

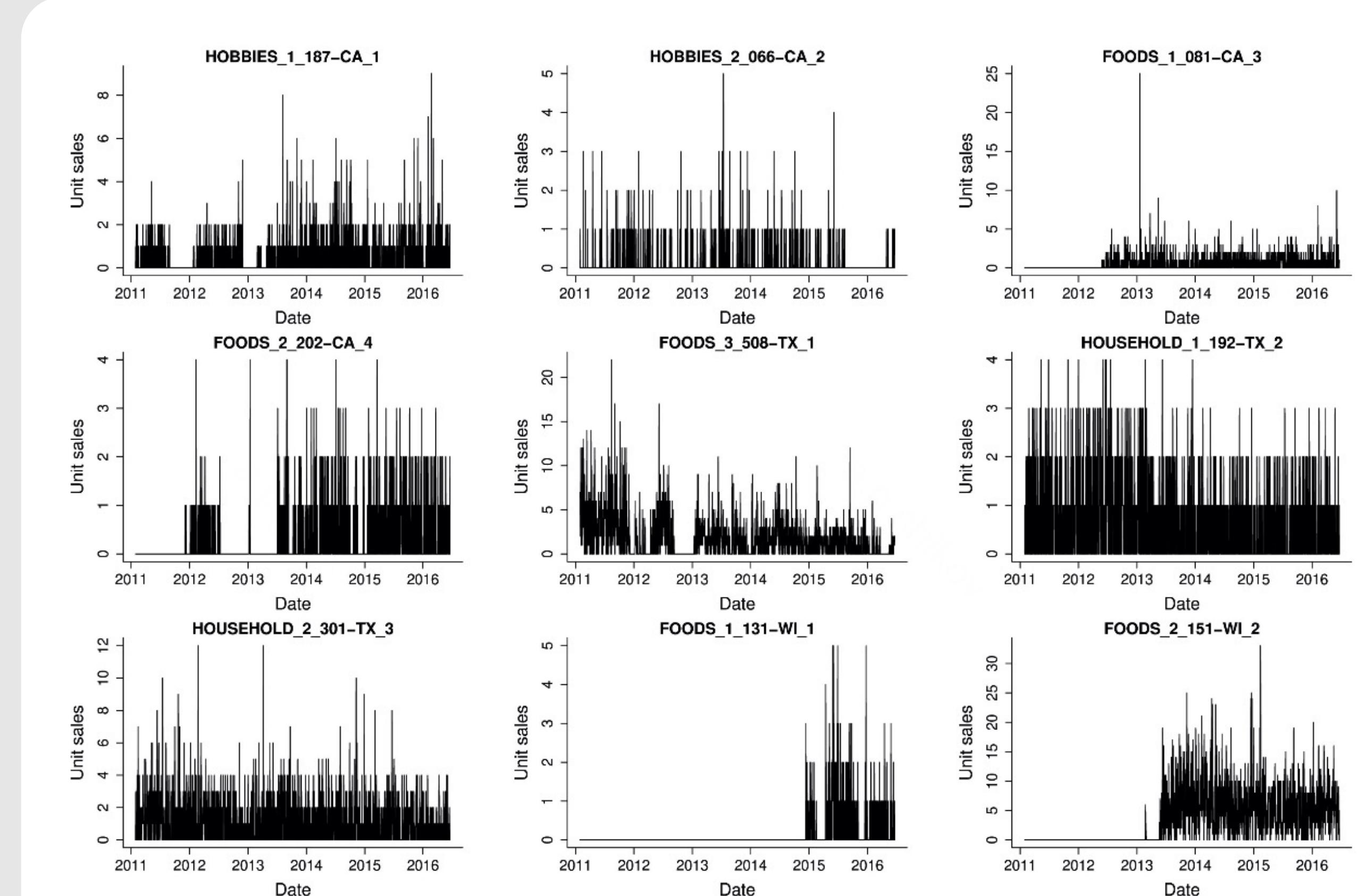
# Табличные методы для прогнозирования. Предсказательные интервалы

ВРЕМЕННЫЕ РЯДЫ IN ACTION

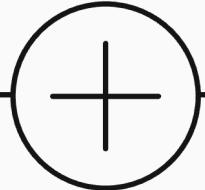
# Классический датасет из ретейла – M4 competition



- Рядов много.
- Разреженность данных.
- Различие в длине рядов в рамках датасета.
- Поведение рядов разное.

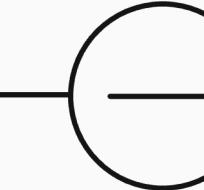


# Проблемы «статистических моделей»



## Достоинства

- Можно хорошо обучиться под конкретный домен или ряд.
- Понятно, как параллельно обучать.



## Недостатки

- Невозможно работать на больших датасетах.
- Доп. данные не используются (не для всех).
- Разреженность данных (в целом и под это подходы есть).
- Различие в длине рядов в рамках датасета.
- Больше требований к количеству и качеству данных.

# Как же быть?



# Сведение к задаче регрессии aka ML

# Формализация

$$Y_t = F(y_{t-1}, y_{t-2}, \dots, X, t)$$

$y_t$  — исторические значения;

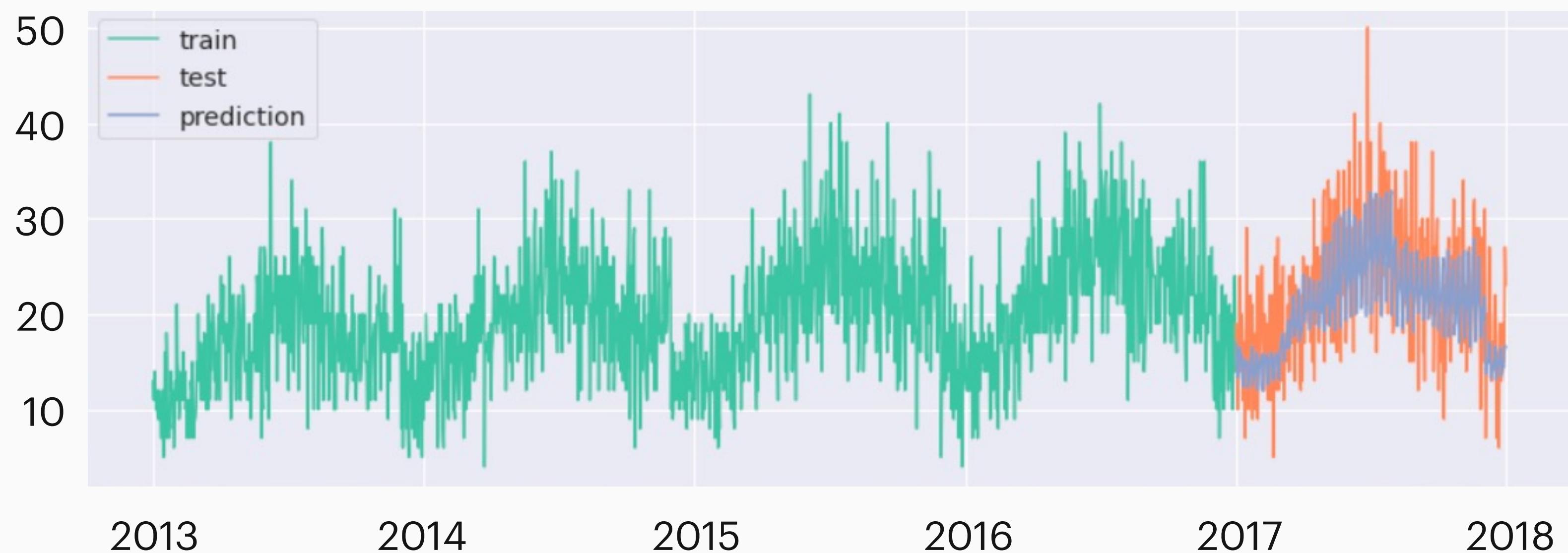
$X$  — признаки ряда;

$t$  — время.



$F$  — ?

Дневные продажи объекта 1 в магазине 1



1. Знаем значение ряда (**зелёные**) до момента времени  $t$ .
2. Хотим предсказать (**синее**) будущие значения ряда (**красные**).

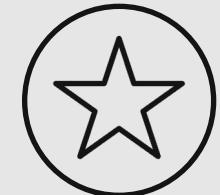
# ML-задача

$$Y_t = F(y_{t-1}, y_{t-2}, \dots, X, t)$$

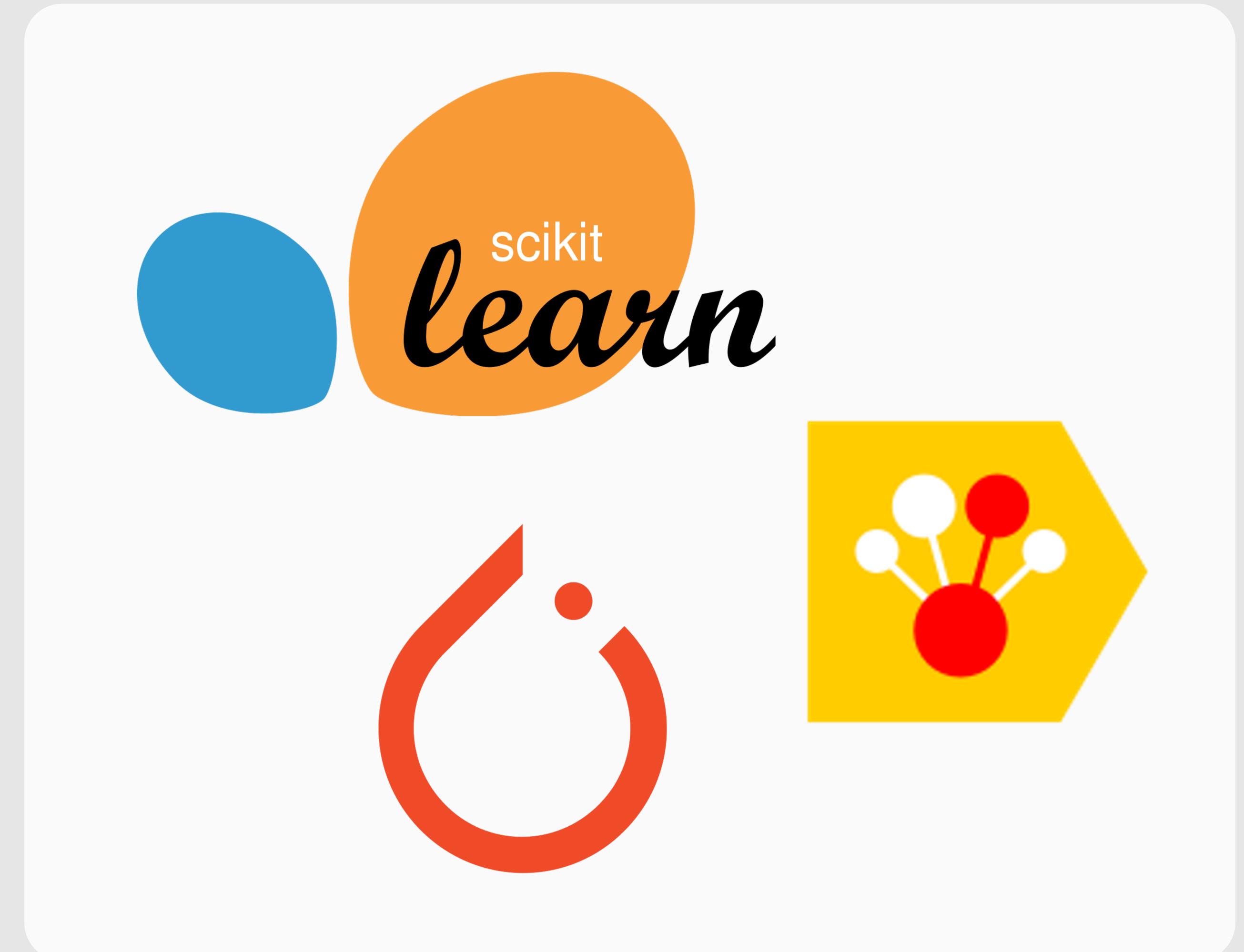
$y_t$  — исторические значения;

$X$  — признаки ряда;

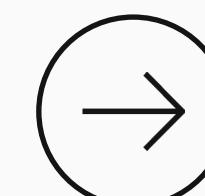
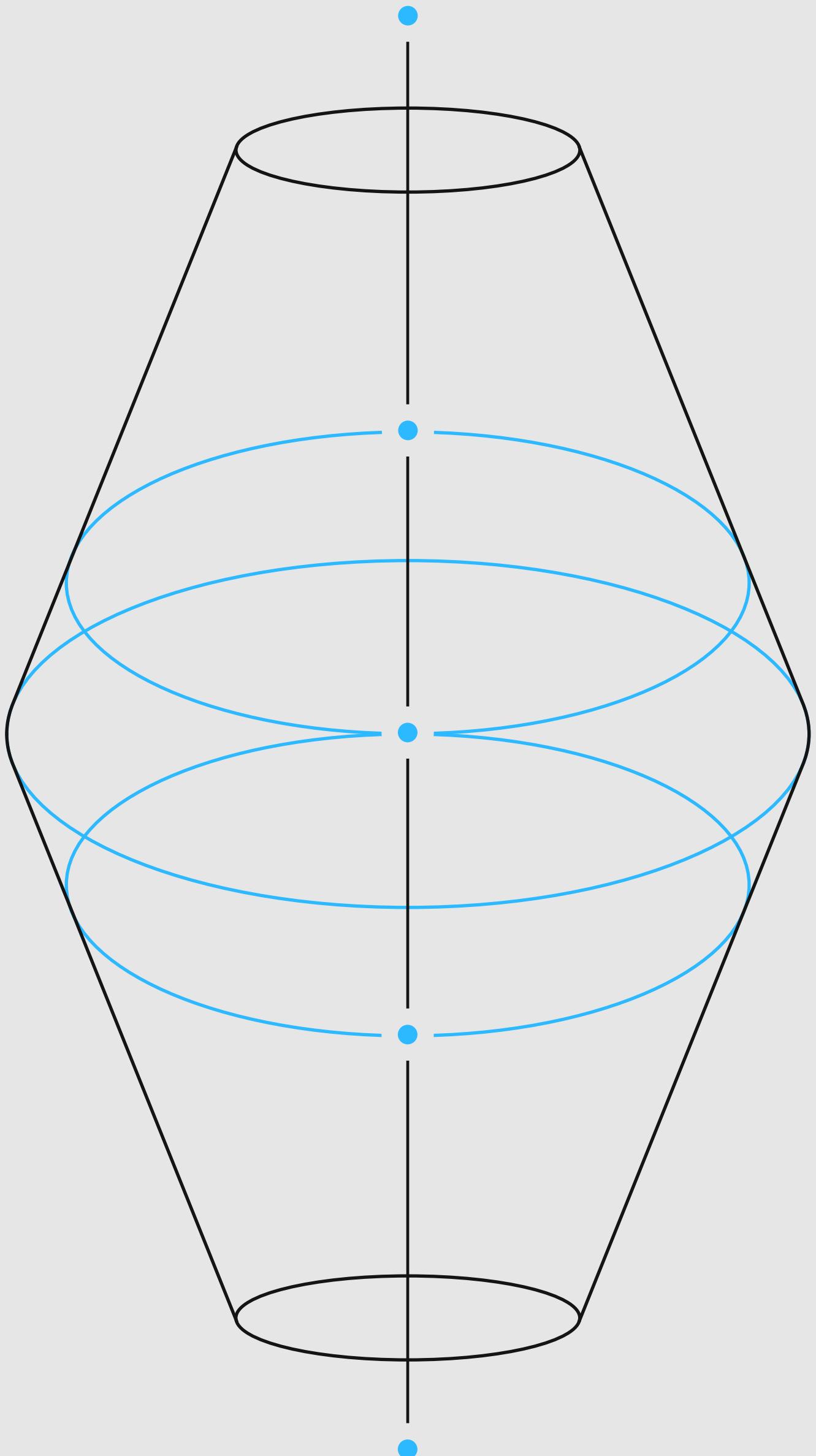
$t$  — время.



Строим функцию  $F$  при помощи ML-модели.

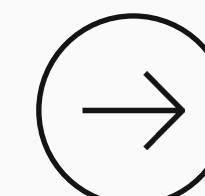


# Стратегии прогнозирования



## Multistep Forecasting

Как прогнозировать на несколько точек вперёд?



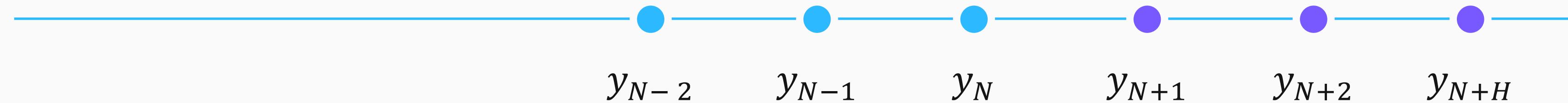
## Panel Forecasting

Как прогнозировать сразу много рядов?

# Как прогнозировать на несколько точек вперёд?

## ПРОБЛЕМА

Модель предсказывает скаляр, а нужно предсказывать вектор =  $(y_{N+1}, y_{N+2}, \dots, y_{N+H})$ .



## РЕШЕНИЕ

Используем одну и ту же модель рекурсивно (aka **рекурсивная стратегия**).

$$\hat{y}_{N+h} = \begin{cases} \hat{f}(y_N, \dots, y_{N-d+1}) & \text{if } h = 1 \\ \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+1}, y_N, \dots, y_{N-d+h}) & \text{if } h \in \{2, \dots, d\} \\ \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+h-d}) & \text{if } h \in \{d+1, \dots, H\} \end{cases}$$

$$\hat{y}_{N+h} = \hat{f}_h(y_N, \dots, y_{N-d+1})$$

# Рекурсивная стратегия

## Обучение

Учим модель предсказывать на шаг вперёд.

## Предсказание

Применяем модель рекурсивно.

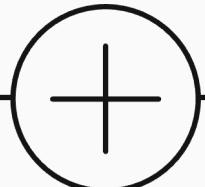
$$y_{t+1} = f(y_t, \dots, y_{t-d+1})$$

Модель 1  
↓  
Модель 1  
↓  
Модель 1

day	lag_5	lag_4	lag_3	lag_2	lag_1
11	5	10	12	12	9
12	10	12	12	9	y_1
13	12	12	9	y_1	y_2

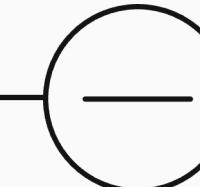
pred
y_1
y_2
y_3

# Рекурсивная стратегия



## Достоинства

- Можно делать прогноз на любой горизонт.
- Требуется обучить одну модель.



## Недостатки

- Происходит накопление ошибок во время прогнозирования.
- Может быть не очень оптимально по скорости инференса.

# Прямая стратегия

## Обучение

Учим  $H$  моделей по модели на горизонт.

## Предсказание

Применяем каждую модель на своих фичах.

$$y_{t+h} = f_h(y_t, \dots, y_{t-d+1})$$

Модель 1

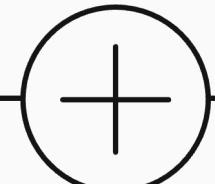
day	lag_5	lag_4	lag_3	lag_2	lag_1
11	5	10	12	12	9
12	10	12	12	9	
13	12	12	9		

Модель 2

pred
y_1
y_2
y_3

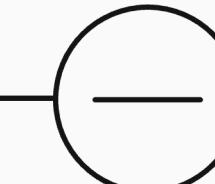
Модель 3

# Прямая стратегия



## Достоинства

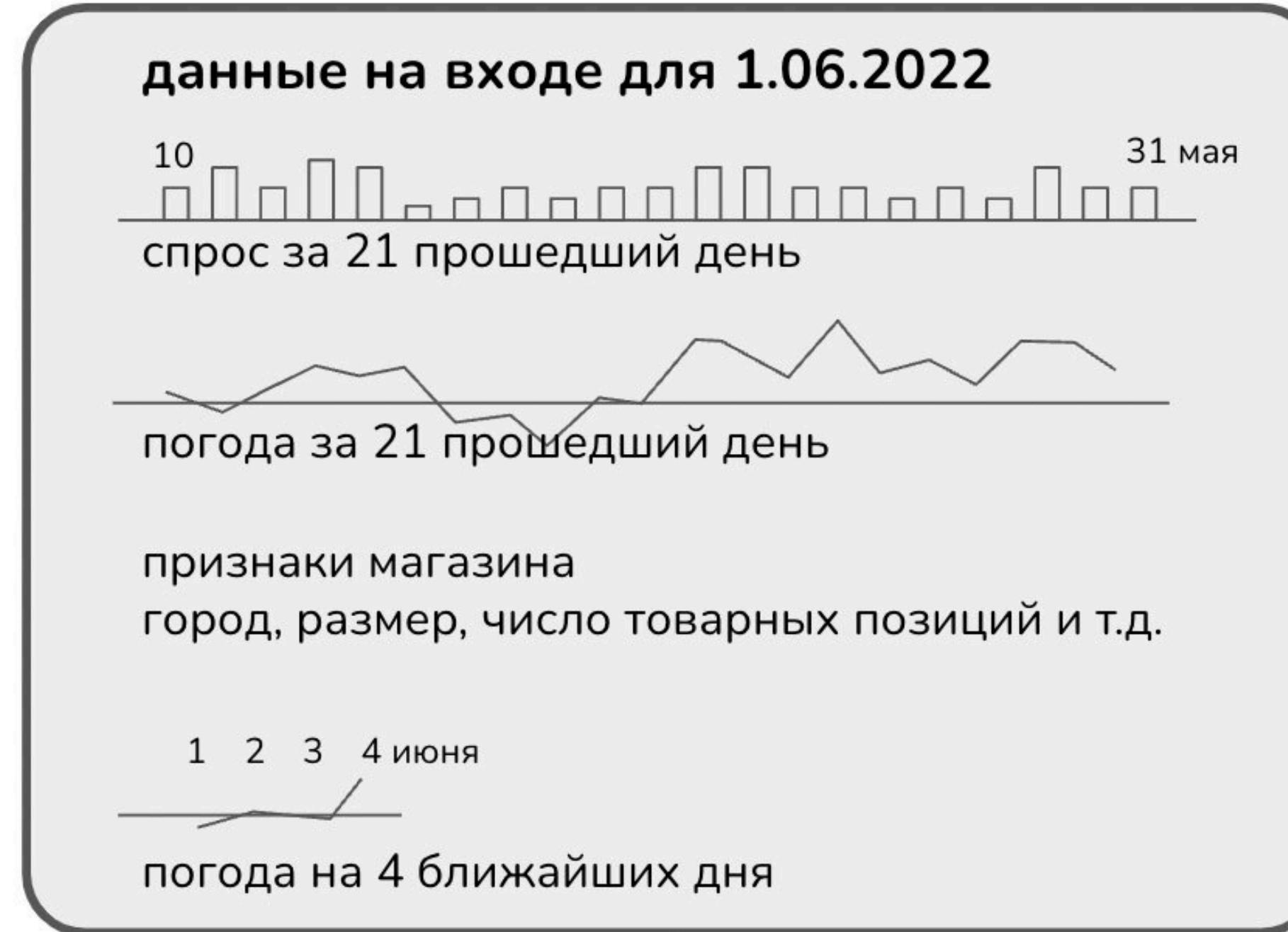
- Нет накопления ошибок во время прогнозирования.
- В целом можно распараллелить.



## Недостатки

- Прогнозы получаются независимо.
- Требуется обучать много моделей.

# Прямая стратегия «СберМаркета»



- Строят 14 моделей для прогнозирования спроса на 14 дней вперёд.
- [Прогнозирование спроса: как «СберМаркет» прогнозирует, что вы закажете вечером в пятницу](#)

# Упрощённая прямая стратегия

## Обучение

Учим одну модель на точку  $H$  горизонта.

