Feature Engineering

Pt 2

Временные ряды

Что такое временной ряд?

Временной ряд — это последовательность наблюдений, упорядоченных во времени. Каждое наблюдение связано с конкретным моментом или периодом времени.

Примеры временных рядов

Финансы
Цены акций, курсы валют, криптовалюты

Метеорология
Температура, осадки, атмосферное давление

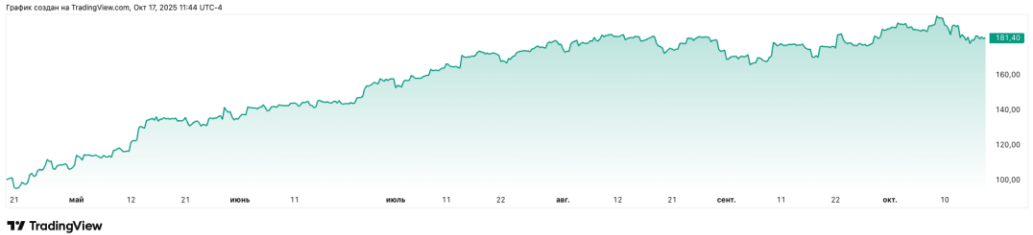
Медицина
ЭКГ, мониторинг пациентов, эпидемиология

Розничная торговля
Продажи, спрос, посещаемость магазинов

IoT и промышленность
Показания датчиков, энергопотребление

Ключевая особенность: Порядок наблюдений имеет критическое значение — нельзя произвольно переставлять точки данных

График стоимости акций NVIDIA



Компоненты временного ряда

Декомпозиция временного ряда — это разложение на составляющие компоненты для лучшего понимания структуры данных. Классическая модель представляет временной ряд как сумму трех компонентов.

01 Тренд (Trend)

Долгосрочное направление изменения временного ряда. Тренд показывает общую тенденцию роста, падения или стабильности. Может быть линейным, экспоненциальным или полиномиальным.

02 Сезонность (Seasonality)

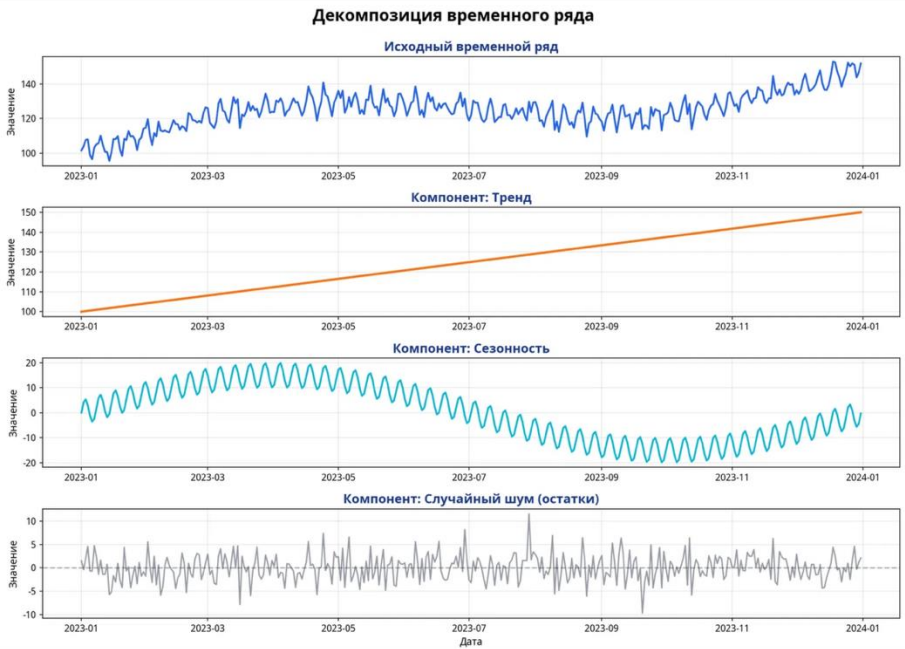
Регулярные, повторяющиеся паттерны с фиксированной периодичностью. Может быть дневной, недельной, месячной или годовой. Например, рост продаж перед праздниками, увеличение трафика в выходные.

03 Случайный шум (Noise)

Непредсказуемые, случайные колебания, которые не объясняются трендом или сезонностью. Остаточная компонента после удаления тренда и сезонности.

