} = \sum_{n=1}^d |w_n|$$

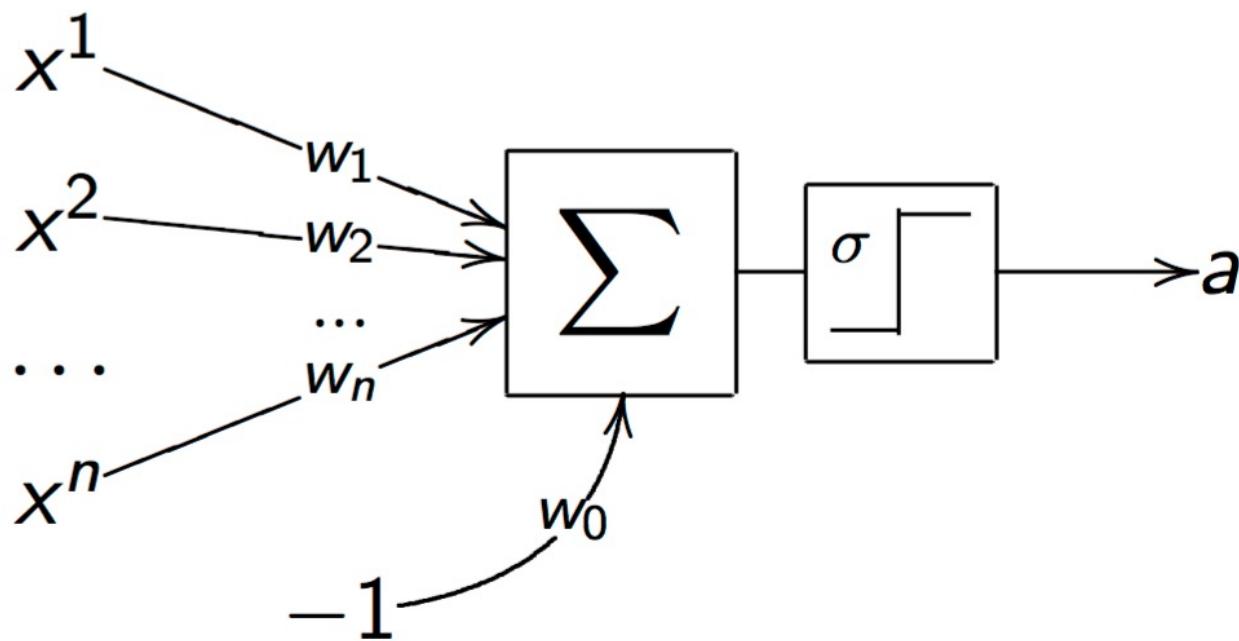
ML алгоритмы

Семейства алгоритмов

- Метод ближайших соседей: похожие объекты относятся к одному классу
- Деревья решений: класс получается в результате последовательных ответов на простые вопросы
- Ансамбли: решение принимается на основе ответов нескольких моделей
- Линейные модели: класс линейно зависит от характеристик объекта
- Нейронные сети: нелинейная комбинация линейных моделей и др.

ML алгоритмы

Линейная модель нейрона



Чтобы получить линейный классификатор, достаточно взять $\sigma = \text{sign}(z)$

Усложняем модель

1. Строим линейную модель над линейной моделью
 $a(x) = W_2 \cdot (W_1 \cdot x + b_1) + b_2 = W \cdot x + b$ – снова линейная модель

ML
алгоритмы

Усложняем модель

1. Строим линейную модель над линейной моделью

$a(x) = W_2 \cdot (W_1 \cdot x + b_1) + b_2 = W \cdot x + b$ – снова линейная модель