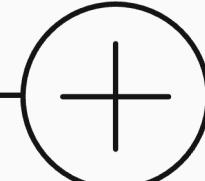
## Предсказание ( $H = 3$ )

Используем одну и ту же модель для каждой точки горизонта.

	<b>day</b>	<b>lag_5</b>	<b>lag_4</b>	<b>lag_3</b>	<b>lag_2</b>	<b>lag_1</b>
Модель 3	11	5	10	12		
Модель 3	12	10	12	12		
Модель 3	13	12	12	9		

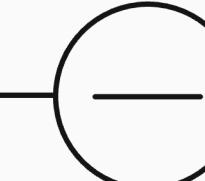
<b>pred</b>
y_1
y_2
y_3

# Упрощённая прямая стратегия



## Достоинства

- Нет накопления ошибок во время прогнозирования.
- Требуется обучить одну модель.



## Недостатки

- Недоиспользование данных.

# Гибридная стратегия

## Обучение

- Учим свою модель на каждую точку горизонта:  $H$  моделей.
- Каждая следующая модель принимает на вход предсказания предыдущих.

## Предсказание ( $H = 3$ )

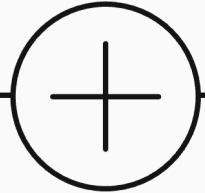
Последовательно получаем предсказания от каждой модели.

Модель 1  
↓  
Модель 2  
↓  
Модель 3

day	lag_5	lag_4	lag_3	lag_2	lag_1	
11	5	10	12	12	9	
12	10	12	12	9	y_1	
13	12	12	9	y_1	y_2	

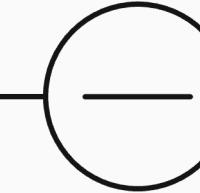
pred
y_1
y_2
y_3

# Гибридная стратегия



## Достоинства

- Нет накопления ошибок во время прогнозирования.
- Выучиваются зависимости между прогнозами.



## Недостатки

- Сложность реализации.
- Требуется обучать много моделей.

# Полезные ссылки



[Стратегии прогнозирования временных рядов в ETNA](#)

A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition

Souhaib Ben Taieb<sup>a</sup>, Gianluca Bontempi<sup>a</sup>, Amir Atiya<sup>c</sup>, Antti Sorjamaa<sup>b</sup>

<sup>a</sup>Machine Learning Group, Département d’Informatique, Faculté des Sciences, Université Libre de Bruxelles, Belgium

<sup>b</sup>Environmental and Industrial Machine Learning Group, Adaptive Informatics Research Centre, Altoo University School of Science, Finland

<sup>c</sup>Faculty of Engineering, Cairo University, Giza, Egypt

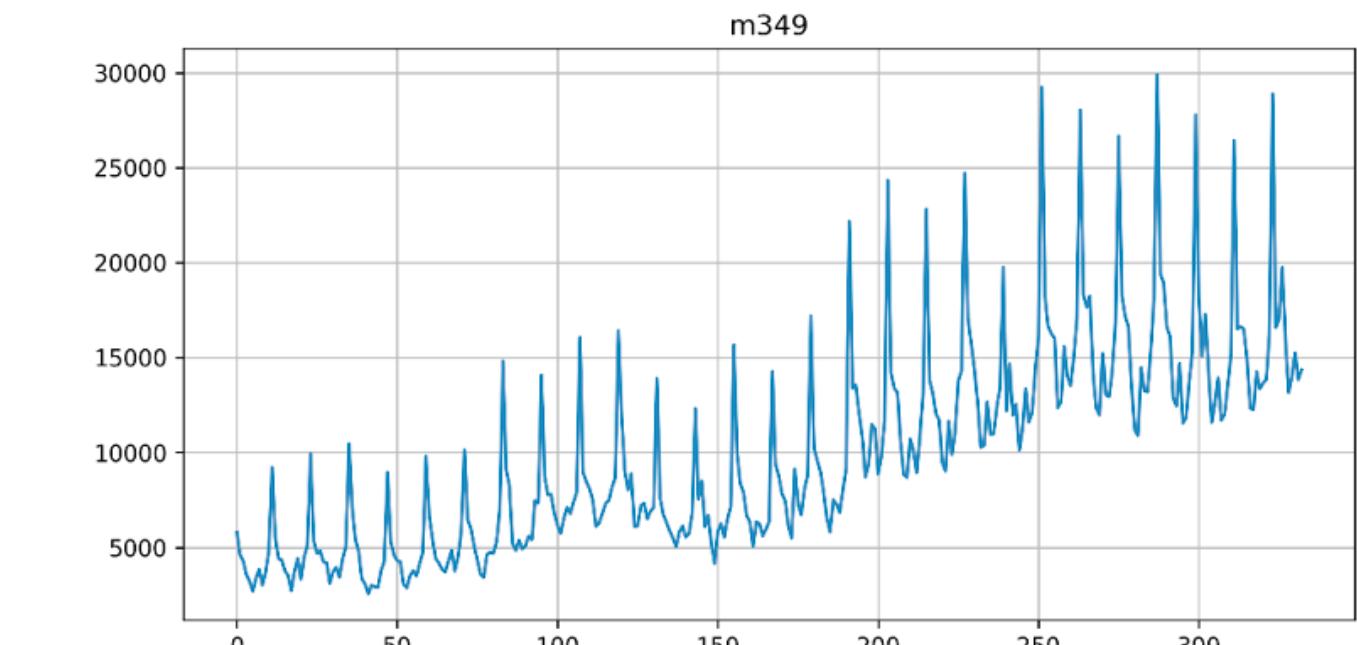
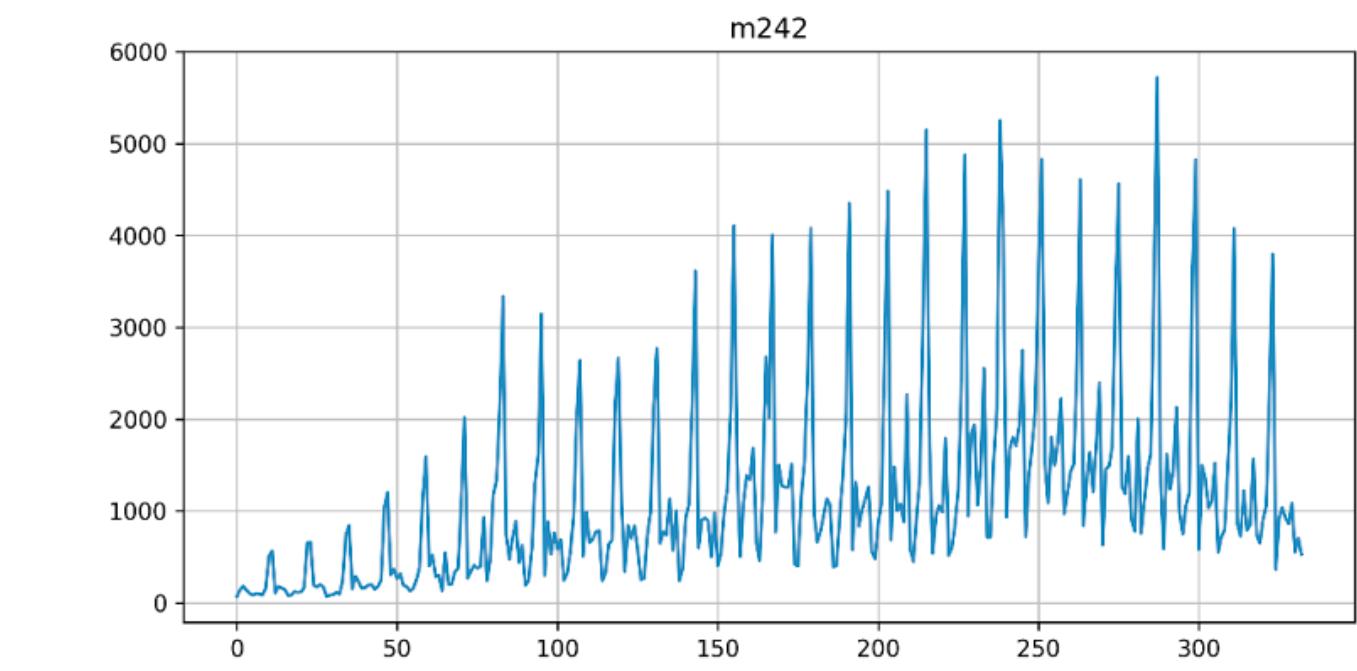
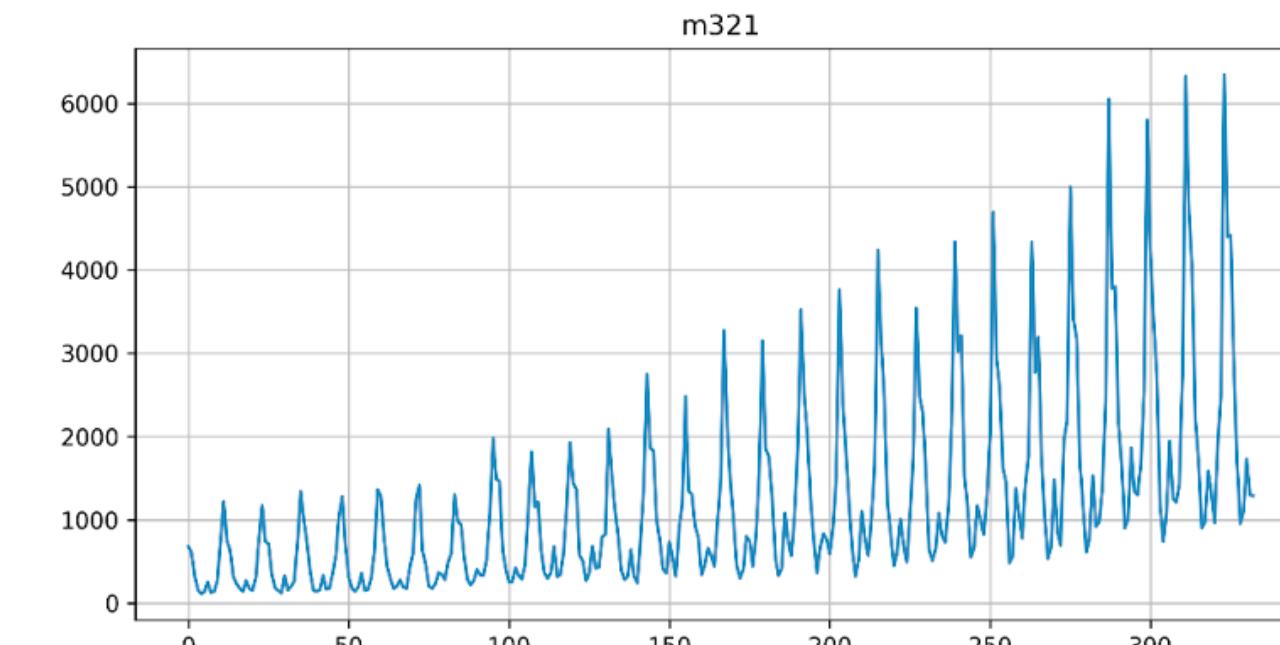
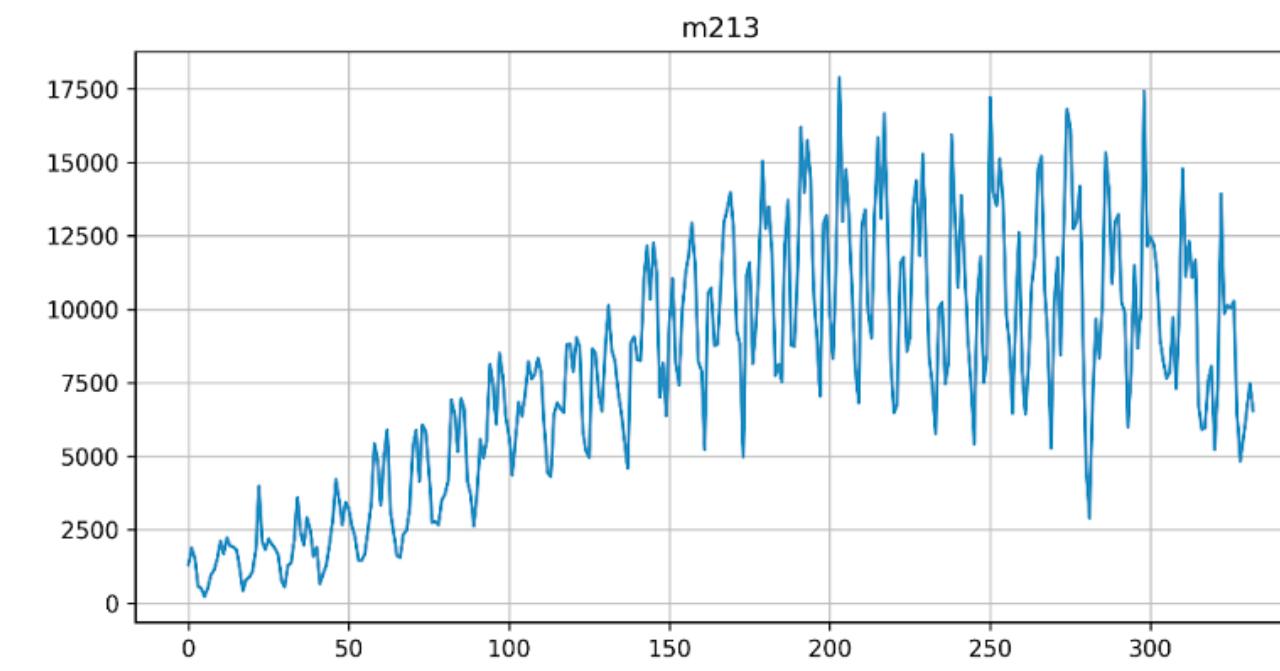
[Статья со сравнением стратегий](#)

# Как прогнозировать сразу много рядов?



## ПРИМЕРЫ

- Прогноз популярности туристических направлений для различных регионов
- Прогноз продаж по различным товарам в магазине



# Как прогнозировать сразу много рядов?



## Локальное моделирование

Отдельная модель — на каждый ряд.

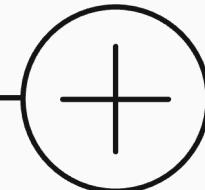
## Глобальное моделирование

- Одна модель — на все временные ряды.
- Модель могла быть обучена как на этих же рядах, так и на других.

## Промежуточный подход

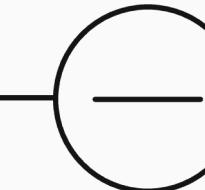
- Одна модель — на группу временных рядов.
- Пример: одна модель — на товары одной категории.

# Локальные модели



## Достоинства

- Можно хорошо обучиться под конкретный домен или ряд.
- Понятно, как параллельно обучать.



## Недостатки

- Медленнее обучаются.
- Сложно использовать в production, если рядов становится много, то и моделей будет много.
- Больше требований к количеству данных и качеству данных.
- Не всегда очевидно, как агрегировать интерпретации моделей.



# Стратегии прогнозирования

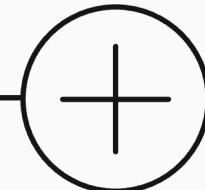
## По точкам

- Рекурсивная
- Прямая
- Гибридная

## По рядам

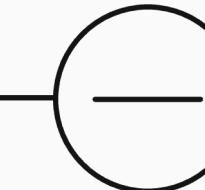
- Локальная
- Глобальная

# Глобальные модели



## Достоинства

- Быстрее обучаются.
- Удобно использовать в production.
- Тolerантны к длине рядов и их качеству.
- Возможный **перенос знаний** между рядами.
- Прогнозирование рядов, которых не было в обучающей выборке, — это **проблема холодного старта**.



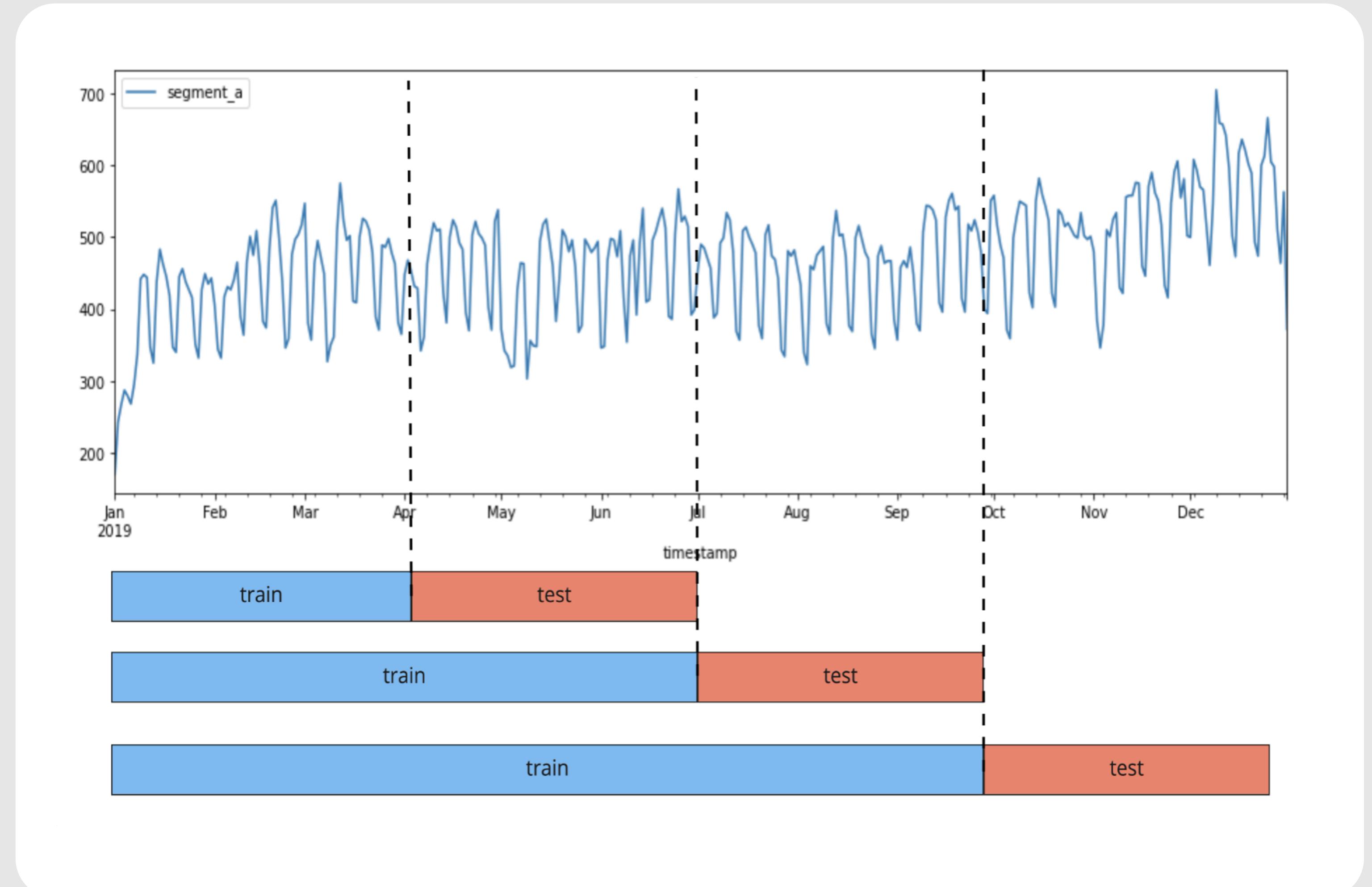
## Недостатки

- Большая вероятность недообучения.
- Ряды могут быть сильно разными по своей природе, из разных доменов, нужен дополнительный анализ.

# Стратегии валидации

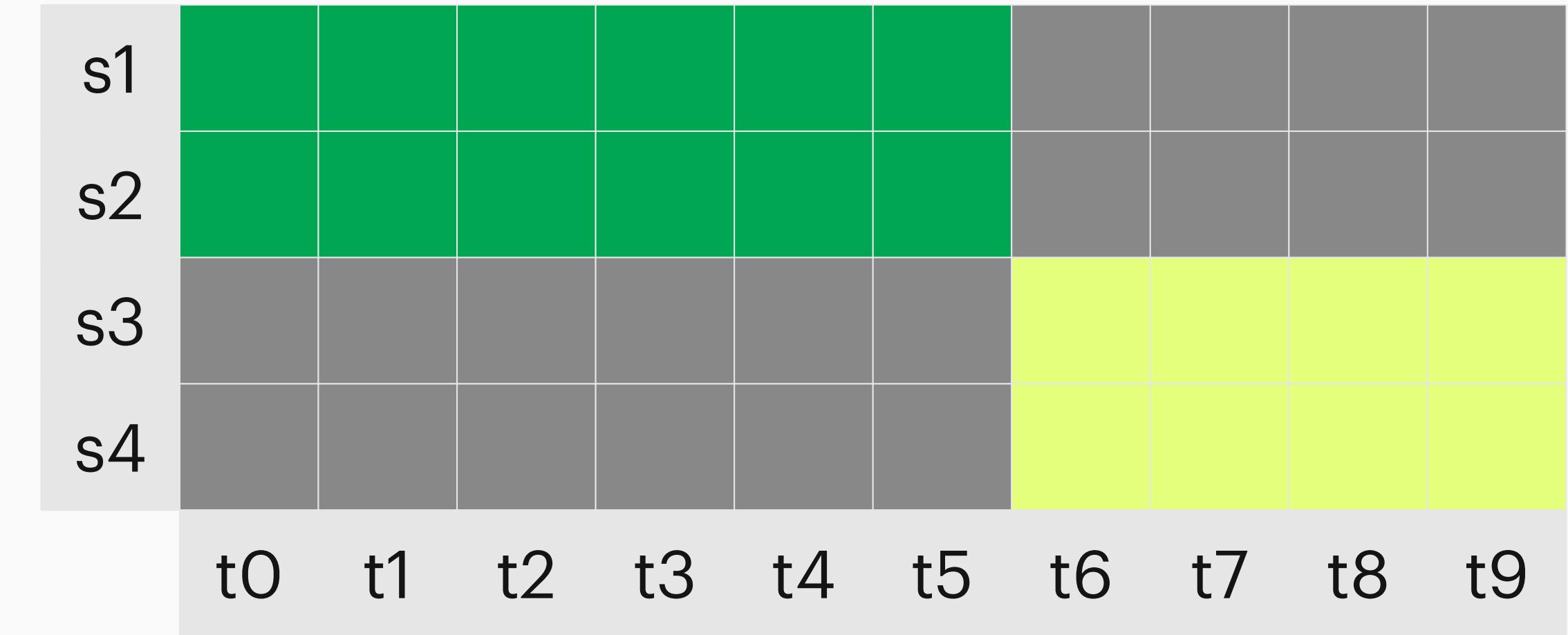
Разделяем фолды по временной  
шкале.

**Не подглядываем в будущее (!).**

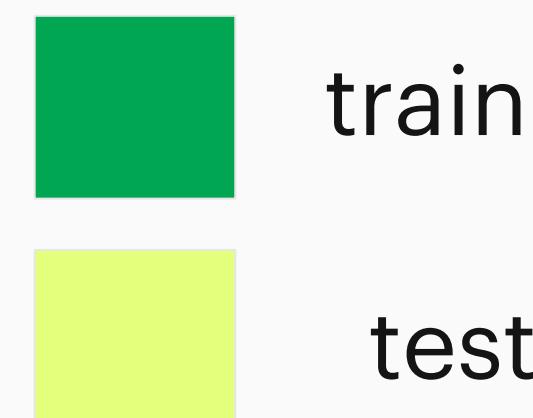


# Стратегии валидации Update

- Оцениваем обобщающую способность глобальных моделей.
- Хотим обучать модель на подвыборке рядов.

