

# Временные ряды: методы и задачи

Мартин Габдушев, ETNA, tinkoff.ai

Яков Малышев, ETNA, tinkoff.ai

2024



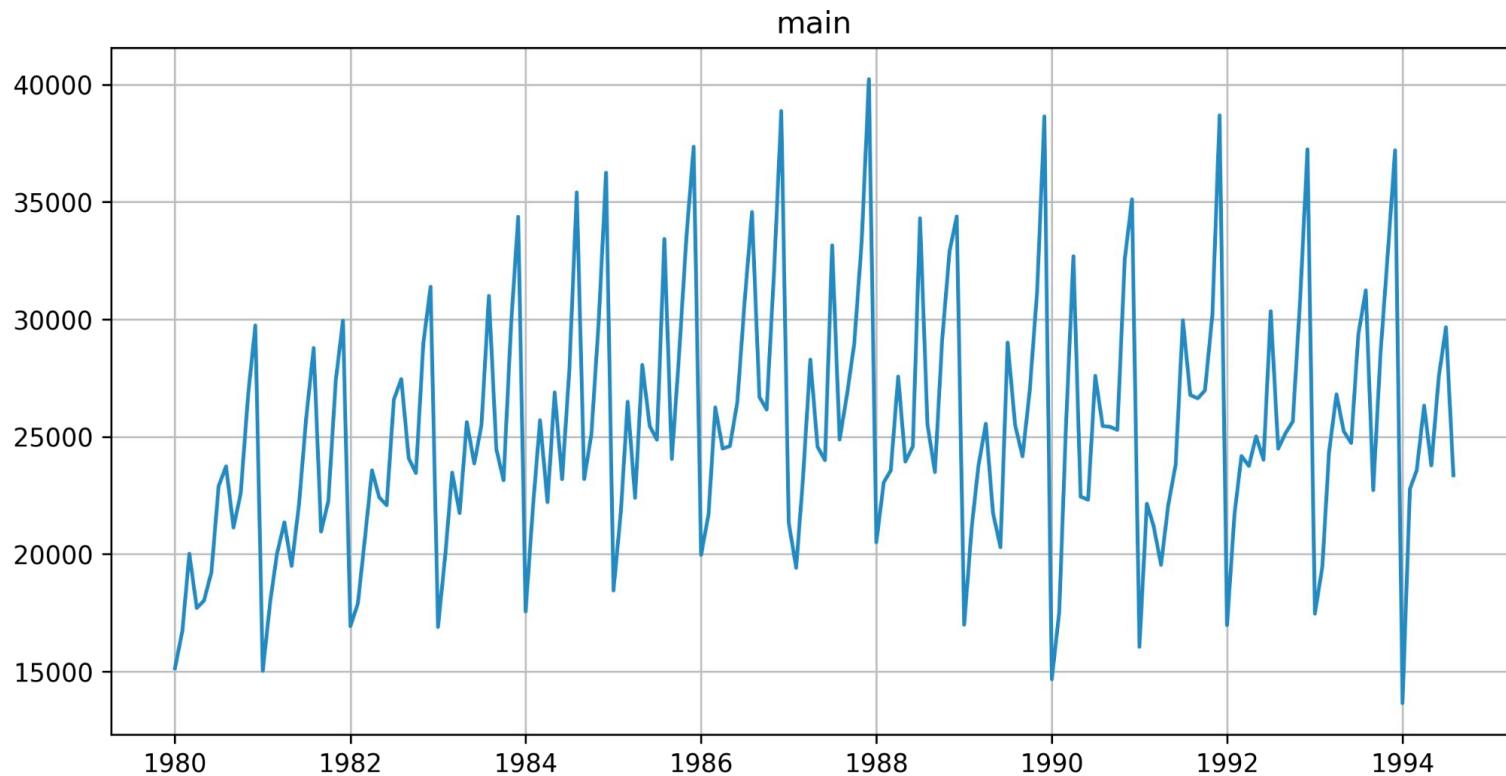
# Определение

Временной ряд — это последовательность замеров некоторых параметров, собранная через некоторые промежутки времени.

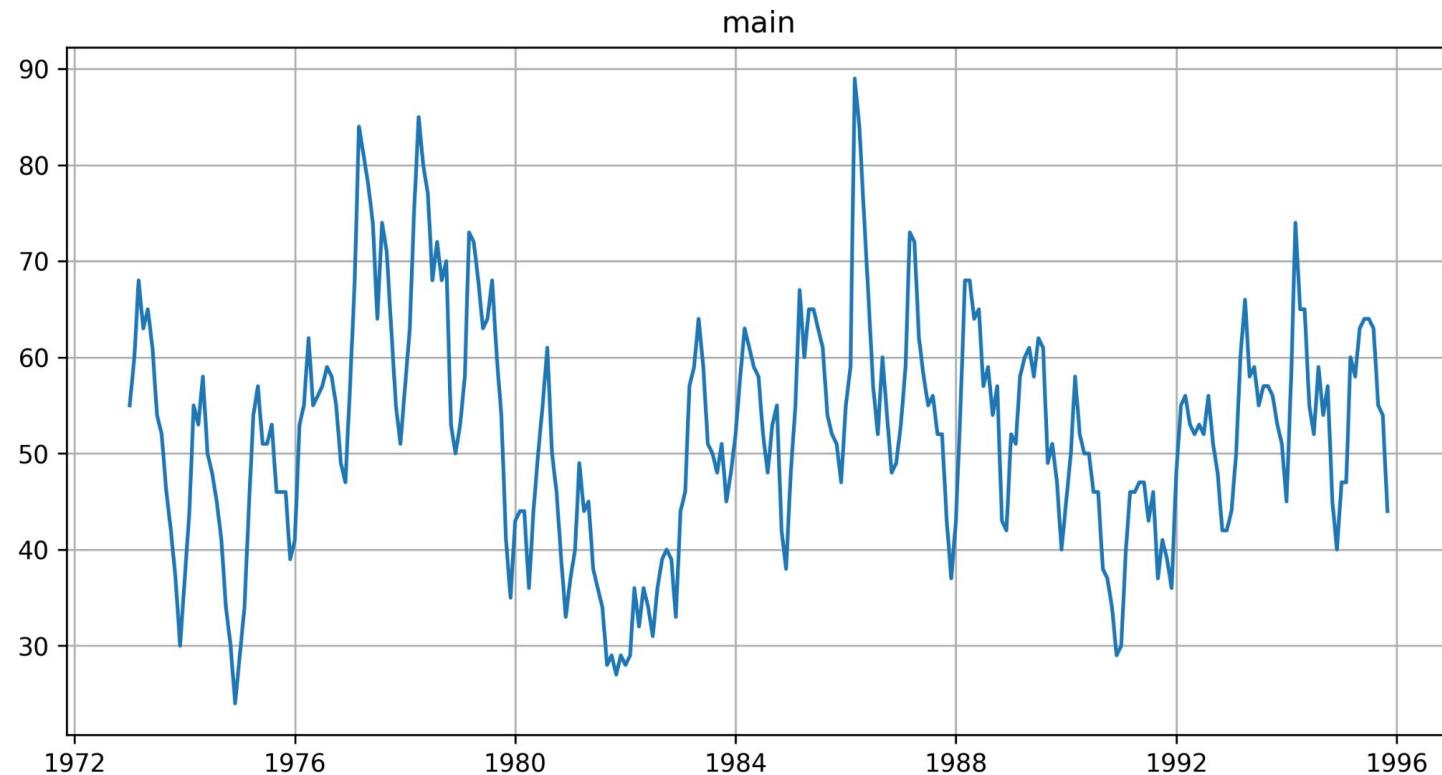
Более формально:  $((y_i, t_i), t_k < t_l \iff k < l)_{i=0}^{\infty}$

- $y_i$  — значения временного ряда
- $t_i$  — временные метки, в простом случае это просто номер

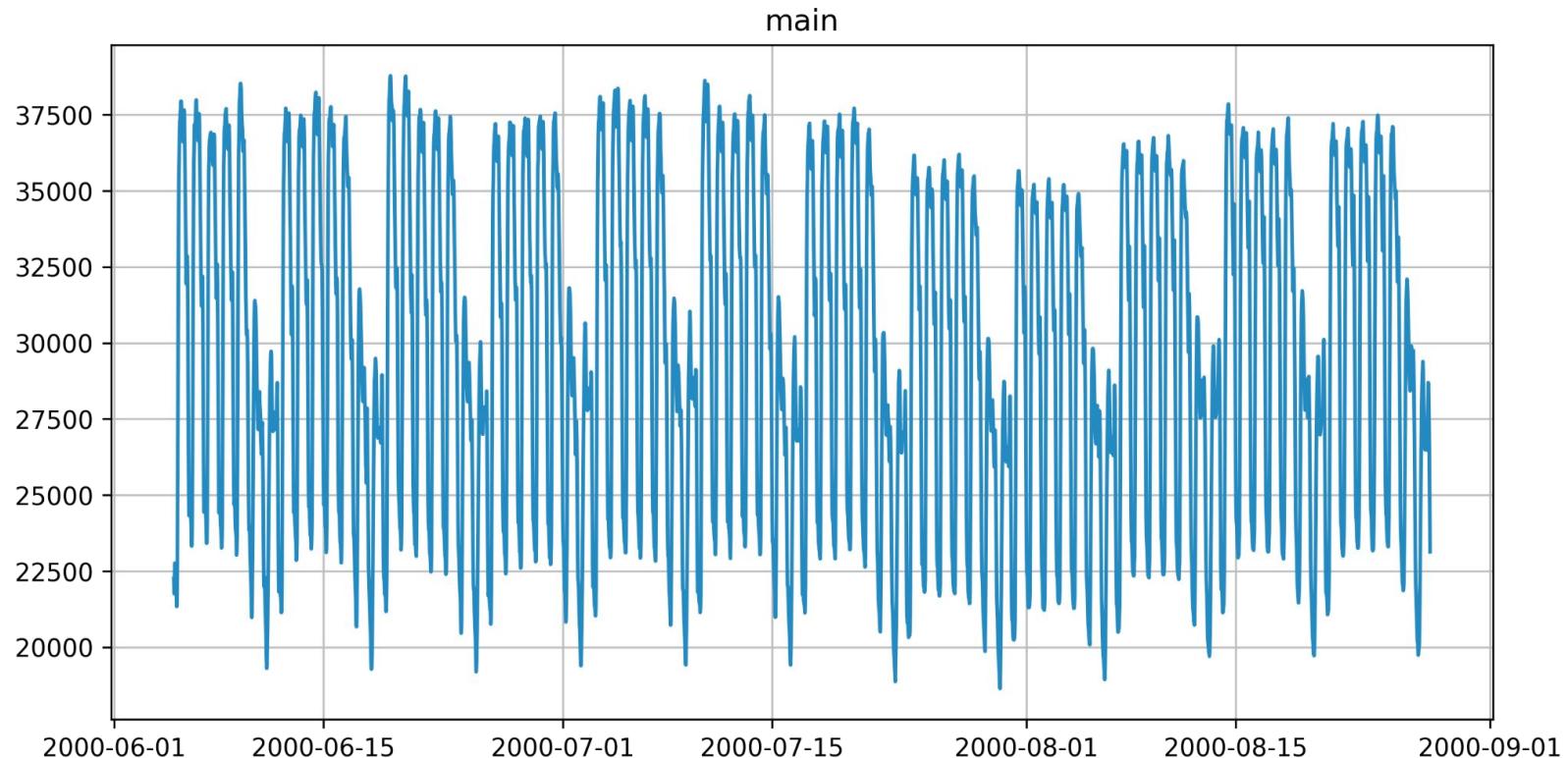
# Пример: месячные продажи вина



# Пример: месячные продажи новых домов в США



# Пример: спрос на электричество в МВт



# Задачи на временных рядах

Прогнозирование

Поиска аномалий

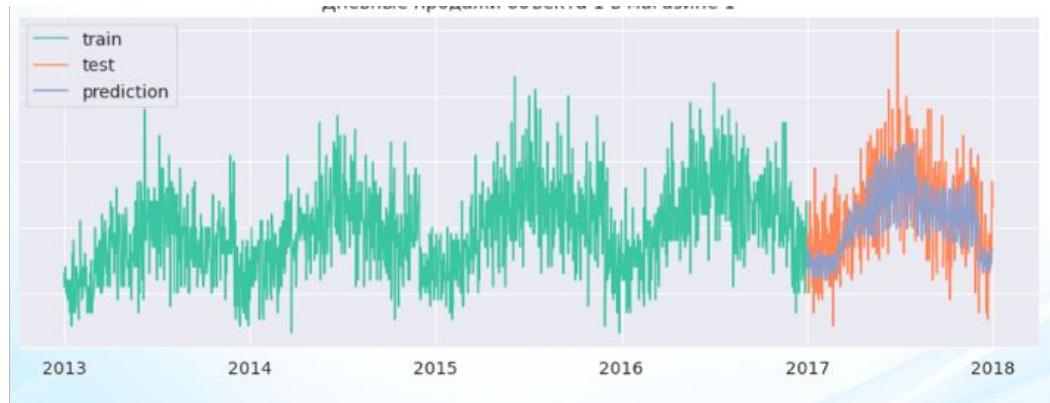
Задачи на  
временных рядах

Сегментация

Классификация

Кластеризация

# Прогнозирование



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

$y_t$  - исторические значения

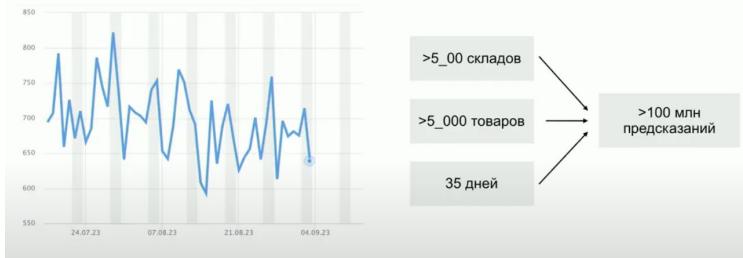
$X$  - признаки ряда

$t$  - время

Предсказываем значения величины в будущие моменты времени

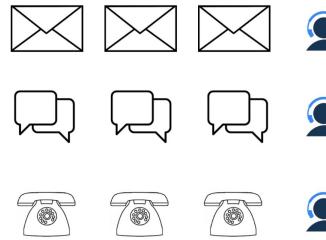
# Прогнозирование(примеры)

**Прогноз спроса** - Сколько будет продано товара X в магазине Y в день D

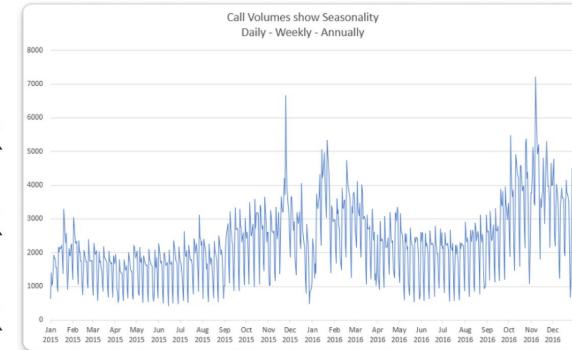


Прогнозируем спрос на товары  
чтобы планировать их закупки

**Работа линий поддержки**

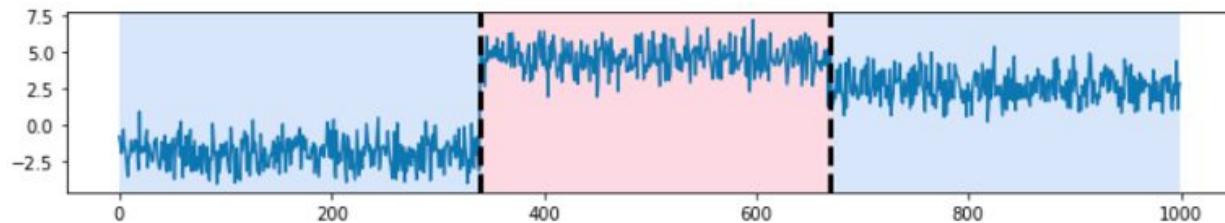


Call Volumes show Seasonality  
Daily - Weekly - Annually



Прогноз нагрузки на кол-центр для  
планирования смен сотрудников

# Сегментация



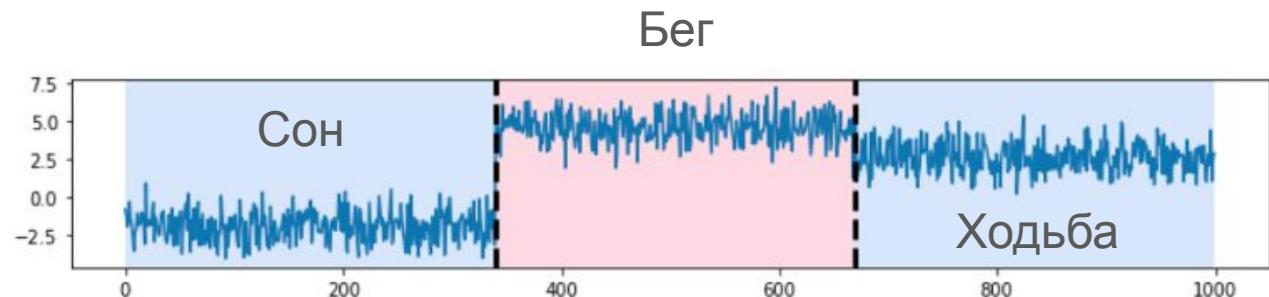
Разделяем временной ряд на отрезки  $y_1, y_2, \dots, y_n$  - временной ряд в рамках которых сохраняется  
консистентное поведение

$S_1, S_2, \dots, S_k$  - сегменты, содержащие последовательные точки  
временного ряда