Формула: $Y(t) = Тренд(t) + Сезонность(t) + Шум(t)$

Декомпозиция временного ряда



Ключевые свойства временных рядов

01 Стационарность

Временной ряд является **стационарным**, если его статистические свойства (среднее, дисперсия, автокорреляция) не изменяются во времени. Стационарные ряды легче моделировать и прогнозировать.

Пример: Курс валюты с трендом — нестационарный, дневные изменения — стационарный

02 Автокорреляция

Корреляция временного ряда с самим собой, сдвинутым на определенное количество периодов (лаг). Показывает, насколько текущие значения зависят от прошлых. Высокая автокорреляция указывает на предсказуемость.

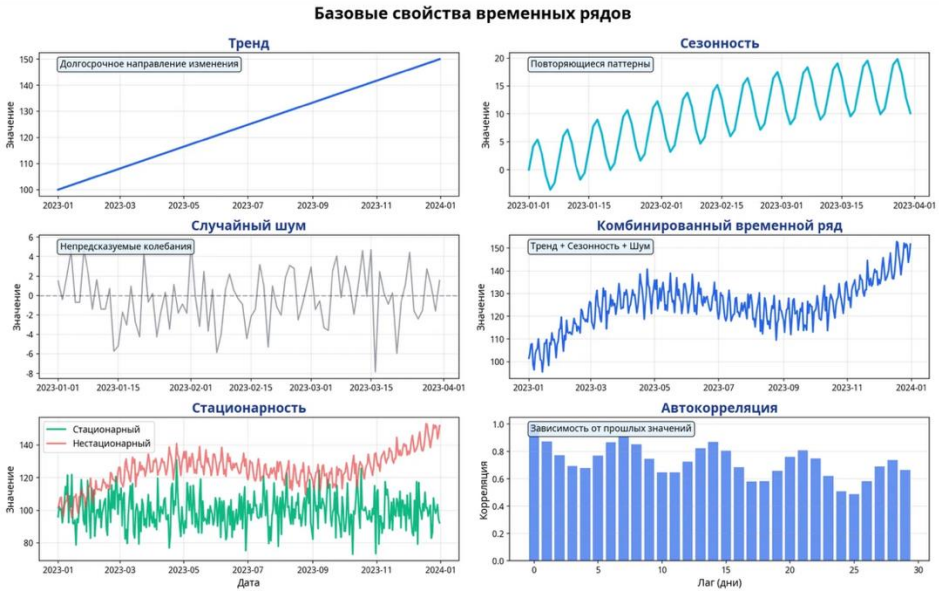
Применение: Определение оптимальных лагов, выявление сезонности

03 Временная зависимость

В отличие от обычных данных, в временных рядах порядок наблюдений критичен. Нельзя использовать случайное перемешивание данных при кросс-валидации — это приведет к утечке данных из будущего в прошлое.

Решение: Использовать TimeSeriesSplit для валидации

Визуализация свойств

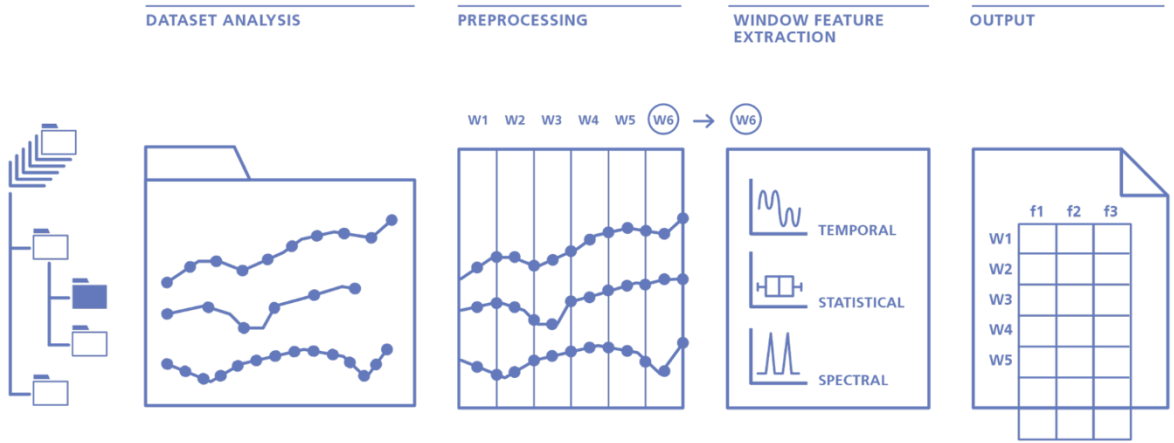


Графики демонстрируют стационарный vs нестационарный ряд и автокорреляционную функцию

