

# Как все начиналось?

1970-е

- ✓ Искусственный интеллект
- ✓ Будущее за многопроцессорными системами
- ✓ Проект 5-е поколение



Сегодня

- ✓ Искусственный интеллект
- ✓ Облачные системы
- ✓ Компании разрабатывают умные системы

Вопрос:

Почему не получилось тогда, и почему должно получиться сейчас?



$X$  — множество *объектов*;

$Y$  — множество *ответов*;

$y: X \rightarrow Y$  — неизвестная зависимость (target function).

**Дано:**

$\{x_1, \dots, x_\ell\} \subset X$  — обучающая выборка (training sample);

$y_i = y(x_i)$ ,  $i = 1, \dots, \ell$  — известные ответы.

**Найти:**

$a: X \rightarrow Y$  — алгоритм, решающую функцию (decision function), приближающую  $y$  на всём множестве  $X$ .

Весь курс машинного обучения — это конкретизация:

- как задаются объекты и какими могут быть ответы;
- в каком смысле « $a$  приближает  $y$ »;
- как строить функцию  $a$ .

Рассмотрим систему несовпадающих точек  $x_i$  ( $i \in 0, 1, \dots, N$ ) из некоторой области  $D$ . Пусть значения функции  $f$  известны только в этих точках:

$$y_i = f(x_i), \quad i = 1, \dots, N.$$

Задача интерполяции состоит в поиске такой функции  $F$  из заданного класса функций, что

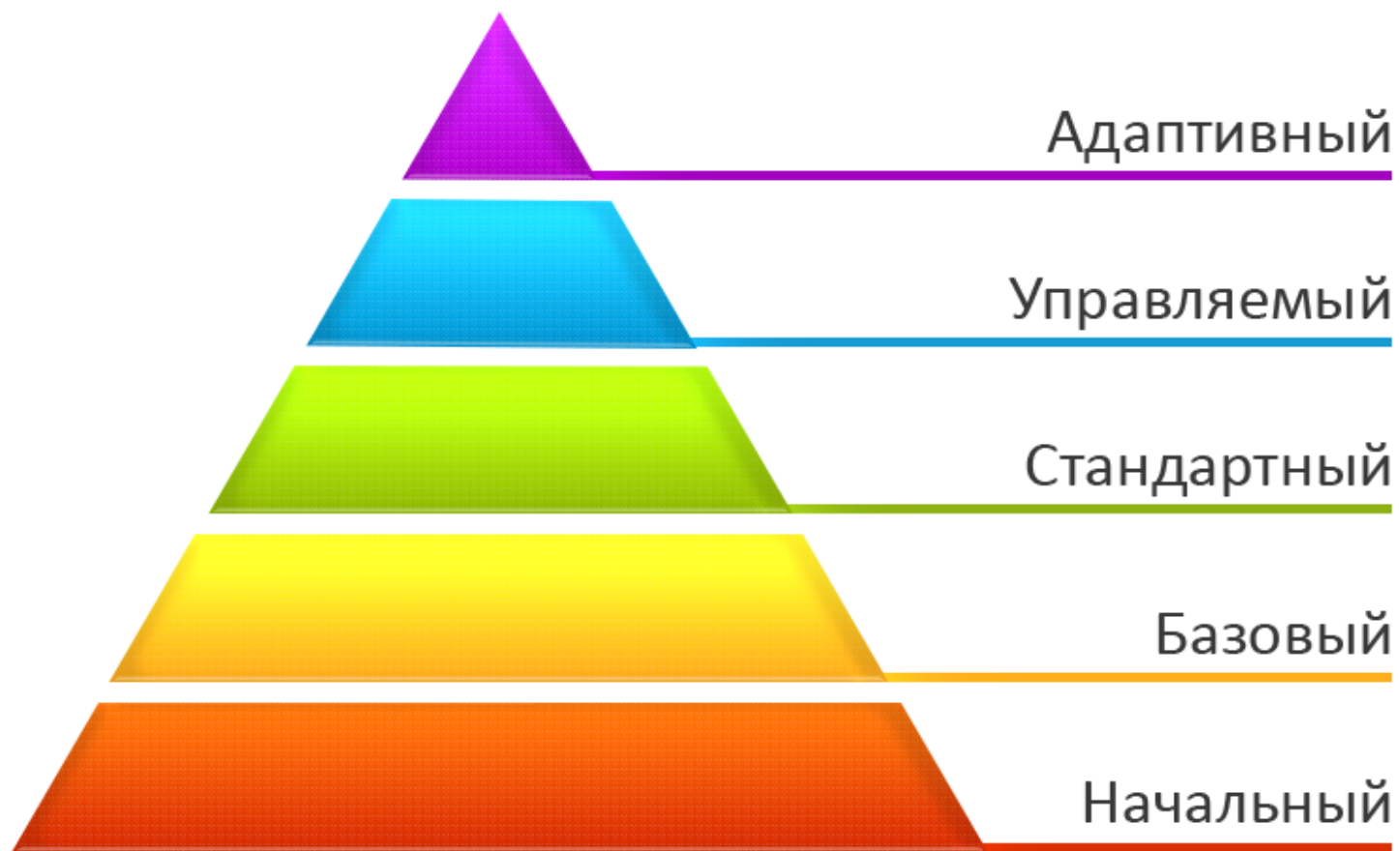
$$F(x_i) = y_i, \quad i = 1, \dots, N.$$

- Точки  $x_i$  называют **узлами интерполяции**, а их совокупность — **интерполяционной сеткой**.
- Пары  $(x_i, y_i)$  называют **точками данных** или **базовыми точками**.
- Разность между «соседними» значениями  $\Delta x_i = x_i - x_{i-1}$  — **шагом интерполяционной сетки**. Он может быть как переменным, так и постоянным.
- Функцию  $F(x)$  — **интерполирующей функцией** или **интерполянт**.

Вопрос:

Чем задача обучения по прецедентам отличается интерполяции?

# Зрелость по Керцнеру



Вопрос:

На каком уровне зрелости сейчас машинное обучение?



# Сложности

Задача интерполяции функции

“в общем виде алгоритмически неразрешима”

# Сложности

Как верифицировать результат?

# Сложности

Как не проспать момент, когда зависимости в системе изменились?

Высокие риски для бизнеса:

- нет гарантии, что при смене зависимости удастся настроить систему с тем же уровнем качества
- Черный лебедь

# Альтернатива

## Data-Driven Development

# Альтернтаива

Рефлексирующие программные системы

[А. Н. Ващенко](#), [Е. Е. Витяев](#), [Н. Г. Загоруйко](#), [А. А. Мальцев](#),  
[Н. Н. Непейвода](#), [Д. Е. Пальчунов](#), [С. Г. Пыркин](#), [А. В. Ткачев](#)