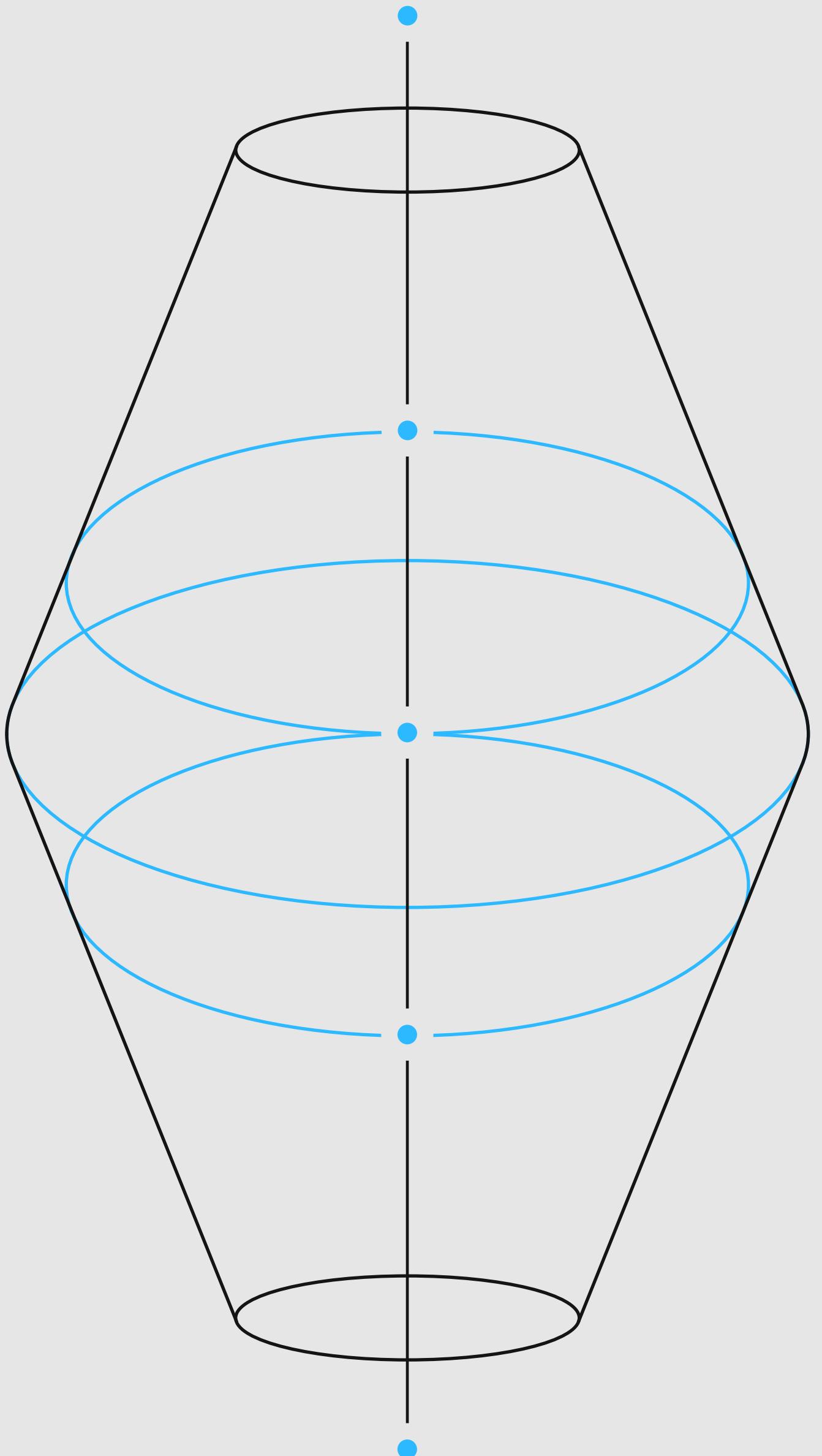


Temporal + Segment



# Генерация признаков

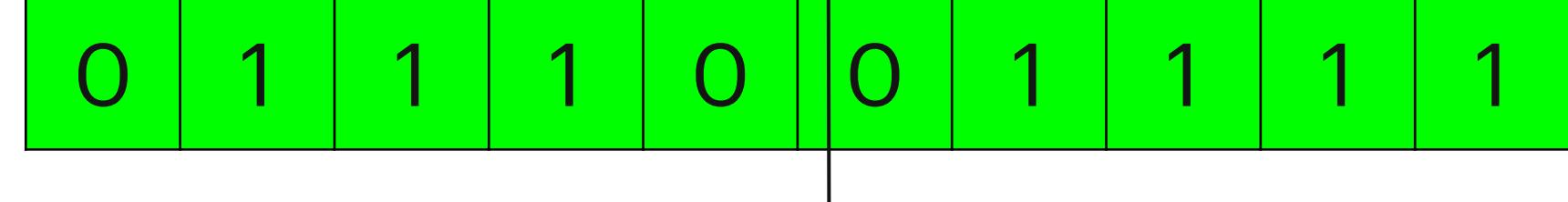
# Обзор признаков



- Признаки из временной метки
- Признаки из истории ряда
- Описательные признаки
- Доменно-специфичные признаки

# Ключевой принцип: не подглядывай в будущее

- Используем только известные данные.
- Например, промо и рекламные блоки покупаются на год вперёд.

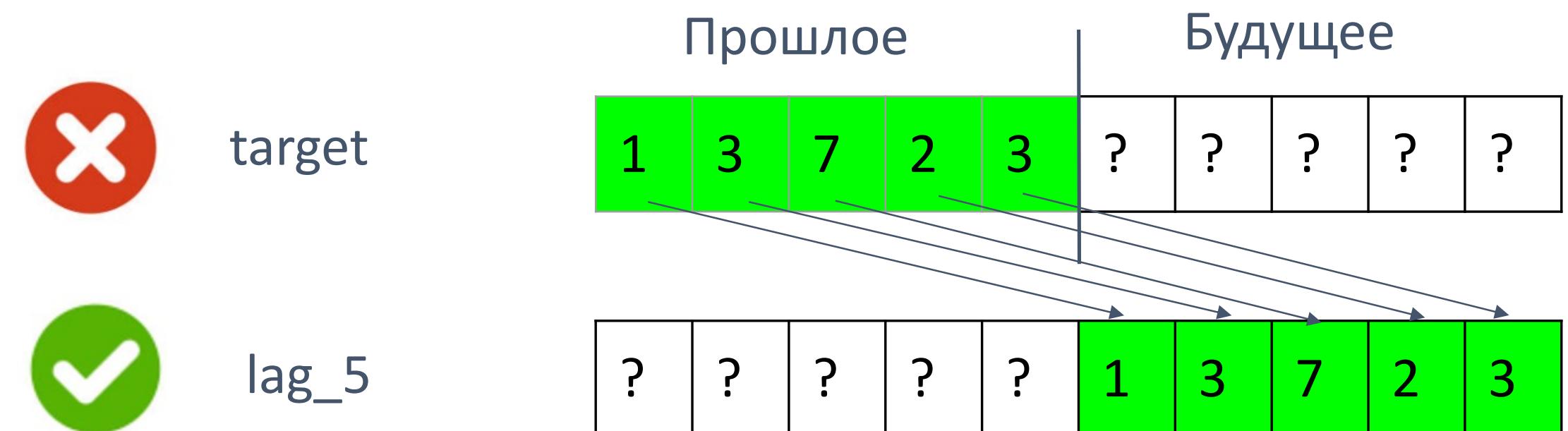


Прошлое

Будущее

# Ключевой принцип: не подглядывай в будущее

- Если не знаем значения в будущем, используем значения из прошлого.
- Например, цены на нефть.



# Ключевой принцип: не подглядывай в будущее

- Если не знаем значения в будущем, используем прогнозы.
- Например, прогноз погоды для продажи мороженого.



Погода



Прогноз  
погоды

Прошлое					Будущее				
1	3	7	2	3	?	?	?	?	?

10	15	20	15	13	10	15	13	16	20
----	----	----	----	----	----	----	----	----	----

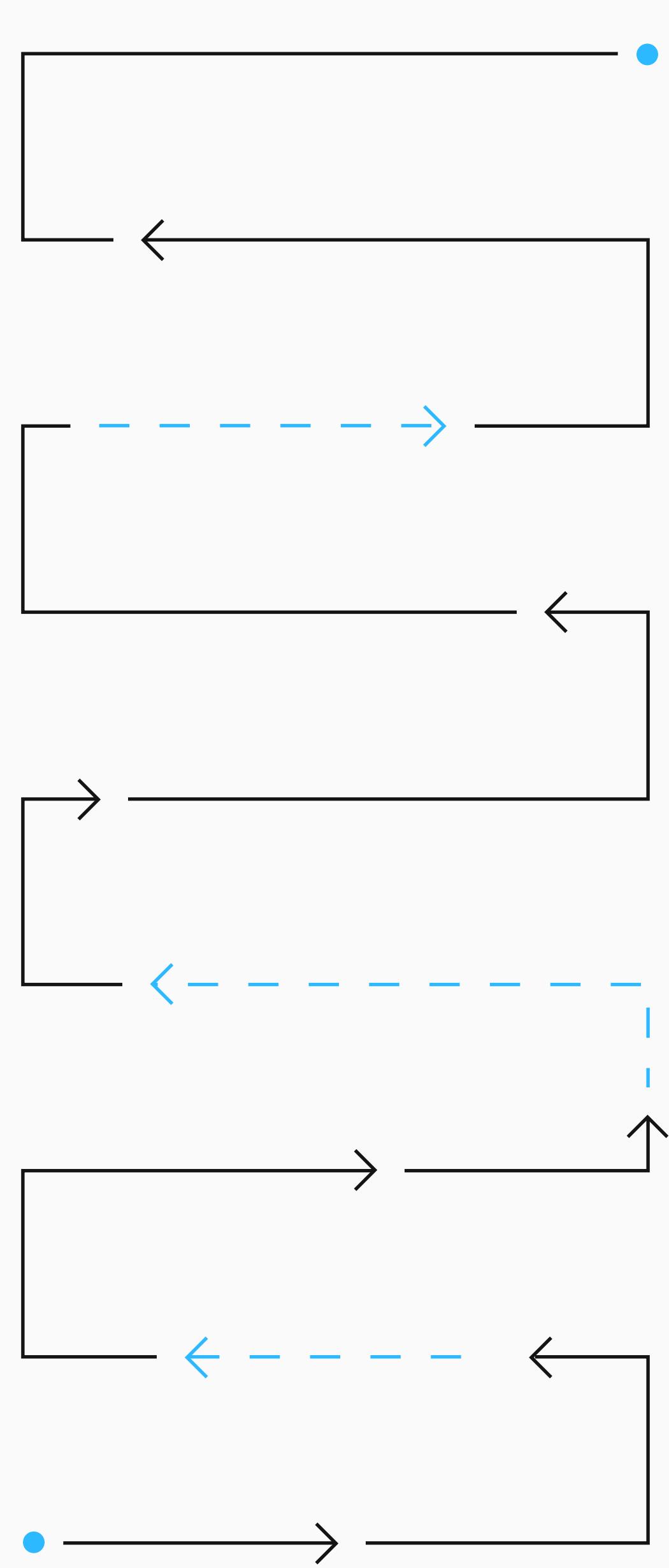


Синоптики

# Ключевой принцип: не подглядывай в будущее



- Известные значения
- Прошлые значения
- Прогнозы, которым доверяем



# Как выглядит датасет для ML

model.fit

model.predict(

Country	Rainfall	Ad spend	$y_{t-3}$	$y_{t-2}$	$y_{t-1}$	$y_t$
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
			21	40	31	?

$\hat{y}_{T+1}$

X (features)

y (target)

# Признаки из временной метки

- **Зачем:** моделируем сезонные паттерны.
- **Пример**
  - Сезонность: недельная.
  - Признак: день недели.

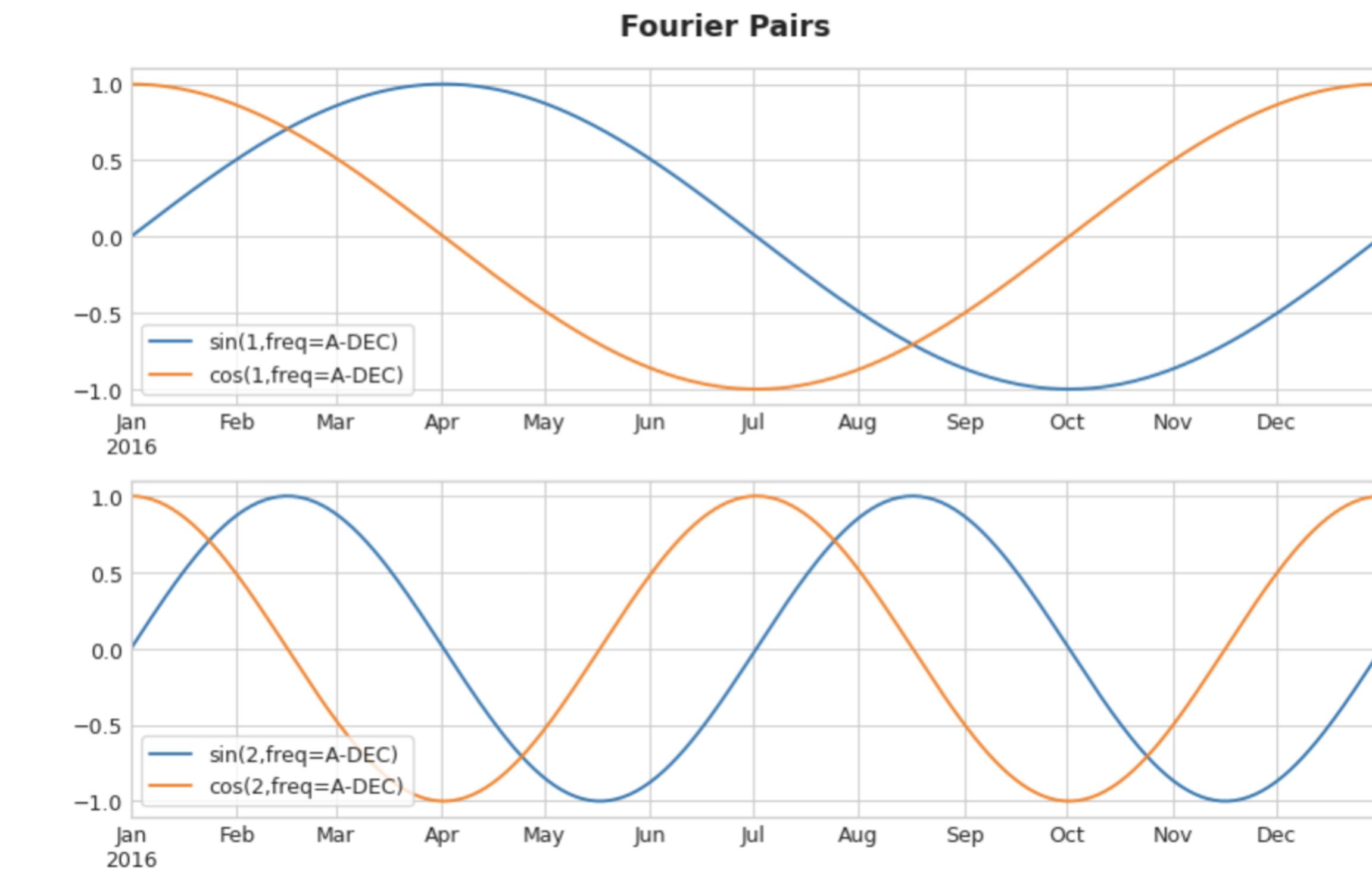
feature	dt_day_number_in_month	dt_day_number_in_week
timestamp		
2020-01-01	1	2
2020-01-02	2	3
2020-01-03	3	4
2020-01-04	4	5
2020-01-05	5	6
...	...	...

**Флаги из даты:** час дня, день недели, номер месяца, ...

# Признаки из временной метки

- **Зачем:** моделируем длинные сезонные паттерны, вам достаточно указать 3–5 гармоник, а не 365 значений, так делают в Prophet.
- **Ряд Фурье**

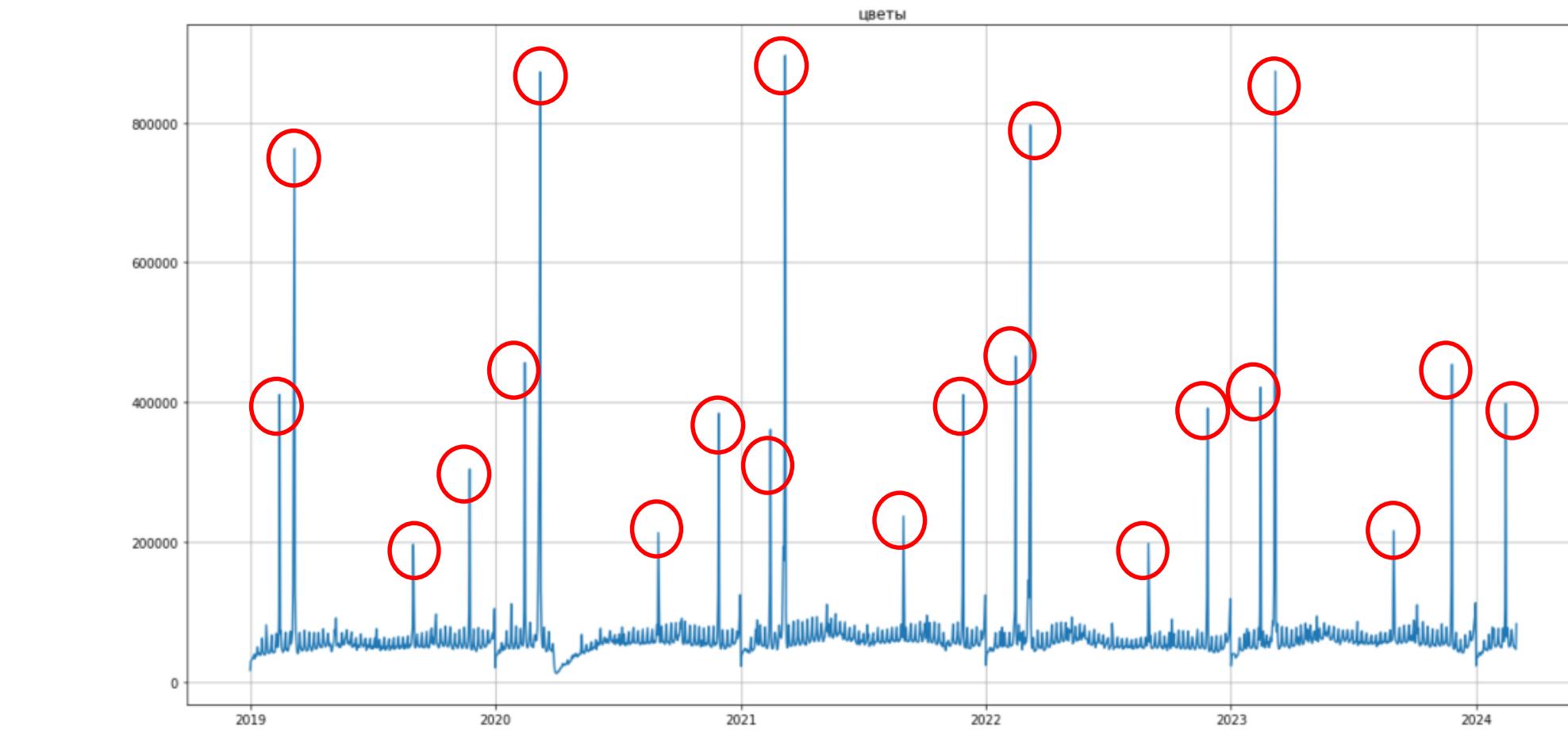
$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx),$$



Годовая vs полугодовая

# Признаки из временной метки

- **Зачем:** моделируем особые даты.
- **Пример**
  - Флаг событий — государственный праздник.
  - Количество дней до праздника или после праздников.



Дата	is_holiday	target
2024-02-13	0	200
2024-02-14	1	1000
2024-02-15	0	150
2024-02-16	0	170

# Признаки ИЗ временной метки

timestamp
0 2020-01-01
1 2020-01-02
2 2020-01-03
3 2020-01-04
4 2020-01-05
...
85 2020-03-26

Флаги из даты: час дня,  
день недели, номер  
месяца, ...

Признаки Фурье

Праздники, особые дни

feature	dt_day_number_in_month	dt_day_number_in_week
timestamp		
2020-01-01	1	2
2020-01-02	2	3
2020-01-03	3	4
2020-01-04	4	5
2020-01-05	5	6
...	...	...

feature	fourier_1	fourier_2	fourier_3	fourier_4	fourier_5	fourier_6
timestamp						
2020-01-01	0.000000	1.000000	0.000000	1.000000	0.000000	1.000000
2020-01-02	0.781831	0.623490	0.974928	-0.222521	0.433884	-0.900969
2020-01-03	0.974928	-0.222521	-0.433884	-0.900969	-0.781831	0.623490
2020-01-04	0.433884	-0.900969	-0.781831	0.623490	0.974928	-0.222521
2020-01-05	-0.433884	-0.900969	0.781831	0.623490	-0.974928	-0.222521
...	...	...	...	...	...	...

feature	holiday
timestamp	
2020-01-01	Новый год
2020-01-02	Новый год
2020-01-03	Новый год
2020-01-04	Новый год
2020-01-05	Новый год
...	...
2020-03-26	NO_HOLIDAY

# Лаги и скользящие статистики

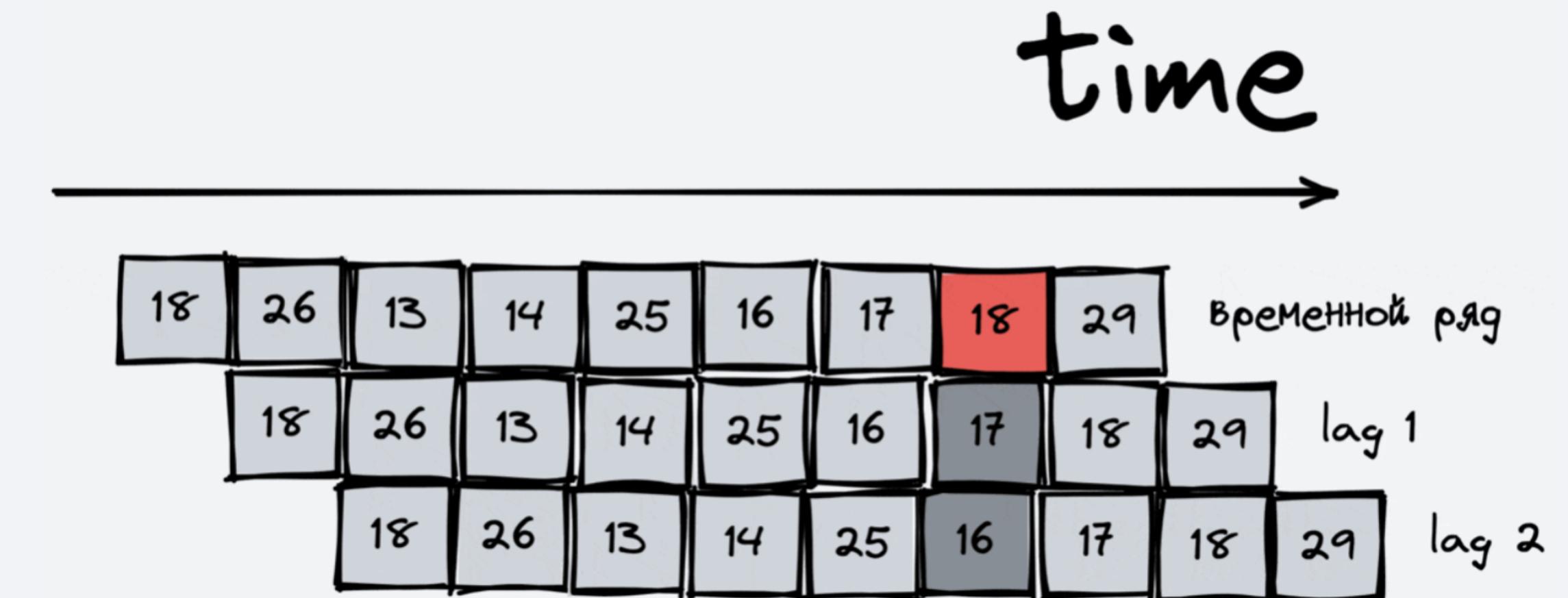
## → Зачем

- моделируем автокорреляционную зависимость;
- сезонные и циклические паттерны;
- зависимость от других переменных, неизвестных на будущее.

