Predicting the "Buy It Again" Moment: Engineering Recurring Purchases

AI Hub - Shopper AI - HardROC Team

allegro

Agenda

1. Business Problem
2. Which categories are recurring one?
3. Intuition and challenges while building the models
4. Time to next purchase prediction
5. Tech Stack

Business Problem

Recommendation Scenarios

Orders history

Similar products recommendations

Complementary products recommendations

Buy it again recommendation

allegro

Motivation and problem decomposition

Motivation

- Most recommendation engines try to predict similar/complementary product based on purchase history
- Many products in retail and e-commerce are bought in recurring cycles
- We should recommend same or equivalent product

Decomposition

- Task consists of two parts:
  - **what** products are recurring
  - predict **time** when customer buys again

Intuition

- What product?
  - FMCG
- When?
  - According to usage pattern specific to person & product category

Which categories are the recurring ones?

Recurring share coefficient definition

$$\textbf{Recurring Share} = \frac{\text{number of clients with } \textbf{more than 1} \text{ purchase}}{\text{number of clients with } \textbf{at least 1} \text{ purchase}}$$

Interpretation:

- percent of clients that bought more than once in the category

Categories examples

Top categories	Recurring share↓
Dry Dog Food	65%
Wet food for dogs	60%
Wet food for cats	59%
Baby food	56%
Coffee Beans	54%


65% of customers in this category bought dry food more than once.

Bottom categories	Recurring share↑
Driving wheels for PlayStation	<1%
Sewage tanks	<1%
Pool Ladders	<1%
Electric tractors for children	<1%
Polishers	<1%



Collectors microsegment

Category	Recurring Share↓
Post stamps from Cuba	63%
Post stamps from Europe, years 1901 - 1945	61%
Post stamps from Australia	61%
Post stamps from Africa	61%
Post stamps from Asia	60%

Similar segments:

- Players of “Magic: The Gathering”
- Phone cards
- Postcard collectors

allegro

Recurring definition - to sum up

$$\text{Category repeat purchase probability} = \frac{\text{number of customers who bought more than once in category*}}{\text{number of customers who bought at least once in category*}}$$

Interpretation: XX% of customers who bought in given category are buying it more than once

Values for selected categories:

- dog food: 65%
- nuts: 40%
- flour: 36%
- Brita water filters: 29%
- cartridges for razors: 17%

Disadvantage: individual customers may buy some products in categories that are globally non-recurring

* in different days so customer who made many orders on the same day is calculated as one purchase

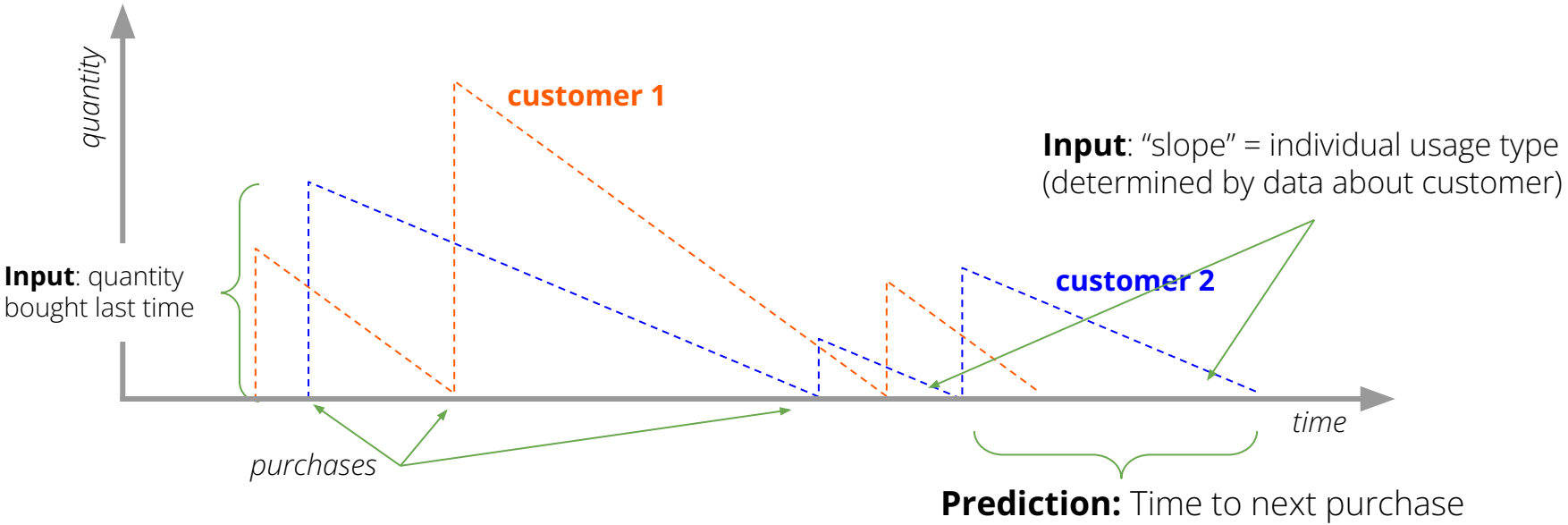
Intuition and challenges while building the models

