Instacart Market Basket Analysis

Георгий Даньщин

План

- Описание задачи
- Описание данных
- Метрика оценки качества
- Оптимизация метрики
- Признаки
- Модели

Задача

- Дана история более, чем 3 млн заказов, сделанных 200k разными пользователями
- Необходимо предсказать какие продукты из тех, которые были заказаны ранее, пользователь выберет в следующем заказе
- Данные по заказам были разбиты на 3 набора
  - prior – история заказов по всем пользователям
  - train/test – последний заказ каждого из пользователей
  - пользователи в train и test не пересекаются

Данные: заказы пользователей

- 3421083 заказов (131209 train, 75000 test, 3214874 prior)
- 206209 пользователей
- для каждого пользователя есть история от 4х до 100 заказов

Колонки:

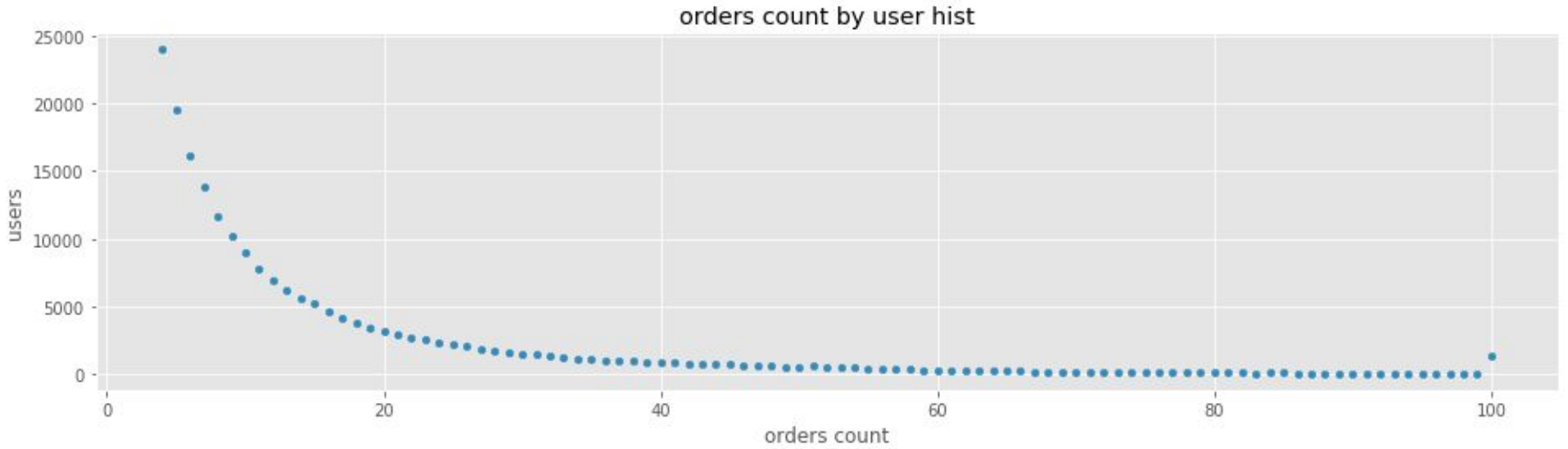
- **order_id**
- **user_id**
- **eval_set** – набор данных [prior|train|test]
- **order_number** – номер заказа пользователя
- **order_dow** – день недели
- **order_hour_of_day** – час, в который был сделан заказ
- **days_since_prior_order** – кол-во дней, прошедших с предыдущего заказа