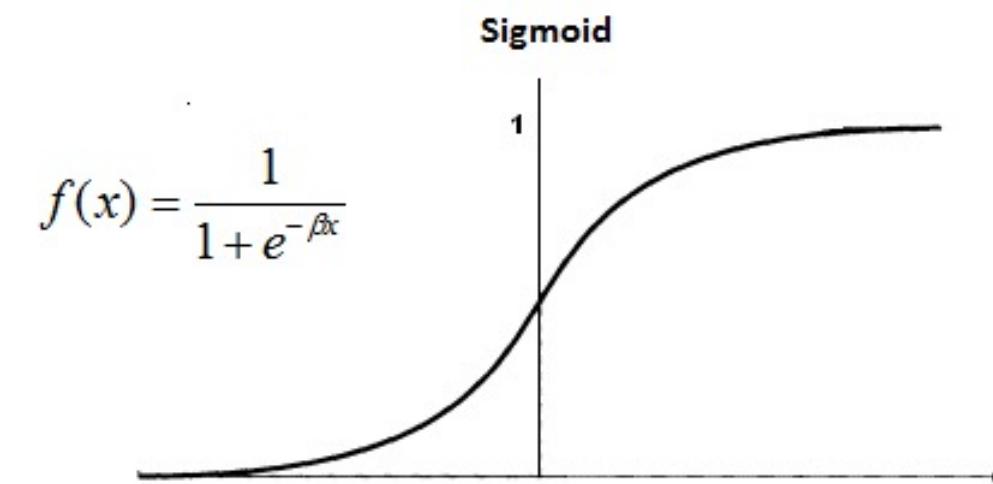
2. Добавляем нелинейность

$a(x) = W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2 \neq W \cdot x + b$ – это уже не линейная модель, значит у нас получилось усложнить

ML
алгоритмы

Усложняем модель

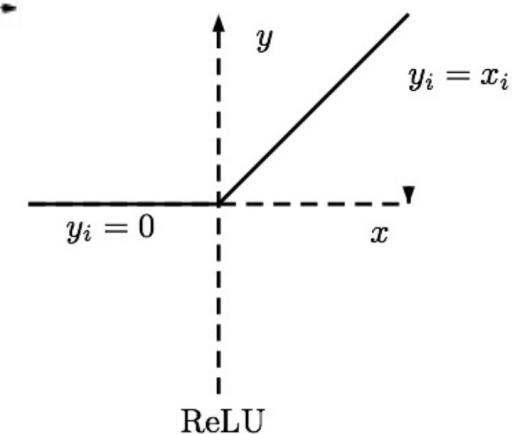
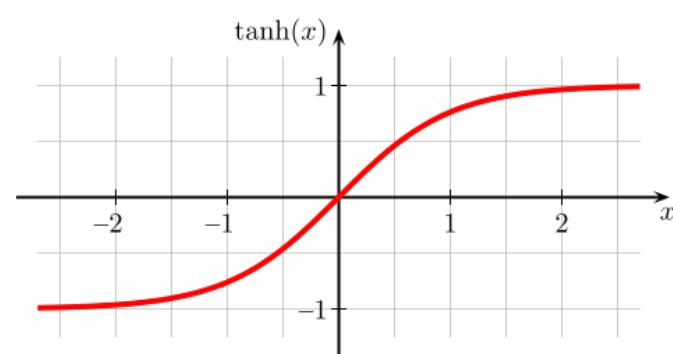
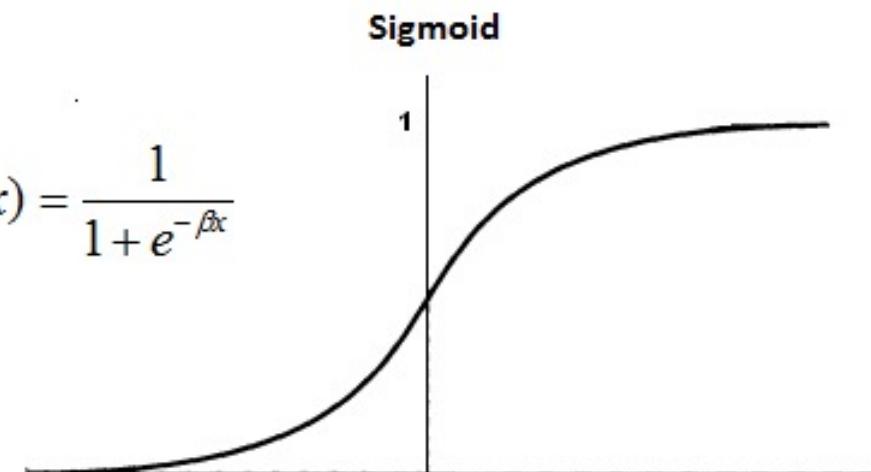
Как ввести нелинейность?