Time to next purchase - intuition

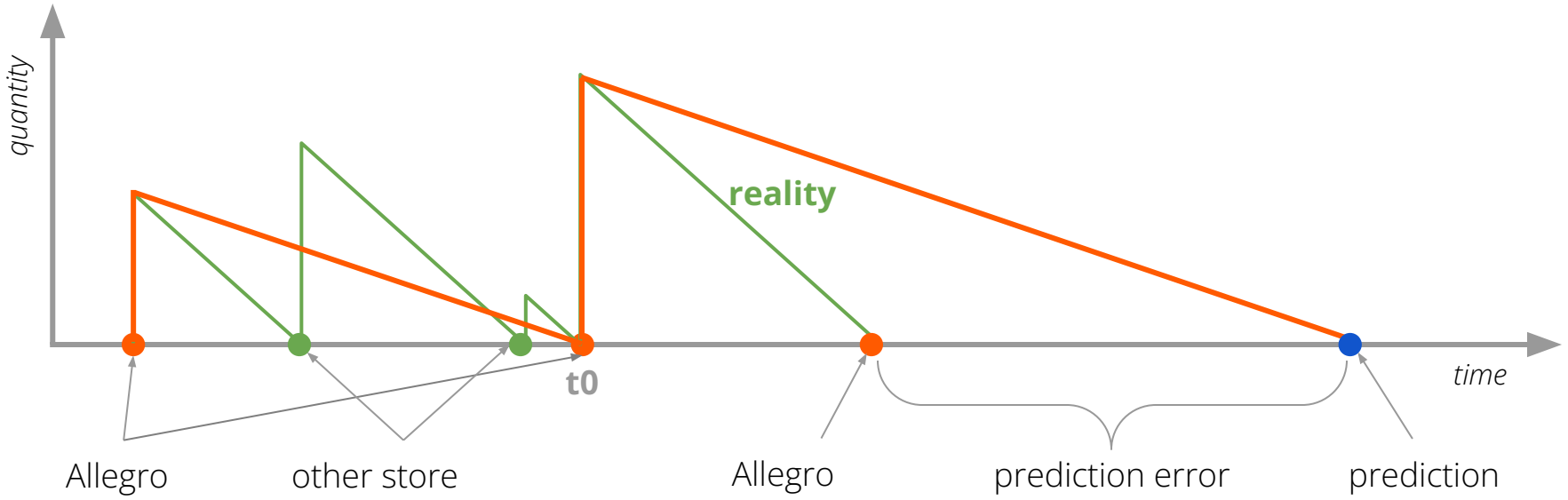
Product category time to next purchase

~

F (time since last purchase, quantity bought last time, individual usage pattern)