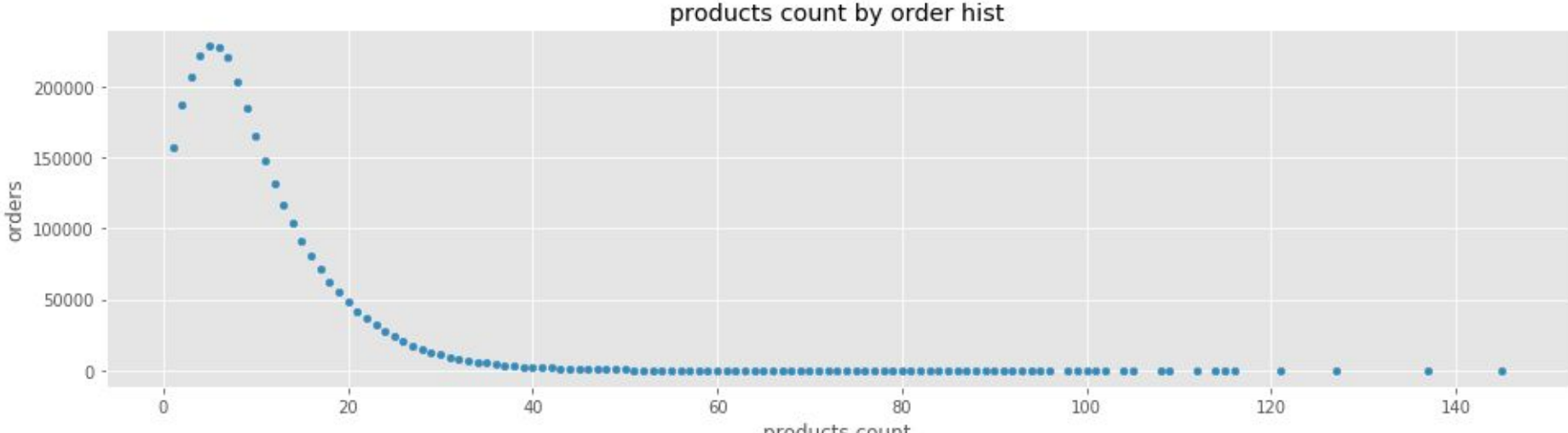
Данные: состав заказов

- в заказе может быть от 1 до 145 продуктов (в среднем 10)

Колонки:

- **order_id**
- **product_id**
- **add_to_cart_order** – порядковый номер добавления продукта в корзину
- **reordered** – заказывался ли этот продукт пользователем ранее


orders count by user hist
users
orders count
25000
20000
15000
10000
5000
0
0
20
40
60
80
100


products count by order hist
orders
products count
200000
150000
100000
50000
0
0
20
40
60
80
100
120
140

Данные: продукты

- 49688 продуктов

Колонки:

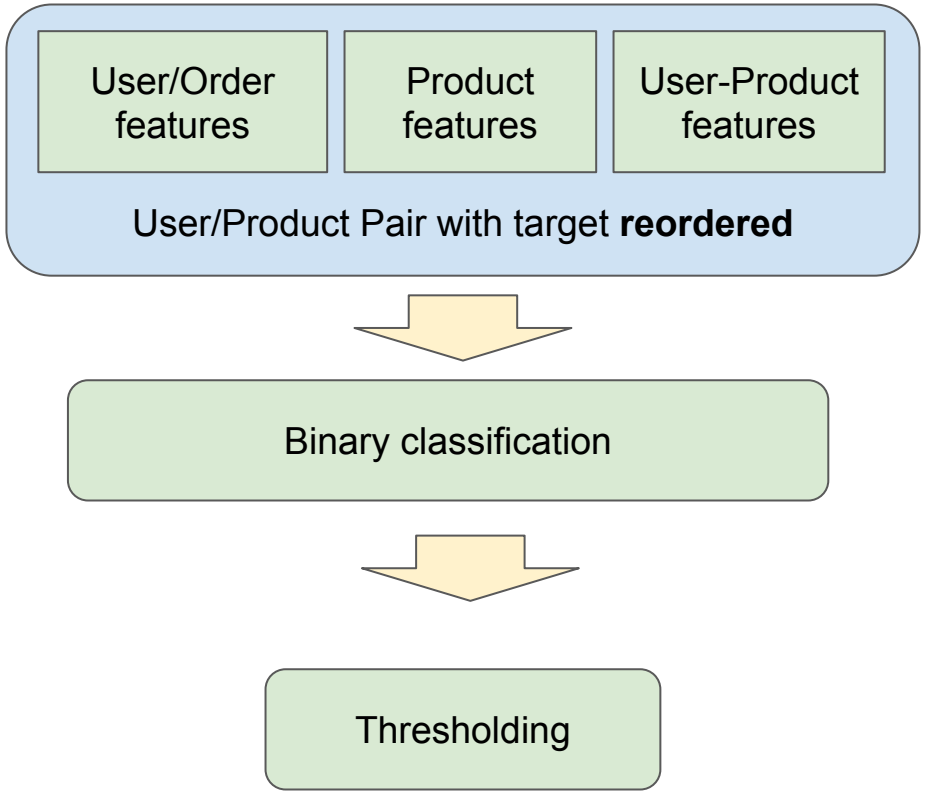
- **product_id**
- **product_name**
- **department_id** – один из 21 отделов (snacks, pantry, beverages, etc.)
- **aisle_id** – одна из 134 категорий товаров (ice cream ice, tea, frozen meals, etc.)