## → Важно: используем лаги, известные на весь горизонт прогноза.

Упрощённая прямая + `horizon=14 => lag>=14.`

Рекурсивная + `step=3 => lag>=3.`



# Лаги и скользящие статистики

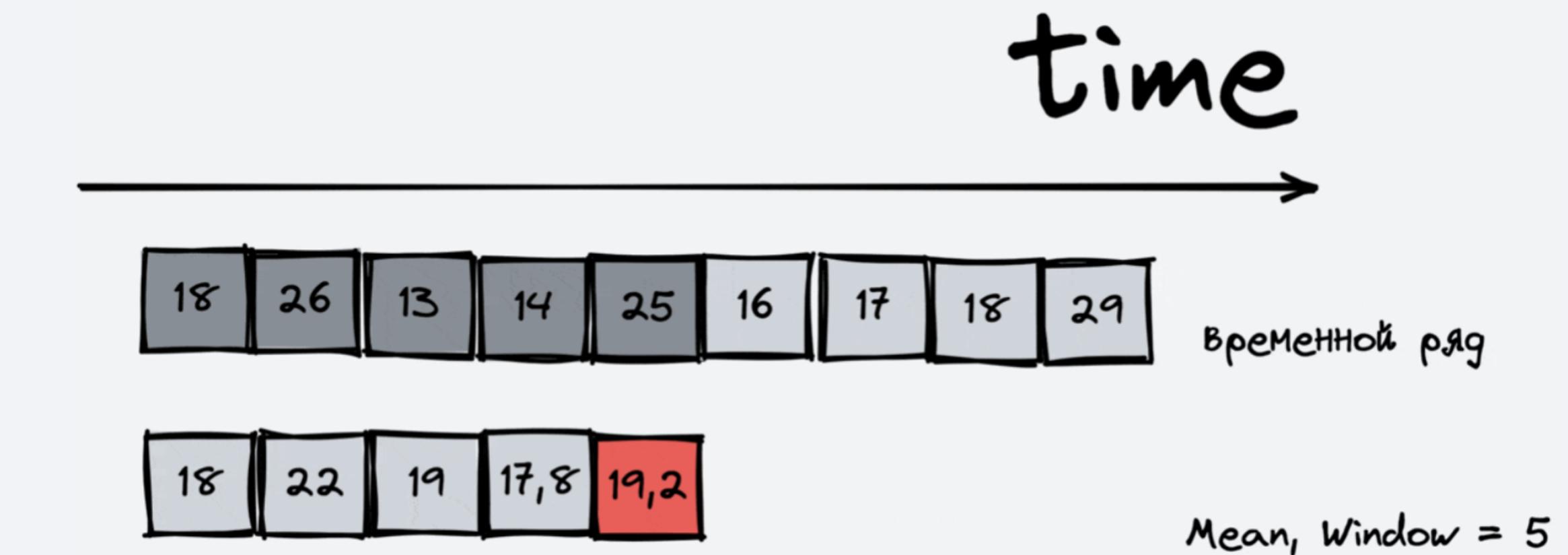
→ **Зачем:** как лаги +

- сглаживаем шум;
- ловим дополнительные паттерны (волатильность, разброс, объёмы, ...).

→ **Формат:**

статистика + окно + сезонность.

→ N. В. Принцип генерации:  
~~statistic(target)~~ → statistic(lag(target)).

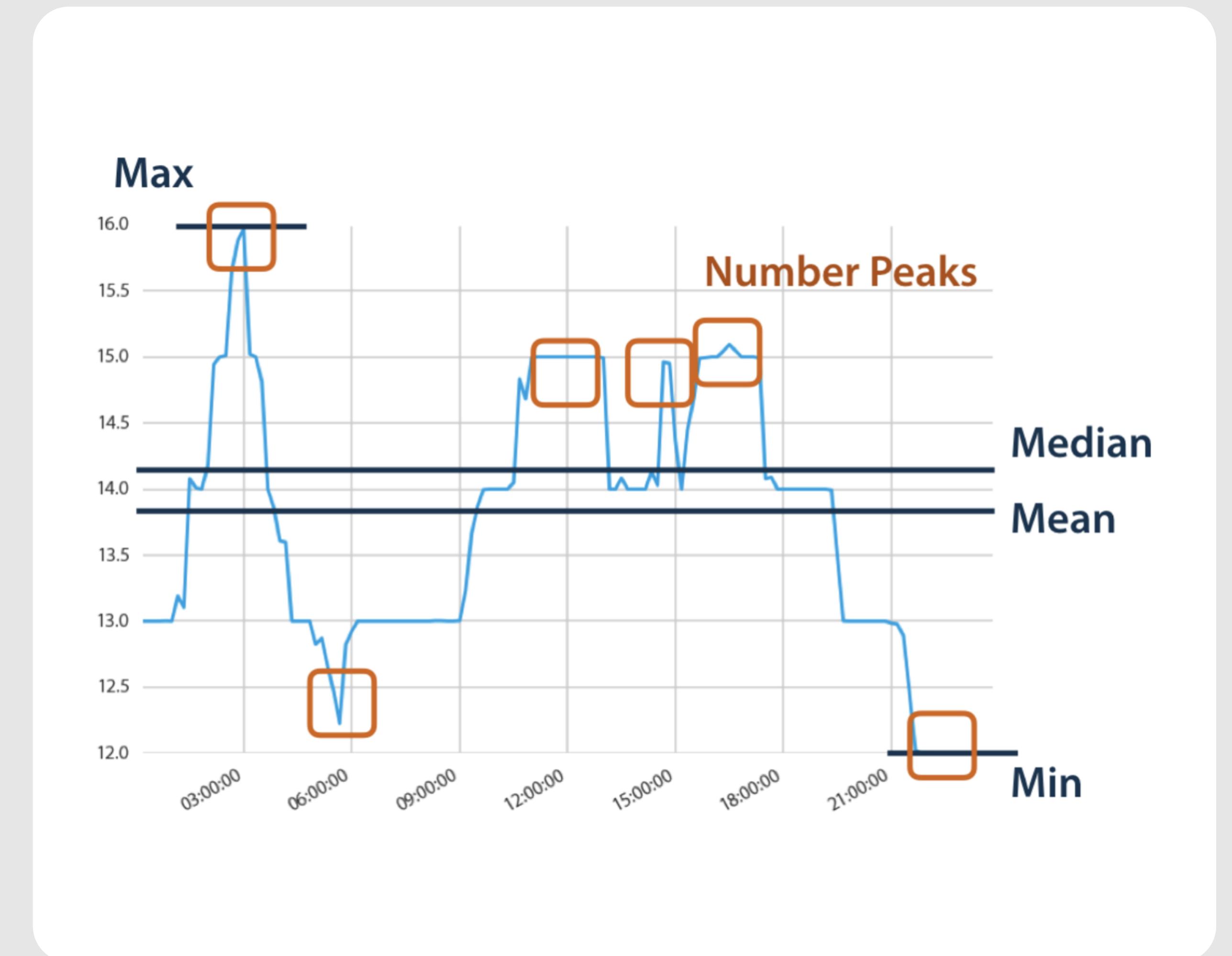


Скользящее среднее

- Статистика: среднее
- Окно: 5
- Сезонность: 1

# Описательные характеристики ряда

- **Зачем:** делаем признаковое описание ряда (не каждой точки), отражающее его характер:
  - трендовость,
  - сезонность,
  - волатильность,
  - ...
- **Когда:** при работе с глобальными моделями.



# Описательные характеристики ряда



Самая популярная библиотека  
генерации ~ 700 признаков



Генерация признаков на polars



*catch22: CAnonical Time-series CCharacteristics  
selected through highly comparative time-series analysis*

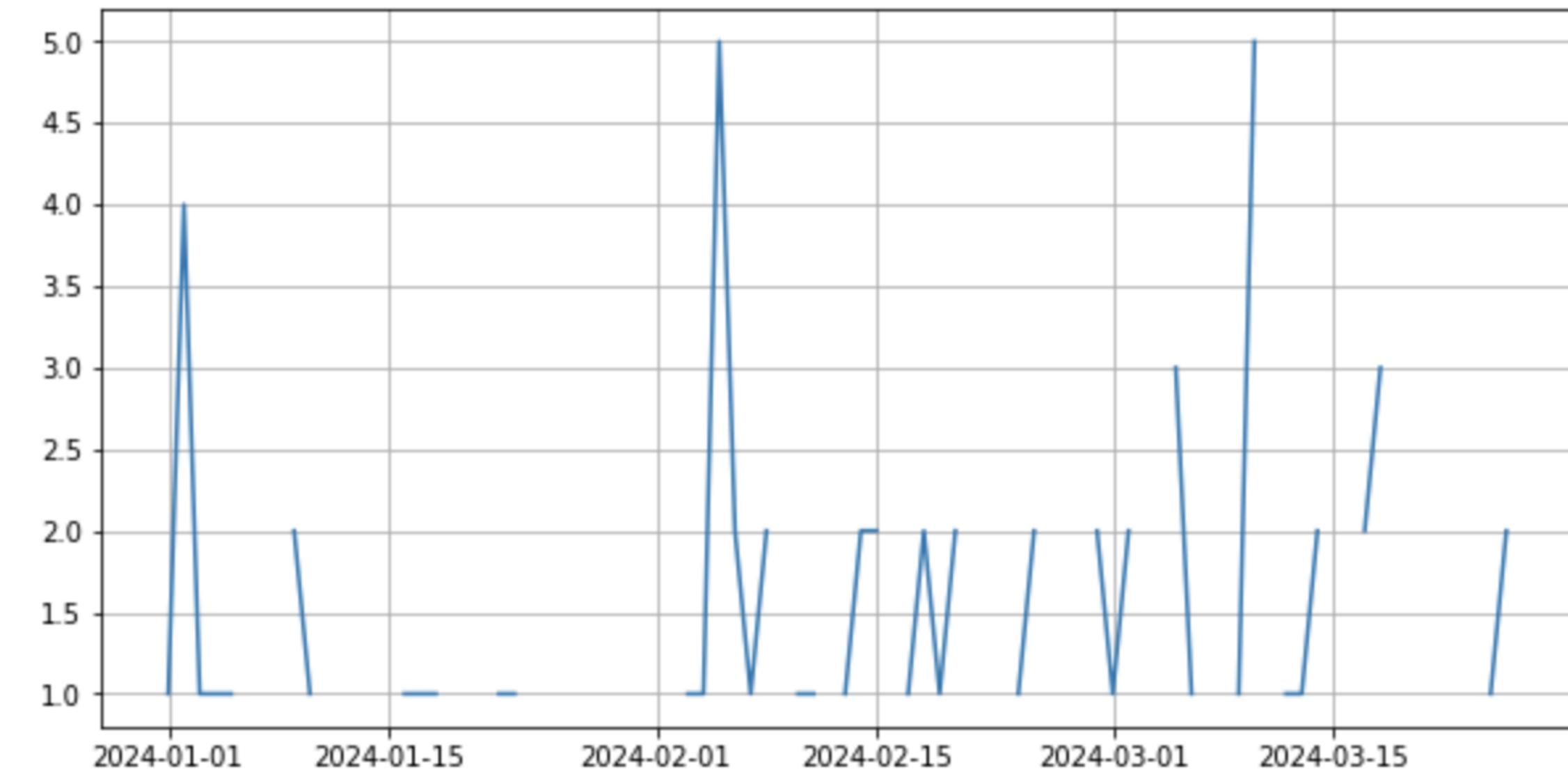
Carl H Lubba · Sarab S Sethi · Philip  
Knaute · Simon R Schultz · Ben D  
Fulcher\* · Nick S Jones\*

Набор 22 наиболее универсальных  
признаков

Обучение  
представлений

# Доменно-специфичные признаки

- **Идея:** привносим знания из домена, не нарушая принципов time series:
  - длину ряда,
  - дни с последнего заказа,
  - среднюю длину промежутков без заказов.
- **Важно:** эффективность и набор зависят от задачи.



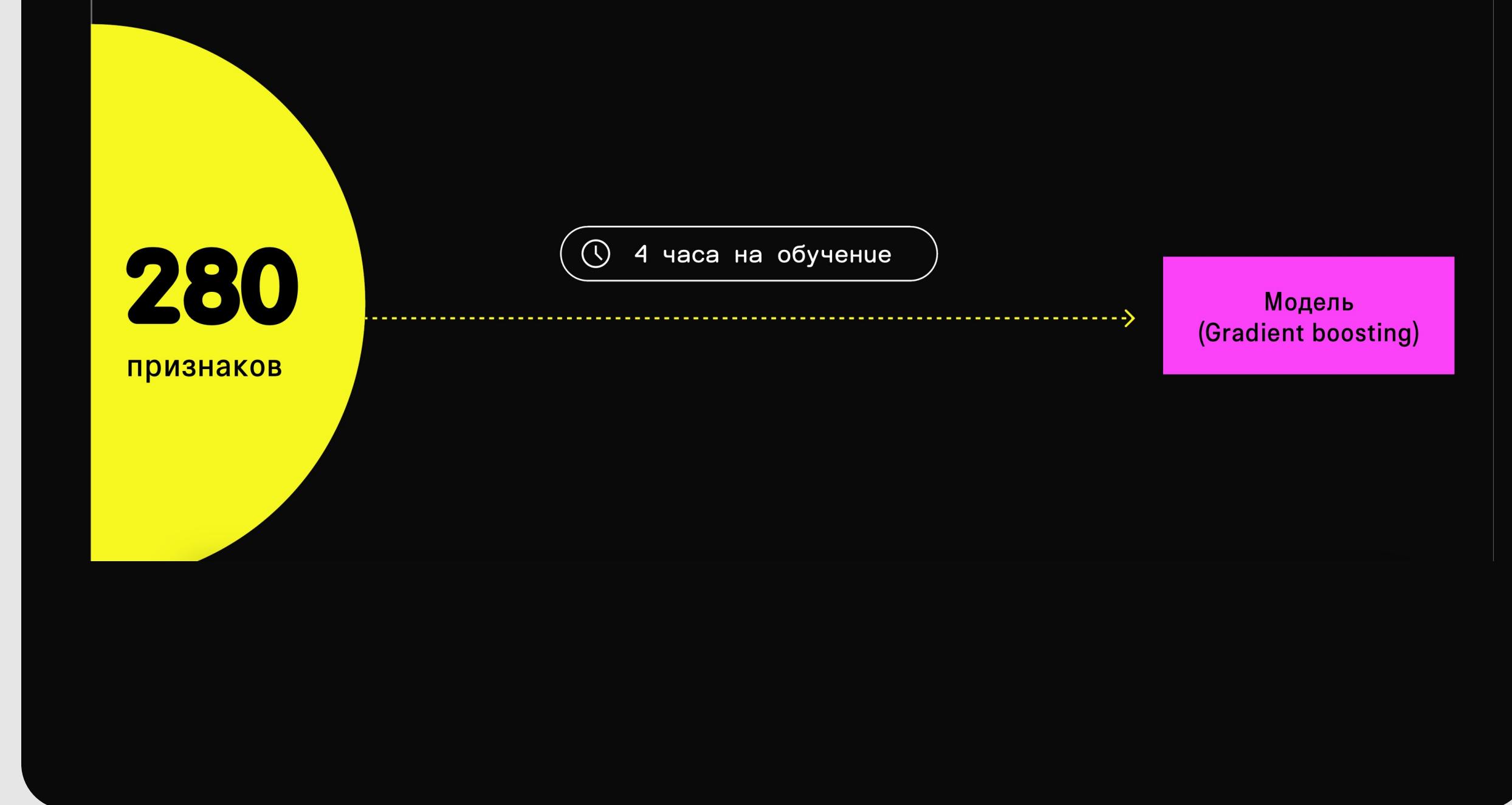
# Доменно-специфичные признаки

Про тонкости прогнозирования спроса  
с примерами полезных признаков

[Прогнозируем спрос на разные виды  
товаров: от комбо до новинок](#)

Отдельные признаки для каждой модели

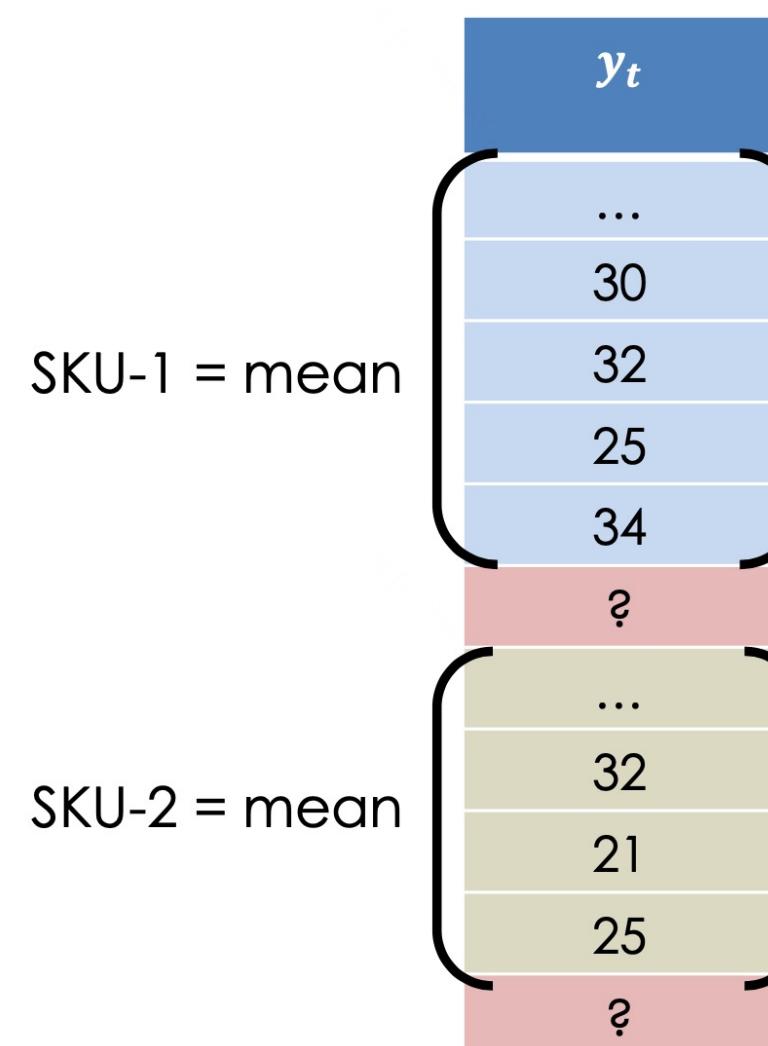
Долгое обучение моделей при полном списке фичей



# Кодирование категориальных признаков

Time	Product ID
...	...
2020-02-13	SKU-1
2020-02-14	SKU-1
2020-02-15	SKU-1
2020-02-16	SKU-1
2020-02-17	SKU-1
...	...
2020-02-14	SKU-2
2020-02-15	SKU-2
2020-02-16	SKU-2
2020-02-17	SKU-2

Как делать **не надо!**



Классический Mean Target Encoding

- (+) Нет лика из train в test.
- (-) Есть лик в рамках точек в train.

Хотим получить числовой признак.

# Кодирование категориальных признаков

Time	Product ID
...	...
2020-02-13	SKU-1
2020-02-14	SKU-1
2020-02-15	SKU-1
2020-02-16	SKU-1
2020-02-17	SKU-1
...	...
2020-02-14	SKU-2
2020-02-15	SKU-2
2020-02-16	SKU-2
2020-02-17	SKU-2

Как делать **не надо!**  
Кодируем скользящим средним.



Хотим получить числовой признак.

# Работа с пропущенными значениями

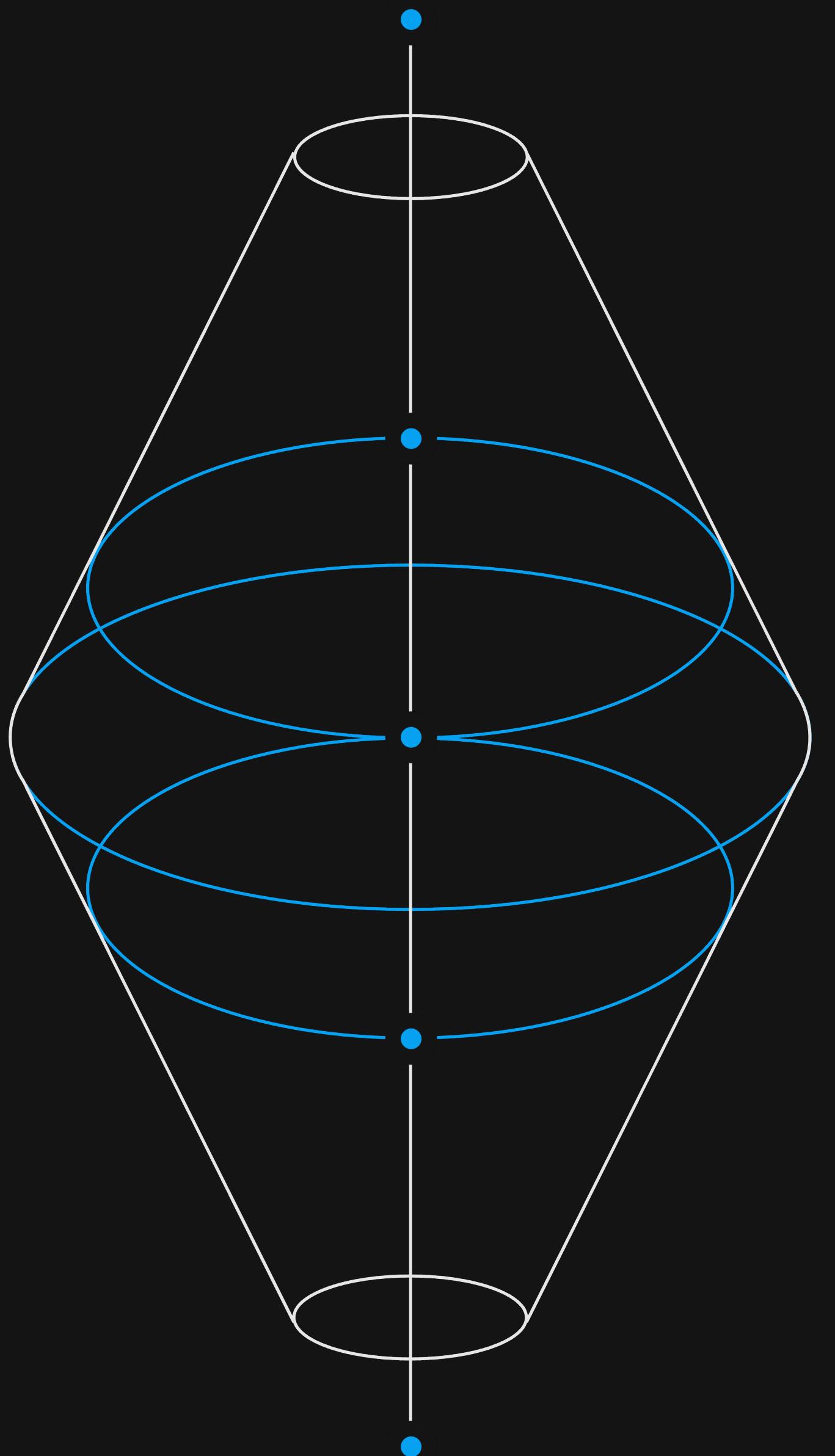
## Варианты замены

1. Константа.
2. Последнее имеющееся значение перед пропуском.
3. Глобальное/скользящее, среднее/медиана.
4. Предсказания модели.