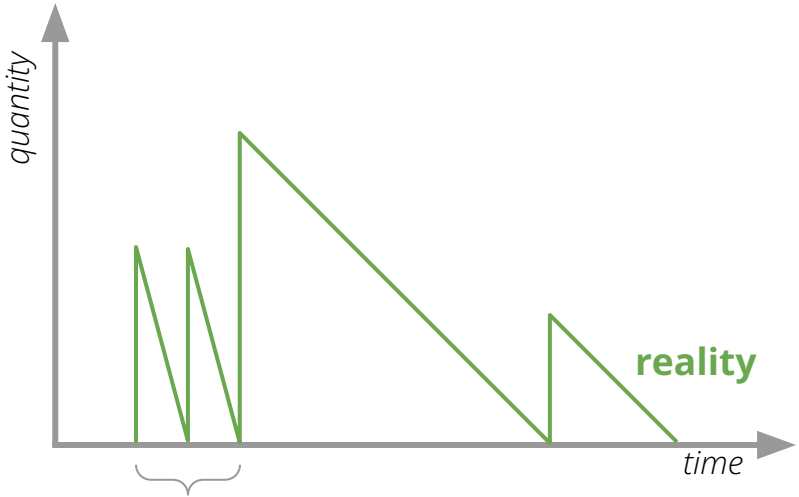


Challenges - “invisible” purchases

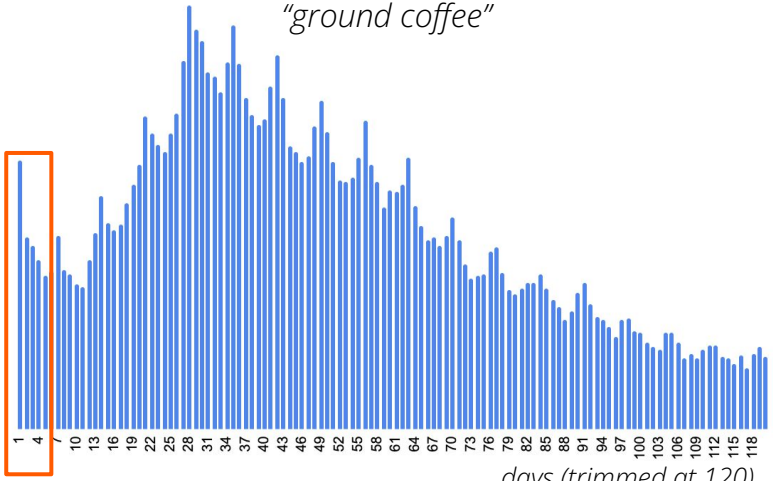


Our solution: use data of customers highly engaged in the modeled category (at least 3 distinct transactional days in the category in last 9 months)

Challenges - anomalies in purchases



Day difference between consecutive orders in "ground coffee"



days (trimmed at 120)

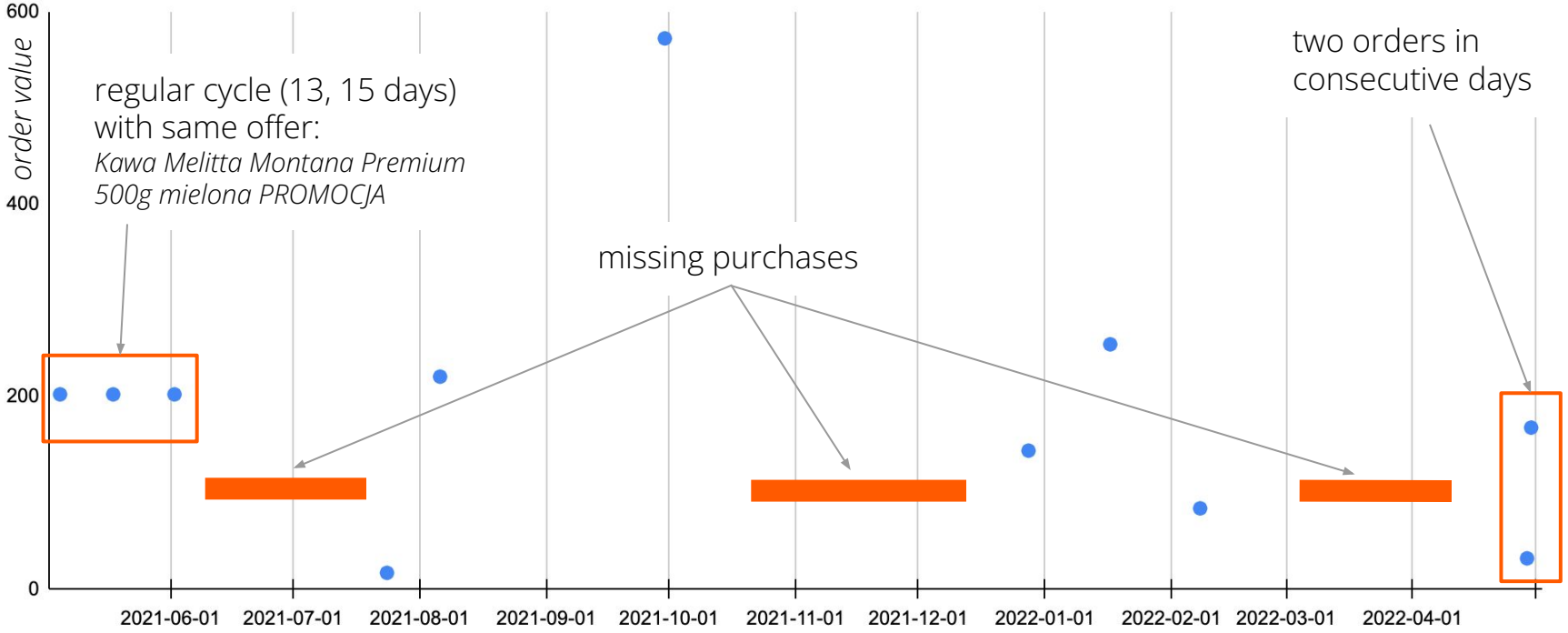
Purchases in same category in short intervals (e.g. buying coffee everyday/every two days)