Метрика

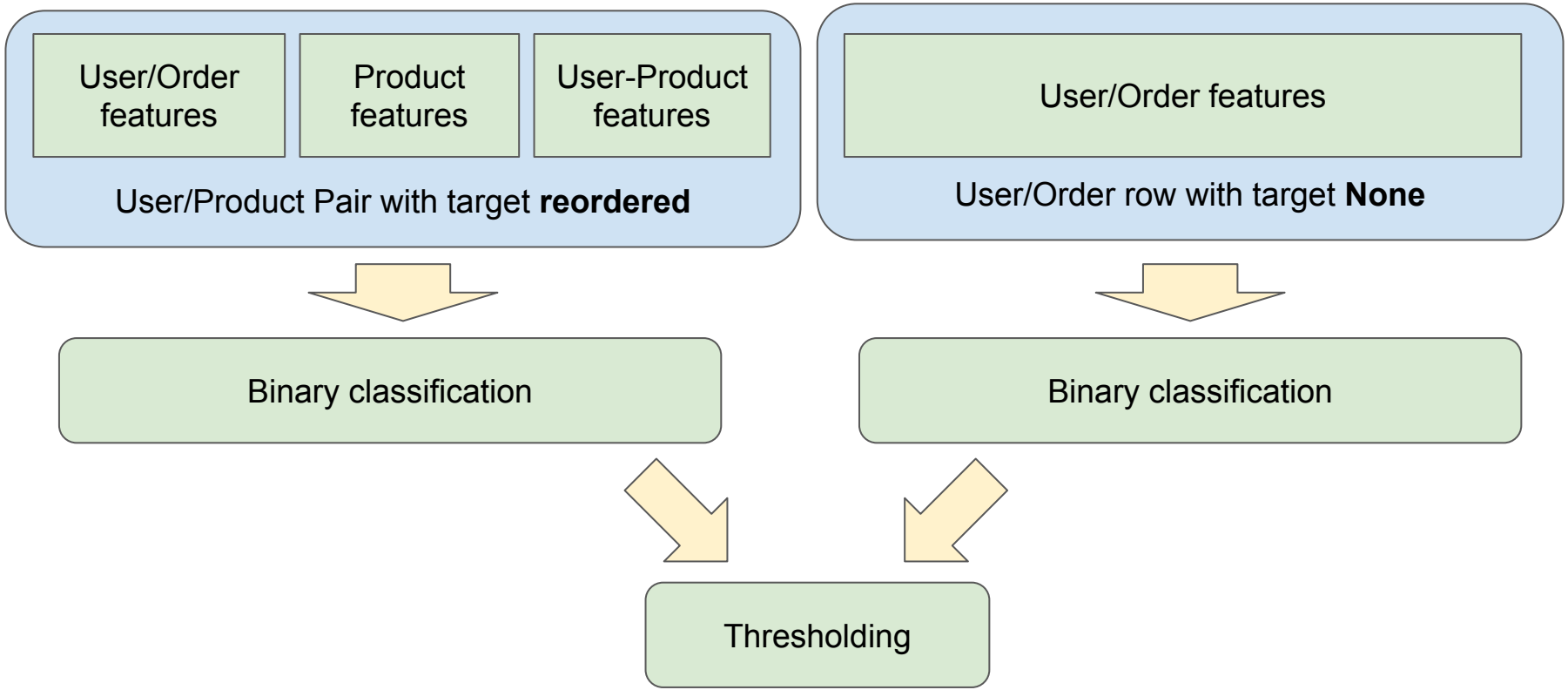
- Для каждого заказа в тесте необходимо предсказать список продуктов из этого заказа среди тех, которые были заказаны ранее
- Если ни один продукт не заказан повторно, нужно предсказать None для этого заказа
- None можно предсказывать вместе с набором product_id
- Качество оценивается с помощью метрики f1_score