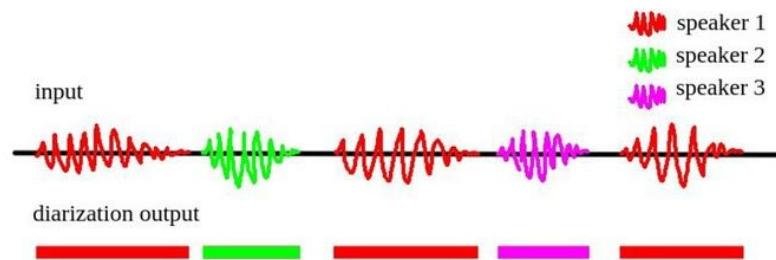
- $F[n, k] = \sum_{S_j \in S} E(S_j)$
- $E(S_j) = \sum_{y_i \in S_j} |y_i - c|$

# Сегментация(примеры)

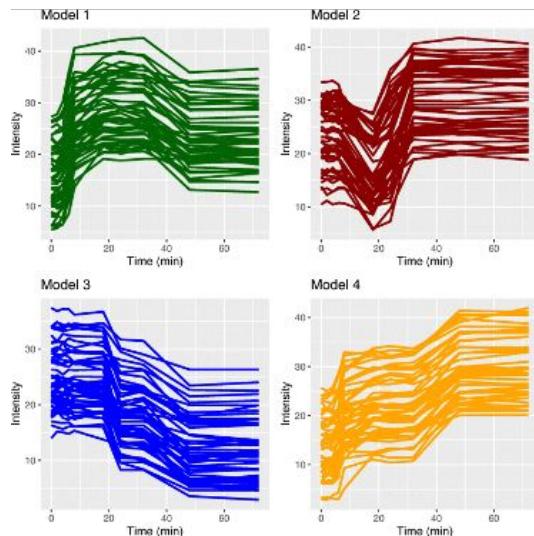
Сегментация по виду активности пользователя в смартчасах



Диаризация спикеров



# Классификация



Для каждого временного ряда  
определяем метку класса

$x_j = y_1, \dots, y_n$  - временной ряд

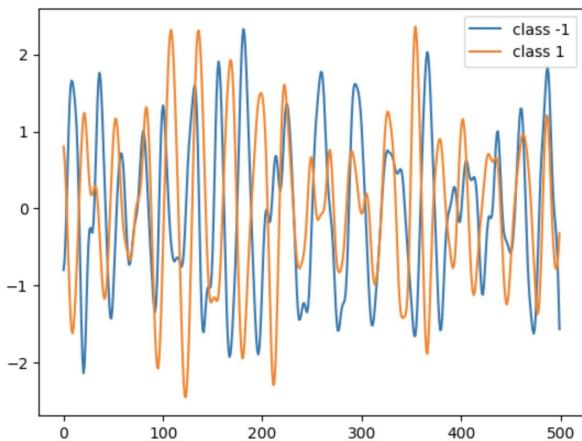
$x = \{x_j\}_{j=1}^N$  - множество временных рядов

$z_j$  - класс временного ряда

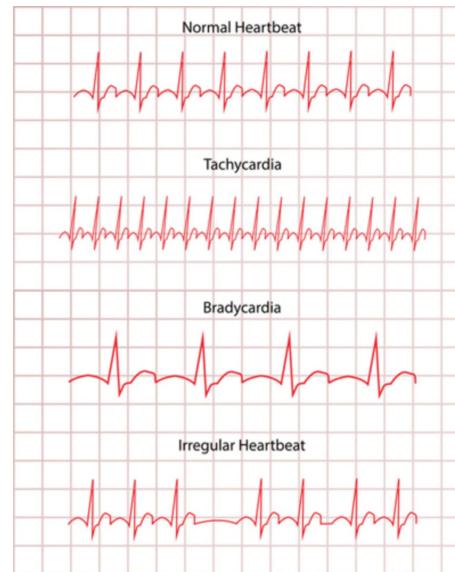
$z_j = F(x_j)$  - функция, определяющая класс для ряда  $x_j$

# Классификация - примеры

Определение неполадок с двигателем по звуку(FordA)



Определение заболевания по ЭКГ



# Паттерны временных рядов

- Тренд
  - Долговременное изменение уровня временного ряда
  - Может быть нелинейным
  - Может меняться со временем
- Сезонный
  - Регулярный повторяющийся паттерн с фиксированным периодом
- \*Циклический
  - Изменения в уровне временного ряда нефиксированного периода
  - Обычно связан с экономическими факторами
  - Уровень меняется в масштабе нескольких лет

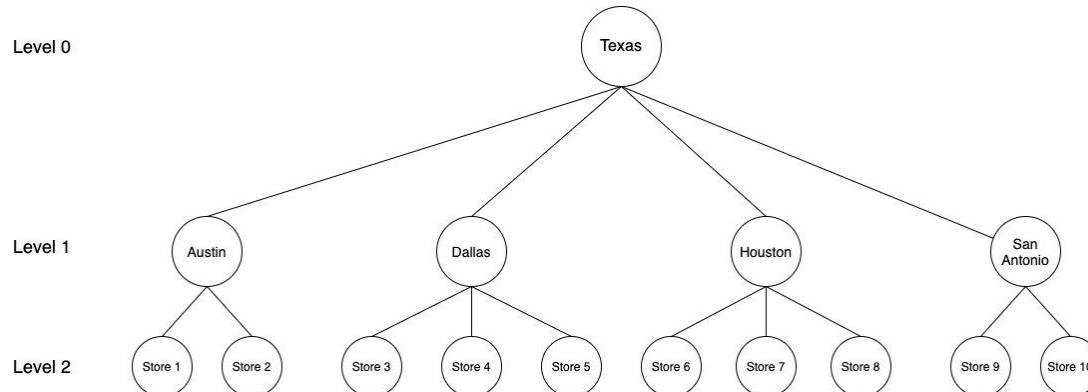
# Какими бывают

- Регулярные (равностоящие)
  - Равные промежутки между временными метками.
  - Пример: ежемесячные продажи вина.
- Нерегулярные (неравностоящие)
  - Неравные промежутки между временными метками.
  - Пример: время подготовки ответа на каждый запрос к серверу.
- Панельные данные
  - Совокупность временных рядов для одних и тех же временных меток
  - Пример: годовые бюджеты стран мира с 2010 по 2020 гг.

# Какими бывают

- Иерархические временные ряды

- Панельные данные, в которых некоторые ряды могут быть представлены, как сумма других рядов.
- *Пример: продажи товара по магазинам, которые могут быть сгруппированы до продаж по городам и совокупных продаж.*



Немного о EDA

# Что такое EDA и зачем это нужно?

EDA (exploratory data analysis) - один из начальных этапов в любом ML проекте, который позволяет получше познакомится с данными, выявить какие-то проблемные места, найти важные закономерности и подготовить данные для дальнейших шагов.

- Изучить, чем представлены данные
- Сбор статистики о данных
- Визуализация данных
- Поиск аномалий, выбросов
- Поиск связи между данными
- Предобработка данных (нормализация, перевод в нужные типы, заполнение пропусков)

# Первое знакомство с данными

segment	segment_0		segment_1		segment_2	
feature	feature	target	feature	target	feature	target
timestamp						
2020-01-01	1.624345	1.624345	-0.012665	-0.012665	-1.444114	-1.444114
2020-01-02	1.012589	1.012589	-1.129975	-1.129975	-1.948580	-1.948580
2020-01-03	0.484417	0.484417	-0.895559	-0.895559	-1.788543	-1.788543
2020-01-04	-0.588551	-0.588551	0.764243	0.764243	-0.912374	-0.912374
2020-01-05	0.276856	0.276856	1.506287	1.506287	-0.596739	-0.596739
2020-01-06	-2.024682	-2.024682	1.314452	1.314452	-2.618940	-2.618940

Какие сущности хранятся в данных?

Сколько пропущенных значений?

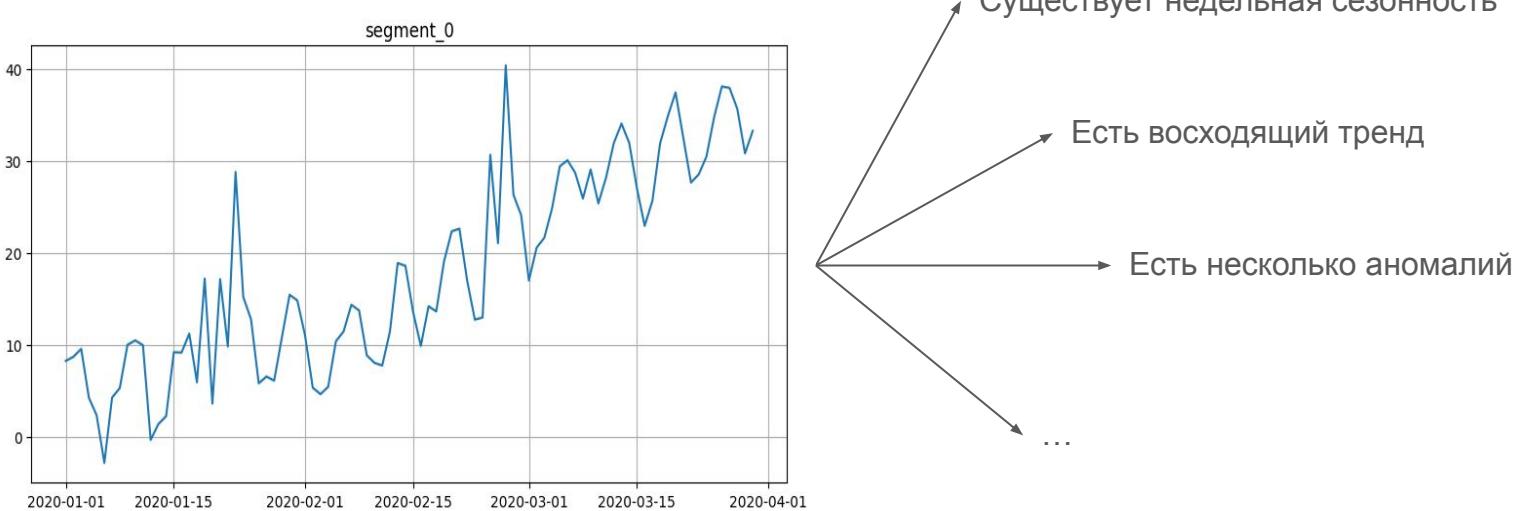
Максимальная и минимальная дата?

Какая частотность данных?

...

# Анализ временного ряда

## Визуализация ряда и первые наблюдения

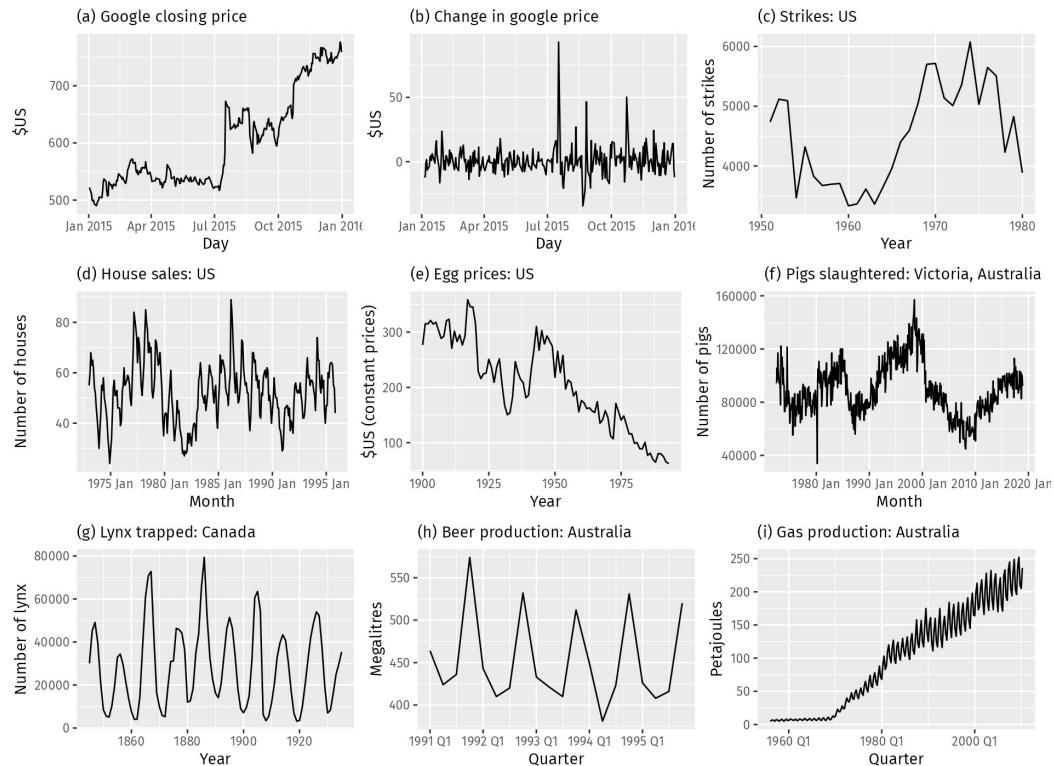


Замечание: Если рядов много, можно построить распределение статистик (среднее, дисперсия).

# Стационарность временного ряда

$\{y_t\}_{t=1}^T$  стационарный временной ряд, если:

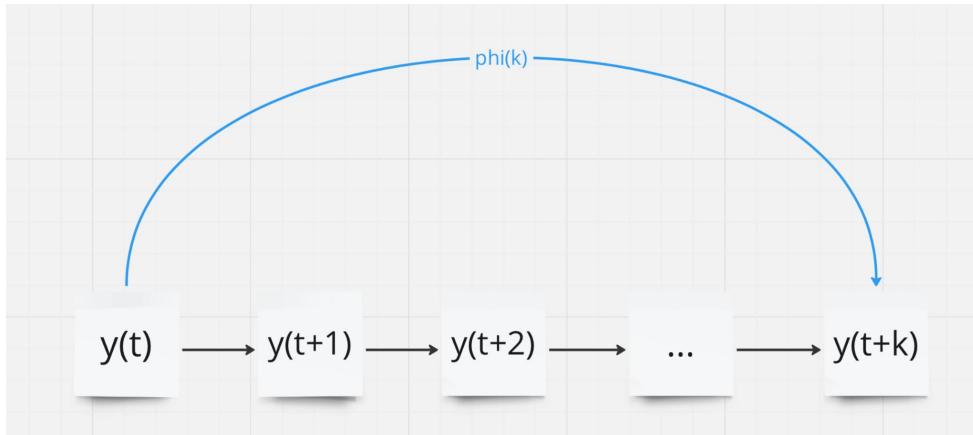
1.  $E[y_t] = \mu$
2.  $D[y_t] = \sigma^2$
3.  $Cov(y_t, y_{t+k}) = \rho(k)$



# ACF и PACF

ACF (autocorrelation function,  $p(k)$ ) - функция, отражающая зависимость между элементами отстоящими друг от друга на  $k$  по времени

PACF (partial autocorrelation function,  $\phi(k)$ ) - функция, отражающая зависимость между элементами отстоящими друг от друга на  $k$  по времени очищенную от влияния промежуточных величин



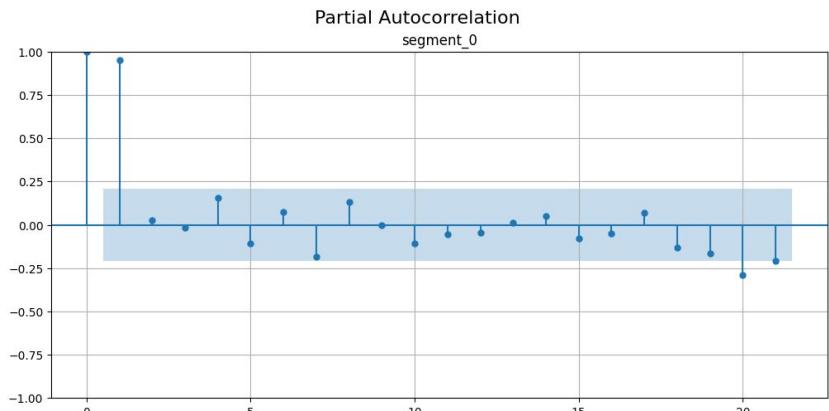
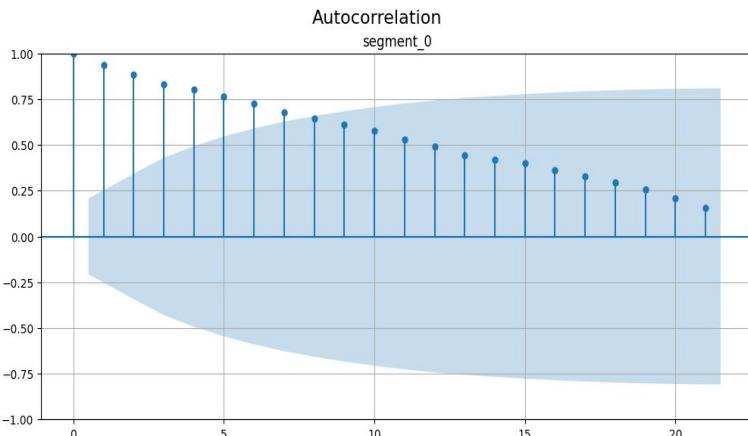
Подробнее можно почитать тут

- <https://en.wikipedia.org/wiki/Autocorrelation>
- [https://en.wikipedia.org/wiki/Partial\\_autocorrelation\\_function](https://en.wikipedia.org/wiki/Partial_autocorrelation_function)

# ACF и PACF

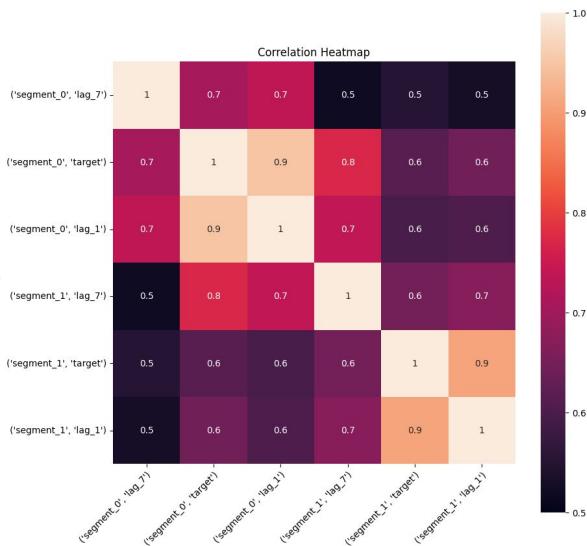
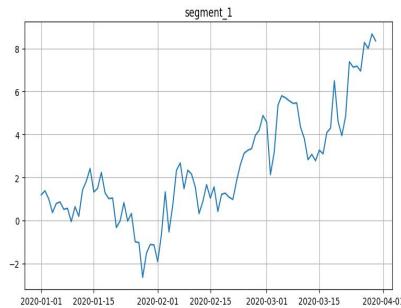
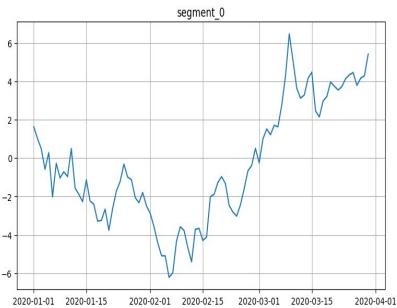
## Примеры визуализации

По данным графикам можно предположить, что мы имеем дело с процессом AR(2)



# Оценка связи рядов

Можно считать корреляцию между рядами (и их лагами), чтобы оценить связь между ними.