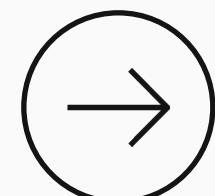
## Как выбрать подход

1. Подход **не** использует **данные из будущего!**
2. По качеству даунстрим-задачи (качество прогноза).
3. По картинке.
4. По логике процесса (например, ночью пропуски — это 0).



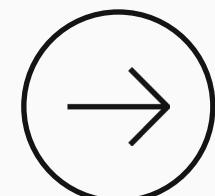


# Выводы



## Не подглядываем в будущее

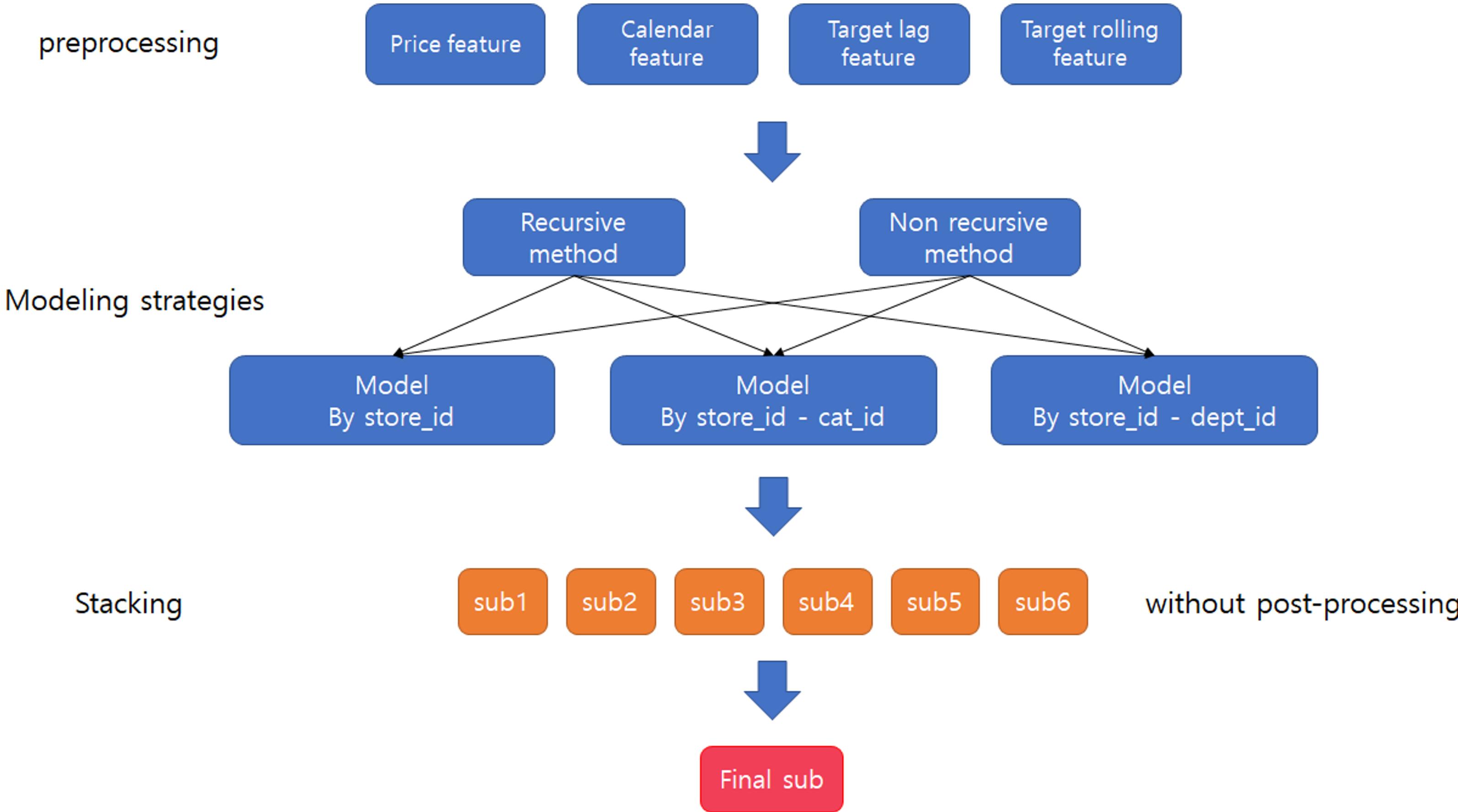
- Известные данные
- Прошлые значения, лаги
- Прогнозы



## Основные группы признаков

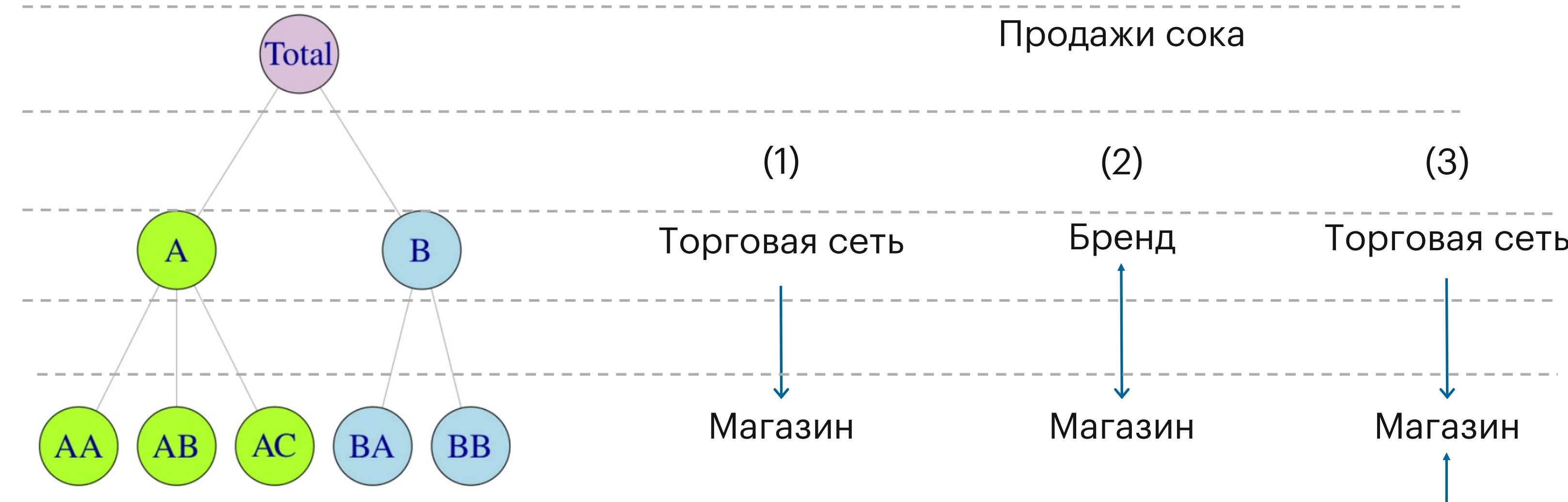
- Признаки на временной метке
- Признаки из прошлых значений переменных
- Описательные признаки ряда
- Доменно-специфичные признаки

# Соревнование M5



# Как использовать иерархическую структуру

# Иерархическая структура



$$y_t = y_{AA,t} + y_{AB,t} + y_{AC,t} + y_{BA,t} + y_{BB,t},$$

$$y_{A,t} = y_{AA,t} + y_{AB,t} + y_{AC,t} \quad \text{and} \quad y_{B,t} = y_{BA,t} + y_{BB,t}.$$

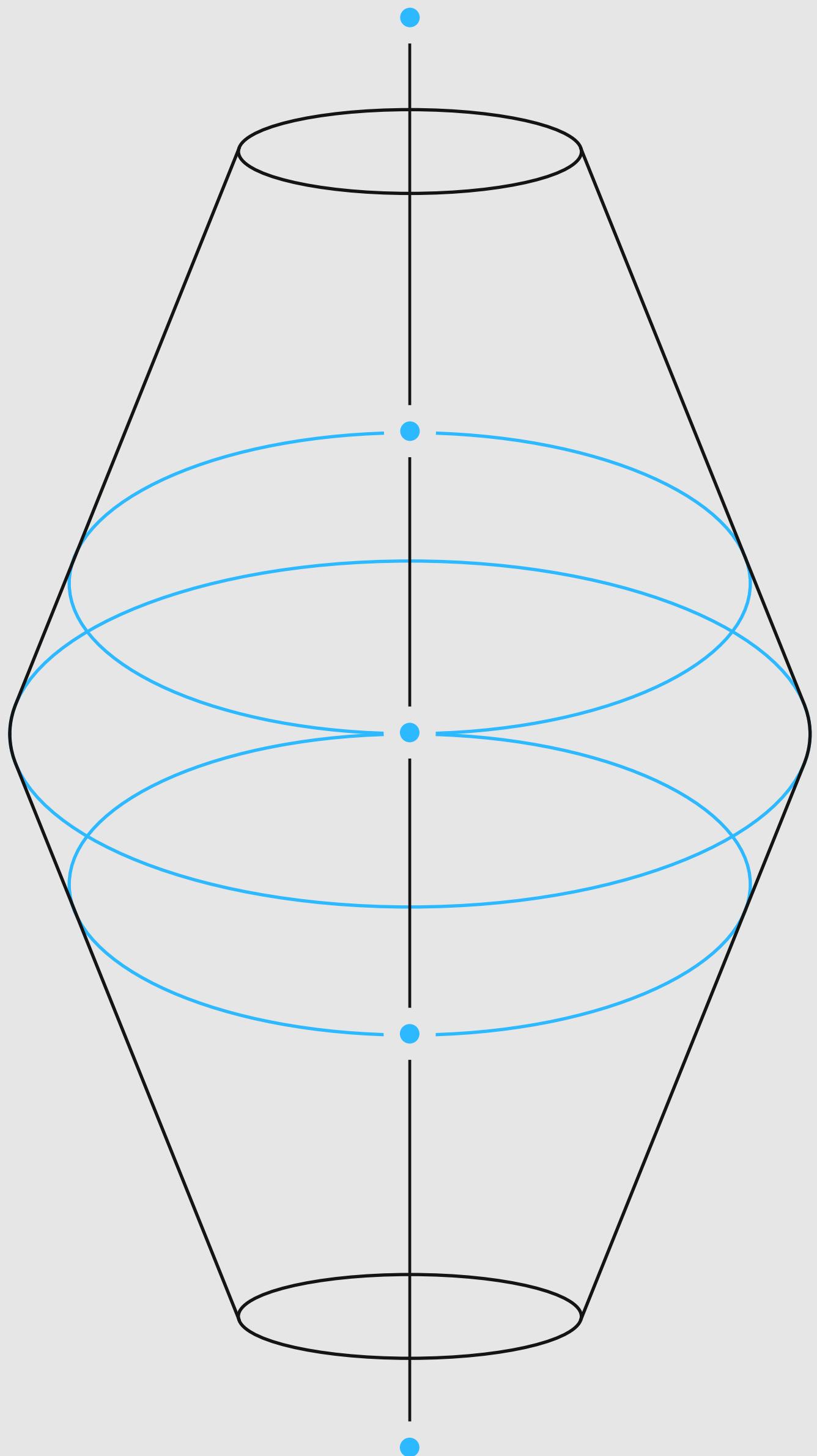
Иерархическая

Групповая

Смешанная

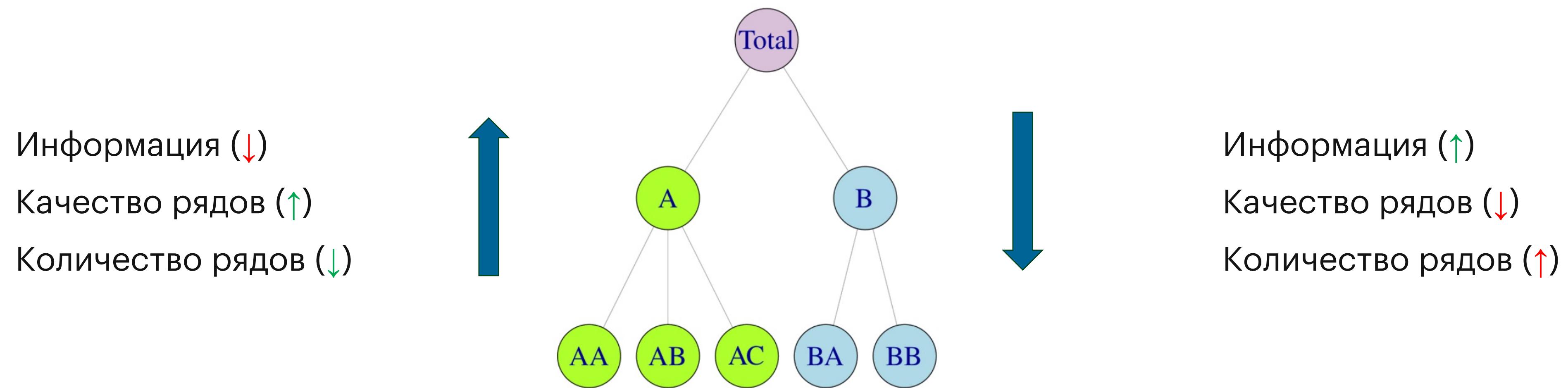
# Постановка задачи

Хотим получить прогноз на уровне  $Y$



- Уровень  $Y$  — в середине иерархии → хотим учесть особенности рядов ниже.
- Уровень  $Y$  очень шумный → хотим прогнозировать ряды выше по уровню и декомпозировать до  $Y$ .
- Хотим использовать информацию со всех уровней.

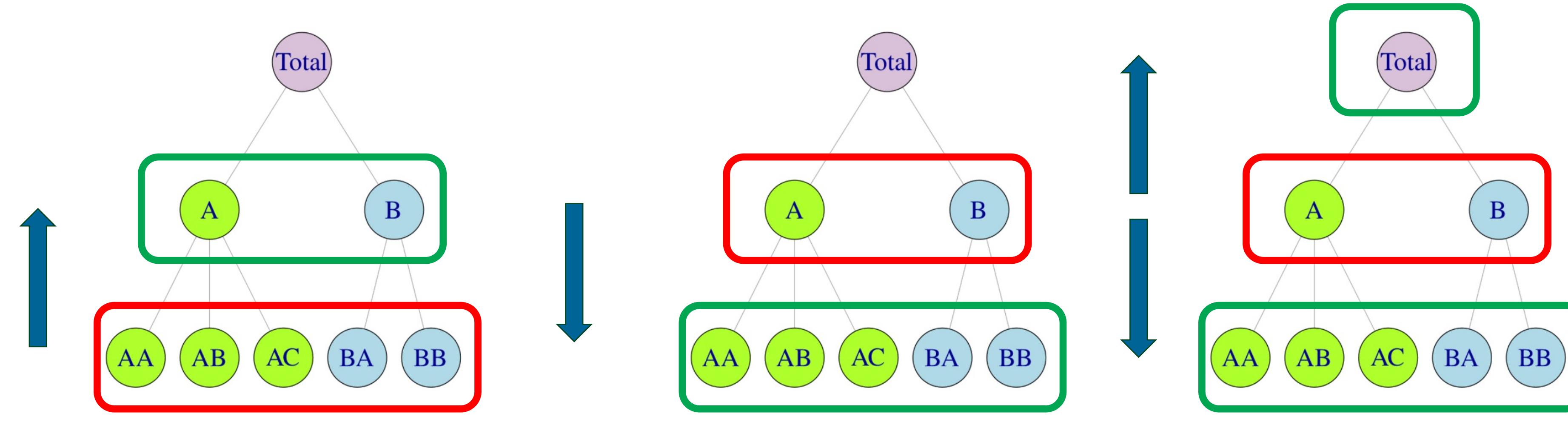
# Основные идеи



**Согласованность** — прогнозы на уровне  $n-1$  должны суммироваться в прогнозы на уровне  $n$  для всех  $n$ .

**Информация** — индивидуальные особенности.  
**Качество рядов** — прогнозируемость.  
**Количество рядов** — вычислительные мощности.

# Базовые подходы



Bottom-up

Top-down

Middle-out

Основная идея: базовый уровень содержит максимум полезной информации.

- Тут прогнозируем (базовый уровень).
- Тут хотим получить прогноз (таргет. уровень).

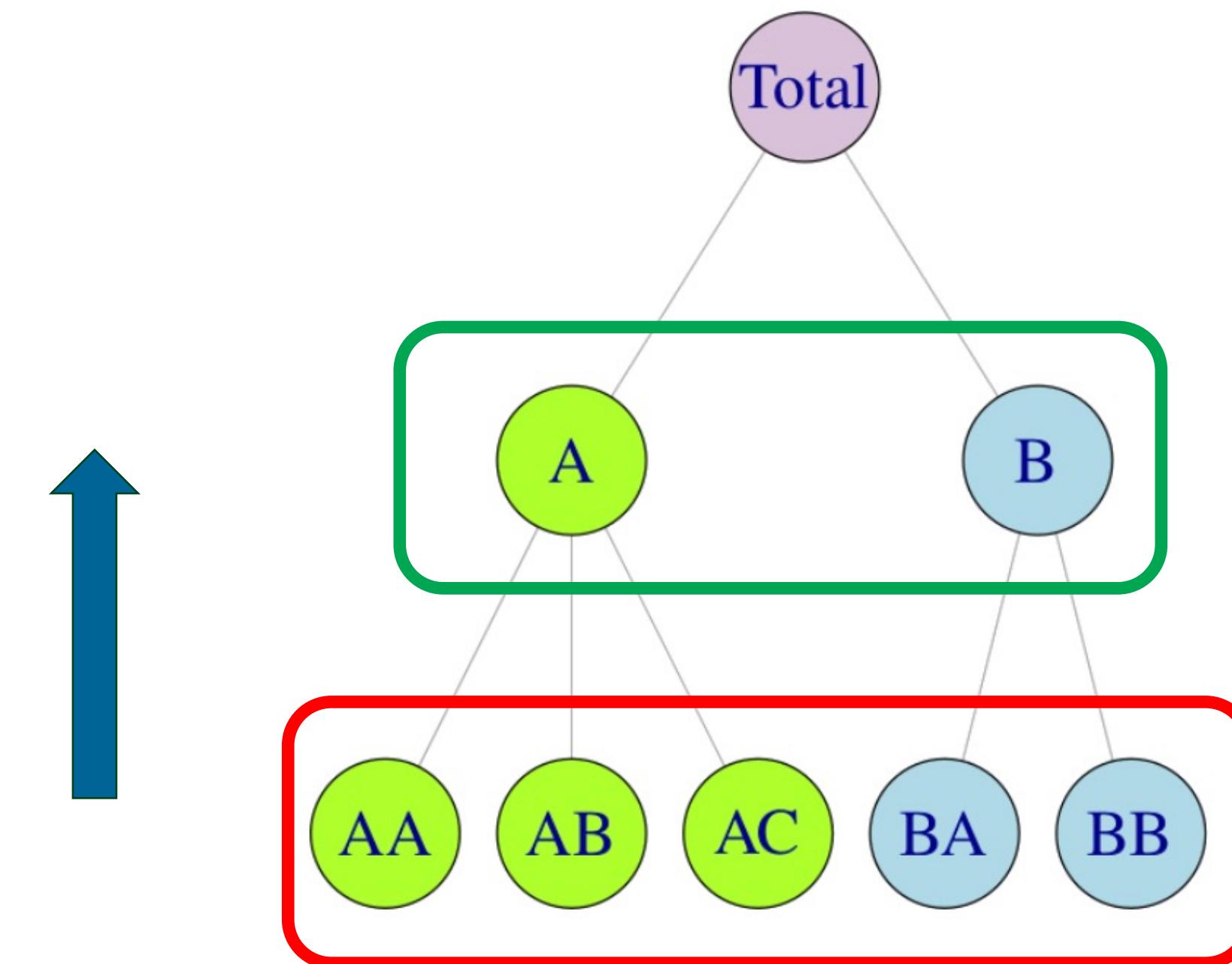
# Bottom-up

## → Свойства

- Автоматическая согласованность.
- Не теряем информацию.
- Шумные ряды.

→ Кейс: хочу прогнозы продаж велосипедов по городам для планирования объёмов выпуска фабрики. В прогнозе учитываем особенности рядов для каждого магазина в городе.

Иерархия: Город → Магазин → Тип велосипеда.



Bottom-up

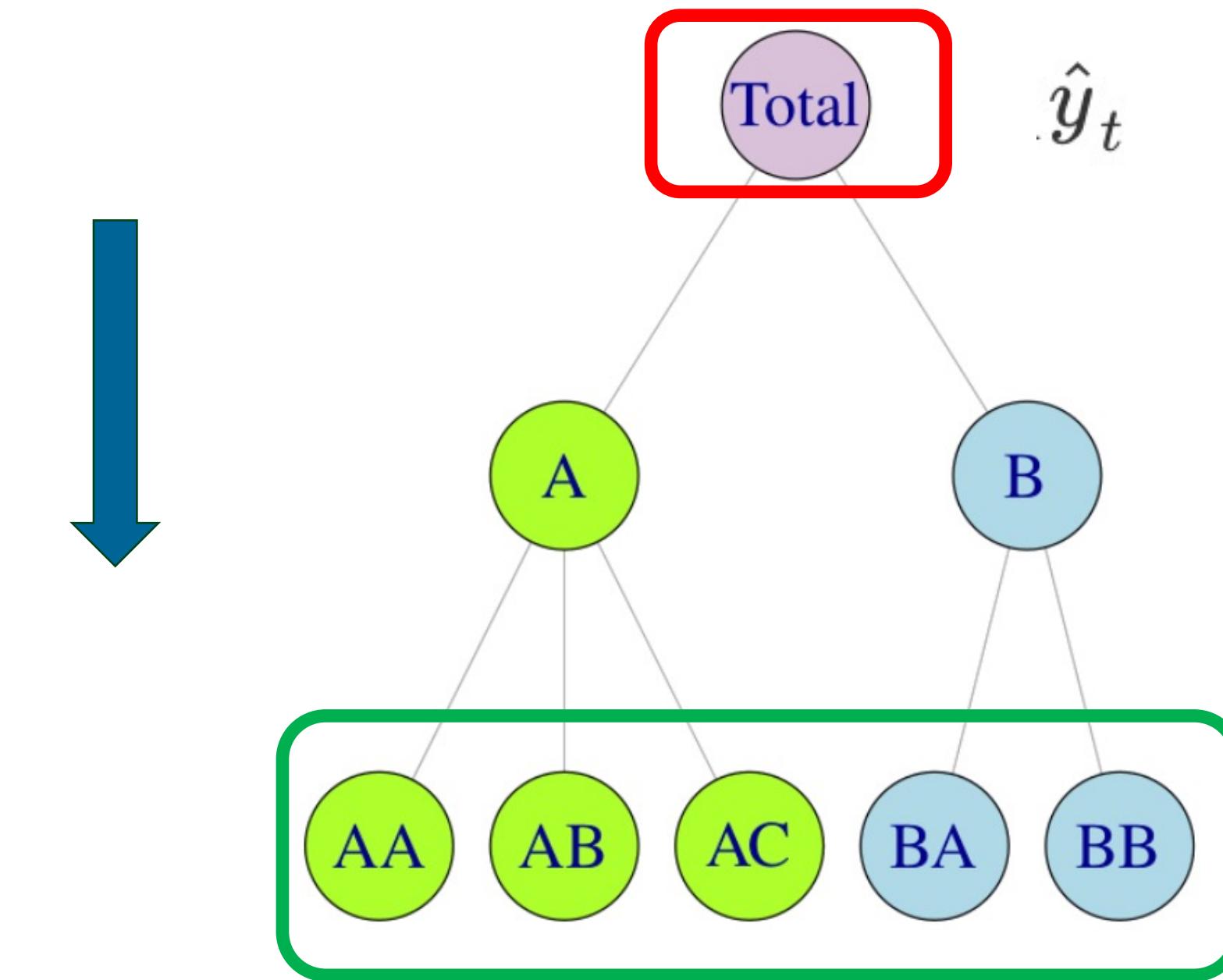
# Top-down

→ Свойства

- Не гарантирована несмешённость.
- Прогнозируем более гладкие ряды + меньше.
- Теряем информацию.