$$\text{FScore}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \frac{2tp_i}{2tp_i + fp_i + fn_i}$$

Pipeline


User/Order features
Product features
User-Product features
User/Product Pair with target **reordered**
Binary classification
Thresholding

Pipeline


User/Order features
Product features
User-Product features
User/Product Pair with target **reordered**
User/Order features
User/Order row with target **None**
Binary classification
Binary classification
Thresholding

Выбор порога

- константный порог для всех пользователей
- отдельная модель для предсказания порога
- expected F-score maximization
  - Thresholding Classifiers to Maximize F1 Score
  - Optimizing F-measure: A Tale of Two Approaches

Expected F-Score maximization

- Для каждого заказа есть предсказанные вероятности продуктов

$$\hat{p}_{\text{None}}, \hat{p}_1, \ldots, \hat{p}_k$$

- Предполагая, что продукты выбираются независимо, для каждого из 2^k возможных вариантов корзин можно вычислить вероятность корзины:

$$b = \{b_i\}_{i=1}^{k} \in B, \quad \forall i \; b_i \in \{0, 1\}, \quad b_{\text{None}} = \{0\}_{i=1}^{k}$$

$$\mathbb{P}(b_{\text{None}}) = \hat{p}_{\text{None}}, \quad \mathbb{P}(b) = \prod_{i=1}^{k} \hat{p}_i^{b_i} (1 - \hat{p}_i)^{1 - b_i}$$

Expected F-Score maximization

b	P(b)	fscore
None	0.07200	0.000
3	0.66239	0.667
2	0.00248	0.000
2, 3	0.01077	0.500
1	0.03165	0.667
1, 3	0.13760	1.000
1, 2	0.00051	0.500
1, 2, 3	0.00224	0.800

p(None) = 0.072, p(1) = 0.172, p(2) = 0.016, p(3) = 0.813, T = 0.15

Expected F-Score maximization

- Для каждого порога T и набора b можно вычислить F-score напрямую, а значит можно вычислить матожидание F-score:

$$\mathbb{E}\mathrm{Fscore}(T) = \sum_{b \in B} \mathbb{P}(b)\mathrm{Fscore}(T, b)$$

- Оптимальным будет порог, на котором матожидание максимально:

$$T^* = \arg\max_T \mathbb{E}\mathrm{Fscore}(\mathrm{T})$$

- Наивная реализация имеет экспоненциальную сложность, но есть алгоритм, вычисляющий порог за квадратичное время по кол-ву меток (пример реализации есть в kernel)