Замечания:

- корреляция улавливает только линейные зависимости
- из корреляции не следует, что ряды зависят/не зависят друг от друга

# Задача прогнозирования

# Постановка задачи прогнозирования

Прогнозирование — это прогнозирование будущих значений ряда.

Пусть имеем временной ряд  $((y_i, t_i))_{i=1}^T$ .

Хотим научиться строить оценки  $\hat{y}_{T+h}$  для  $y_{T+h}, h = 1, \dots, H$

Величина  $H$  — горизонт прогнозирования.

# Пример: прогноз прошлым годом

Данные: месячные продажи вина.

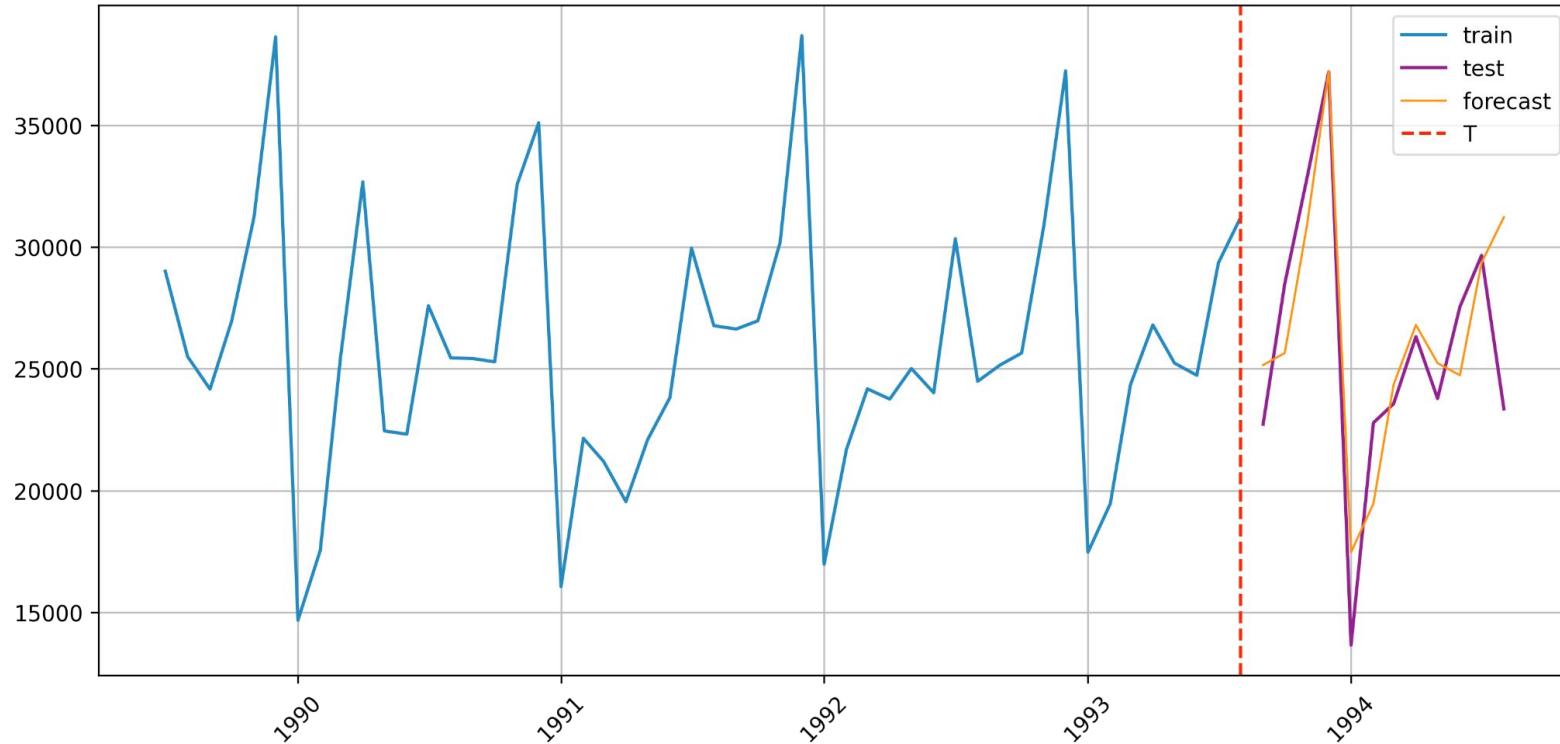
Возьмем  $T = 164$  и  $H = 12$ , хотим сделать прогноз на год вперед.

В качестве оценки возьмем значение за прошлый год, т.е.

$$\hat{y}_{T+h} = y_{T+h-H}, \quad h = 1, \dots, H$$

# Пример: данные для прогноза

main



# Построения прогноза

Хотим:  $\hat{y}_{T+1}, \dots, \hat{y}_{T+H}$

Построим функцию  $f$ , использующую **данные на момент времени  $T$** .

- Значения ряда:  $y_1, \dots, y_T$
- Временные метки:  $t_1, \dots, t_T$
- Будущие временные метки:  $t_{T+1}, \dots, t_{T+H}$

В нашем примере получим  $f$ :

$$f(y_1, \dots, y_T) = (y_{T-H+1}, \dots, y_T)$$

# Иные данные для прогноза

- Статические переменные
  - Для каждого временного ряда не меняются во времени
  - Примеры: прогноз годового дохода испытуемых
    - год рождения,
    - рост.
- Исторические экзогенные переменные (aka доп. данные)
  - Известны значения до момента времени T.
  - Пример: прогноз цены акций нефтедобывающей компании
    - курс доллара,
    - цена на нефть.

# Иные данные для прогноза

- Известные на будущее экзогенные переменные (aka регрессоры)
  - Известны значения до момента времени  $T+N$ .
  - Пример: дневной прогноз продаж мороженого
    - день месяца,
    - прогноз погоды.

# Метрики

# Ошибка прогноза

Ошибка прогноза - разность между наблюдениями из теста и спрогнозированными значениями.

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h}$$

Качество прогноза оценивается с помощью агрегации (анализа) ошибок прогноза разными способами.

# Абсолютные ошибки

- MAE, RMSE

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- Имеют масштаб исходных данных
- Не могут использоваться для сравнения прогнозов с разными единицами измерений (масштабом значений)

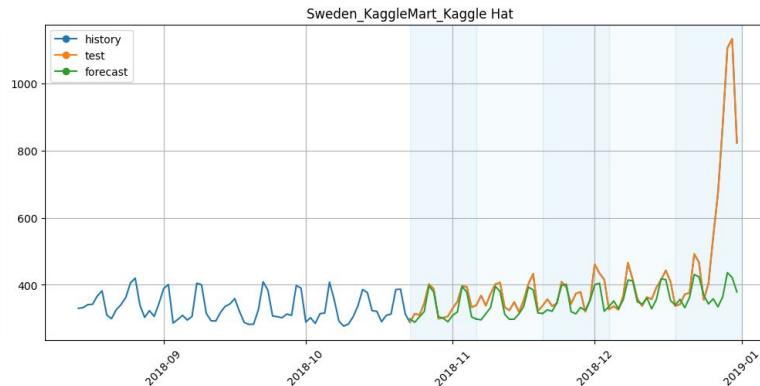
+

- Простота
- Широко используются
- Могут применяться к любому масштабу

-

- Не всегда сравнимы
- Сложность интерпретации (MSE/RMSE)

# Абсолютные ошибки



MAE: 15.89  
RMSE: 23.5

MAE: 62.6  
RMSE: 90.4

# Процентные ошибки

- MAPE, SMAPE, WAPE

$$MAPE(y, \hat{y}) = \frac{100}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i|}$$

$$SMAPE(y, \hat{y}) = \frac{2 * 100}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

$$WAPE(y, \hat{y}) = \frac{100}{n} \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i|}$$

- Не зависят от масштаба данных
- Могут быть использованы для сравнения
- Нестабильное поведение (деление на 0) в случае близости  $y_t$  к нулю

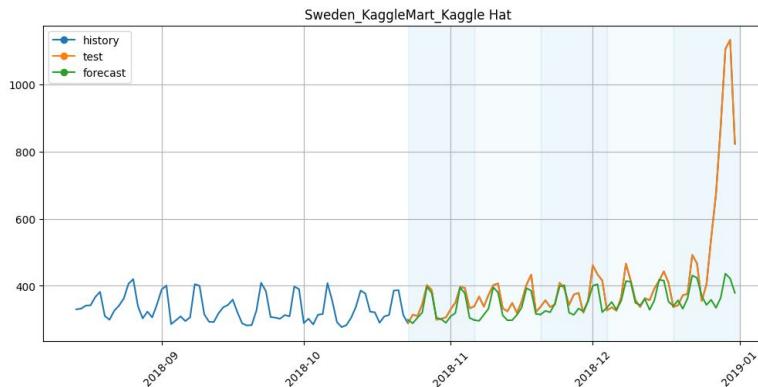
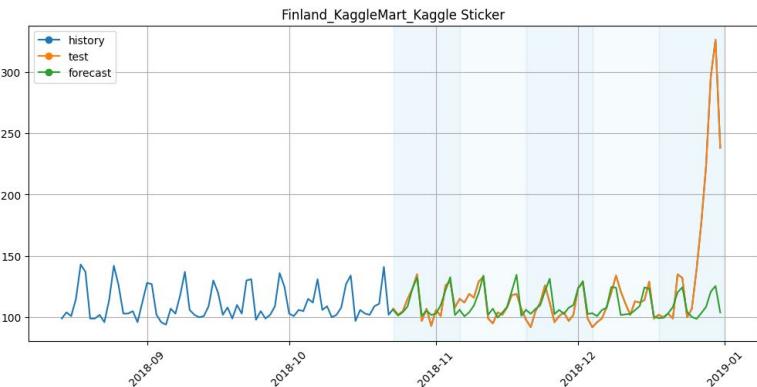
+

- Интерпретируемость
- Могут использоваться для сравнения

-

- Может возникать деление на 0
- Не симметричны относительно ошибок прогнозов

# Процентные ошибки



MAPE: 9.15

SMAPE: 10.66

WAPE: 0.111

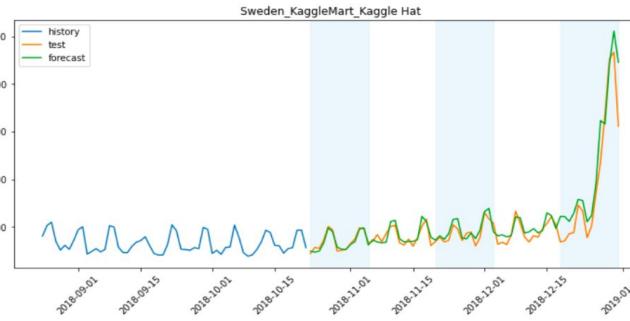
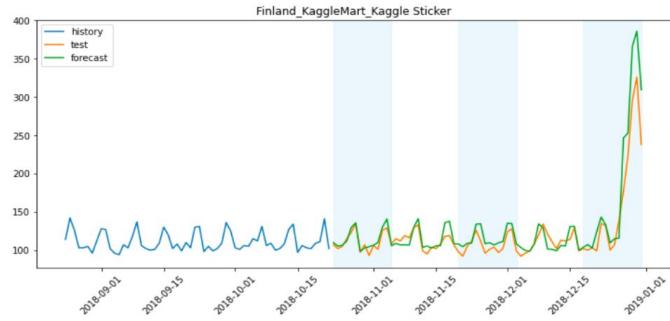
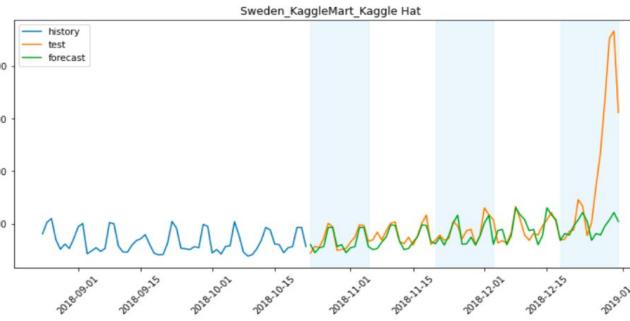
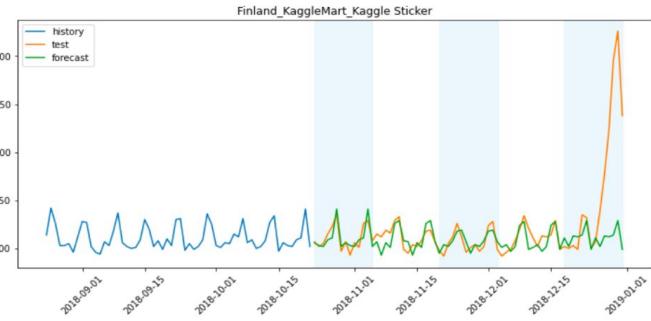
MAPE: 10.3

SMAPE: 12.4

WAPE: 0.125

# Процентные ошибки

## 11.5 vs 6.3 SMAPE



# Масштабированные ошибки

- MASE, rMAE

$$MASE(y, \hat{y}) = \frac{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|}{\frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}|}$$

$$rMAE(y, \hat{y}) = \frac{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|}{\bar{y}}$$

- Альтернатива процентным ошибкам
- Не зависят от масштаба
- Масштабирование на наивный прогноз ( $<1$  - лучше,  $>1$  - хуже)
- MASE: Требует in-sample данные. Для сравнения они должны совпадать
- rMAE: Масштабирует на out-of-sample (test)

+

- Могут использоваться для сравнения
- Интерпретируемость

-

- Неустойчивы вблизи 0
- Сложнее в расчете
- Могут требовать дополнительные данные
- Не ограничены сверху

# Итоги

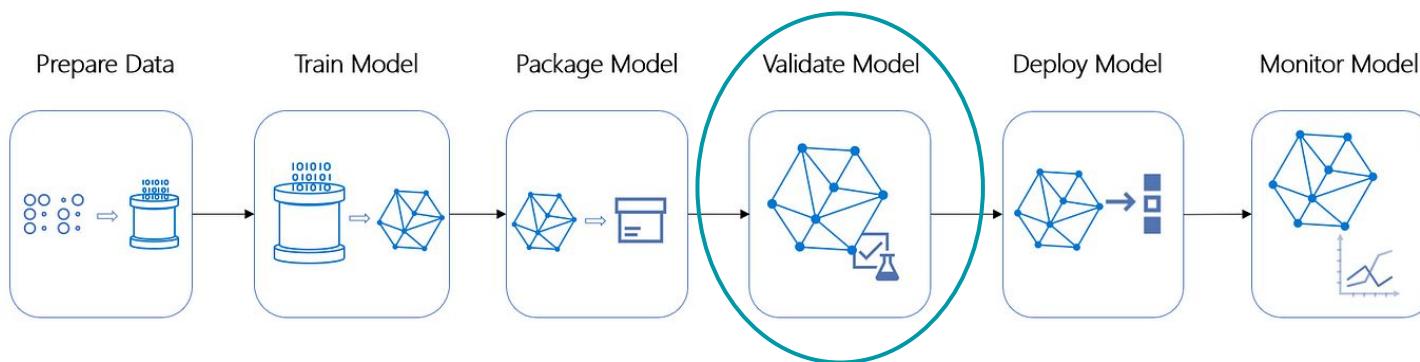
- Нет единой универсальной метрики
- Метрика выбирается под задачу
- От выбора метрики зависит выбор модели
- Если у вас набор временных рядов, то метрики можно сагрегировать

# Валидация на временных рядах

# Кросс-валидация

Цель: оценить качество алгоритма в offline

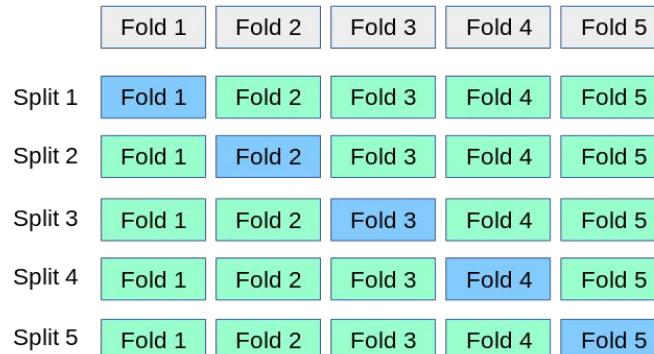
Основной принцип:  
максимально приближаем процедуру к проду