Что такое Feature Engineering для временных рядов

Feature Engineering для временных рядов — это процесс создания **информативных признаков** из исходных временных данных для улучшения качества прогнозных моделей. В отличие от традиционных методов (ARIMA), которые автоматически используют лаговые значения, современный подход машинного обучения требует явного создания признаков, захватывающих временные зависимости, тренды и сезонность.

Эффективное конструирование признаков позволяет моделям выявлять сложные нелинейные зависимости в данных, что особенно важно для задач прогнозирования в финансах, энергетике, розничной торговле и других областях. Правильно созданные признаки могут повысить точность прогнозов на **20-30%** по сравнению с базовыми моделями.



Четыре ключевые категории признаков для временных рядов

После освоения базовых методов конструирования признаков мы переходим к более специализированным техникам. Эти продвинутые методы позволяют извлекать скрытые паттерны из сложных типов данных: циклических временных признаков и временных рядов.

01

Лаговые признаки (Lag Features)

Использование прошлых значений временного ряда в качестве предикторов. Лаговые признаки захватывают автокорреляцию и позволяют модели помнить" историю. Например, цена биткоина вчера, неделю назад и месяц назад могут быть сильными предикторами сегодняшней цены.

02

Скользящие статистики (Rolling Statistics)

Вычисление агрегированных метрик в скользящем окне: среднее, стандартное отклонение, минимум, максимум. Эти признаки сглаживают шум и выявляют локальные тренды. Скользящее стандартное отклонение измеряет волатильность рынка.

03

Временные признаки (Time-based Features)

Извлечение компонентов из даты: день недели, месяц, квартал, признак выходного дня. Циклическое кодирование (sin/cos) сохраняет периодическую природу времени. Критично для учета сезонности и календарных эффектов.

04

Разностные признаки (Difference Features)

Вычисление изменений между текущим и предыдущим значением (абсолютное и процентное). Помогает работать со стационарными рядами и выявлять динамику изменений. Особенно полезно для финансовых данных.

Автокорреляция временных рядов

Автокорреляция — это корреляция временного ряда с самим собой, сдвинутым на определенное количество периодов (лаг). Показывает степень зависимости текущих значений от прошлых.

Ключевые концепции

Что показывает: Насколько текущие значения предсказуемы на основе прошлых наблюдений

Диапазон значений: От -1 (отрицательная связь) до +1 (положительная связь), 0 — нет связи

Лаг (lag): Временной сдвиг между наблюдениями (например, лаг 1 = предыдущий день)

ACF график: Визуализирует автокорреляцию для разных лагов, помогает выявить сезонность

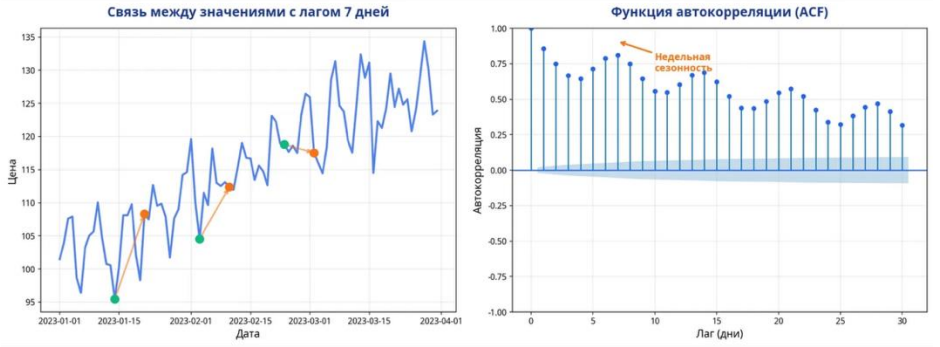
Применение: Определение оптимальных лагов для моделей, выявление периодических паттернов

Формула: $\rho(k) = Corr(Y_t, Y_{t-k})$

где k — лаг, Y_t — значение в момент t

Важно: Высокая автокорреляция на определенном лаге указывает на то, что этот лаг можно использовать как признак для прогнозирования

Визуализация автокорреляции



Слева: связь между значениями с лагом 7 дней.
Справа: ACF график показывает автокорреляцию для разных лагов

Лаговые признаки захватывают временные зависимости

Описание метода

Лаговые признаки (Lag Features) создаются путем сдвига значений временного ряда назад на определенное количество периодов. Это один из самых важных типов признаков для временных рядов, так как он позволяет модели использовать историческую информацию для прогнозирования будущего.