Possible reasons:

- returns
- long delivery time for first order
- volume underestimation
- our marketing actions

Our solution: remove customers with irregularities
Goal: aggregate very close purchases

Real customer - ground coffee orders



Challenges - data quality

Ground coffee category examples

Offer title	Weight with package	Size (selected from list)	Weight	Status
KAWA MIELONA ARABICA 100% ESPRESSO FAIR TRADE BIO	[brak]	inna wartość	300	OK
Kawa mielona illy Moka Classico **6x250g**	[brak]	250 g	1500	OK
KAWA MIELONA ARABICA 100% FAIR TRADE BIO **500 g**	0.56	500 g	560	Different units, inconsistency
Kawa mielona z wanilią BIO **125g**	[brak]	200 g	140	Inconsistency
Kawa mielona Kahlua HAZELNUT **340 G**	[brak]	inna wartość	[brak]	no weight

Our solution: heuristics and use median weight/count/volume in category instead
Goal: extract data from offer titles - regular expressions/NER models (tbc)

Time to next purchase prediction

First approach - survival analysis

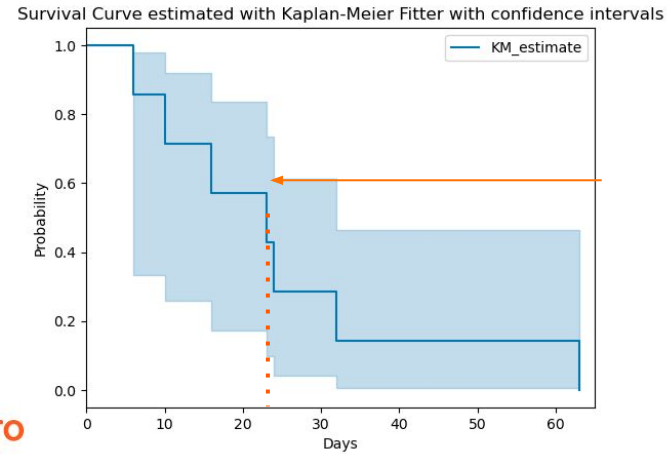
"How long until the next purchase?"

Per category approach:

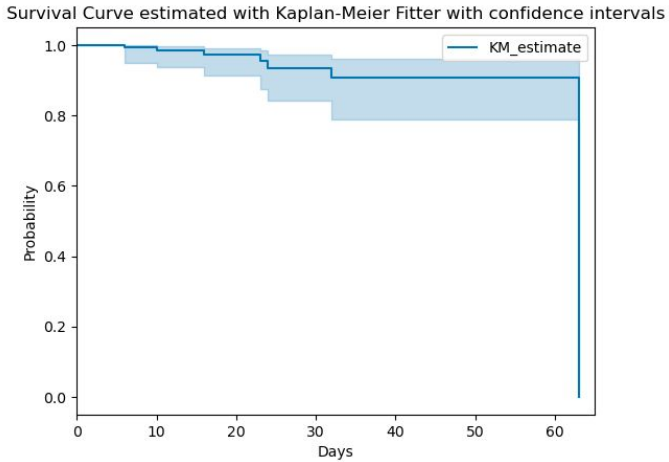
- Probability of the purchase in the following days
- Setting thresholds for campaigns
- Wide confidence interval for the entire category

Per customer approach:

- Probability of the purchase in the following days
- Setting thresholds for campaigns
- Narrow confidence interval for most of the regular buyers



selected threshold



Survival Analysis - Challenges

Analytical:

- customer based approach is more accurate, but with high variance
- setting thresholds for customers/segments/categories
- constant model re-training required based on the entire history of orders

Technical:

- duplication of rows in the data (we need description of every day for each customer)
- 1 customer = 1 model