# Кросс-валидация

Табличные данные:

- IDD сэмплы -> разделяем на train и test случайно
- K-Fold – делим K раз, убираем зависимость от разделения



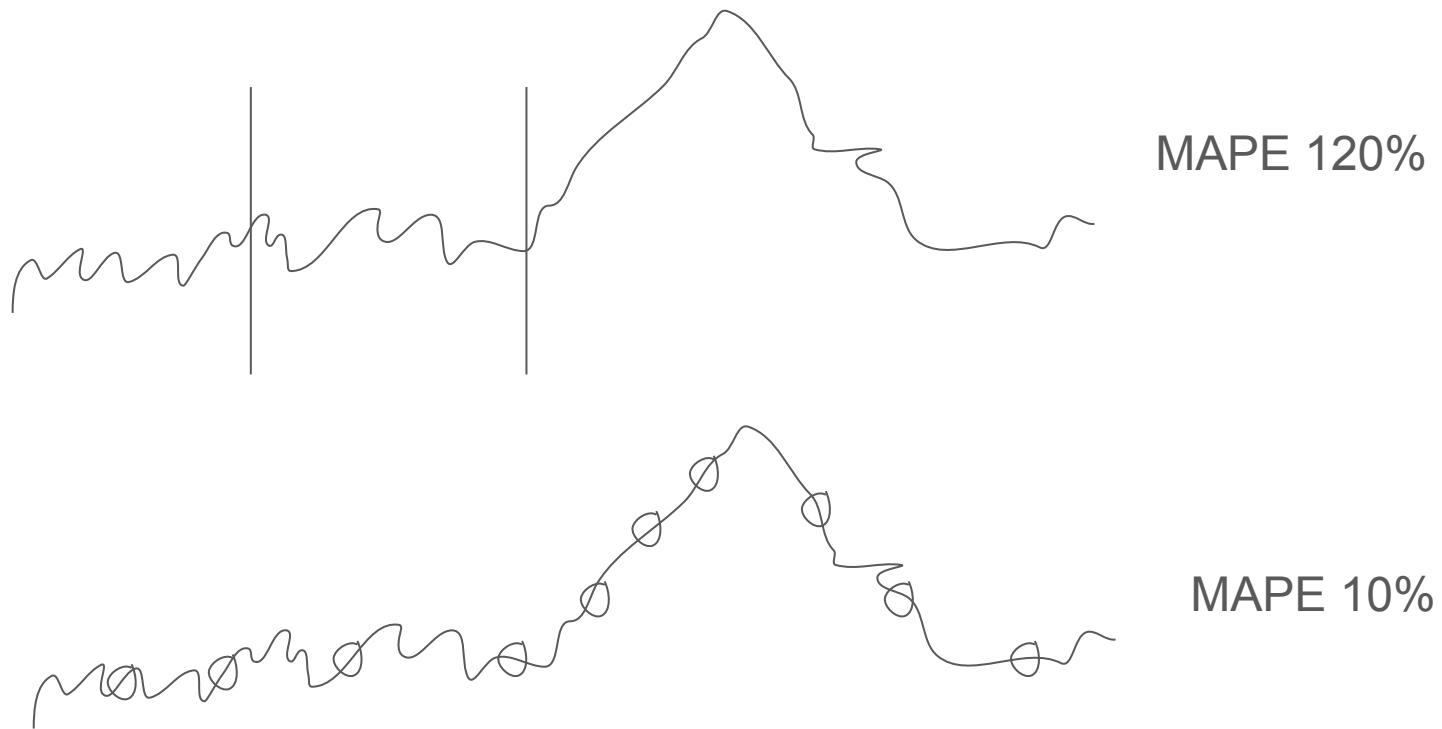
Fold 1

Обучение

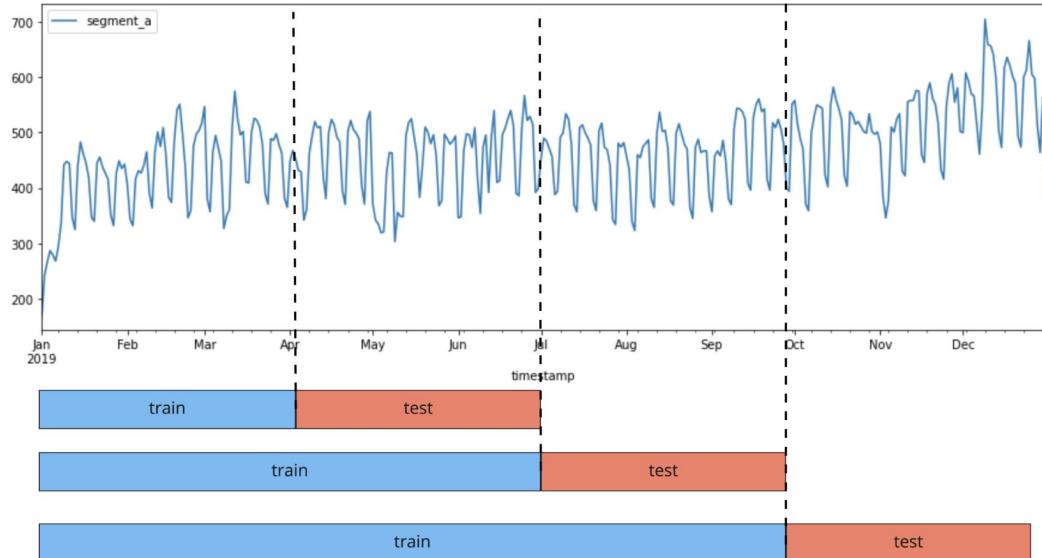
Fold 3

Оценка качества

# Чем плох K-Fold

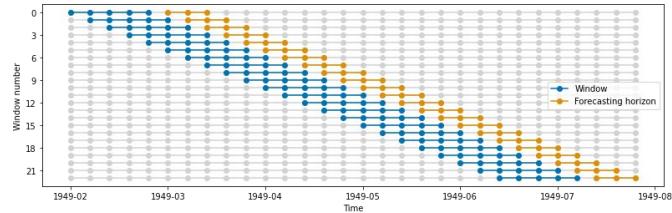


# Общий принцип

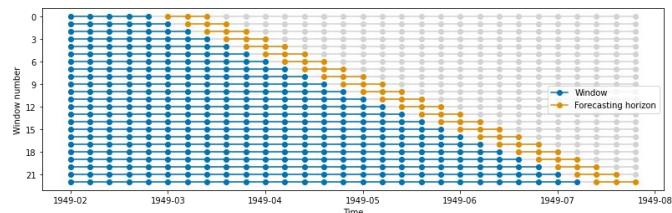


Разделяем на фолды  
по временной шкале

# Разбиение по времени



Constant window



Expanding window

- Распределение таргета меняется
- Доступны только свежие данные

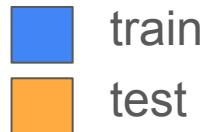
- Распределение таргета не меняется
- Чем больше данных, тем лучше

# Разбиение времени и сегментам

s1										
s2										
s3										
s4										
	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9

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

Temporal + Segment



\*Глобальная модель – одна модель, предсказывающая все сегменты

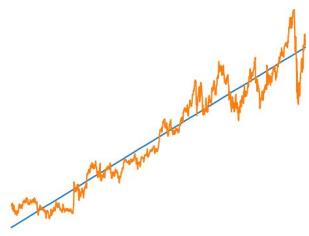
# Итоги

- Кросс-валидация нужна для оценки оценки качества моделей в offline
- Во временных рядах есть явная зависимость от времени – нельзя делать на фолды случайно
- Разделение данных можно делать
  - По временной шкале
  - По инстансам временных рядов
- Валидация настраивается под способ применения модели

# Преобразование таргета

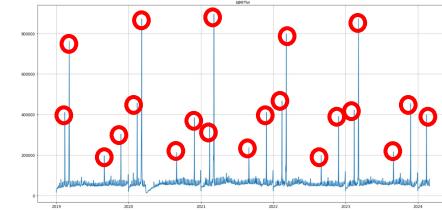
# Мотивация

Нестационарность

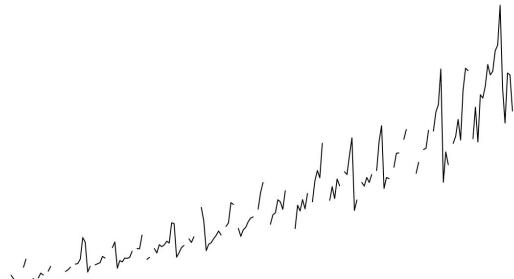


Цель: упростить структуру  
рядов для модели

Влияние внешних факторов

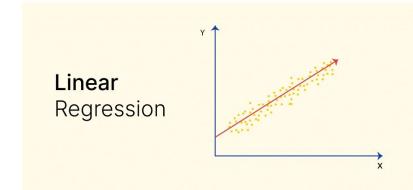


Аномалии сборки данных

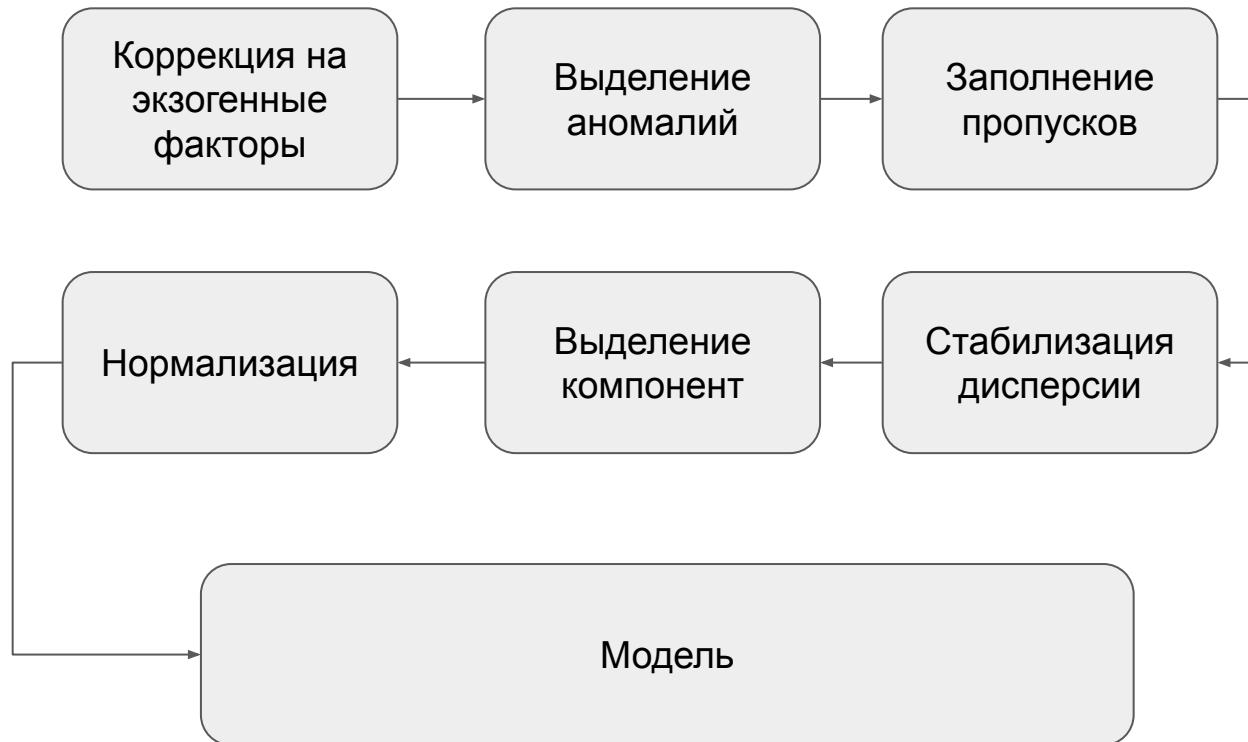


Особенности модели

$N(0, 1)$  →



# Виды преобразований



# Коррекция на экзогенные факторы

**Пример:** хотим спрогнозировать выручку сети супермаркетов на месяц вперед



От чего зависит выручка:

1. Количество покупателей( $n$ )
2. Количество дней в месяце( $t$ )
3. Цены на товары( $p$ )
4. Другие факторы( $r$ )

Идея: разложим сложную функцию на более простые зависимости:

$$\begin{aligned} F &= F(n, t, p, r) = \\ &= F(n) * F(t) * F(p) * F(r) \end{aligned}$$

# Коррекция на экзогенные факторы

Легко моделируется

$F(n, t, p, r)$  – сложная, непонятно как прогнозировать

- $F(n)$  – линейный рост в 10 раз
- $F(t)$  – каждый день зарабатываем примерно одинаково
- $F(p)$  – корректируются каждый год на годовую инфляцию
- $F(r)$  – календарь скидок и акций влияет на спрос

# Коррекция на экзогенные факторы

Календарь

Динамика пользователей

	январь(31)	февраль(28)	март(31)	апрель(30)	май(31)
Выручка за месяц	310	2800	31000	300000	3100000
Средняя выручка за день	10	100	1000	10000	100000
Среднедневная выручка на 1000 пользователей	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>

Количество покупателей	1000	10_000	100_000	1_000_000	10_000_000
------------------------	------	--------	---------	-----------	------------

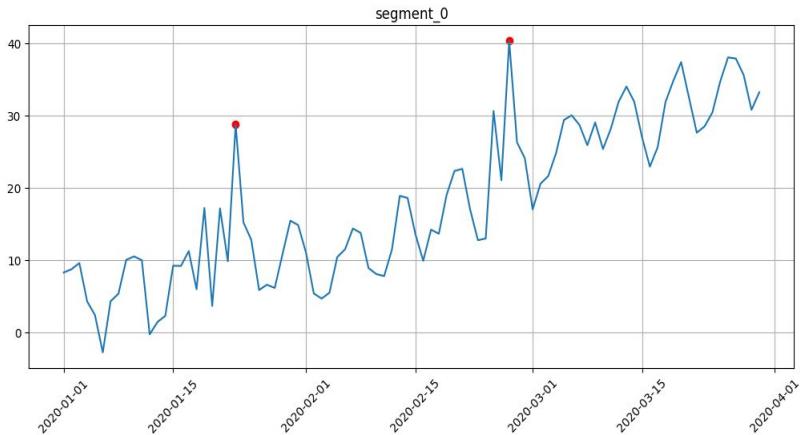
# Выделение аномалий

Аномалия - это отклонение или необычное поведение, которое не соответствует ожидаемому или типичному шаблону в данном временном ряду.

В рамках задачи прогнозирования требуется обнаружить аномальные значения и предобработать их, потому что они могут снизить качество прогнозных моделей

Примеры методов для обнаружения аномалий:

1. Медианный метод
2. Метод на основании плотности
3. Метод на доверительных интервалах



# Работа с пропущенными значениями

Варианты замены пропущенных значений в ряду:

1. Заменить константой (часто подходит нулевое значение)
2. Заменить последним имеющимся значением перед пропуском
3. Заменить первым имеющимся значением после пропуска
4. Заменить на глобальное/скользящее среднее/медиану



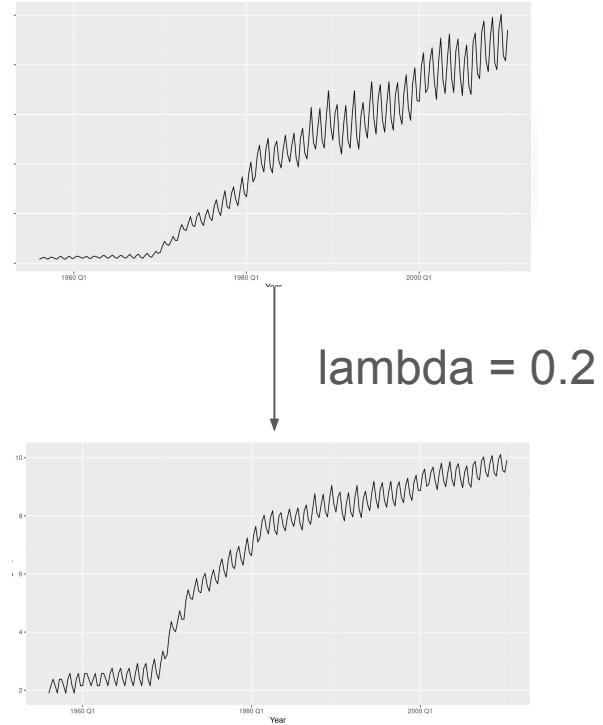
# Стабилизация дисперсии

Зачем: приводим  
данные к нормальному  
распределению

Пример: линейная  
регрессия предполагает  
константность  
дисперсии

Преобразование Бокса-Кокса

$$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ (\text{sign}(y_t)|y_t|^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$$



# Разложение ряда на компоненты

$y_t = S_t + T_t + R_t$  - аддитивная модель

$y_t = S_t * T_t * R_t$  - мультипликативная модель

$S_t$  - сезонная компонента

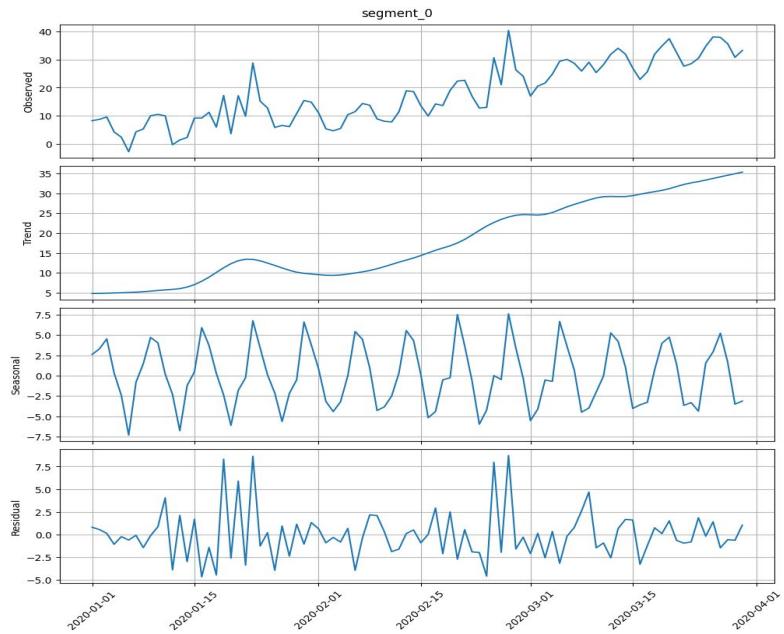
$T_t$  - трендовая компонента

$R_t$  - остаточная компонента

Зачем: может быть полезно для понимания ряда, можно сделать ряд стационарным

Некоторые виды разложений:

1. Выделение тренда регрессией
2. Выделение тренда с помощью MA
3. STL разложение (пример на картинке)



# Нормализация

Зачем: приводим данные к одному масштабу для глобальных моделей

Важно: делаем ряд стационарным перед нормализацией!