Выбор лагов зависит от специфики данных: для финансовых рынков часто используют лаги 1, 7, 30 дней (вчера, неделя, месяц), для розничных продаж — лаги соответствующие дням недели и сезонным периодам.

Применение

Прогнозирование цен акций, криптовалют, спроса на товары, энергопотребления, трафика

Пример кода на Python

```
import pandas as pd # Создание лаговых признаков
df['price_lag_1'] = df['price'].shift(1) # Вчера
df['price_lag_7'] = df['price'].shift(7) # Неделю назад
df['price_lag_30'] = df['price'].shift(30) # Месяц назад #
Результат: # date price price_lag_1 price_lag_7
price_lag_30 # 2023-01-01 30000 NaN NaN NaN # 2023-01-02
30500 30000 NaN NaN # 2023-01-08 31200 31100 30000 NaN #
2023-01-31 32500 32400 31800 30000
```

Важно: Лаговые признаки создают NaN значения в начале временного ряда. Необходимо удалить эти строки или использовать методы заполнения пропусков.

Скользящие статистики для выявления трендов

Скользящие статистики (Rolling Statistics) вычисляются в окне фиксированного размера, которое "скользит" по временному ряду. Скользящее среднее (Moving Average) — один из самых популярных индикаторов в техническом анализе, позволяющий отфильтровать краткосрочные колебания и увидеть общий тренд.

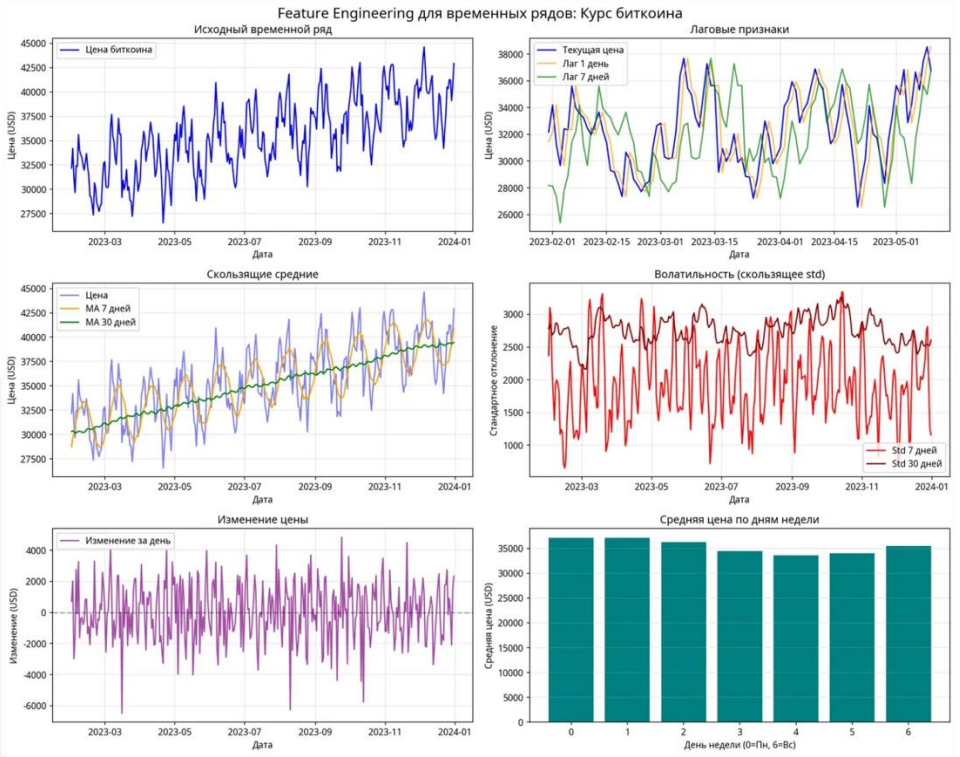
Скользящее стандартное отклонение измеряет **волатильность** — важный показатель риска в финансах.