ML алгоритмы

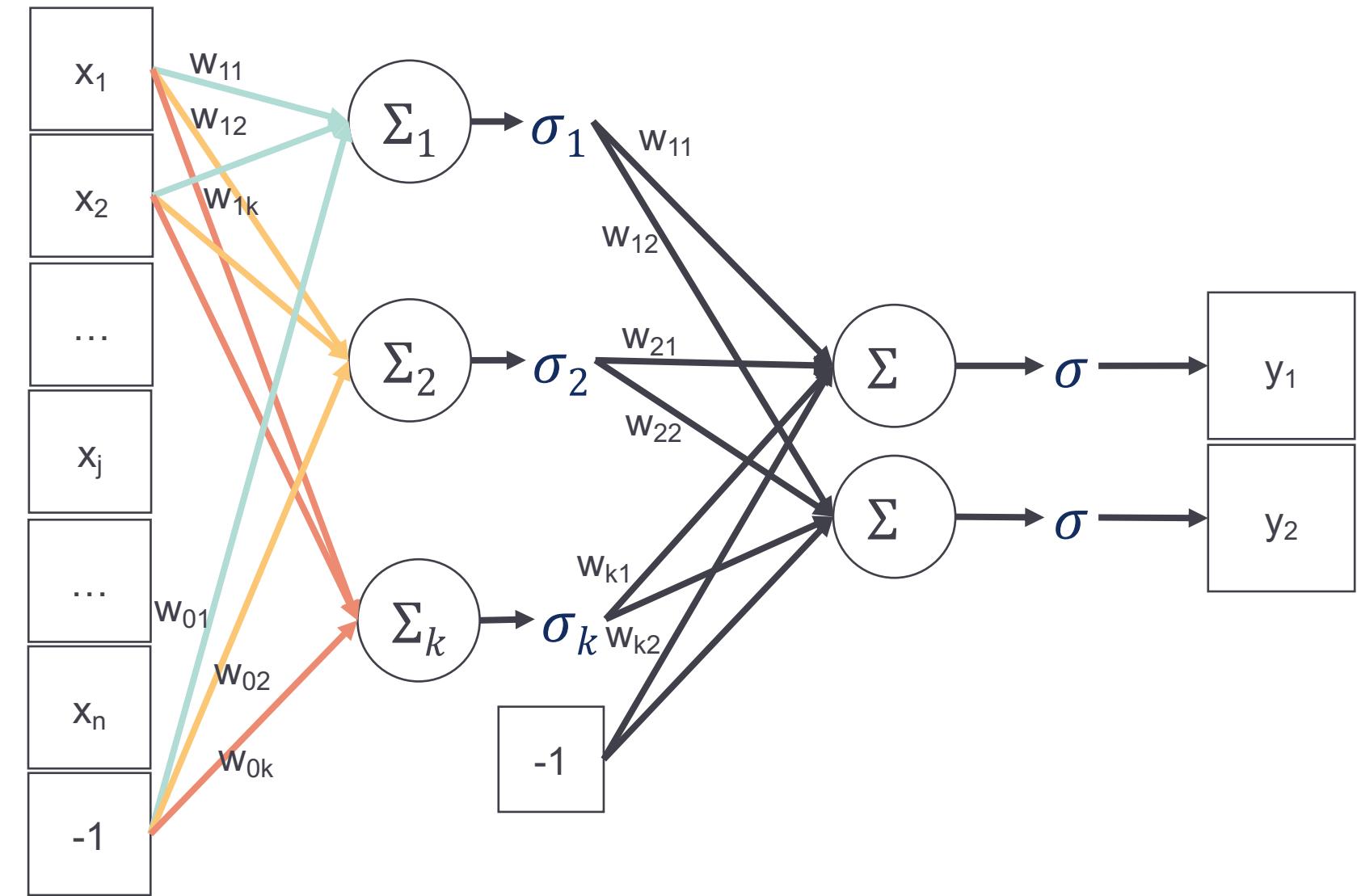
Усложняем модель

Как ввести нелинейность?