StandartScaler – приводим данные к  $N(0,1)$

$$X'_i = \frac{X_i - \mu}{\sigma} = \frac{X_i - X_{\text{mean}}}{X_{\text{std}}}$$

MinMaxScaler – проецируем данные в  $[0,1]$

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

MinMaxScaler аналогично (1) при наличии аномалий

$$X_{\text{scale}} = \frac{x_i - x_{\text{med}}}{x_{75} - x_{25}}$$

# Выводы

- Цель преобразования: подготовить данные для подачи в модель
- Чем больше эффектов мы учтем, тем проще будет сделать прогноз
- Эффектов может быть много
  - Внешние факторы
  - Аномалии и пропуски
  - Тренд и сезонность
  - ...

# Экспертные прогнозы

# Экспертные прогнозы

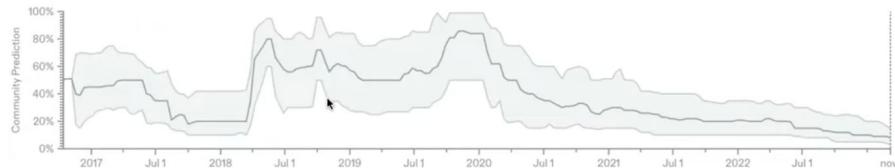
Предсказания толпы: Wisdom of the Crowd

SpaceX Lands People on Mars by 2030

Created by Greg • Opened Oct 13, 2016 • Resolves Jan 2, 2030

^ 884 v Open Closes Jan 1, 2030

▼ Forecast Timeline



Total Predictions

6.24k

Community Prediction 8%

Total Forecasters

3795

Community



- 2018 - запуск Tesla в космом, 2019-2020 - пилотируемые запуски к МКС и успешные испытания Starship в 2020

## Rule based или heuristic based подходы

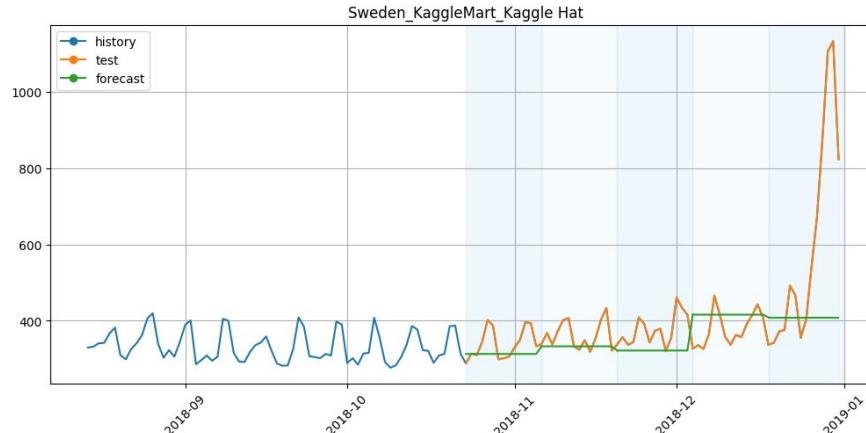
- Данных очень мало
- Эксперты и их знания, которые сложно переложить на модели - такое часто встречается в анализе спроса или финансовом моделировании
- Простые процессы, которые легко формализовать через `if else` правила

```
if timestamp in night:  
    return 0  
elif timestamp in morning:  
    return 5  
elif timestamp in day:  
    return 20  
elif timestamp in evening:  
    return 10  
else:  
    raise ValueError("Unknown timestamp")
```

# Статистические подходы

# Простое экспоненциальное сглаживание

- Наивный прогноз (хороший бэйзлайн)  $\hat{y}_{T+h} = y_T$
- Прогноз средним  $\hat{y}_{T+h} = \frac{1}{T} \sum_{t=1}^T y_t$
- Подходит для моделирования рядов без ярко выраженных тренда либо сезонности



# Простое экспоненциальное сглаживание

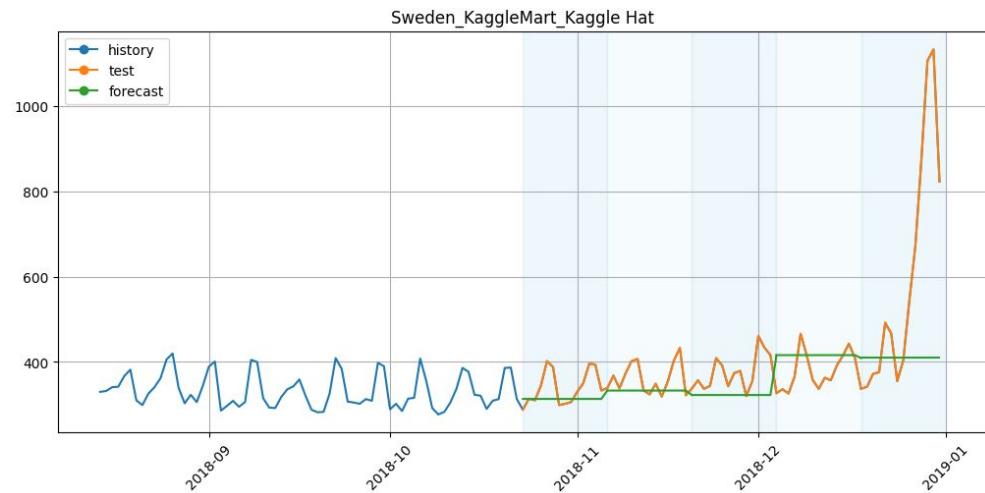
- Затухание веса наблюдения с течением времени
- Параметр  $\alpha \in [0, 1]$ 
  - Меньшие значения дают больший вес более далеким наблюдениям
  - Большие увеличивают влияние более поздних наблюдений
- Форма взвешенного среднего

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$$

- Компонентная форма

Forecast equation  $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation  $\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$ ,



# Модель Хольта

- Расширяет простую модель экспоненциального сглаживания с помощью компоненты тренда
- Подходит для моделирования рядов с ярко выраженным трендом

Forecast equation       $\hat{y}_{t+h|t} = \ell_t + h b_t$

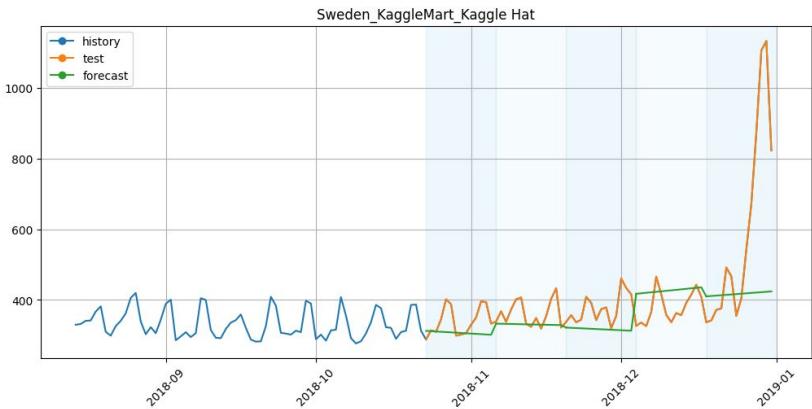
Level equation       $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$

Trend equation       $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1},$

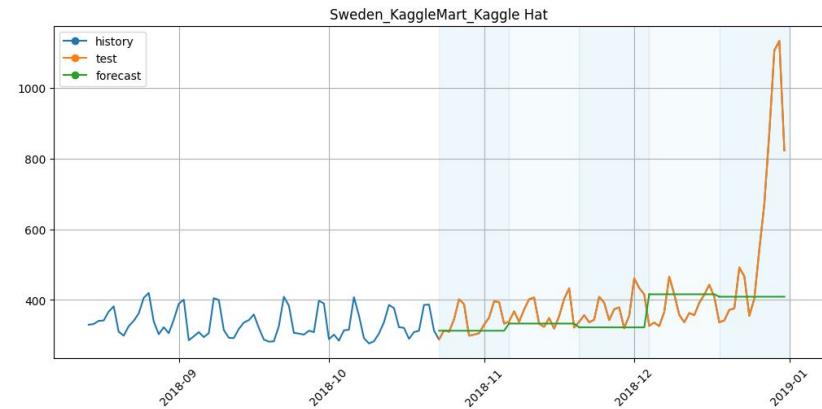
$$\beta^* \in [0, 1]$$

- Тренд растет неограниченно, можно использовать затухание

# Модель Хольта



Растущий тренд



Затухающий тренд

# Модель Хольта - Винтерса

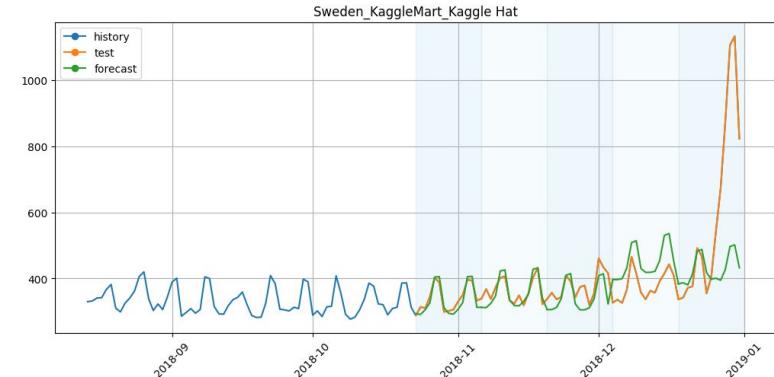
- Модификация модели Хольта с помощью включения сезонной компоненты
- Подходит для моделирования рядов с трендом и сезонностью
- Период сезонности устанавливается относительно частотности ряда

$$\hat{y}_{t+h|t} = \ell_t + h b_t + s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m},$$



# ETS

- ETS(Error, Trend, Seasonal) - обобщение моделей экспоненциального сглаживания
  - Error
    - $A$  - аддитивные
    - $M$  - мультипликативные
  - Trend
    - $N$  - отсутствует
    - $A$  - аддитивный
    - $A_d$  - затухающий
  - Seasonal
    - $N$  - отсутствует
    - $A$  - аддитивная
    - $M$  - мультипликативная
- Максимизация правдоподобия
- Автоматический подбор компонент на основе информационных статистик

# ETS

- ETS(A,N,N) - простое экспоненциальное сглаживание

$$\begin{array}{ll} \text{Forecast equation} & \hat{y}_{t+1|t} = \ell_t \\ \text{Smoothing equation} & \ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}, \end{array} \xrightarrow{\hspace{10em}} \begin{array}{l} y_t = \ell_{t-1} + \varepsilon_t \\ \ell_t = \ell_{t-1} + \alpha \varepsilon_t. \end{array}$$

- ETS(A, A, N) - модель Хольта

$$\begin{array}{ll} \text{Forecast equation} & \hat{y}_{t+h|t} = \ell_t + h b_t \\ \text{Level equation} & \ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}, \end{array} \xrightarrow{\hspace{10em}} \begin{array}{l} y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t \\ \ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t \\ b_t = b_{t-1} + \beta \varepsilon_t, \end{array}$$

# ETS

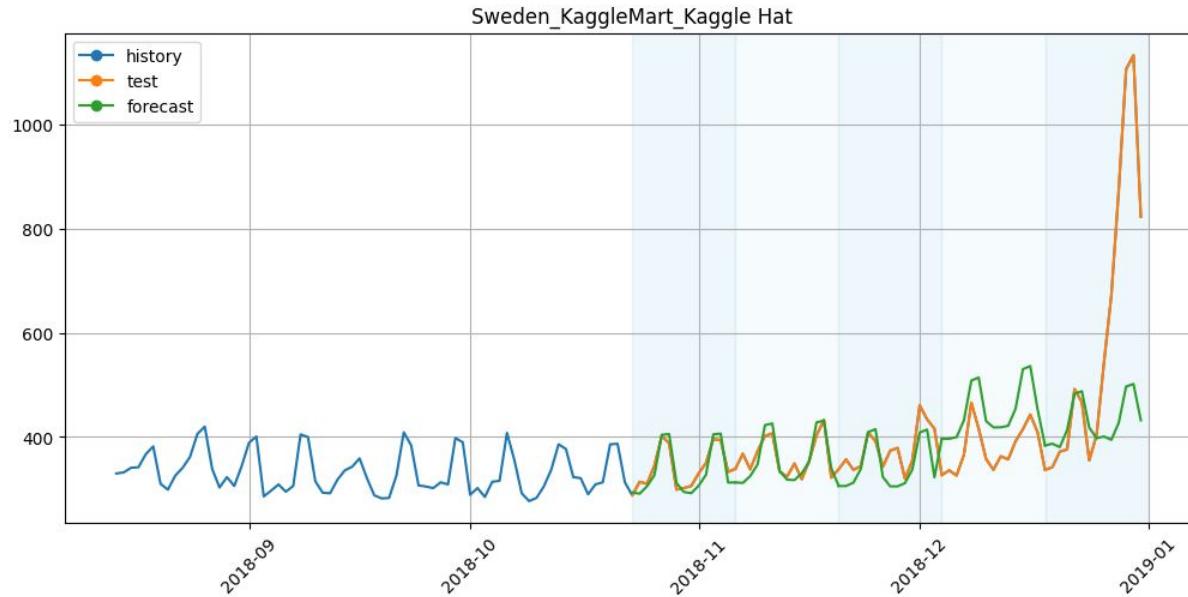
## ADDITIVE ERROR MODELS

Trend		Seasonal		
	N	A	M	
<b>N</b>	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = \ell_{t-1} s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / \ell_{t-1}$	
<b>A</b>	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + b_{t-1})$	
<b>A<sub>d</sub></b>	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = \phi b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + \phi b_{t-1})$	

## MULTIPLICATIVE ERROR MODELS

Trend		Seasonal		
	N	A	M	
<b>N</b>	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha \varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1} s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha \varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma \varepsilon_t)$	
<b>A</b>	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha \varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha \varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma \varepsilon_t)$	
<b>A<sub>d</sub></b>	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha \varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1}) s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha \varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma \varepsilon_t)$	

# ETS



ETS(A, A, A)

# Авторегрессионная модель

- AR(p) - линейная комбинация предыдущих значений
- p - параметр, количество предыдущих значений

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

- Подходит для моделирования рядов случайного блуждания со смещением и без

$$y_t = y_{t-1} + \varepsilon_t$$

$$y_t = c + y_{t-1} + \varepsilon_t$$

# Модель скользящего среднего

- MA(q) - линейная комбинация предыдущих ошибок
- q - параметр, количество предыдущих ошибок

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q},$$

- Отличие от скользящего сглаживания: прогнозирует будущие значения (не константный прогноз), а не выделяет циклы/тренды в исторических наблюдениях

## ARIMA

- ARIMA(p,d,q) - авторегрессионное интегрированное скользящее среднее
- p - порядок AR, q - порядок MA, d - порядок дифференцирования
- ARIMA(p,1,q) -  $y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$
- ARIMA(p, d, q) в компактной форме:

$$(1 - \phi_1 B - \cdots - \phi_p B^p) (1 - B)^d y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

# ARIMA

- SARIMA(p, d, q, P, D, Q, m) - сезонность учитывается в модели напрямую с помощью дополнительного сезонного ARIMA процесса
- 
- SARIMA(1,1,1)(1, 1, 1) с периодом сезонности 4:

$$(1 - \phi_1 B) (1 - \Phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B) (1 + \Theta_1 B^4)\varepsilon_t$$

- SARIMAX - использование экзогенных переменных

$$y_t = \beta_t x_t + u_t$$
$$\phi_p(L)\tilde{\phi}_P(L^s)\Delta^d\Delta_s^D u_t = A(t) + \theta_q(L)\tilde{\theta}_Q(L^s)\epsilon_t$$

- Метод максимального правдоподобия
- AutoARIMA - автоматическая оценка порядка модели с помощью информационных статистик