→ Кейс: хочу прогнозы объёмов грузоперевозок между всеми парами станций в неделю. Можем спрогнозировать по регионам, а потом спуститься до уровня станций.

Иерархия: Грузоперевозчик → Тип груза → Тип вагона → Регион отправления → Регион прибытия → Станция отправления → Станция прибытия.



$$p_1\hat{y}_t, p_2\hat{y}_t, p_3\hat{y}_t, p_4\hat{y}_t, p_5\hat{y}_t$$

P – ?

# Top-down

Average historical proportions

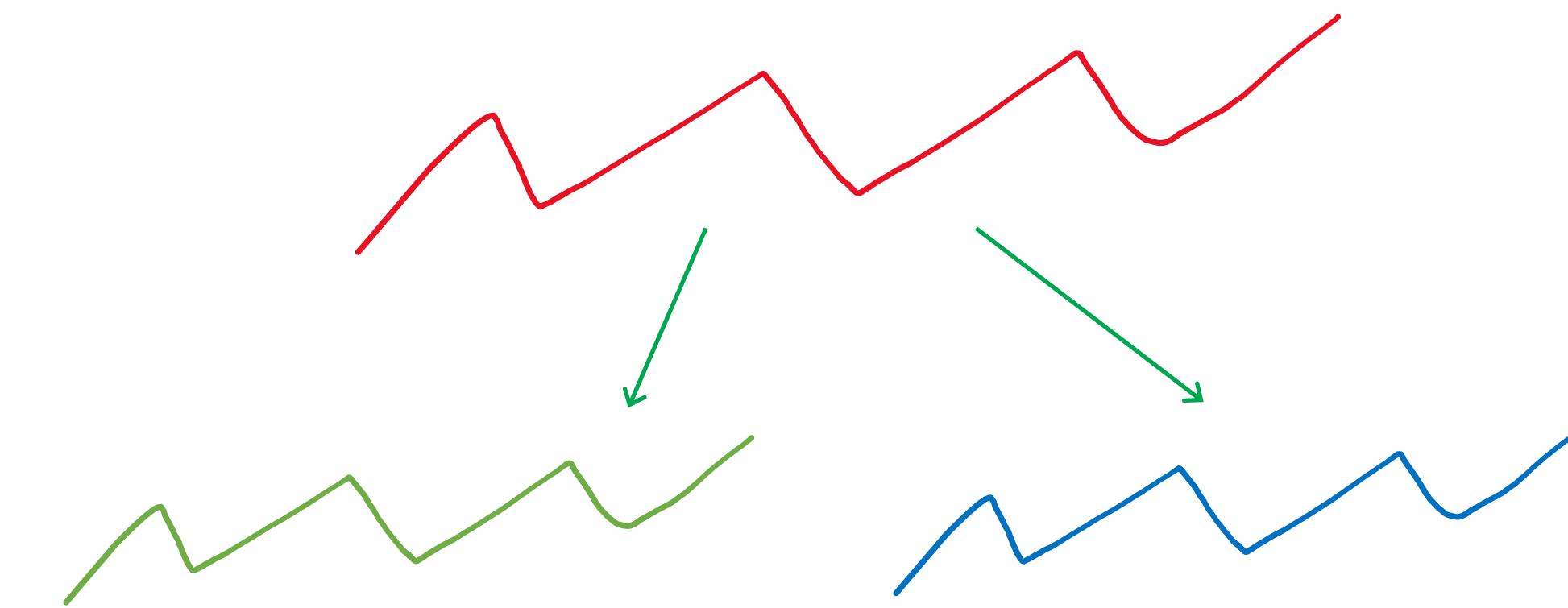
$$p_j = \frac{1}{T} \sum_{t=1}^T \frac{y_{j,t}}{y_t}$$

Можно придумывать и более изощрённые формулы.

1. Прогнозировать пропорции  $\hat{p}_{it}$ .
2. Можно сглаживать и считать по дням недели.

Proportions of the historical averages

$$p_j = \sum_{t=1}^T \frac{y_{j,t}}{T} \Big/ \sum_{t=1}^T \frac{y_t}{T}$$



# Матричная нотация

$S$  –  $(N \times K)$  матрица суммирования;

$N$  – количество рядов на всех уровнях;

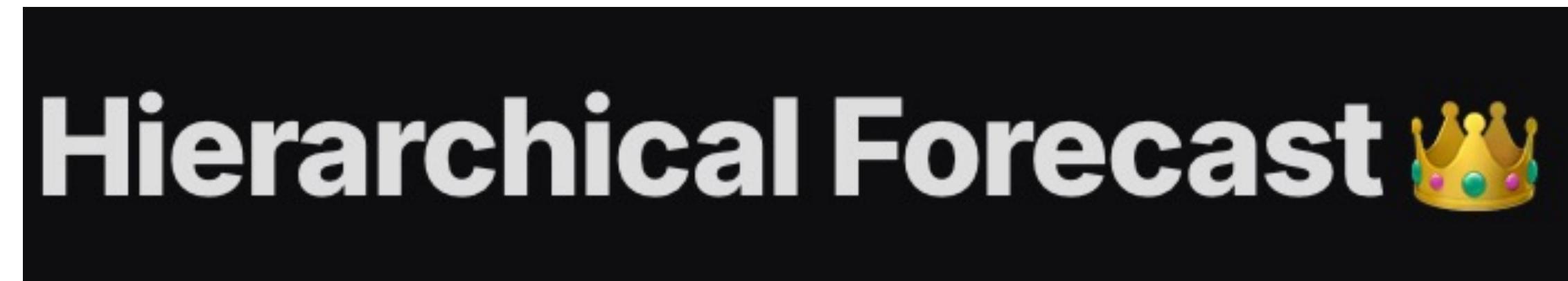
$K$  – количество рядов на самом низком уровне.

$S$  позволяет перейти от самого низкого уровня к любому другому.

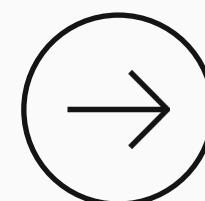
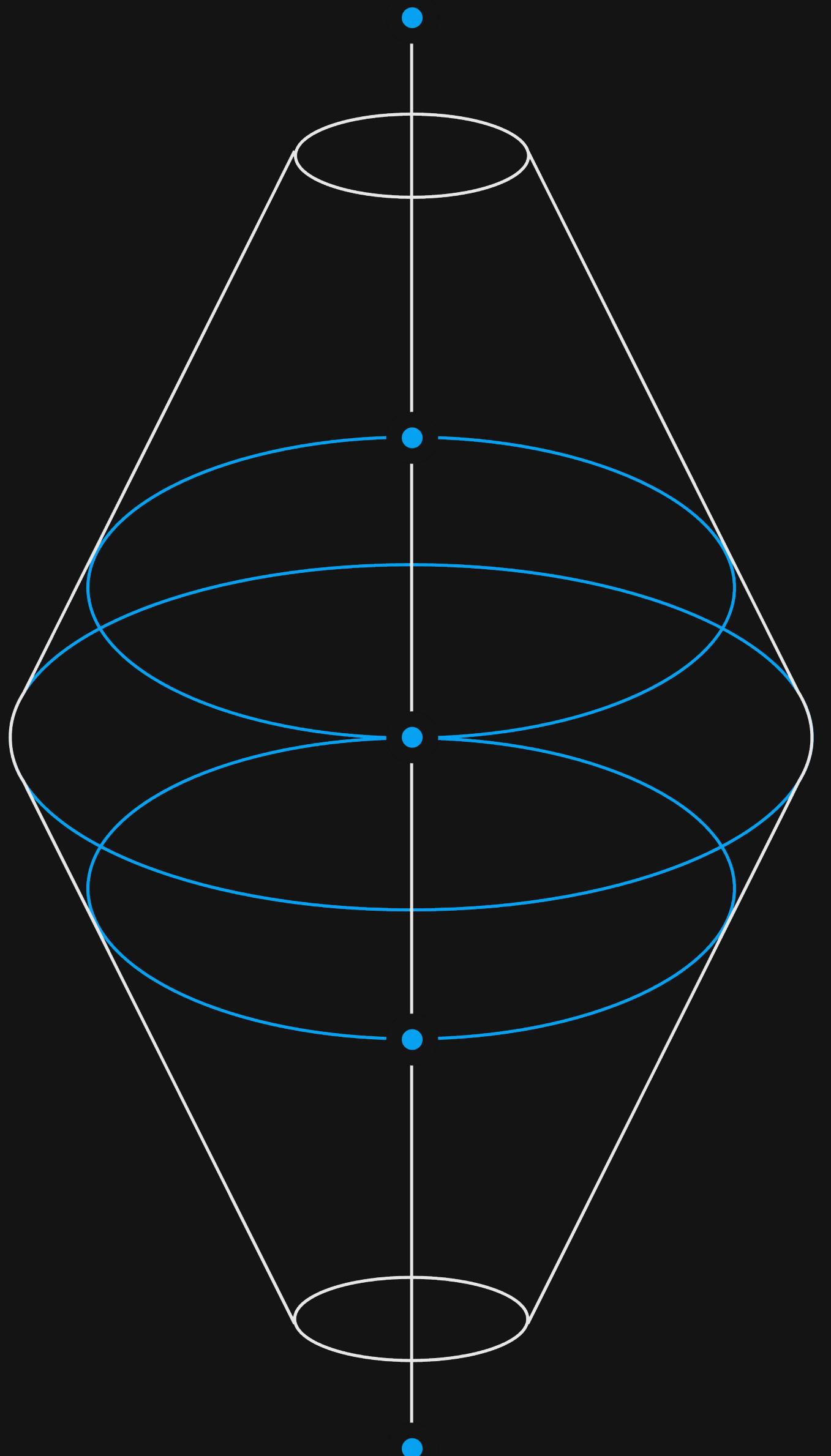
$$\begin{bmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{AA,t} \\ y_{AB,t} \\ y_{AC,t} \\ y_{BA,t} \\ y_{BB,t} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{AA,t} \\ y_{AB,t} \\ y_{AC,t} \\ y_{BA,t} \\ y_{BB,t} \end{bmatrix}$$

$$\mathbf{y}_t = S\mathbf{b}_t,$$

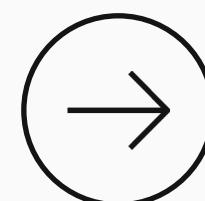
# Библиотеки с поддержкой



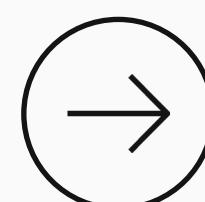
# Итоги



Хотим учитывать особенности рядов на разных уровнях иерархии.

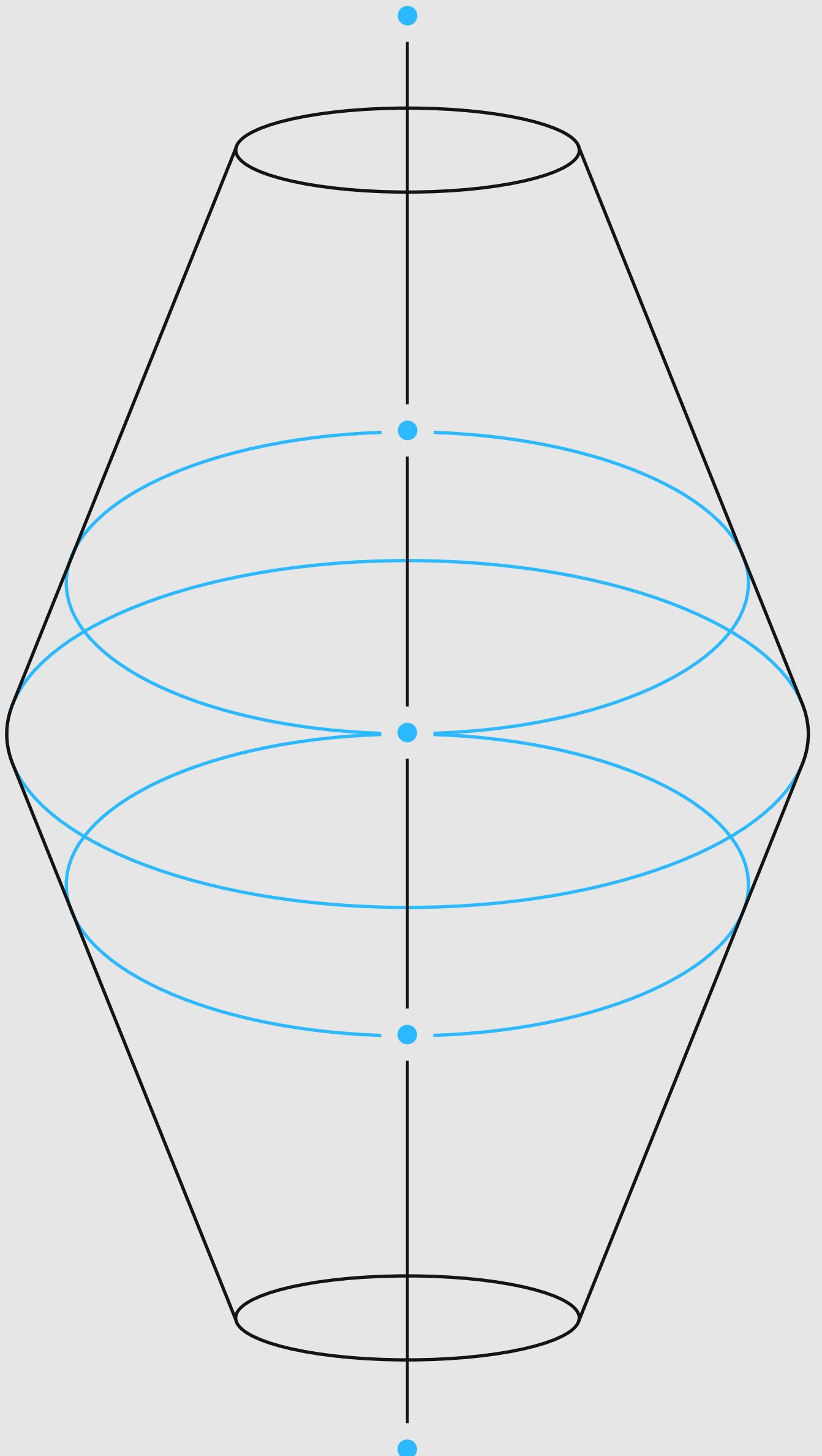


Выбираем наиболее информативный и прогнозируемый исходный уровень.



Можем прогнозировать сразу на всех уровнях для получения более точных прогнозов.

# Предсказательные интервалы



# Примеры применения

- Построение оптимистичных и пессимистичных сценариев в бизнес-моделировании
- Оптимизация закупки на складах, safety stock
- Проверка на аномальность бизнес-метрик

# Источники случайности

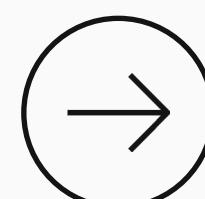
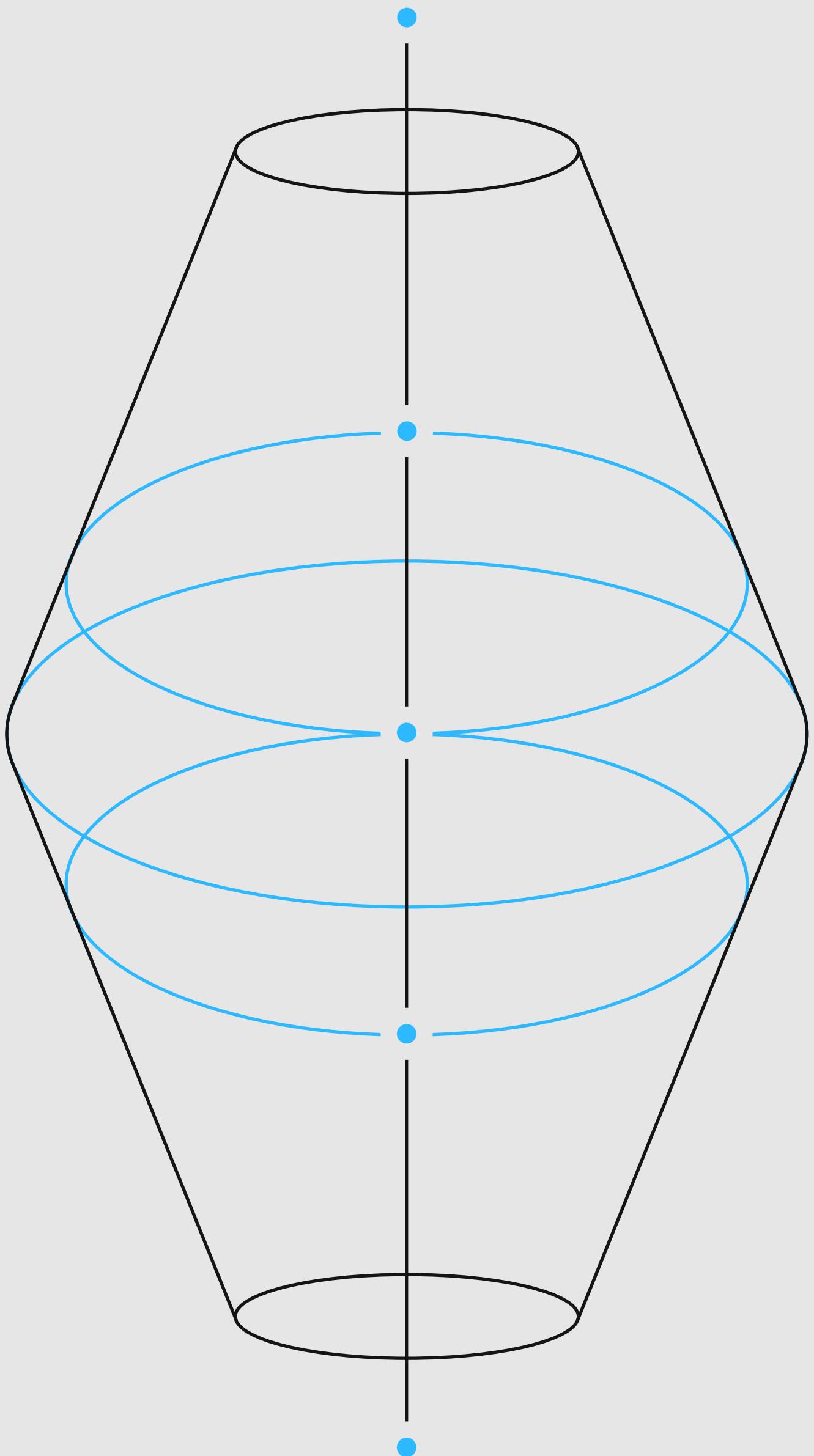
## Aleatoric

Случайная ошибка измерений

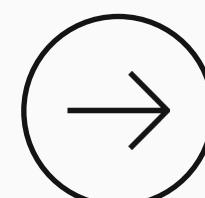
## Epistemic

- Случайность в оценке параметров
- Неопределённость в выборе модели для задачи
- Неопределённость из-за изменения DGP (процесса генерации данных) в будущем

# Стратегии прогнозирования



**Прямая:** сводим к табличному ML и используем методы для регрессий для построения интервалов.



**Рекурсивная:** предсказываем на шаг вперёд, вызываем модель на предсказанных данных.



# Подходы построения интервалов

## Model Specific

- Точечные модельные оценки
- Моделирование квантилей
- Байесовские оценки

## Model Agnostic

- Jacknife
- CV-оценка
- Конформные интервалы

# Model Specific

# Точные модельные оценки

ПРИМЕР

Линейная регрессия

$$\begin{aligned} m(x) &\equiv \mathbb{E}[Y|X = x] = \beta_0 + \beta_1 x \\ Y|X = x &\sim N(m(x), \sigma^2) \\ &= m(x) + N(0, \sigma^2) \\ &= \hat{m}(x) + N\left(0, \frac{\sigma^2}{n}\left(1 + \frac{(x - \bar{x})^2}{ns_X^2}\right)\right) + \underbrace{N(0, \sigma^2)}_{\text{noise}} \end{aligned}$$

parameter estimation

# Точные модельные оценки

ПРИМЕР

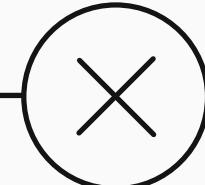
Линейная регрессия

$$s_{pred}^2 = \hat{\sigma}^2 \frac{n}{n-2} \left( 1 + \frac{1}{n} + \left( 1 + \frac{(x - \bar{x})^2}{ns_X^2} \right) \right)$$

В пределе:

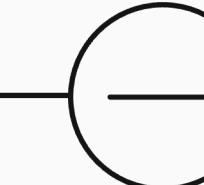
$$\frac{Y - \hat{m}(x)}{s_{pred}(x)} | X \sim N(0, 1)$$

# Точные модельные оценки



## Ограничения

- Условия на нормальность остатков.
- Асимптотическая нормальность.

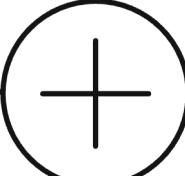


## Недостатки

Точные методы работают только явно вычисляемыми моделями.

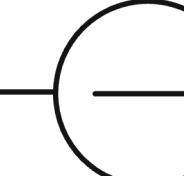
# Моделирование квантилей

Меняем MSE функцию на QuantileLoss:  $E((y - \hat{y})^2) \rightarrow E(\rho(y - \hat{y}))$



## Преимущества

Работает для большого класса моделей:  
градиентный бустинг, линейные  
и глубокие модели.



## Недостатки

Обучаем отдельную модель на квантиль  
либо модифицируем loss, если это  
возможно.