Признаки

- признаки пользователя
- признаки продукта
- признаки пары пользователь/продукт

Признаки пользователя

- кол-во дней, прошедших с первого заказа
- кол-во дней, прошедших с последнего заказа
- статистики по интервалам между заказами пользователя
- разница/отношение кол-ва дней, прошедших с последнего заказа и среднего интервала между заказами
- статистики по размеру заказов
- средний день недели/час заказов
- разница между текущим днём недели/часом и средним
- различные агрегации по фичам продукта и фичам пары пользователь/продукт

Признаки продукта

- доля заказов с продуктом
- доля пользователей, которые заказывали продукт
- кол-во, доля reorder’ов для продукта
- кол-во, доля пользователей, совершавших reorder
- средний нормализованный порядковый номер добавления в корзину
- вероятность заказа продукта в первый раз, вероятность reorder’а, их отношение
- размер категорий (aisle, department) продукта
- вероятности reorder’а для категорий

Признаки пары пользователь/продукт

- кол-во/доля заказов данного продукта пользователем
- кол-во дней/заказов, прошедших с последней покупки выбранного продукта
- статистики кол-ва дней/заказов между покупками продукта
- разница/отношение между кол-вом дней/заказов, прошедших с последней покупки продукта и средним кол-во дней/заказов
- средний нормированный порядковый номер добавления заказа в корзину
- такие же признаки, вычисленные для aisle и department
- tfidf по паре пользователь/продукт
- восстановленное svd разложение tfidf матрицы пользователь/продукт

История заказов конкретного продукта

user_id	order_id	order_number	target
72812	361096	1	1
72812	485157	2	0
72812	1816060	3	0
124747	1181461	1	0
124747	2527579	2	1
124747	941001	3	1
141122	3098547	1	1
141122	1241431	2	0

Заказы, в которых возможен reorder для продукта

user_id	order_id	order_number	target
72812	361096	1	1
72812	485157	2	0
72812	1816060	3	0
124747	1181461	1	0
124747	2527579	2	1
124747	941001	3	1
141122	3098547	1	1
141122	1241431	2	0

Product regressions

- Для каждого продукта из prior, который заказывался не менее 100 раз (около 20k продуктов) выберем все заказы, для которых был возможен reorder этого продукта
- Обучим отдельную логистическую регрессию для каждого продукта на следующих данных:
  - X – one hot encoding по продуктам из n (от 1 до 5) предыдущих заказов
  - y – reorder
- Предсказания регрессии будем использовать в качестве признаков для train и test

order_id	order_lag_1	order_lag_2	order_lag_3	target
3245	[4123, 34, 1]	[4123, 34]	[1]	0
1239	[123, 3, 1, 23]	[123, 3, 1, 23]	[123, 4, 2, 23]	1
4322	[5]	[5, 6]	[5]	1

Markov chains

- Для каждой пары заказ/продукт из prior введём 2 состояния:
  - 0 – продукт был заказан этим пользователем в предыдущем заказе
  - 1 – продукт не был заказан этим пользователем в предыдущем заказе
- Для каждого состояния вычислим вероятности, усредняя по всем заказам, в которых был возможен reorder:

$$\mathbb{P}(\text{reorder}|\text{state}, \text{user})$$

$$\mathbb{P}(\text{reorder}|\text{state}, \text{product})$$

$$\mathbb{P}(\text{reorder}|\text{state}, \text{user}, \text{product})$$

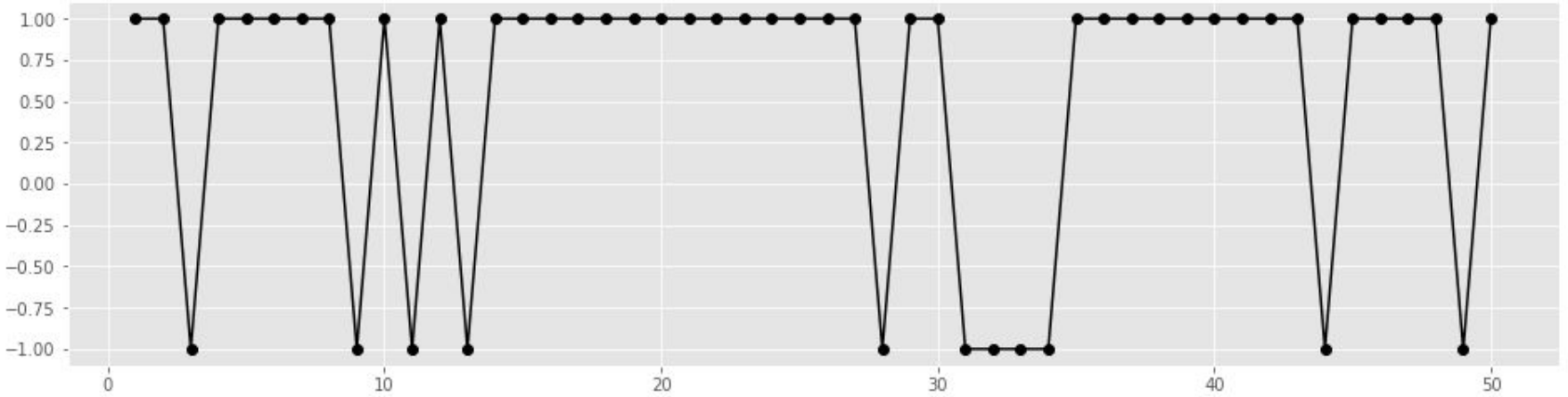
- Все вероятности будем считать используя аддитивное сглаживание
- Вычисленные вероятности будем использовать в виде фичей для train и test, подставляя их в соответствии с состоянием/пользователем/продуктом

Markov chains

- Другой вариант задания состояний:
  - 0 – продукт никогда не заказывался пользователем
  - 1 – продукт был заказан в предыдущем заказе в первый раз
  - 2 – продукт был в предыдущих заказах, но не в прошлом
  - 3 – продукт был заказан в предыдущем заказе не в первый раз
- Расширим состояния ещё больше, декартово умножая состояния за n предыдущих заказов (от 1 до 5ти)
- Такие же признаки посчитаем для категорий (aisle/department)

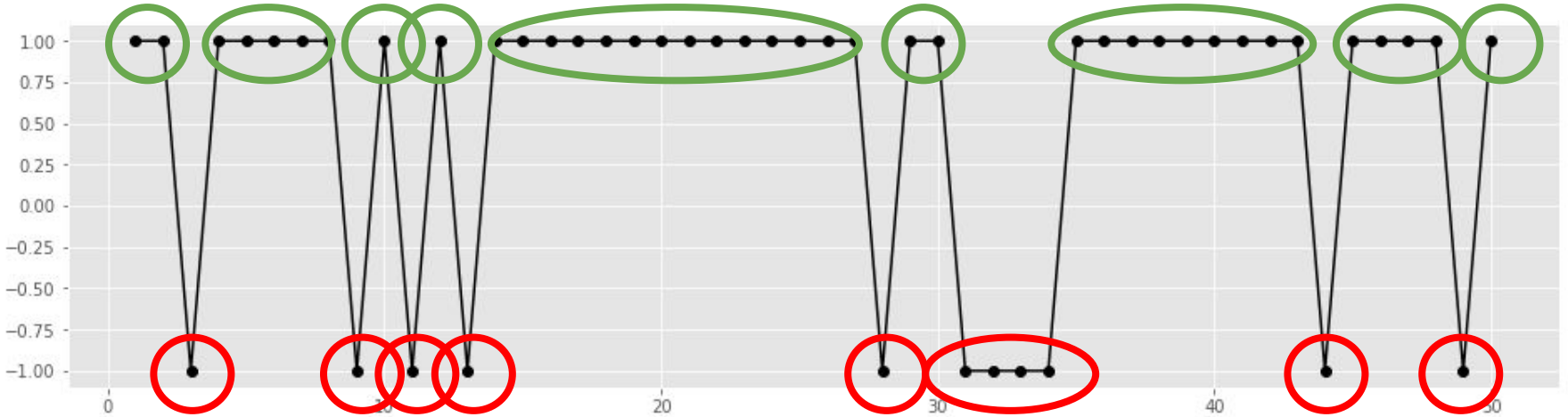
Streaks

- Для каждой пары пользователь/продукт построим график по всем заказам пользователя:
  - x – порядковый номер заказа
  - y = 1, если продукт был в заказе, -1, если не было