ML алгоритмы

Собираем полносвязную нейронную сеть

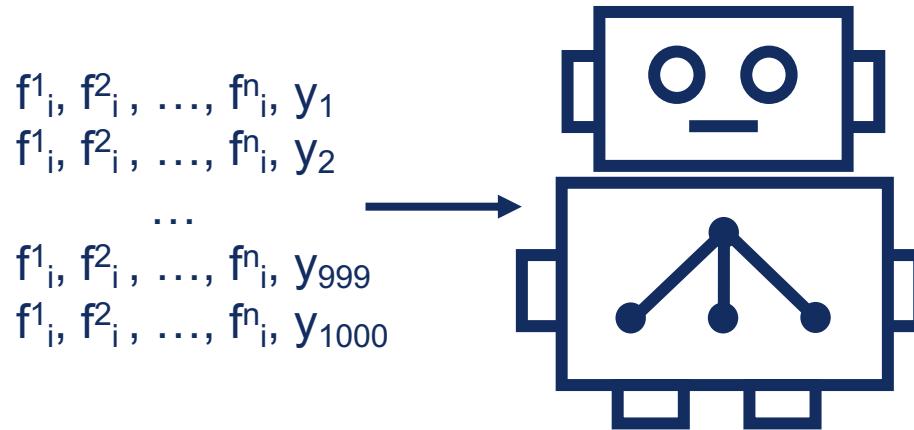


Жизненный цикл модели

Жизненный цикл модели

Модель

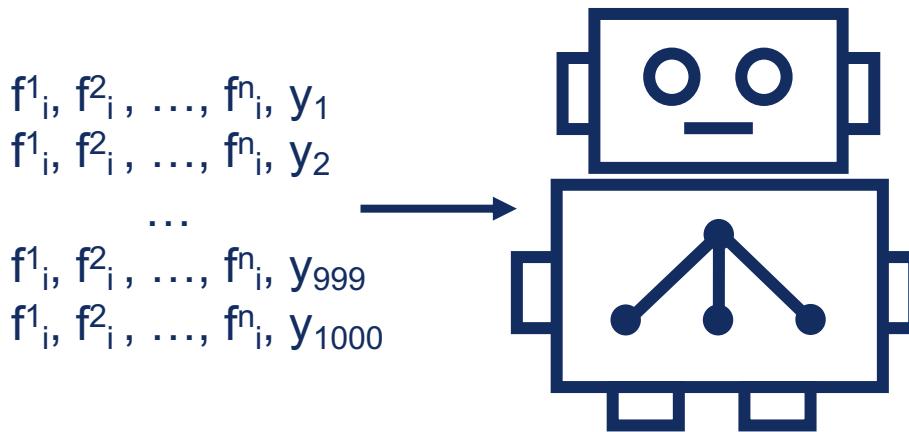
Обучение модели:



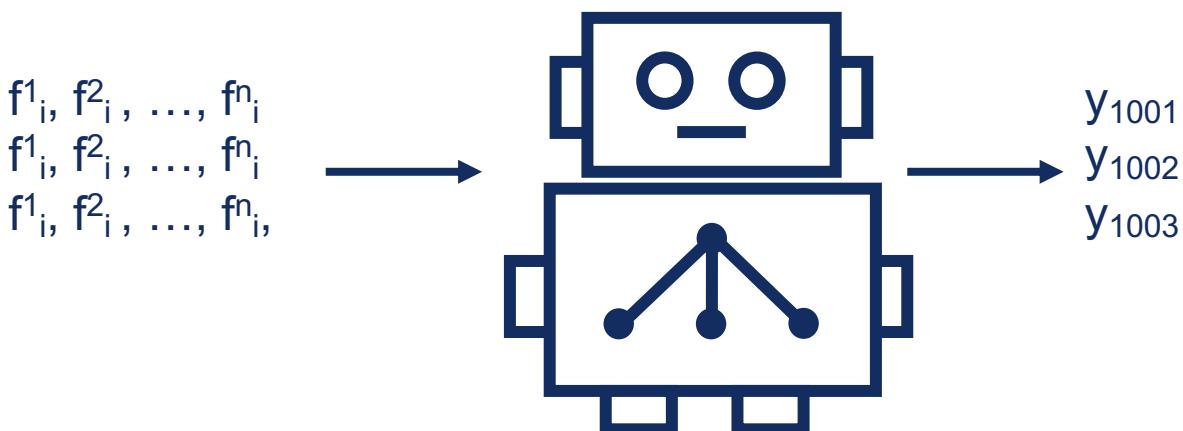
Жизненный цикл модели

Модель

Обучение модели:



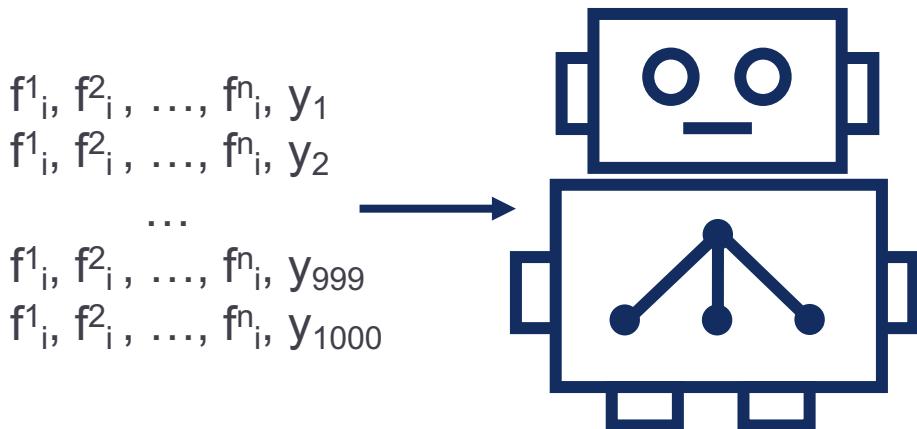
Применение модели:



Жизненный цикл модели

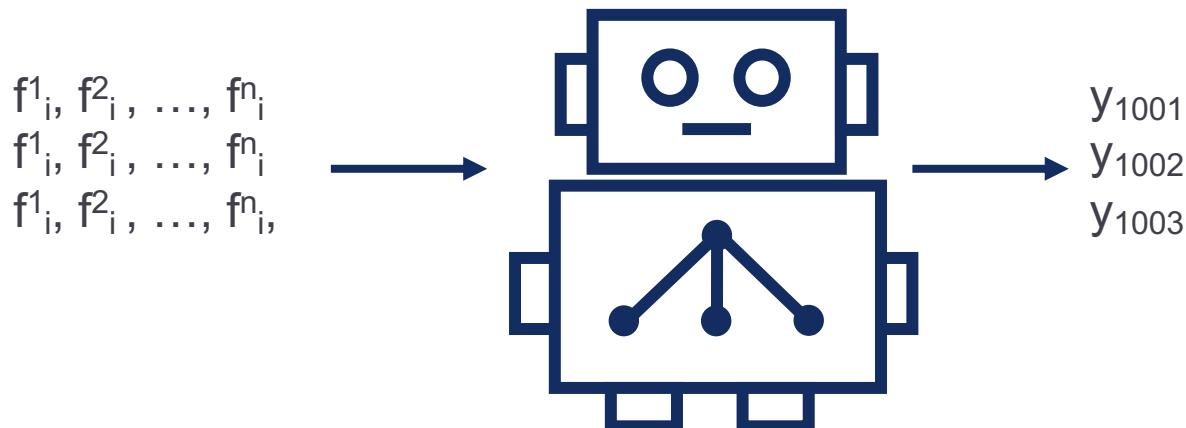
Модель

Обучение модели:



$Q_{\text{train}}(a, X)$ – ошибки модели $a(x)$ на обучающей выборке

Применение модели:



$Q_{\text{test}}(a, X)$ – ошибки модели $a(x)$ на тестовой выборке

Жизненный цикл модели

Модель

- $Q_{\text{train}}(a, X)$ – ошибки модели $a(x)$ на обучении
- $Q_{\text{test}}(a, X)$ – ошибки модели $a(x)$ на teste

?

О чём говорят:

- Ошибка на обучении высокая?
- Ошибка на обучении низкая?

Жизненный цикл модели

Модель

- $Q_{\text{train}}(a, X)$ – ошибки модели $a(x)$ на обучении
- $Q_{\text{test}}(a, X)$ – ошибки модели $a(x)$ на teste