# Информационные критерии

- Akaike's Information Criterion (AIC)

$$AIC = 2k - 2\ln(L)$$

$L$  - значение логарифмической функции правдоподобия

$k$  - количество параметров модели

- Скорректированный AIC

$$AIC_c = AIC + \frac{2k^2 + 2k}{n - k - 1}$$

$n$  - число примеров в обучающей выборке

- Байесовский информационный критерий

$$BIC = k\ln(n) - 2\ln(L)$$

# Prophet

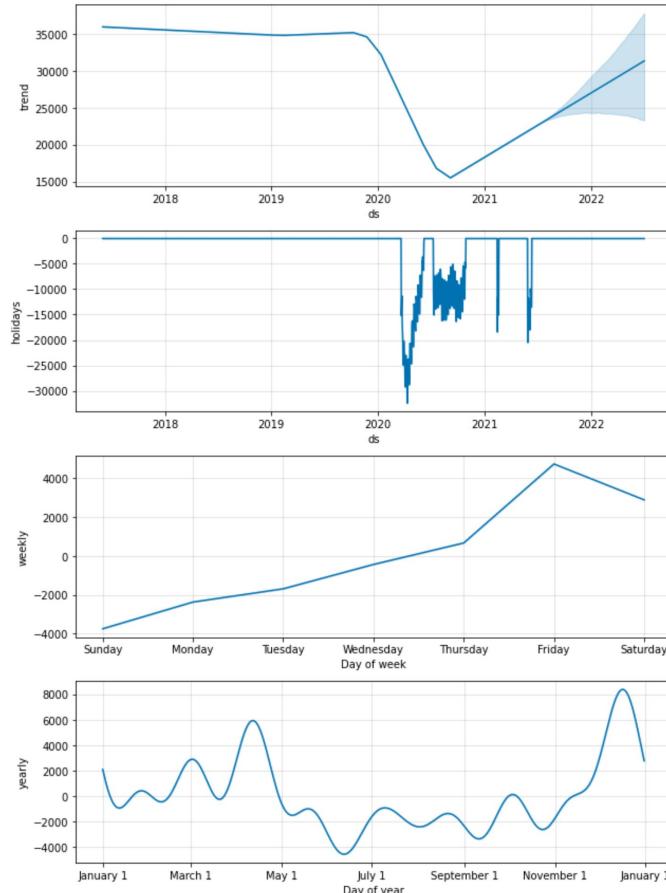
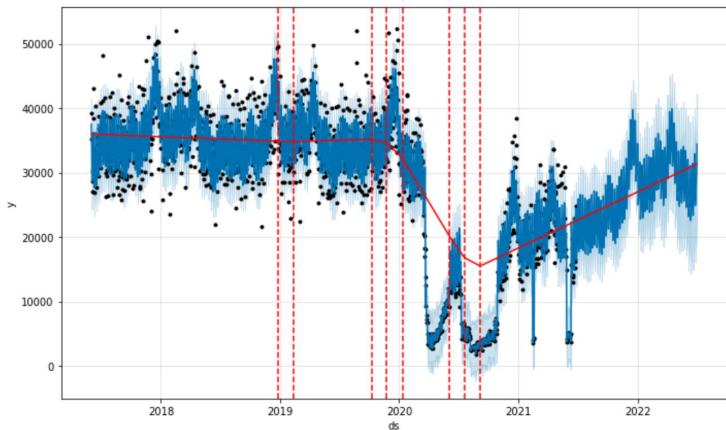
- Моделирование ряда с помощью аддитивных компонент: тренд, сезонность, эффекты праздников

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

- Поддерживает пропуски (не требует интерполяции)
- Автоматическое (и ручное) выделение точек смены тренда
- Модель тренда:
  - Кусочно линейный
  - Логистический рост

# Prophet

- Сезонность - моделирование периодичности с помощью ряда Фурье
- Моделирование праздников и событий
- Возможность добавления экзогенных переменных



# TBATS

BATS - Box-Cox transform, ARMA errors, Trend, and Seasonal components  
(Box-Cox + аддитивная ETS с различными периодами сезонности + ARMA errors)

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^\omega - 1}{\omega}; & \omega \neq 0 \\ \log y_t & \omega = 0 \end{cases}$$

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t$$

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t,$$

- Применима только к положительным данным
- Есть возможность учитывать несколько сезонных процессов разной периодичности
- Учитывает возможную корреляцию ошибок с помощью ARMA
- Не учитывает не целочисленные периоды сезонности

# TBATS

TBATS - тригонометрическая сезонность + BATS

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-1}^{(i)} + d_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t$$

$$s_{j,t}^{*(i)} = -s_{j,t-1} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t,$$

- Моделирование сезонности с помощью Фурье
- Может моделировать не целочисленные периоды
- Учитывает нелинейные эффекты

# Выводы

- Стат. методы могут дать довольно высокое качество прогнозов, стоит попробовать
- У стат. методов более строгие предпосылки к данным
- Довольно много хороших реализаций на python

# Временные ряды как табличный ML

# Задача прогнозирования временные ряды как табличные данные

Идея: почему бы не применить к временным рядам те же методы, что и к табличным данным, с которыми мы умеем работать?

Основные вопросы:

- Как строить прогнозы?
- Сколько моделей учить?
- Какие признаки можно построить?

# Как строим прогнозы?

## Стратегии многошагового прогнозирования

Пусть горизонт прогнозирования  $H$

Если модель предсказывает скаляр:

- Рекурсивная стратегия
- Прямая стратегия
- Гибридная стратегия

Если модель предсказывает вектор: можем сразу вернуть вектор.

# Рекурсивная стратегия

- Обучение
  - Учимся предсказывать на 1 шаг вперед
- Предсказание
  - Предсказываем всегда только 1 шаг вперед
  - В последующих итерациях используем предсказанные значения, как истинные

day	lag_5	lag_4	lag_3	lag_2	lag_1	pred
11	5	10	12	12	9	$y\_1$
12	10	12	12	9	$y\_1$	$y\_2$
13	12	12	9	$y\_1$	$y\_2$	$y\_3$

# Рекурсивная стратегия

## Преимущества

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

## Недостатки

- Происходит накопление ошибок во время прогнозирования
- Может быть вычислительно затратным - пересчитываем признаки на каждом шагу с учетом предсказанных наблюдений

# Прямая стратегия

- Обучение
  - Учим свою модель на каждую точку горизонта:  $N$  моделей
- Предсказание
  - Используем свою модель для прогнозирования каждой точки горизонта

day	lag_5	lag_4	lag_3	pred
11	5	10	12	$y_1$
12	10	12	12	$y_2$
13	12	12	9	$y_3$

Пример данных для модели 3-го шага

# Прямая стратегия

## Преимущества

- Нет накопления ошибок во время прогнозирования
- Можем использовать больше лагов

## Недостатки

- Прогнозы получаются независимо
- Требуется обучать много моделей
- Не можем выйти за горизонт без хаков

# Упрощенная прямая стратегия

- Обучение
  - Учим одну модель на точку Н горизонта
- Предсказание
  - Используем одну и ту же модель для каждой точки горизонта

day	lag_5	lag_4	lag_3	pred
11	5	10	12	y_1
12	10	12	12	y_2
13	12	12	9	y_3

Пример данных для модели 3-го шага

# Упрощенная прямая стратегия

## Преимущества

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

## Недостатки

- Недоиспользование данных
- Можем прогнозировать только на заданный горизонт

# Гибридная стратегия

- Обучение
  - Учим свою модель на каждую точку горизонта:  $N$  моделей
  - Каждая следующая модель принимает на вход предсказания предыдущих
- Предсказание
  - Последовательно получаем предсказания и сохраняем их, чтобы передать в следующие модели

# Гибридная стратегия

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

pred
y_1

Данные для предсказания моделью 1

day	lag_5	lag_4	lag_3	lag_2	lag_1
12	10	12	12	9	y_1

pred
y_2

Данные для предсказания моделью 2

day	lag_5	lag_4	lag_3	lag_2	lag_1
10	12	12	9	y_1	y_2

pred
y_3

Данные для предсказания моделью 3

# Гибридная стратегия

## Преимущества

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

## Недостатки

- Сложность реализации
- Требуется обучать много моделей

# Прогнозирование панельных данных

На практике часто имеем панельные данные, где много рядов.

Примеры:

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

# Прогнозирование панельных данных

## Подходы

- Локальное моделирование
  - Отдельная модель на каждый временной 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 [1]
- 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

# Глобальные модели

Достоинства:

- Быстрее обучаются
- Удобно использовать в production
- Тolerантны к длине рядов и к их качеству
- Возможный перенос знаний между рядами
- Можно использовать для прогнозирования ряда, который не был в обучающей выборке - проблема холодного старта

Недостатки:

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

# Что работает и что выбрать

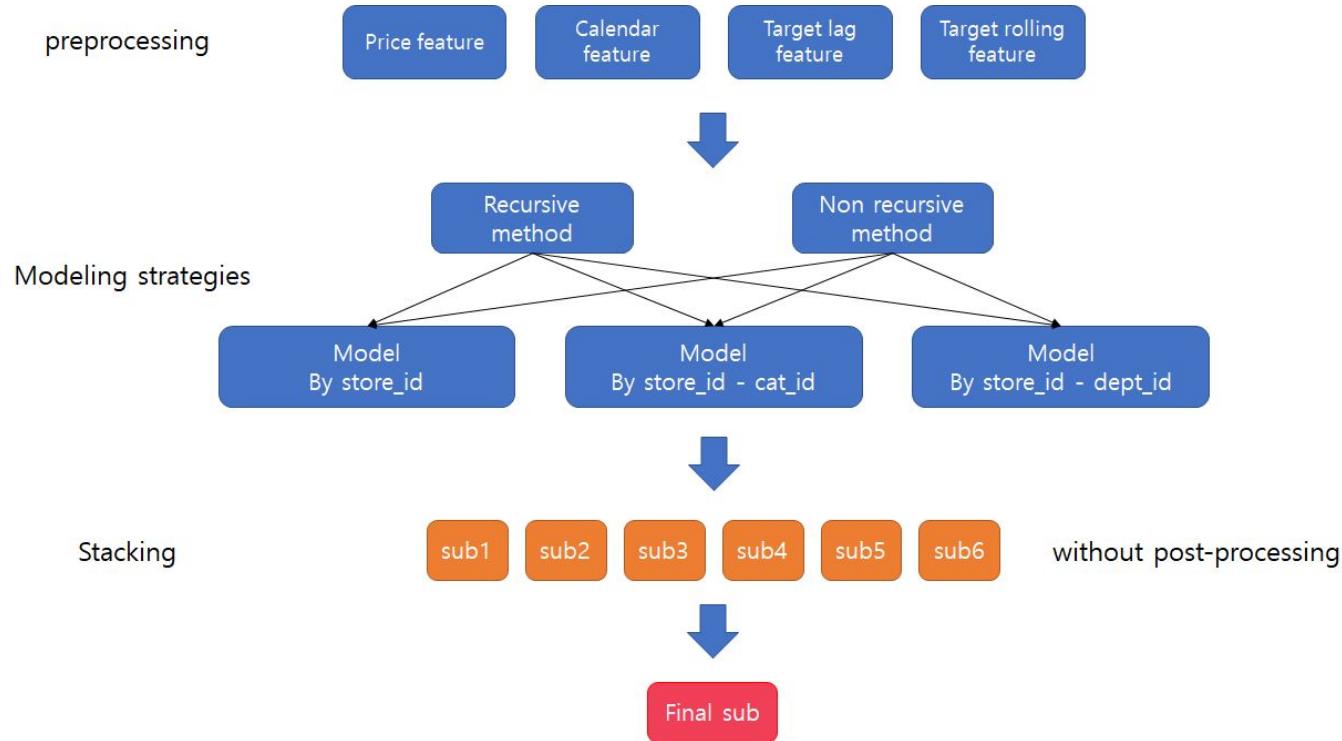
Если участвуете в соревнованиях:

- Имеет смысл ансамблировать несколько подходов - это скорее всего позволит выбрать максимум на private board

Если вы работаете работу:

- В случае классических подходов - выбор понятен - локальные модели
- В случае табличного ML - смотрим на глобальные подходы

# Соревнование M5



# Обзор признаков

Признаки из  
временной метки

Признаки из истории  
ряда

Кодирование  
категориальных  
признаков

Описательные  
признаки ряда

Примеры домено-  
специфичных  
признаков

## Ключевой принцип

## Не подглядывай в будущее

## Используем известные данные



0 1 1 1 0 0 1 1 1 1 1

## Прошлое

## Будущее



## Календарь промо

# Ключевой принцип

Используем прошлые значения



target

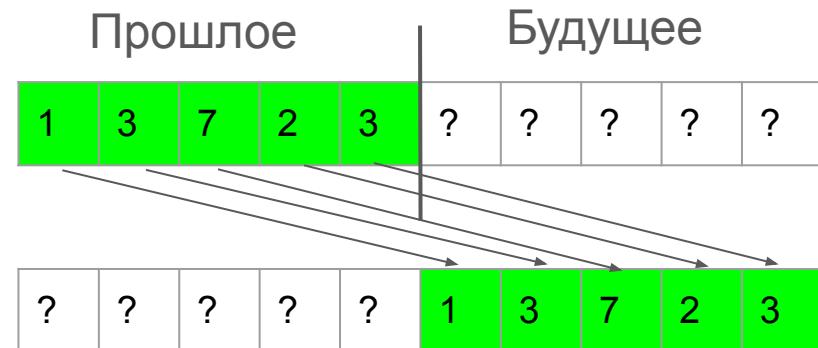


lag\_5

Не подглядывай в будущее



Цена на нефть -> Цена на нефть 5 дней назад



# Ключевой принцип

Не подглядывай в будущее

Используем  
прогноз



Погода -> Прогноз погоды



Погода



Прогноз  
погоды

Прошлое

1	3	7	2	3	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---

Будущее

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



Синоптики

# Признаки из временной метки

**Зачем:** моделируем  
сезонные паттерны

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
...	...	...

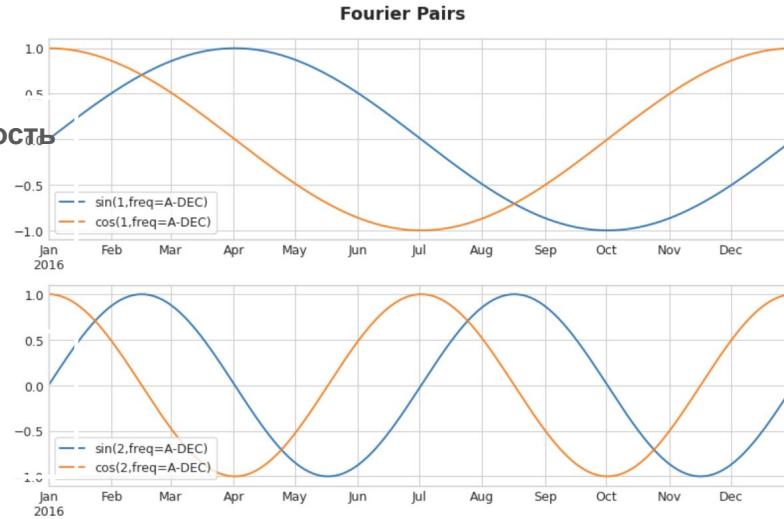
**Флаги из даты:** час дня, день  
недели, номер месяца, ...

# Признаки из временной метки

**Зачем:** моделируем  
длинные сезонные  
паттерны

Годовая сезонность  
(1 раз в год)

Полугодовая  
сезонность  
(2 раза в год)

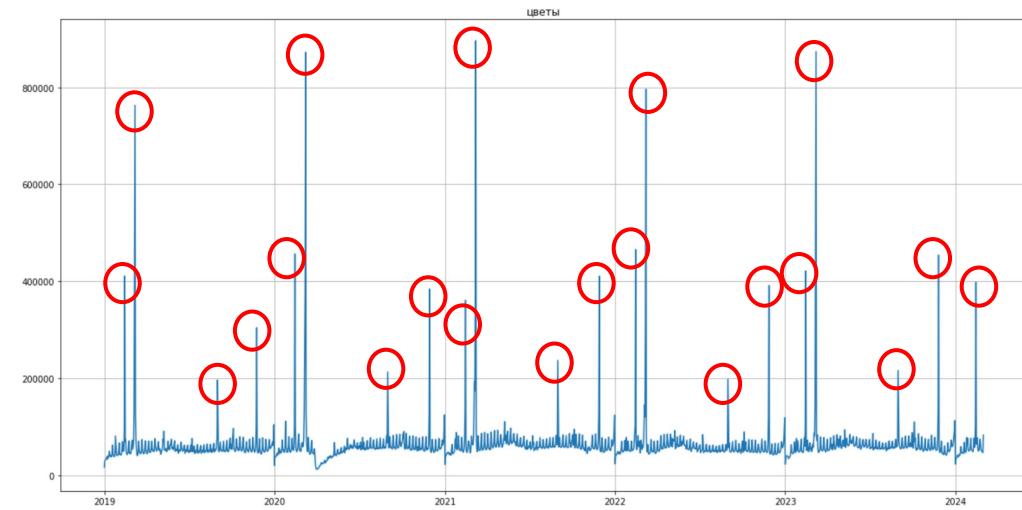


Признаки Фурье

# Признаки из временной метки

Зачем: моделируем  
особые даты

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



Флаги праздников

# 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, ...

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
...	...	...

Признаки Фурье

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
...	...	...	...	...	...	...