Код Python

```
# Скользящее среднее
df['rolling_mean_7'] = df['price']
.rolling(window=7).mean()
df['rolling_mean_30'] = df['price']
.rolling(window=30).mean()

# Волатильность
df['rolling_std_7'] = df['price']
.rolling(window=7).std()

# Диапазон
df['rolling_min_7'] = df['price']
.rolling(window=7).min()
df['rolling_max_7'] = df['price']
.rolling(window=7).max()
```



Применение: Технический анализ, детекция аномалий, измерение риска

Разностные признаки помогают работать со стационарными рядами

Разностные признаки (Difference Features) вычисляют изменение значений между текущим и предыдущим периодом. Многие алгоритмы машинного обучения лучше работают со стационарными рядами, где среднее и дисперсия постоянны во времени.

Процентное изменение (pct_change) особенно полезно для финансовых данных, так как оно нормализует изменения относительно текущего уровня цены. Это позволяет сравнивать динамику активов с разными ценовыми уровнями.

Код Python

```python
# Абсолютное изменение
df['price_diff_1'] = df['price'].diff(1)
df['price_diff_7'] = df['price'].diff(7)
# Процентное изменение
df['price_pct_1'] = df['price'].pct_change(1)
df['price_pct_7'] = df['price'].pct_change(7)
```

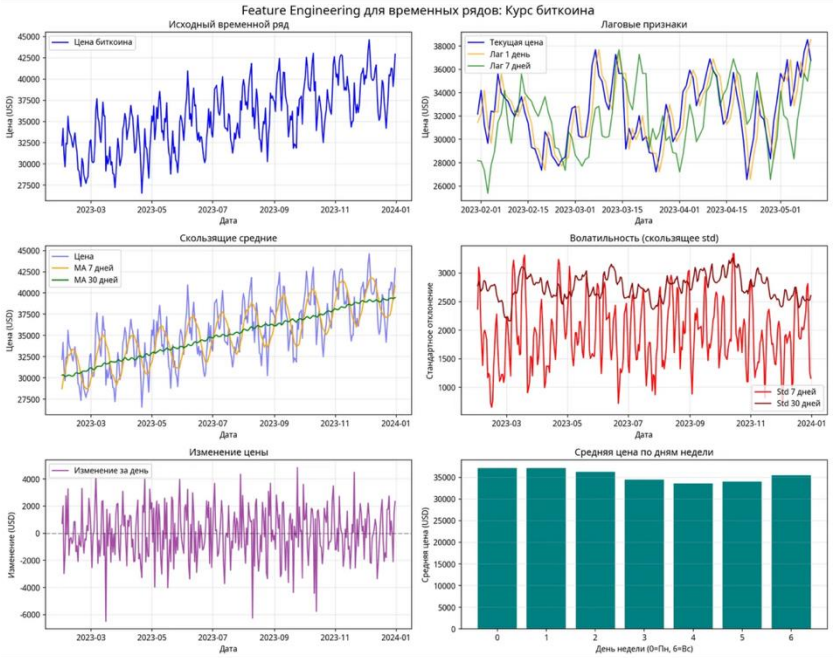


График изменения цены биткоина (дневные колебания)

Циклические признаки

Проблема циклических признаков

Временные признаки, такие как час дня (0-23) или месяц года (1-12), являются циклическими. При обычном кодировании модель не понимает, что значение 23 (23:00) и 0 (00:00) находятся рядом друг с другом, а не на противоположных концах шкалы.