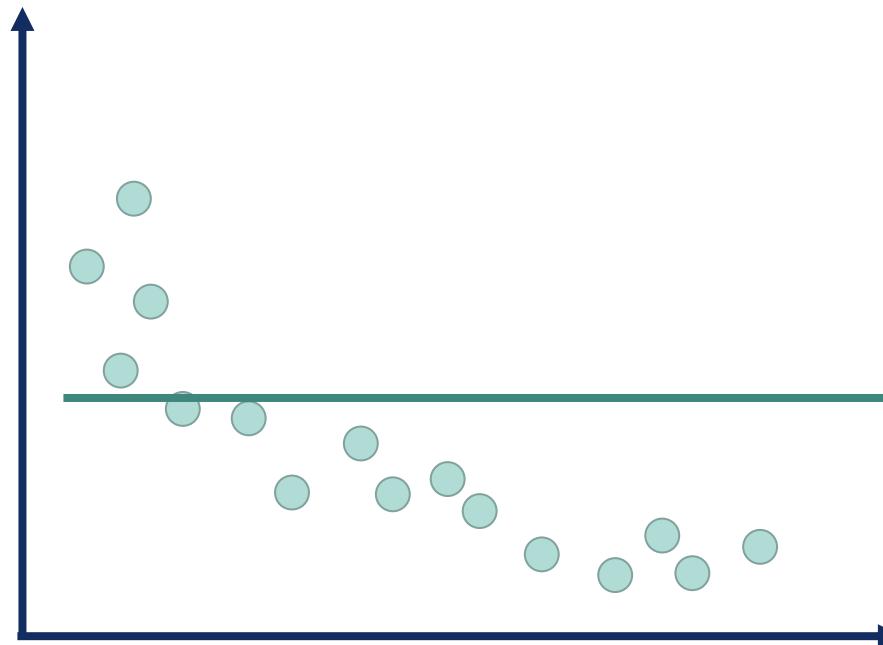
?

О чём говорят:

- Ошибка на обучении высокая?
- Ошибка на обучении низкая?
- Ошибка на обучении немного ниже, чем на teste?
- Ошибка на обучении существенно ниже, чем на teste?