# Моделирование квантилей

ПРИМЕР

Sklearn и градиентный бустинг

• • •

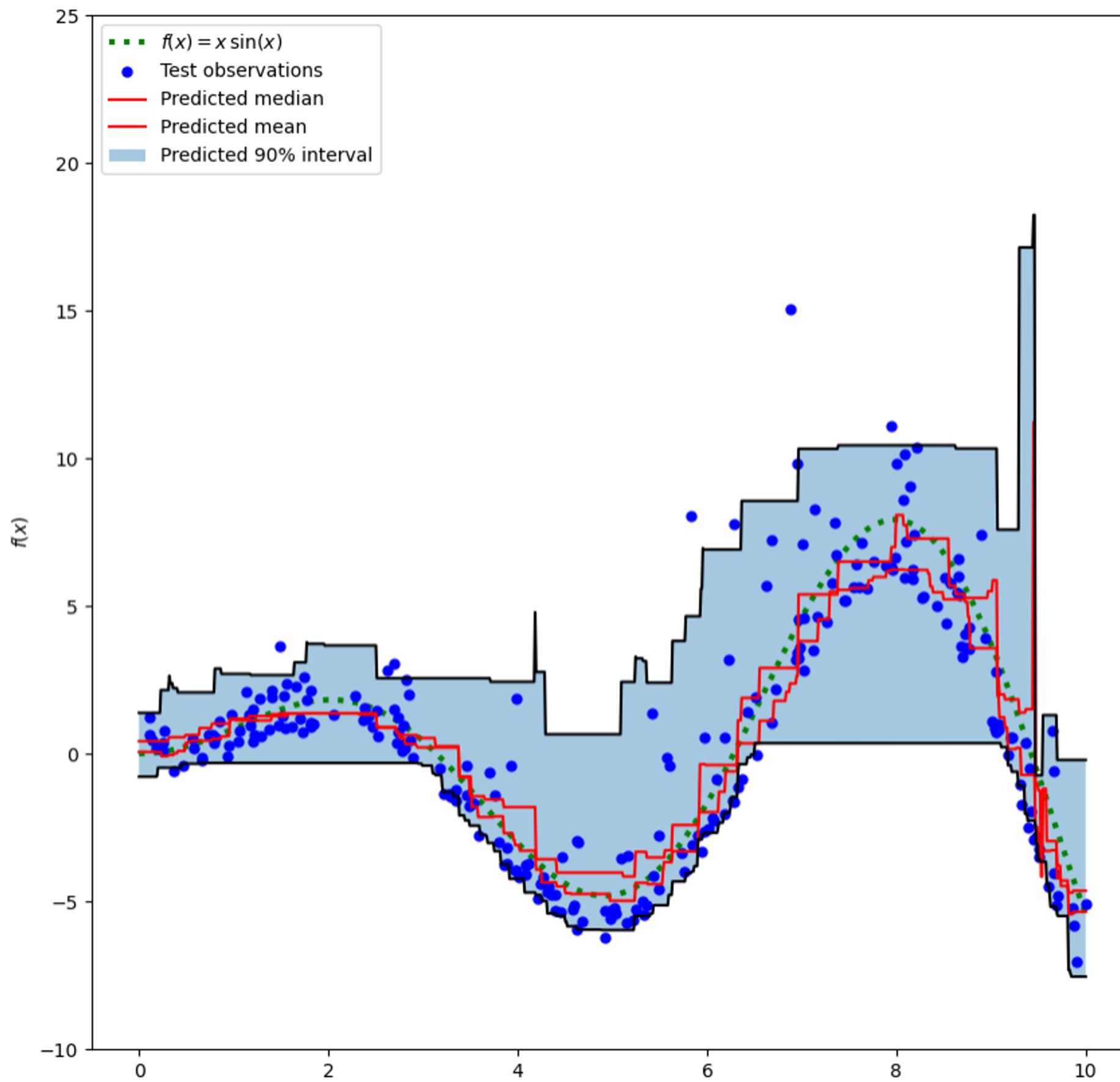
```
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_pinball_loss, mean_squared_error

all_models = {}

for alpha in [0.05, 0.5, 0.95]:
    gbr = GradientBoostingRegressor(loss="quantile", alpha=alpha)
    all_models["q %1.2f" % alpha] = gbr.fit(X_train, y_train)

y_pred = all_models["mse"].predict(xx)
y_lower = all_models["q 0.05"].predict(xx)
y_upper = all_models["q 0.95"].predict(xx)
y_med = all_models["q 0.50"].predict(xx)
```

# Моделирование квантилей



# Вероятностные модели: частотный подход



## ПРИМЕР

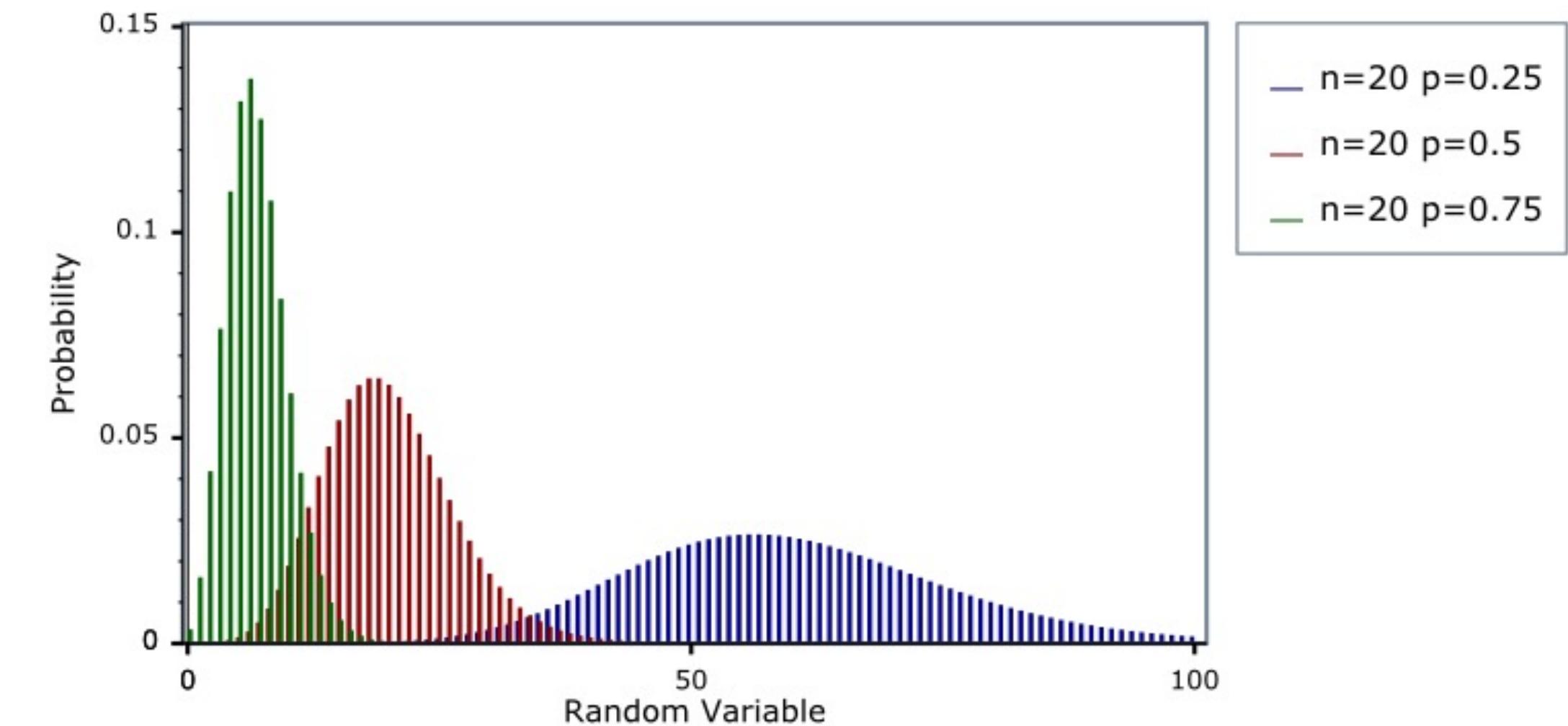
DeepAR и метод максимального правдоподобия

$$\mathcal{L} = \sum_i \sum_t \log l(z_{i,t} | \theta(h_{i,t})), h(\cdot) - \text{RNN}$$

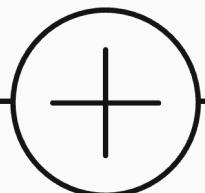
$$l_G(z|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-(z-\mu)^2/(2\sigma^2))$$

$$l_{\text{NB}}(z|\mu, \alpha) = \frac{\Gamma(z + \frac{1}{\alpha})}{\Gamma(z+1)\Gamma(\frac{1}{\alpha})} \left(\frac{1}{1+\alpha\mu}\right)^{\frac{1}{\alpha}} \left(\frac{\alpha\mu}{1+\alpha\mu}\right)^z$$

## Negative Binomial Distribution

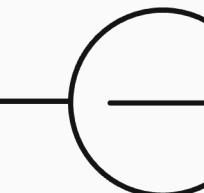


# Вероятностные модели: частотный подход



## Преимущества

Не требует дополнительных вычислений.



## Недостатки

- Важно корректно задать семейство распределений.
- В основном оцениваем неопределённость, связанную с шумом, чтобы оценить неопределённость в параметрах, можем семплировать данные.

# Вероятностные модели: байесовский вывод



$$p(\theta|D) = \frac{p(\mathbf{y}, \theta|\mathbf{X})}{p(\mathbf{y}|\mathbf{X})} = \frac{p(\mathbf{y}|\mathbf{X}, \theta)}{\int p(\mathbf{y}|\mathbf{X}, \theta)p(\theta)d\theta}$$

Предсказание:

$$\begin{aligned} p(y_{n+1}|\mathbf{x}_{n+1}, D) &= \int p(y_{n+1}|\mathbf{x}_{n+1}, \theta)p(\theta|D)d\theta \\ \hat{y}_{n+1} &= \mathbb{E}[y_{n+1}|\mathbf{x}_{n+1}, D] \end{aligned}$$

# Вероятностные модели: байесовский вывод



## ПРИМЕР

Линейная регрессия и румс (Н. В. Линейную регрессию можно проинтегрировать.)

• • •

```
import pymc3 as pm
import arviz as az
from pymc3 import HalfCauchy, Model, Normal, sample

SIGMA_PRIOR = 10

with Model() as model:

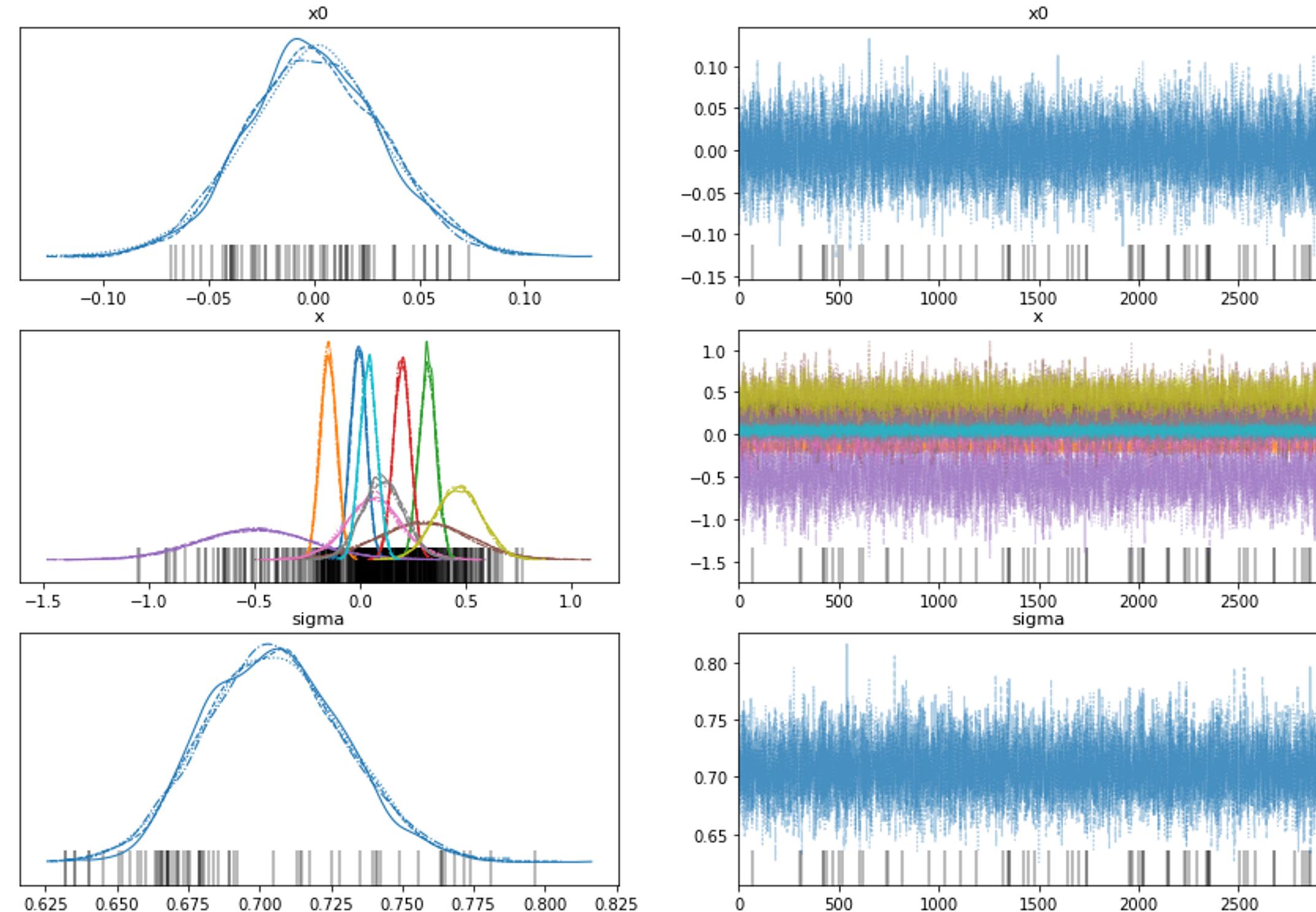
    sigma = HalfCauchy("sigma", beta=SIGMA_PRIOR)
    intercept = Normal("x0", 0, sigma=SIGMA_PRIOR)
    x_coeff = Normal("x", 0, sigma=SIGMA_PRIOR, shape=(X.shape[1]))
    likelihood = Normal(
        "y", mu=intercept + pm.math.dot(X, x_coeff), sigma=sigma, observed=y
    )
    trace = sample(3000, return_inferencedata=True)
```

# Вероятностные модели: байесовский вывод

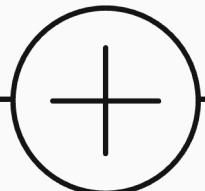


ПРИМЕР

Линейная регрессия и румс (Н. В. Линейную регрессию можно проинтегрировать.)

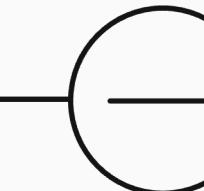


# Вероятностные модели: байесовский вывод



## Преимущества

Полноценные апостериорные распределения на параметры и предсказания.



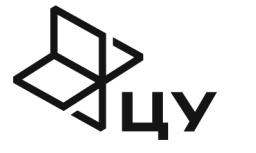
## Недостатки

- Важно корректно задать семейство распределений.
- В большинстве случаев приходится использовать аппроксимации: МCMC или Variational Inference.



# Model Agnostic

# CV-оценка



Input: significance level  $\alpha$ , train set  $\mathcal{T}$ , validation set  $\mathcal{V}$ , inference point  $x_{n+1}$

Output: prediction interval for  $x_{n+1}$

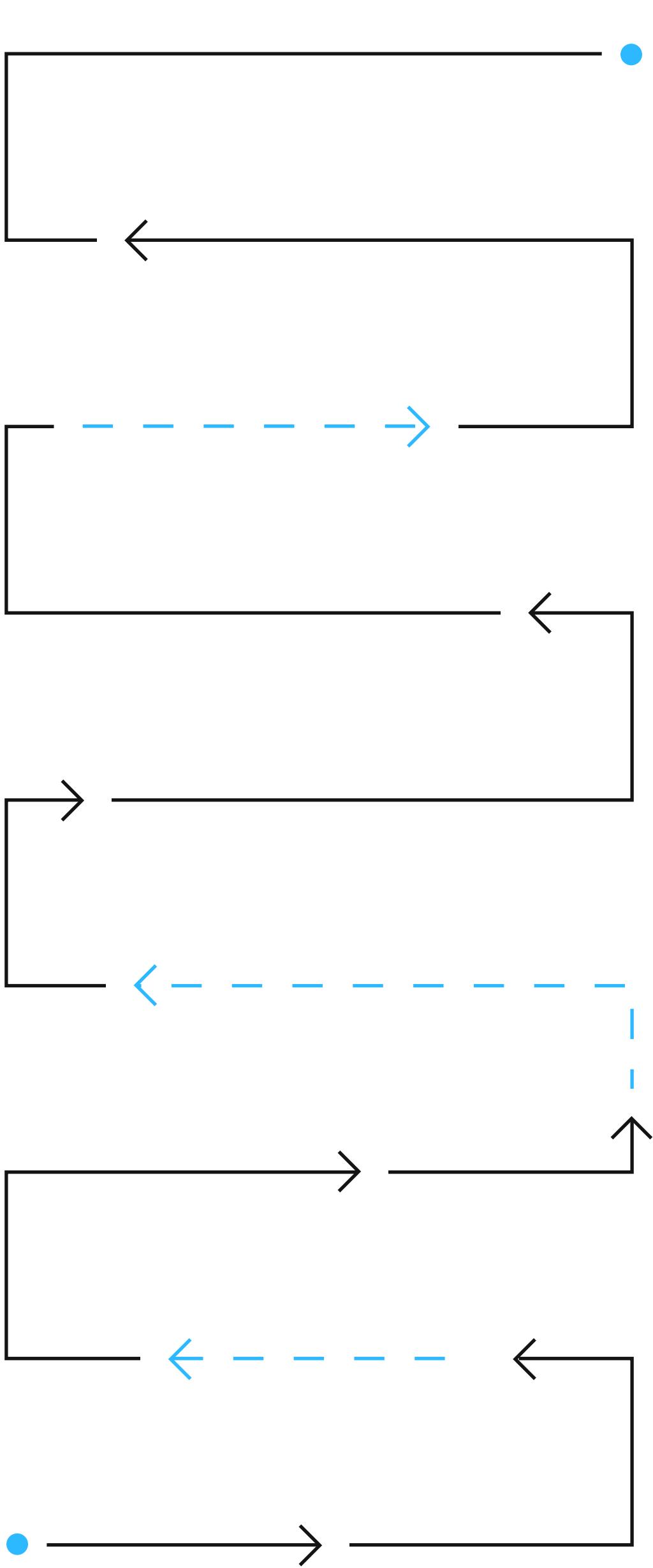
Train  $\hat{m}$  on  $\mathcal{T}$

Loop for each  $(x_i, y_i)$  in  $\mathcal{V}$ :

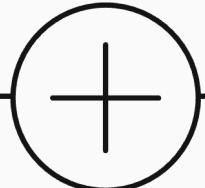
residuals  $\leftarrow$  residuals  $\cup$   $(y_i - \hat{m}(x_i))$

std  $\leftarrow$  std(residuals)

return  $\hat{m}(x_{n+1}) \pm z_{1-\frac{\alpha}{2}} \text{ std}$

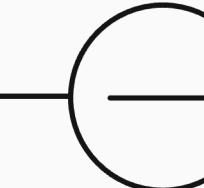


# CV-оценка



## Преимущества

- Model Agnostic.
- Стабильные интервалы.



## Недостатки

- Необходимо валидационное множество.
- Используем асимптотическую нормальность.
- Работающая эвристика.

# Конформные интервалы



## Алгоритм

Input: significance level  $\alpha$ , nonconformity measure  $A$ , train set  $\mathcal{T}$ , calibration set  $\mathcal{V}$ , inference point  $x_{n+1}$   
Output: prediction interval for  $x_{n+1}$

Train  $\hat{m}$  on  $\mathcal{T}$

Loop for each  $(x_i, y_i)$  in  $\mathcal{V}$ :

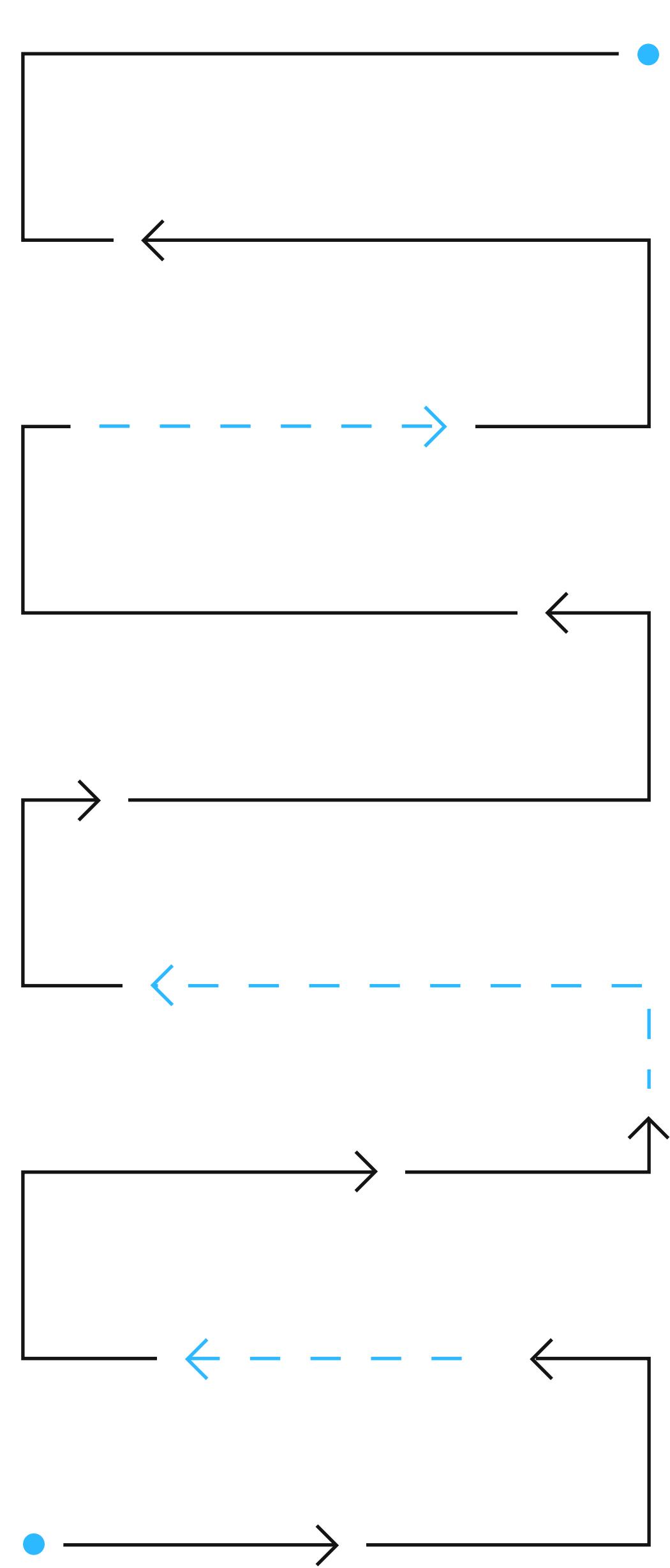
    measures  $\leftarrow$  measures  $\cup A(x_i, y_i)$

Critical value:  $\alpha^* \leftarrow (\text{the } ((1 - \alpha)(1 + \frac{1}{|\mathcal{V}|})) \text{ quantile of measures set})$

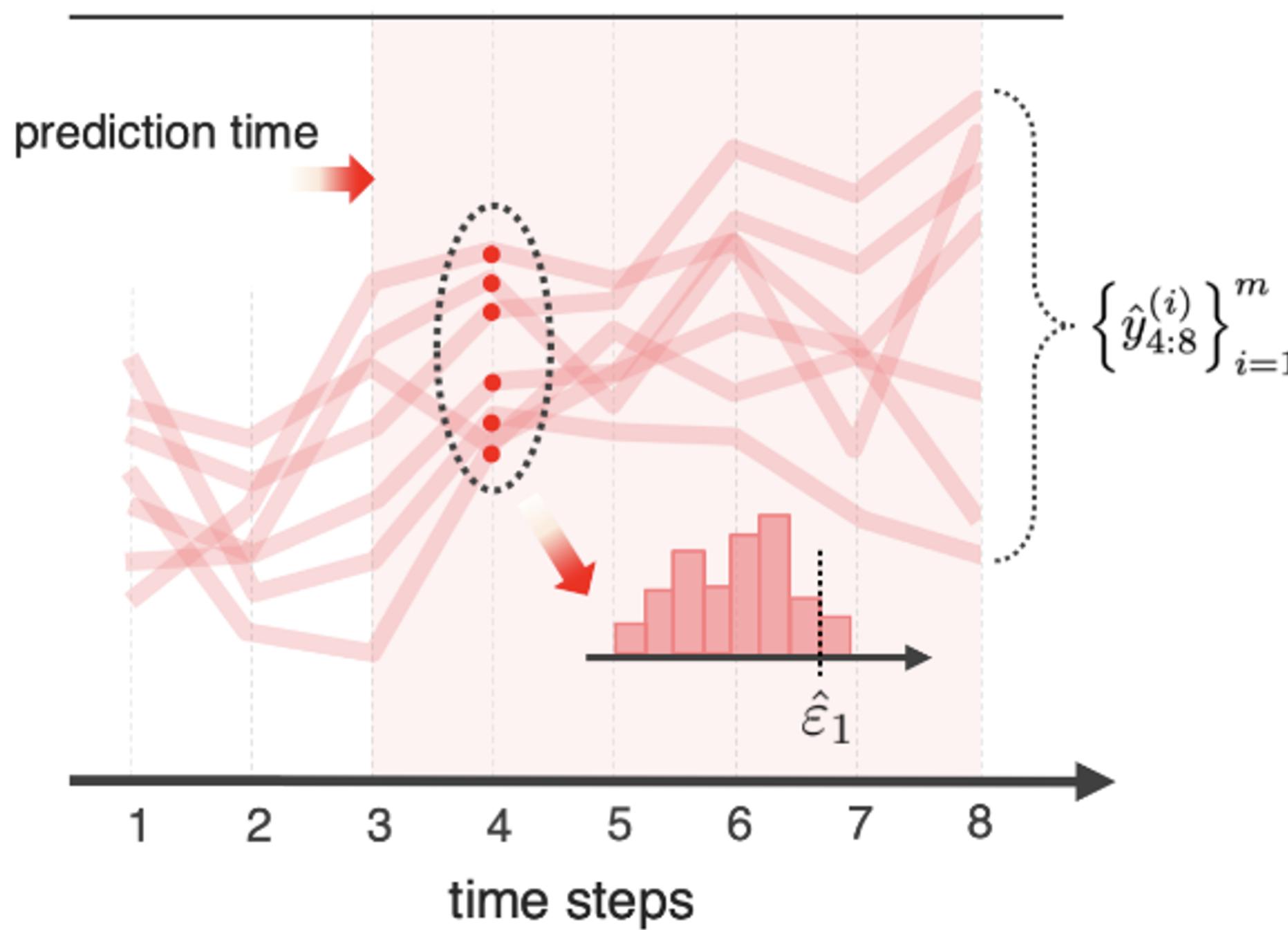
Loop for each  $y \in \mathbb{R}$ :