Решение: синус и косинус

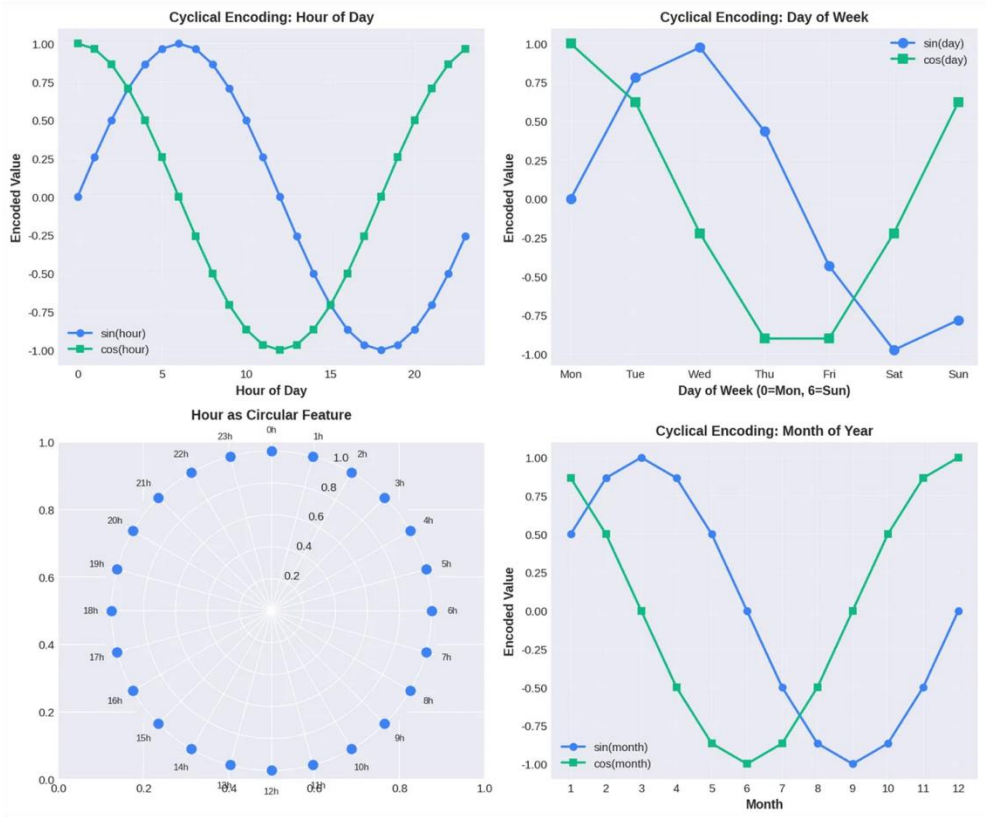
Используя тригонометрические функции, мы преобразуем циклические признаки в две координаты на единичной окружности. Это сохраняет циклическую природу данных и позволяет модели корректно интерпретировать близость значений.

Формулы преобразования

$$hour\_sin = \sin(2\pi \times hour / 24)$$
$$hour\_cos = \cos(2\pi \times hour / 24)$$

$$day\_sin = \sin(2\pi \times day / 7)$$
$$day\_cos = \cos(2\pi \times day / 7)$$

$$month\_sin = \sin(2\pi \times month / 12)$$
$$month\_cos = \cos(2\pi \times month / 12)$$



Преимущества

Две координаты (sin и cos) однозначно определяют положение на окружности. Расстояние между точками отражает реальную близость во времени. Модель легко обучается на таких признаках.

Примеры тригонометрических признаков

Медицинские приложения

1. Время приема пациентов

Анализ загруженности клиники по часам дня. Модель может выявить паттерны: утренние часы (7-9) и вечерние (18-20) имеют схожую нагрузку, хотя числовые значения далеки.

```
# Час приема: 8 утра
hour = 8
hour_sin = np.sin(2 * np.pi * hour / 24)
hour_cos = np.cos(2 * np.pi * hour / 24)
# Результат: (0.866, 0.5)
```

2. День недели госпитализации

Прогнозирование риска осложнений в зависимости от дня недели. Воскресенье (6) и понедельник (0) кодируются как близкие значения, что отражает реальность работы медучреждений.

Бизнес-приложения

1. Сезонность продаж

Прогнозирование продаж с учетом месяца года. Декабрь (12) и январь (1) кодируются как соседние месяцы, что важно для новогодних распродаж и сезонных трендов.

```
# Месяц: декабрь
month = 12
month_sin = np.sin(2 * np.pi * month / 12)
month_cos = np.cos(2 * np.pi * month / 12)
# Результат: (0.0, 1.0)
```

2. Час покупки онлайн

Предсказание вероятности покупки в зависимости от времени суток. Поздний вечер (22-23) и раннее утро (0-1) показывают схожее поведение пользователей — низкую активность.