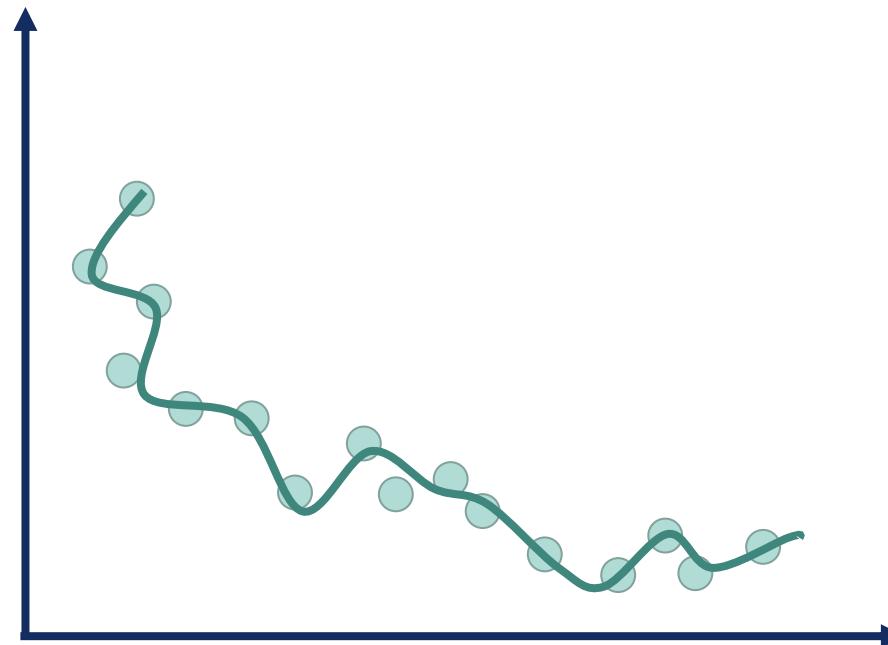
Жизненный цикл модели

Недообучение



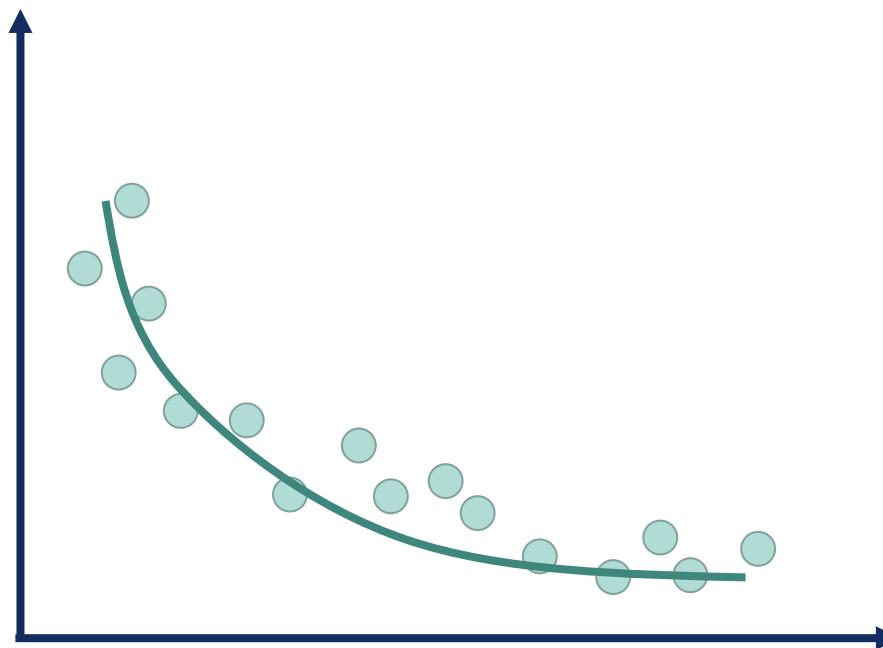
Жизненный цикл модели

Переобучение



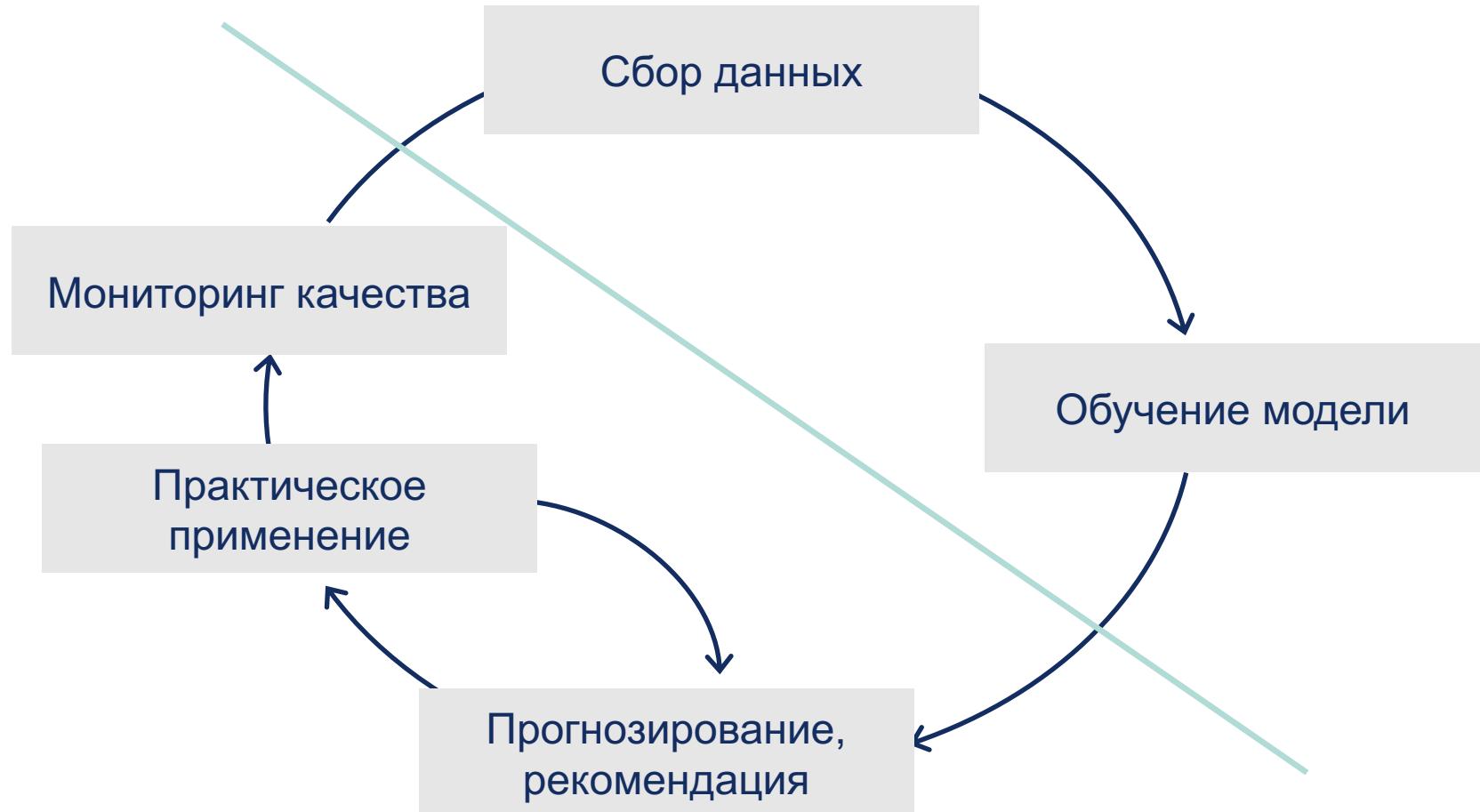
Жизненный цикл модели

Качественная модель



Жизненный цикл модели

Индустриальное применение



Инструменты для анализа данных

Data analysis tools and technologies

1. Operation systems
2. Code repository
3. Software engineering
4. Administration
5. Data bases, SQL
6. ETL, Pipelines
7. Visualization, dashboarding
8. Distributed computing
9. Cloud Platforms

OS

OS Unix

- Terminal
- Bash
- Remote server
- Setting up, updating the system
- Virtual environment
- Package managers: apt-get, aptitude

OS

Mac OS

- Terminal
- Bash
- RDP
- Setting up, updating the system
- Virtual environment
- Development kit: xcode, homebrew

OS

OS Windows

- Terminal
- Putty
- WSL - Windows Subsystem for Linux
- RDP
- Setting up, updating the system
- Virtual environment

Code repository

Control version systems

- git, svn, cvs
- most popular: git, github, gitlab
- Setting up a repository
- Groups, members, access rights
- Cloning, push/pull
- Branches
- Code review & pull requests

Software engineering

Software developments

- Scripting language (solid knowledge)
- Compiled language (basic understanding)
- Development environment (IDE)
- Testing approaches
- Debugging
- Code style
- Software system architecture

Software engineering

Scripting language (Python)

Python (preferably), R

- Programming paradigms
- Syntax
- Standard Template Library
- Libraries for Data Science
- Interactive mode, scripting mode, package mode

Compiled programming language

- At least read C++/Java code

Software engineering

Python libraries

- Standard libraries: os, math, collections, datetime, json, etc
- Pandas, Numpy, Scipy, Scikit-learn
- Matplotlib, Seaborn, Plotly