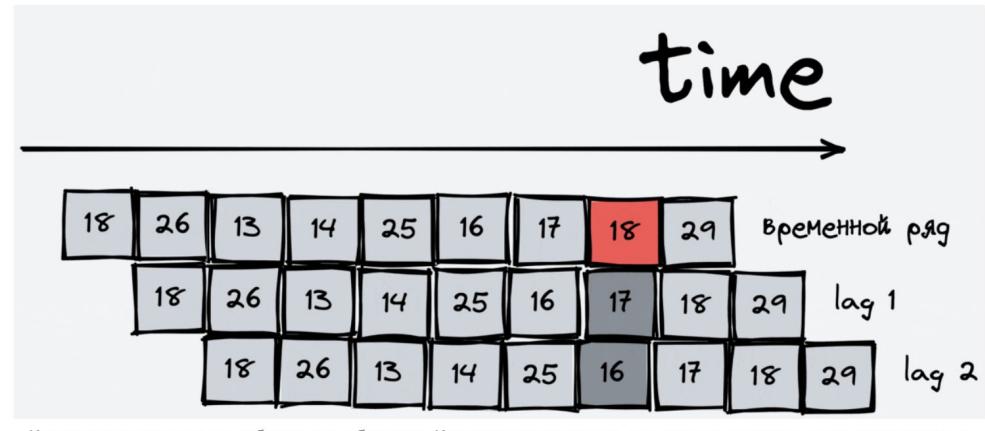
Праздники, особые дни

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



Лаги

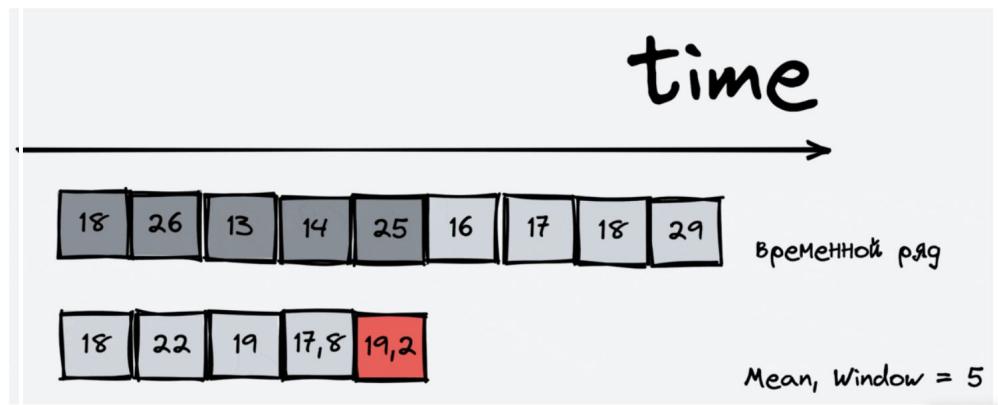
**Важно:** используем лаги, известные на весь горизонт прогноза( $\text{horizon}=14 \Rightarrow \text{lag} \geq 14$ )

# Лаги и скользящие статистики

Зачем:

Как лаги +

- Сглаживаем шум
- Ловим дополнительные паттерны(волатильность, разброс, объемы, ...)

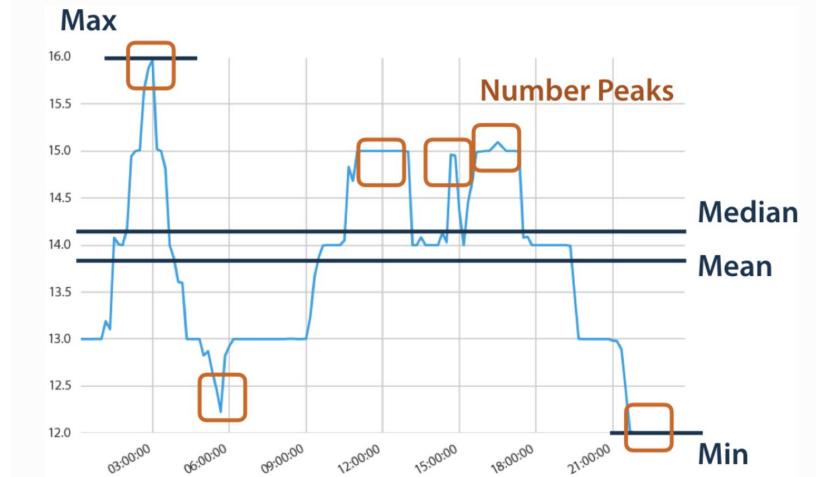


Скользящие статистики

# Описательные характеристики ряда

**Зачем:** делаем признаковое описание ряда(не каждой точки), отражающее его характер

- Трендовость
- Сезонность
- Волатильность
- ...



# 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

# Кодирование категориальных признаков

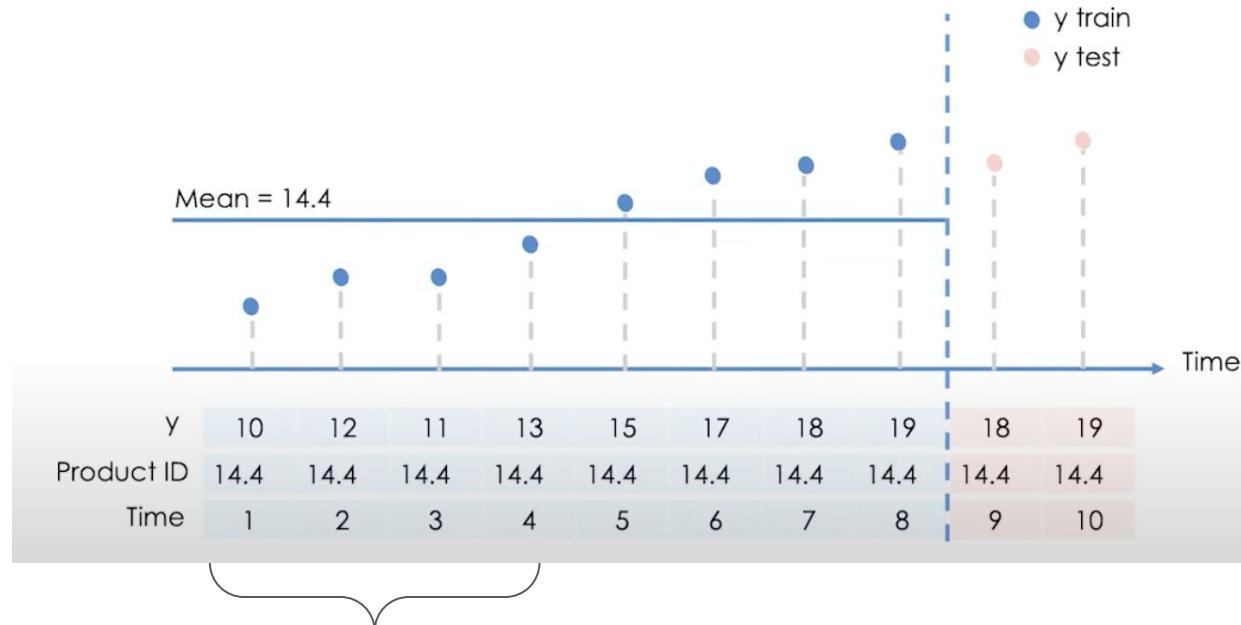
One hot encoding

daypart	morning	afternoon	evening	other
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	0
1	1	0	0	0

- N категорий -> N-1 признаков
- Отдельный признак для новых категорий

# Кодирование категориальных признаков

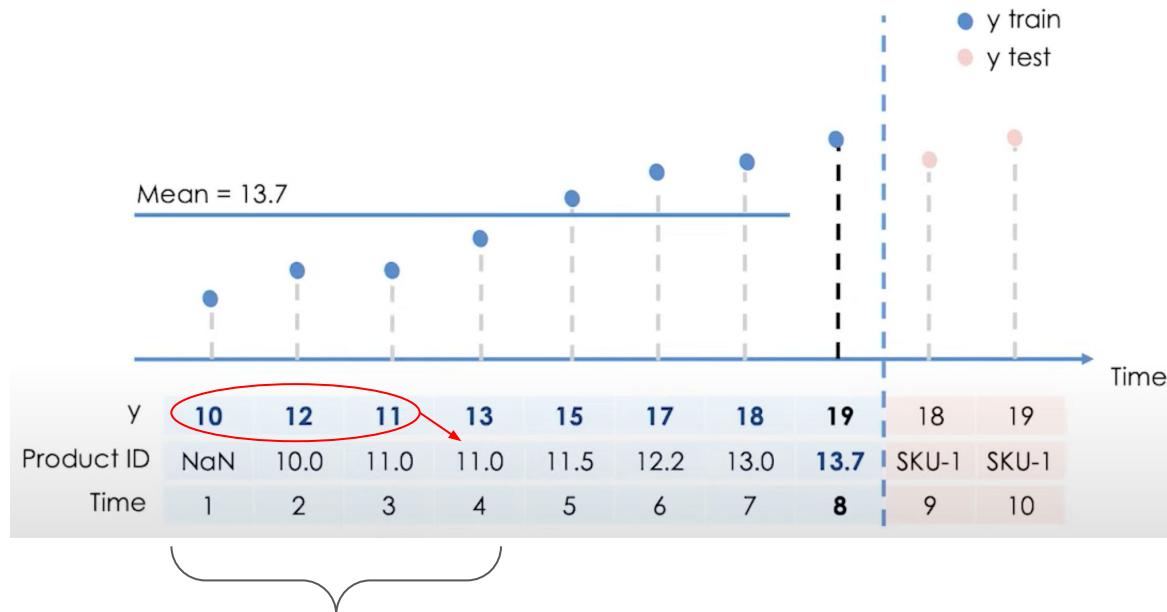
## Mean target encoding



Тут еще не знаем что таргет бывает выше 13 - look-ahead bias

# Кодирование категориальных признаков

Mean target encoding

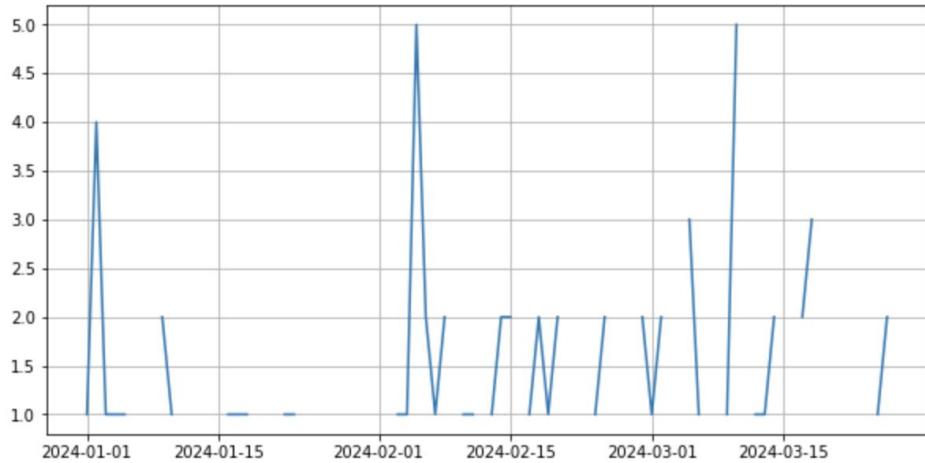


Кодируем скользящим  
средним

# Домено-специфичные признаки

Идея: привносим знания из домена, не нарушая принципов time series:

- Длина ряда
- Дней с последнего заказа
- Средняя длина промежутков без заказов
- ....



Заказы товара

# Выводы

- Основной принцип – не поглядываем в будущее, используем только известные данные
- Для построения признаков на тесте нужен обучающий датасет - поэтому не всегда удобно использовать стандартные библиотеки для ML
- Основные группы признаков
  - a. Признаки из временной метки
  - b. Признаки из прошлых значений переменных
  - c. Описательные признаки ряда
  - d. Домено-специфические признаки

Lag

DateFlags

Quantile  
SpecialDays

OneHotEncoder

SegmentEncoder

PredictionIntervalOutliers

DensityOutliers

MedianOutliers

GaleShapleyFeatureSelection

LinearTrend

TimeFlags  
YeoJohnson

STL

MAD

MaxAbsScaler

Log

Max

Mean

Median

Min

Holiday

TimeSeriesImputer

MinMaxScaler

LabelEncoder

Trend

TreeFeatureSelection

Fourier

MeanSegmentEncoder

AddConst

ResampleWithDistribution

BinsegTrend

MRMRFeatureSelection

Differencing

RobustScaler

FilterFeatures

StandardScaler

TheilSenTrend

# Нейросетевые подходы

# Какие модели рассмотрим

MLP

DeepAR

TFT

MLP-based

RNN-based

Transformer-based

NBeats

DeepState

Wavenet

TS specific

State space models based

CNN-based

# Многослойный перцептрон MLP

Подготовка  
данных

ML модели

Dates	Store ID	Product ID	Product name	# of Sales
02-12-2021	001	RP01	Almond milk	21
10-12-2021	005	RS21	Oat milk	15
18-01-2022	004	RK32	Hazelnut milk	9



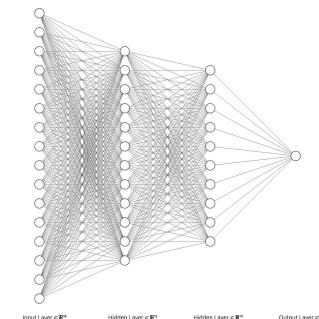
MLP

Dates	Store ID	Product ID	Product name	# of Sales
02-12-2021	001	RP01	Almond milk	21
10-12-2021	005	RS21	Oat milk	15
18-01-2022	004	RK32	Hazelnut milk	9

Модель



CatBoost



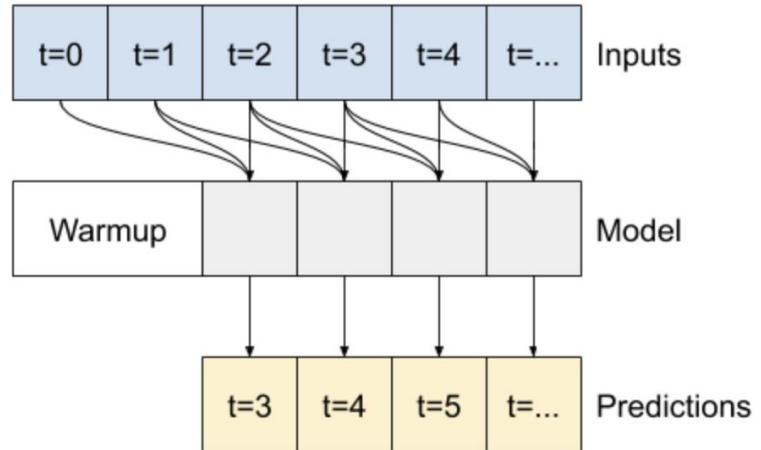
# Многослойный перцептрон MLP

Почему вам могут понадобиться MLP модели:

- Меньше проблем с экстраполяцией чем у деревьев и в то же время модель сильнее чем просто линейная регрессия
- Проще интегрировать свою функцию потерь, чем в деревья или градиентный бустинг - иногда это может быть важно
- Легко переделать в вероятностную модель, поменяв функцию потерь и последний слой

# CNN based

- Это расширение классических подходов со скользящим окном - но теперь параметры обучаемые и мы можем использовать несколько слоев для увеличение контекста (receptive field)
- В целом - можно пробовать любые архитектуры из speech recognition - например - WaveNet



# Рекуррентные нейронные сети RNN

Мы же используем сети - зачем нам строить признаки руками?

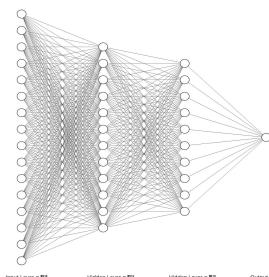
MLP

Подготовка  
данных

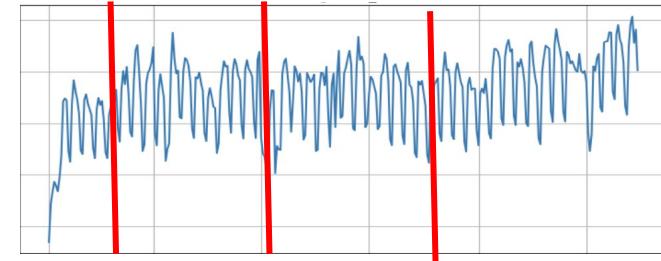
Dates	Store ID	Product ID	Product name	# of Sales
02-12-2021	001	RPO1	Almond milk	21
10-12-2021	005	RS21	Oat milk	15
18-01-2022	004	RK32	Hazelnut milk	9



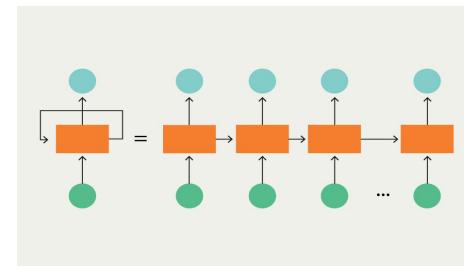
Модель



RNN



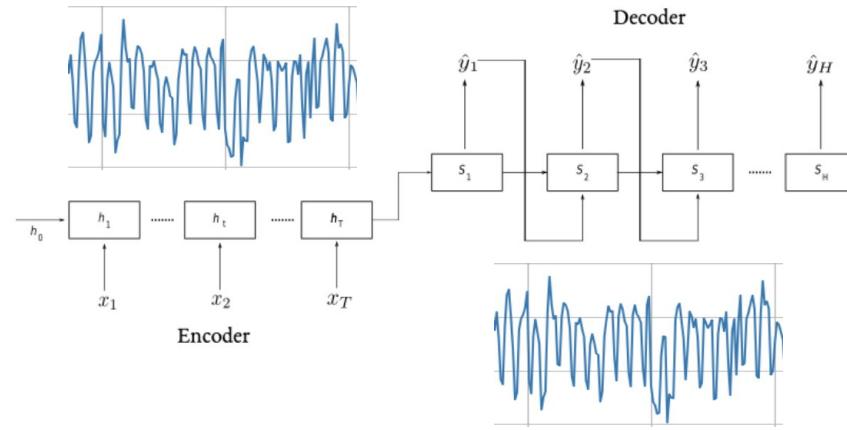
encoder decoder



Обучение = Инференс

# Рекуррентные нейронные сети RNN

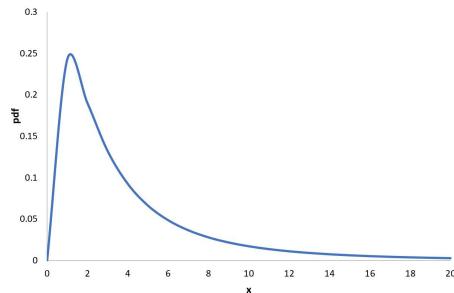
Авторегрессионно  
предсказываем  
значения