Streaks

- Назовём streak’ом группу заказов, идущих подряд, в которой по конкретному продукту было принято одно и то же решение (купить/не купить)



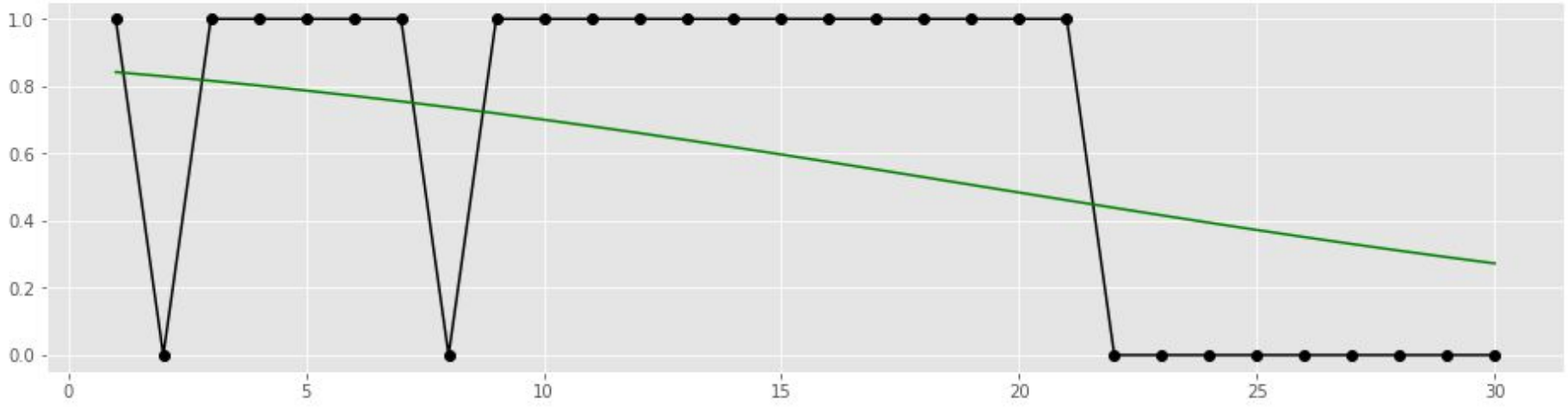
Streak

Признаки:

- длина последнего streak в днях и заказах, умноженная на знак streak (+1/-1)
- статистики по длинам streaks в днях и заказах
- статистики по длинам положительных/отрицательных streaks
- разница/отношение длины последнего streak и средней длины
- разница/отношение длины последнего streak и средней длины того же типа (положительный/отрицательный)
- статистики по длинам streaks по всем пользователям, и разница с последним

Trends

- для каждой пары пользователь/продукт построим тренд (регрессия таргета на номер заказа) за последние 10/20/30/50/100 дней
- в качестве фичей возьмём наклон и предсказание тренда



Модели

- Локальная валидация считалась по 5ти фолдам, разбитым по пользователям
- p(None) предсказывалось отдельным xgb на фичах пользователя
- Ансамбль из 7 xgboost: 0.4073408 public, 0.4058049 private
- Ансамбль из 7 lightgbm: 0.4071269 public, 0.4062934 private

Ансамбль

1. все признаки (239)
2. без streak (203)
3. без markov chains (213)
4. без product regressions (234)
5. без trend (227)
6. без streak и markov chains (177)
7. без product regressions и markov chains (208)