$\alpha_y \leftarrow A(x_{n+1}, y)$

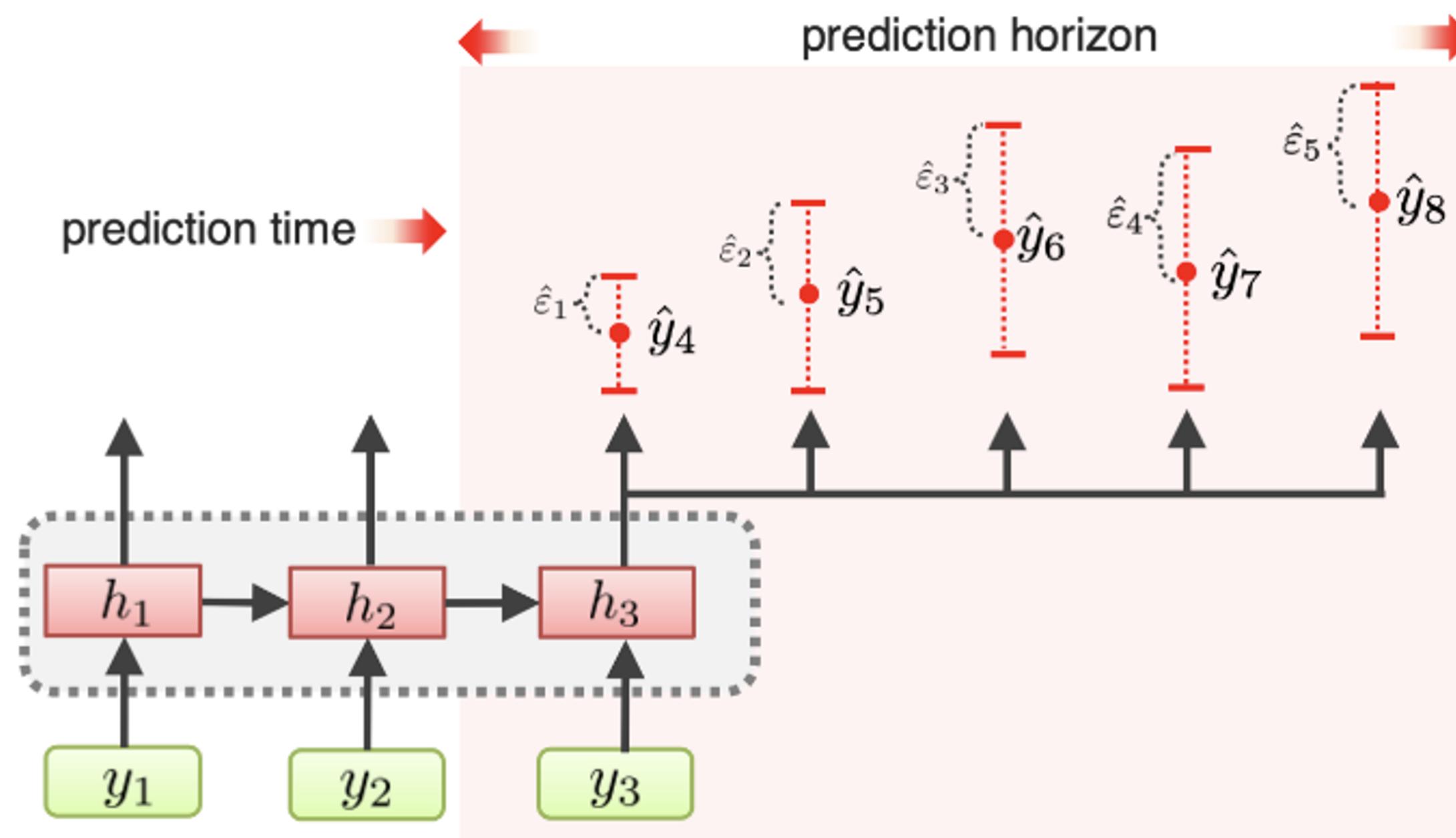
return  $\{y \in \mathbb{R} : \alpha_y \leq \alpha^*\}$



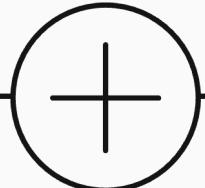
**(a) Construction of critical nonconformity scores**



**(b) Forecasting on a new time-series**

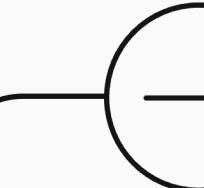


# Конформные интервалы



## Преимущества

- Model Agnostic.
- Больше гарантий на покрытие.



## Недостатки

- Допущение взаимозаменяемости — более слабый аналог независимости.
- Выбор меры конформности.
- Валидационное множество.

# Как измерять качество

- Игра вокруг настройки оптимальной Coverage и Width.
- Coverage должен стремиться к теоретическому уровню  $(1 - \alpha)$  сверху.
- Ширина Width достаточная, чтобы таким интервалом можно было пользоваться.

$$\text{Coverage} = \frac{1}{n} \sum (\hat{y}_{i\text{upper}} \geq y_i) * (\hat{y}_{i\text{lower}} \leq y_i)$$

$$\text{Width} = \frac{1}{n} \sum (\hat{y}_{i\text{upper}} \geq \hat{y}_{i\text{lower}})$$

# Как измерять качество: другие метрики



Quantile Loss

Scaled Quantile Loss – M5

Scaled Interval Score – M4

$$\text{MSIS} = \frac{1}{n} \frac{\sum((\hat{y}_{iupper} - \hat{y}_{ilower}) + \frac{2}{\alpha}(\hat{y}_{ilower} - y) * [y < \hat{y}_{ilower}] + \frac{2}{\alpha}(y_i - \hat{y}_{iupper}) * [y_i > \hat{y}_{iupper}])}{\text{insample seasonal model mae}}$$

Как у других

# Моделирование интервалов в Kats



Input: significance level  $\alpha$ , train set  $\mathcal{T}$ , inference point  $x_{n+1}^h$

Output: prediction interval for  $x_{n+1}^h$

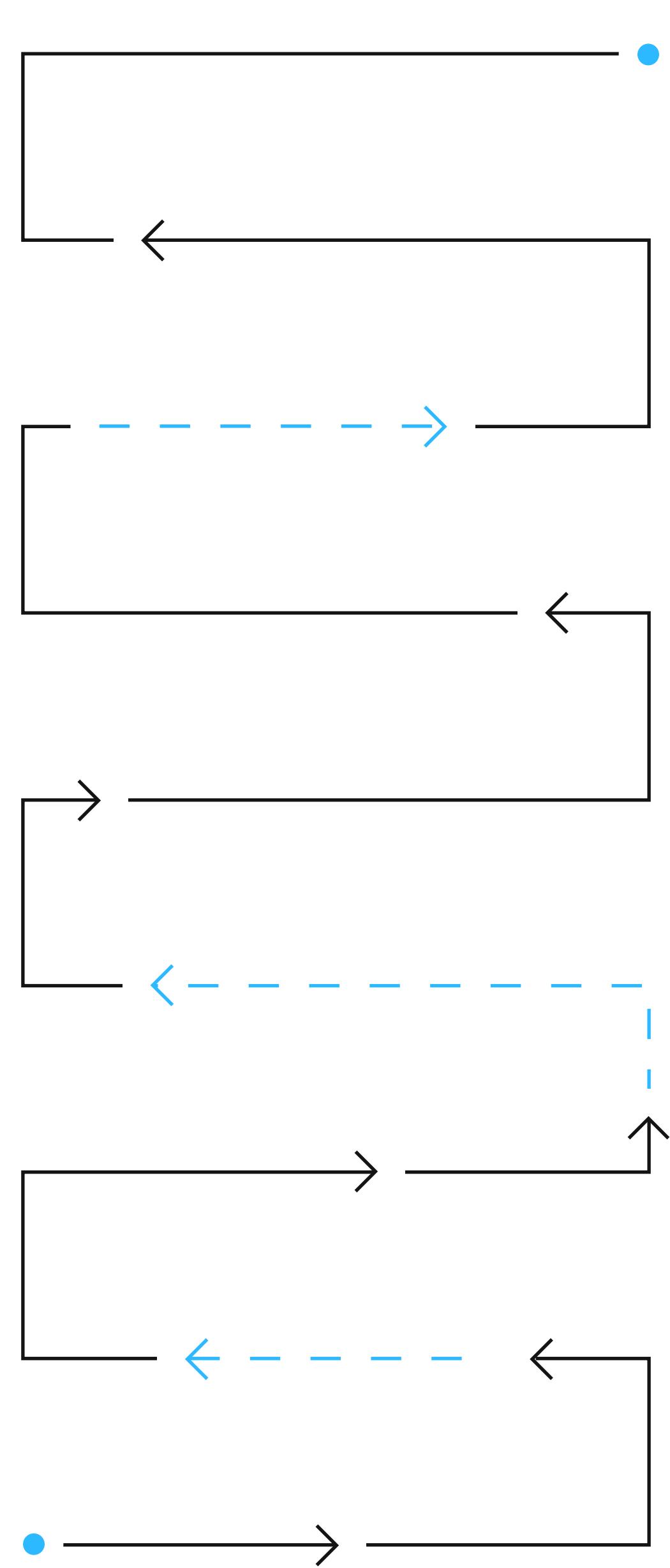
Run K-fold CV for a given model and data each fold contains  $h$  time ahead points

For each horizon, calculate the std of residuals terms

Fit linear model:  $\text{std} \sim h$

Estimate the  $\hat{\text{std}}_h$  for each horizon for the true future

Return:  $\hat{m}(x_{m+1}^h) \pm z_{1-\frac{\alpha}{2}} \hat{\text{std}}_h$



# Моделирование интервалов в Prophet



• • •

```
model {
    // Priors
    k ~ normal(0, 5);
    m ~ normal(0, 5);
    delta ~ double_exponential(0, tau);
    sigma_obs ~ normal(0, 0.5);
    beta ~ normal(0, sigmas);

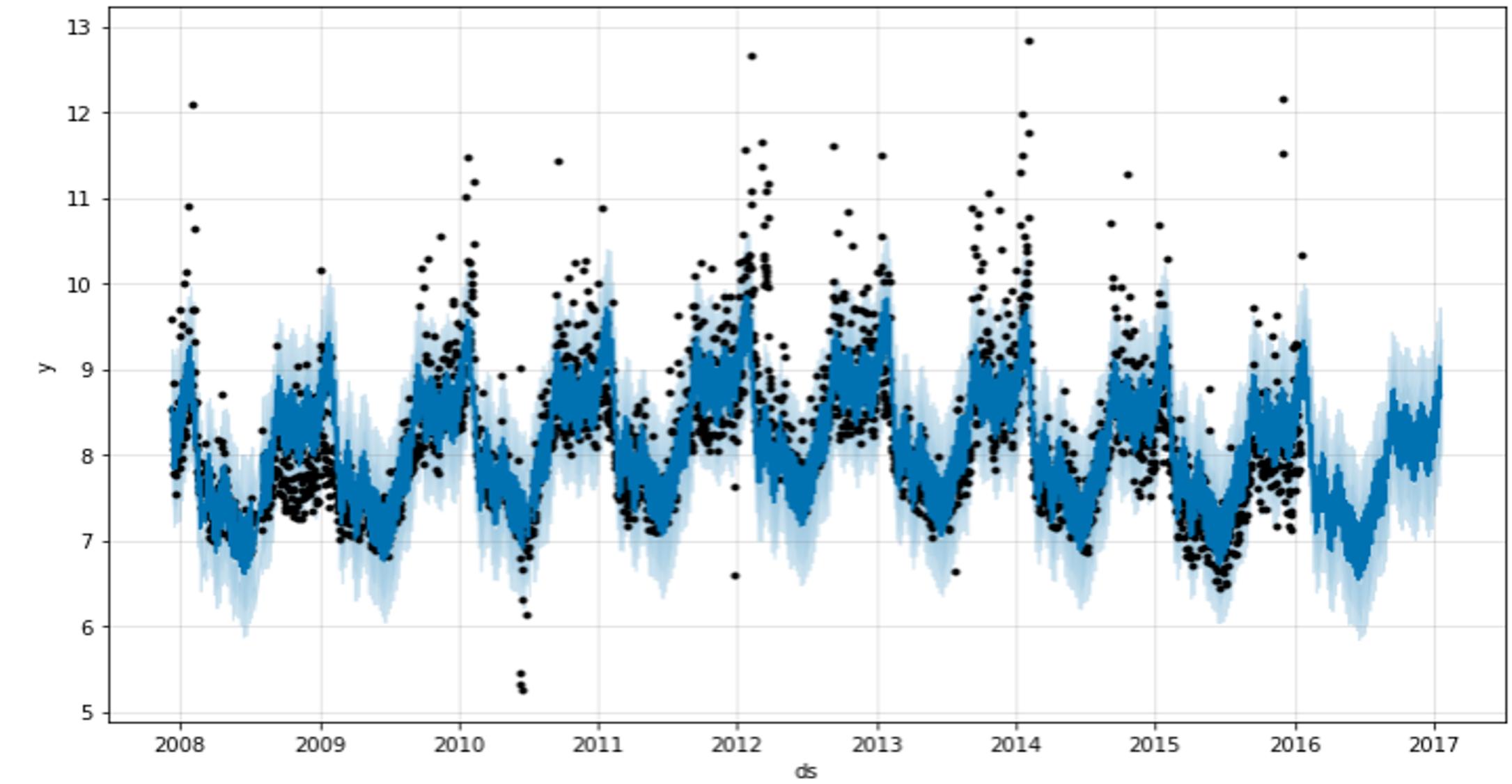
    // Likelihood
    y ~ normal_id_glm(
        x_sa,
        trend .* (1 + x_sm * beta),
        beta,
        sigma_obs
    );
}
```

# Моделирование интервалов в Prophet



• • •

```
# changepoint_prior_scale 0.001
m = Prophet(changepoint_prior_scale=0.001)
forecast = m.fit(df).predict(future)
fig = m.plot(forecast)
```

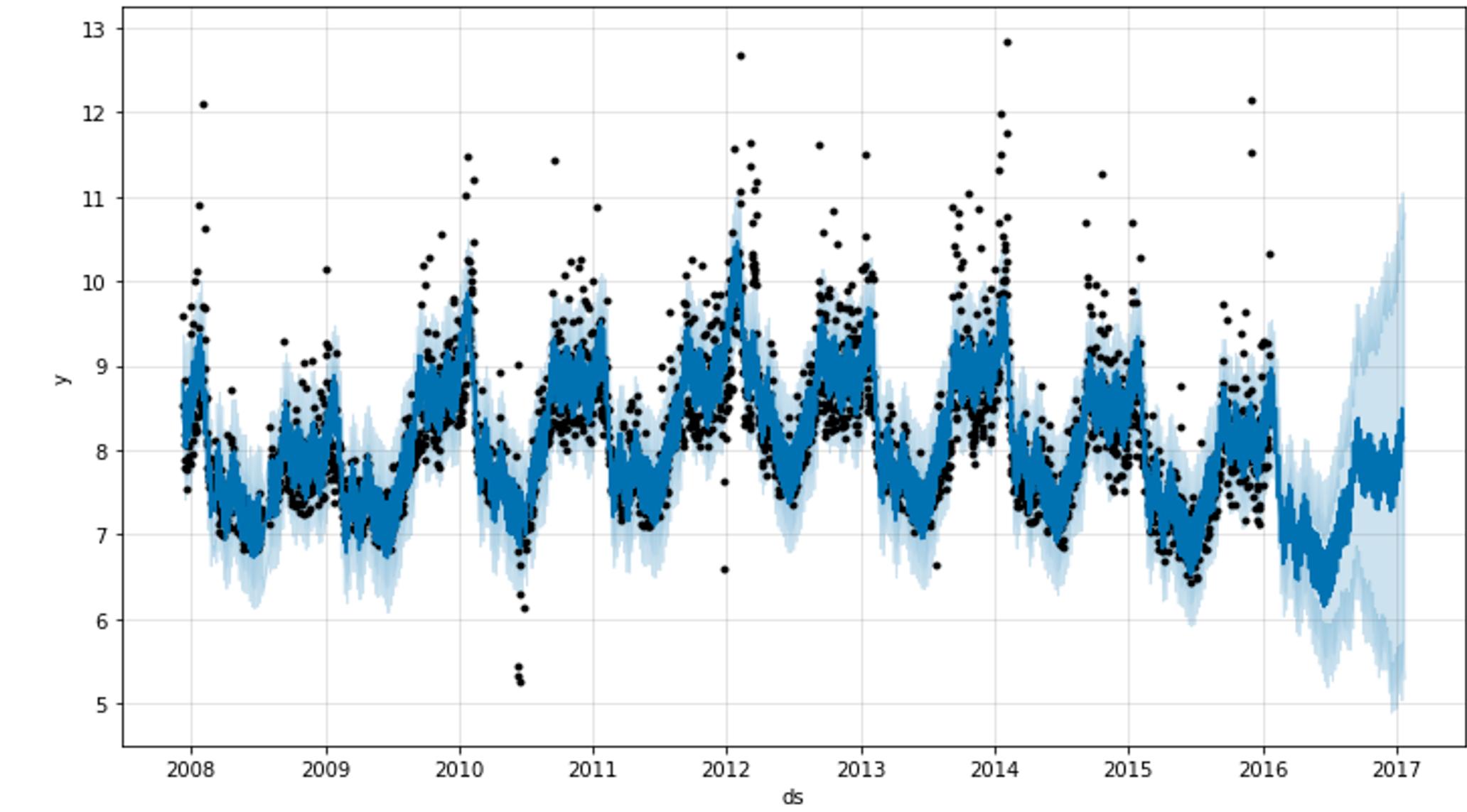


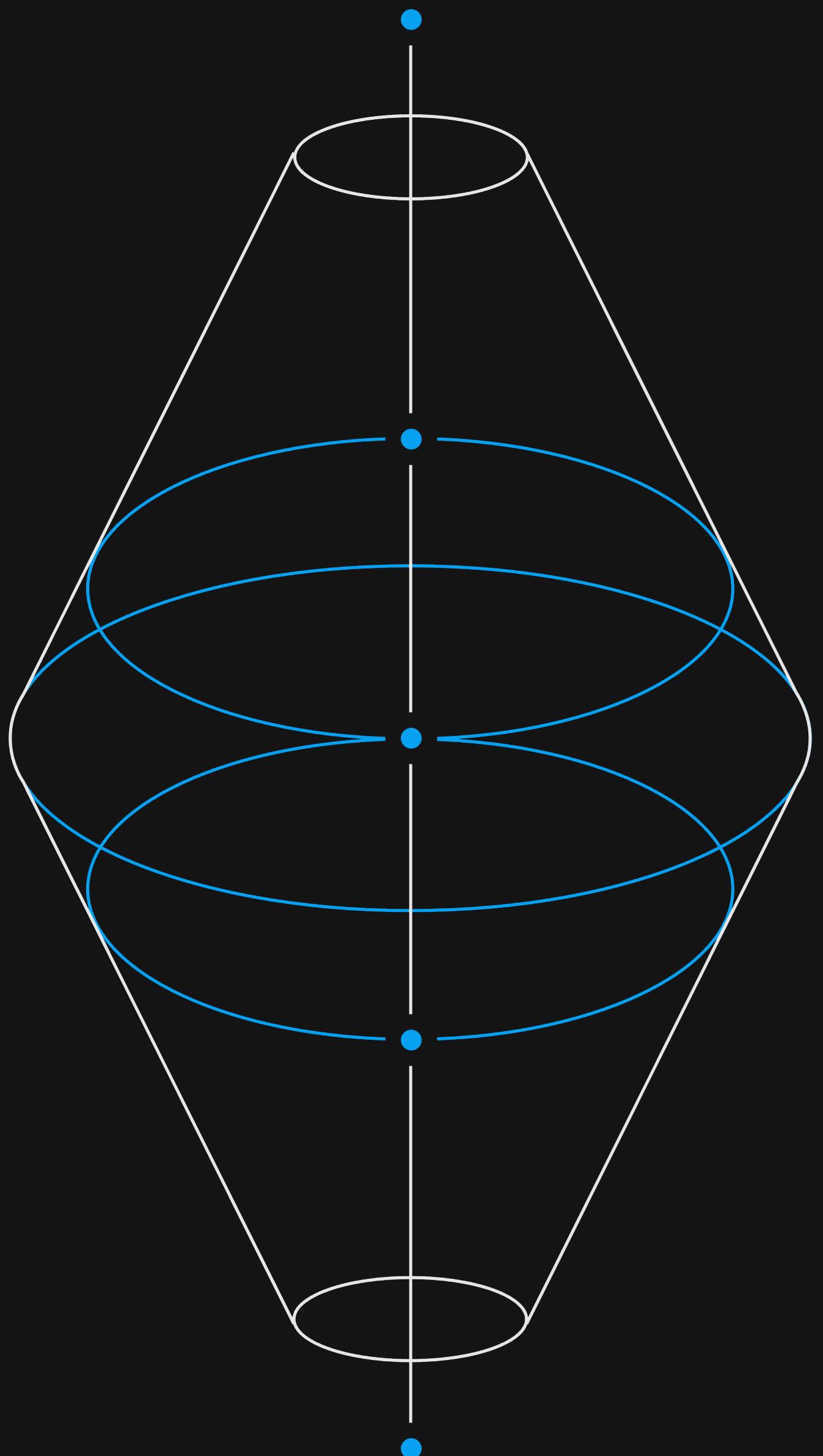
# Моделирование интервалов в Prophet



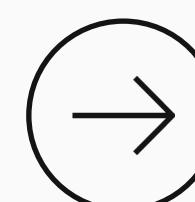
• • •

```
# changepoint_prior_scale 0.5
m = Prophet(changepoint_prior_scale=0.5)
forecast = m.fit(df).predict(future)
fig = m.plot(forecast)
```

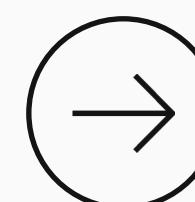




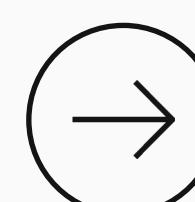
# Итого



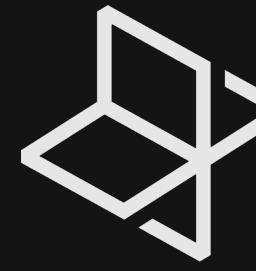
Мы рассмотрели основные способы построения предсказательных интервалов.



Выбор подхода сильно зависит от выбранной модели, данных и вычислительных ресурсов.



Важно оценивать не только качество прогнозов, но и качество предсказательных интервалов, если вы их собираетесь использовать.



ЦЕНТРАЛЬНЫЙ  
УНИВЕРСИТЕТ

# Вопросы