$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

# DeepAR

1. Sampling – боремся со скошенностью распределения



3. Probabilistic – предсказываем параметры распределения, а не значения ряда

2. Scaling – боремся с разницей в масштабах

Прямое преобразование(таргет)

$$\nu_i = 1 + \frac{1}{t_0} \sum_{t=1}^{t_0} z_{i,t},$$

Обратное преобразование(параметры)

$$\mu = \nu_i \log(1 + \exp(o_\mu))$$

$$\ell(z_{i,t} | \theta(\mathbf{h}_{i,t}, \Theta))$$

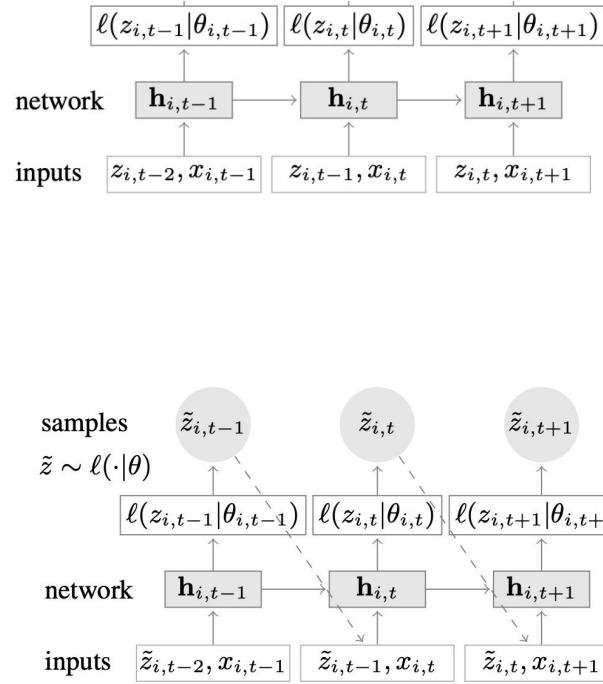
# Инференс

# Обучение

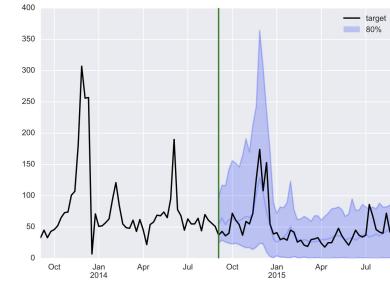
# DeepAR

Предсказываем  
параметры  
распределения

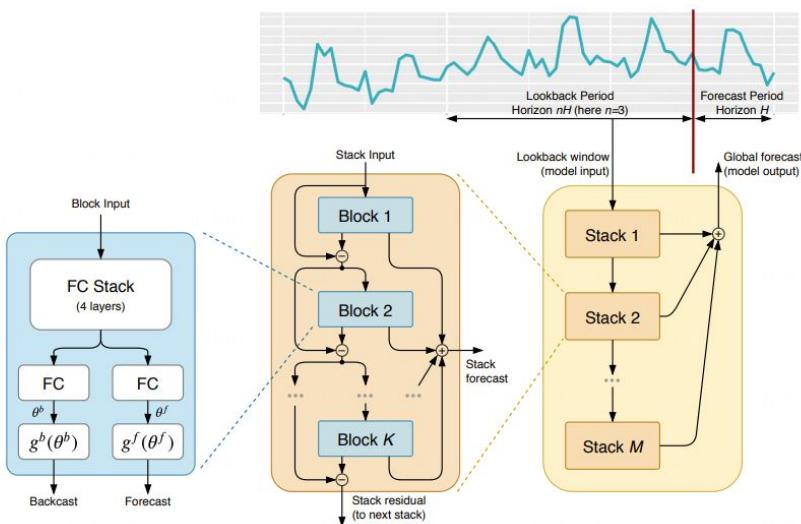
Авторегрессионно  
сэмплируем из  
распределения



$$\mathcal{L} = \sum_{i=1}^N \sum_{t=t_0}^T \log \ell(z_{i,t}|\theta(\mathbf{h}_{i,t}))$$



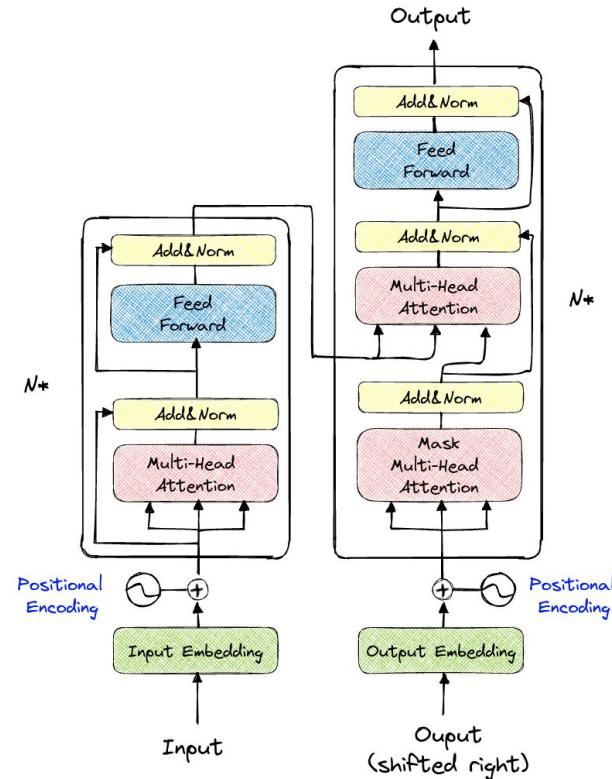
# N-BEATS



- Архитектура основана на остатках прогнозирования и residual связях сети
- Основным блоком является полносвязный слой
- Использование базисов для построения backcast/forecast
  - generic  $\hat{\mathbf{y}}_\ell = \mathbf{V}_\ell^f \theta_\ell^f + \mathbf{b}_\ell^f, \quad \hat{\mathbf{x}}_\ell = \mathbf{V}_\ell^b \theta_\ell^b + \mathbf{b}_\ell^b$
  - trend  $\hat{\mathbf{y}}_{s,\ell} = \sum_{i=0}^p \theta_{s,\ell,i}^f t^i$ .
  - seasonality  $\hat{\mathbf{y}}_{s,\ell} = \sum_{i=0}^{[H/2]-1} \theta_{s,\ell,i}^f \cos(2\pi i t) + \theta_{s,\ell,i+[H/2]}^f \sin(2\pi i t)$
- Интерпретируемость прогнозов (в случае интерпретируемых базисов)

# Transformers

- Позволяет использовать большой контекст
- В основном используется для предсказания на шаг вперед - в авторегрессионном режиме - но можно и предсказывать вектор сразу
- Attention можно использовать для интерпретации модели
- Пример: Temporal Fusion Transformer



# Foundation models

# Foundation models

Что есть в других модальностях:

- ChatGPT
- Gemini
- Midjourney
- DALL-E
- Grok
- Claude Opus
- Command R+

Все это картинки и текст + есть немного моделей для звука

# Foundation models - текущий статус

Таймлайн:

- [NBeats](#) - 2020
- TimeGPT api - конец лета 2023
- [LLMTime, Lag-Llama, PreDcT](#) - осень 2023
- [Amazon Chronos, Salesforce Moirai](#) - весна 2024

# Foundation models - какие основные блоки

- Transformer - Базовая архитектура
- RevIN - для нормализации
- RoPE - в качестве позиционных эмбедингов
- Patching - для учета временных зависимостей
- Вероятностное прогнозирование - дополнительная регуляризация + вероятностные прогнозы из коробки
- Дискретизация входа или использование лагов - как токенизация на входных слоях
- Freq-Mix - Freq-Mask аугментация и использование синтетических данных

# Foundation models - данные

Основные источники:

- Kaggle
- [Monash Time Series Forecasting Repository](#)

freq	Data	
	Сумма по полю N segments	Сумма по полю N samples
10T	1427	73487061
15T	386	52881715
1D	88	14167
1H	107	152297
30T	6820	169834677
5T	624	5756339
AS-JAN	1163	30997
B	8	48808
D	408318	421796366
H	158284	747873905
M	3621	285950
MS	1032328	67614233
QS-JAN	24000	2406108
QS-OCT	1183	79548
T	346	186704873
W	152500	44699602
W-MON	766	420099
W-SUN	115704	19092987
Total Result	1907673	1793179732

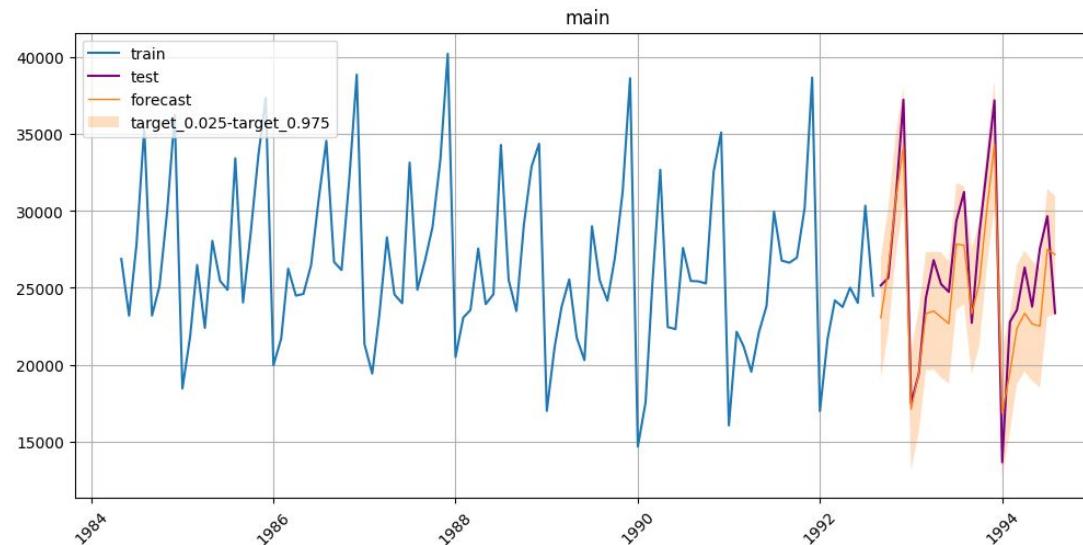
# Foundation models - будущее

- Явных победителей нет - у всех моделей есть проблемы с zero-shot прогнозированием
- Есть явные проблемы с наличием единых бенчмарков
- Открытых данных не так много
- Модели не больше 1В

# Предсказательные интервалы

# Предсказательные интервалы

**Предсказательные интервалы** – оценка интервала, в который будущее наблюдение попадет с определенной вероятностью при условии исторических наблюдений.



# Предсказательные интервалы

- Отражают неопределенность прогноза
- Позволяет оценить возможные траектории прогнозов
- Построение оптимистичных и пессимистичных сценариев при финансовом моделировании
- Оптимизация закупки на складах - safety stock
- Проверка на аномальность бизнес метрик

# Пример

- 95% интервал при условии нормального распределения будущих наблюдений

$$\hat{y}_{T+h|T} \pm 1.96 \hat{\sigma}_h,$$

- Оценка дисперсии одношагового прогноза

$$\hat{\sigma} = \sqrt{\frac{1}{T - K - M} \sum_{t=1}^T e_t^2},$$

- Оценка дисперсии многошагового прогноза наивной модели

$$\hat{\sigma}_h = \hat{\sigma} \sqrt{h}$$

# Источники случайности

- Aleatoric
  - Случайная ошибка в измерениях
- Epistemic
  - Случайность в оценки параметров модели
  - Случайность в выборе модели для конкретной задачи
  - Случайность из-за изменения DGP в будущем

# Предсказательные интервалы

## Подходы построения интервалов

- model specific
  - Точные модельные оценки
  - Моделирование квантилей
  - Байесовые оценки
- model agnostic
  - CV оценка
  - Jackknife
  - Конформные интервалы

# Оценка качества интервала

- Width

$$Width(y, lower, upper) = \frac{\sum_{i=0}^{n-1} |upper_i - lower_i|}{n}$$

- ширина интервала (меньше лучше, при обеспечении покрытия)
- зависит от масштаба данных
- использует только прогноз для расчёта

- Coverage

$$Coverage(y, lower, upper) = \frac{\sum_{i=0}^{n-1} [y_i \geq lower_i] * [y_i \leq upper_i]}{n}$$

- Процент покрытых точек из test
- Не зависит от масштаба данных
- Должен быть как можно ближе к заданному значению покрытия

# Оценка качества интервала

Другие метрики:

- Quantile loss
- Scaled quantile loss
- Mean Scaled Interval score

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

# 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

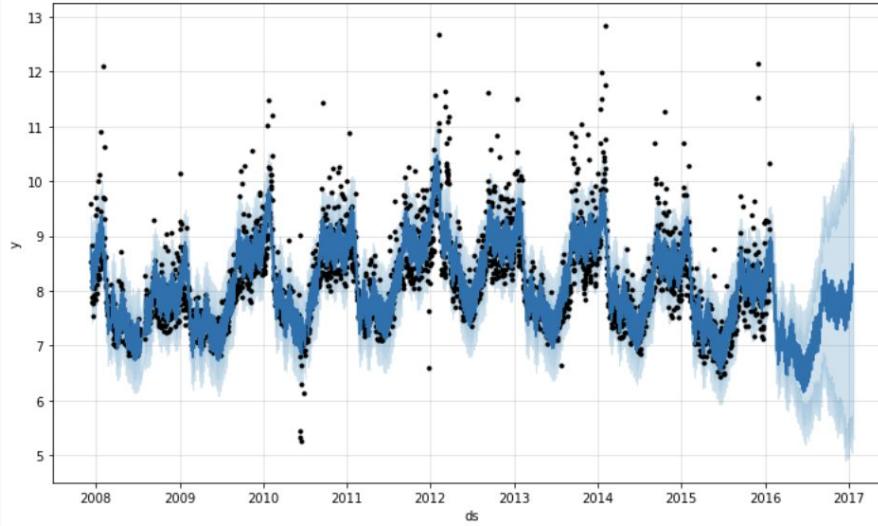
Return:  $\hat{m}(x_{n+1}^h) \pm z_{1-\frac{\alpha}{2}} \hat{\text{std}}_h$

# Моделирование интервалов в Prophet

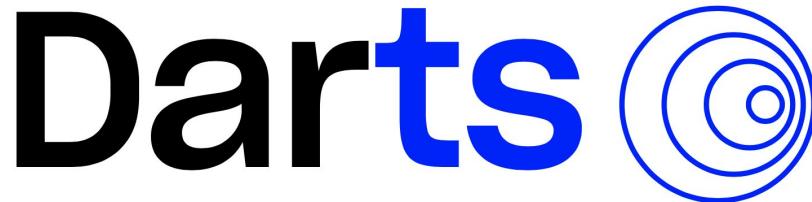
```
model {  
    //priors  
    k ~ normal(0, 5);  
    m ~ normal(0, 5);  
    delta ~ double_exponential(0, tau);  
    sigma_obs ~ normal(0, 0.5);  
    beta ~ normal(0, sigmas);  
  
    // Likelihood  
    y ~ normal_id_glm(  
        X_sa,  
        trend .* (1 + X_sm * beta),  
        beta,  
        sigma_obs  
    );  
}
```

# Моделирование интервалов в Prophet

```
1 # Python
2 m = Prophet(changepoint_prior_scale=0.5)
3 forecast = m.fit(df).predict(future)
4 fig = m.plot(forecast)
```



# Open source



# Nixtla

Набор библиотек для предсказания большого количества рядов:

- StatsForecast: библиотека с эффективной реализацией статистических методов: ARIMA, Holt и пр.
- MLForecast: библиотека с реализацией классических ML подходов к прогнозированию.
- NeuralForecast: библиотека с реализациями DL моделей.
- TimeGPT: GPT-like модель для zero-shot прогнозирования



## Sktme

Самый популярный фреймворк для работы с временными рядами.

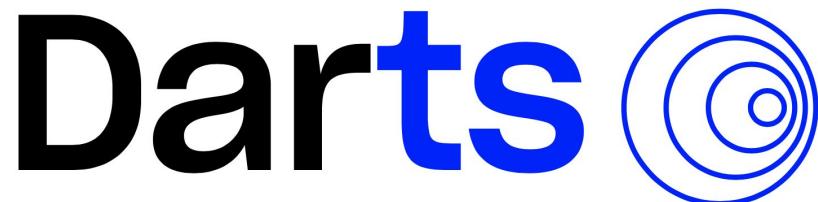
- Решает все основные задачи
- Содержит основные модели для прогнозирования
- Есть API для генерации признаков на временных рядах



## Darts

Одна из популярных библиотек

- Решаемые задачи:
  - Прогнозирование
  - Детектирование аномалий
- Содержит модели всех типов: статистические, ML, DL.
- Хорошая поддержка работы с доп. данными.



# GluonTS

Одна из первых библиотек для временных рядов, поддерживаемая Amazon

- Библиотека вероятностного прогнозирования временных рядов, предназначенная для работы с DL-моделями.
- Содержит реализацию большого количества SOTA-подходов.
- Большое количество инструментов для быстрого создания новых моделей, а также их оценки, анализа и тестирования.



# ETNA

- Библиотека для прогнозирования от компании Тинькофф.
- Решаемые задачи
  - Прогнозирование
  - Кластеризация
  - Классификация
  - Аномалии
- Содержит модели всех типов: статистические, ML, DL.
- Широкий набор средств для генерации признаков.



# Что выбрать и на что обращать внимание

- Посмотреть на количеству и частоту обновлений
- Устраивает вас производительность - некоторые библиотеки удобно использовать для экспериментов, но могут возникнуть сложности на больших объемах данных
- Насколько понятные инструкции и насколько просто добавить свою модель или признак
- Есть ли функциональность для удобных экспериментов - сохранение моделей и гиперпараметров, логирование в системы трекинга экспериментов

# Полезные материалы

- [Forecasting: Principles and Practice \(3rd ed\)](#)
- [Предсказание неопределённости и предсказательные интервалы](#)
- [Стратегии прогнозирования рядов](#)
- [Facebook Prophet via rymc3](#)
- [ETNA Tutorials: EDA, Modeling, Backtesting, etc](#)