- Python packages for popular ML tools: LightGBM, XGBoost, Catboost, Tensorflow, VowpalWabbit
- Pytorch, Keras
- Keras-RL, Openai

Data Bases

Data sources

- File systems
- SQL DB
- noSQL DB

SQL & noSQL DB

- Access rights
- Reading/Writing
- Replicas
- Temporary tables
- Querying

Administration

Demo services development

- flask or/and django (python)
- Virtual machines
- Containers (Docker)

DevOps (MLOps in our case!)

- Reproducible experiments (DVC)
- Service development (Mlflow, Kubeflow, etc)
- Model Versioning
- Model Monitoring (Evidently AI, Greate expectations, SageMaker, etc)

Python pipelines

- Scikit-learn pipelines
- Construct a pipeline from the given estimators (name, transform)
- Construct a feature union from the given transformers

ETL, Pipelines

Airflow

- Schedule and monitor workflow
- More info: <https://airflow.apache.org/>

MLflow

- An open source platform to manage the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry
- Model tracking
- Model deploying
- Model registry
- More info: <https://mlflow.org/>

Visualization

Python

- Matplotlib
- Seaborn
- Plotly and Dash

BI visualization tools

- Tableau, Looker, etc

Distributed computing

Distributed filesystems

- HDFS
- Data storing, partitioning

Computing

- MapReduce
- Spark, SparkML
- HQL (Hive)
- Pig
- ...

Cloud platforms

Remote work

- Code and data transferring
- Jupyter hub
- Keys generation
- ssh, scp, ...
- Session managers: tmux, screen

Amazon, GCP, Digital Ocean

- Virtual servers
- Dedicated servers
- Data storages (S3, etc)
- ML platforms and tools (AzureML, Kubernetes, Sagemaker)

Infrastructure and tools

To take away

1. Operation systems
2. **Code repository**
3. **Software engineering**
4. Administration
5. Data bases, SQL
6. ETL, **Pipelines**
7. **Visualization, dashboarding**
8. Distributed computing
9. Cloud Platforms

Машинное обучение: базовые концепции машинного обучения

Спасибо!
Эмели Драль