Summary:

- Hard to maintain
- A lot of models

Duplication of rows

Input data

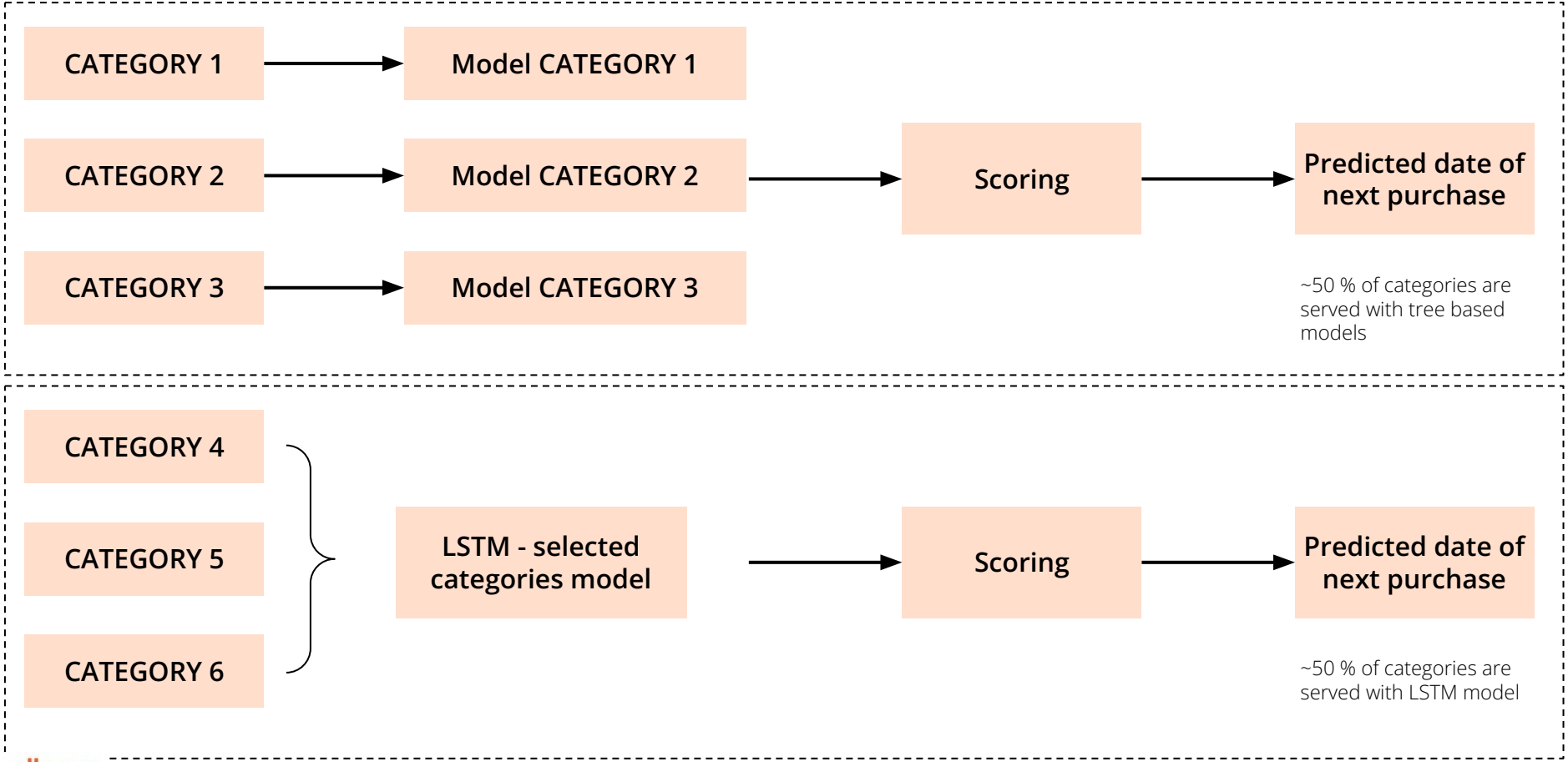
	Data/Customer				
Customer	**Time**	**Quantity**	**Usage speed**	**Order**	**Customer data**
123	3 days	2 kg	2/17	1	.....
123	2	0 kg	2/17	0	.....
123	1	0 kg	2/17	0	.....
123	8	3 kg	4/17	1	.....
123	...	...	...	...	.....

Dane wyjściowe

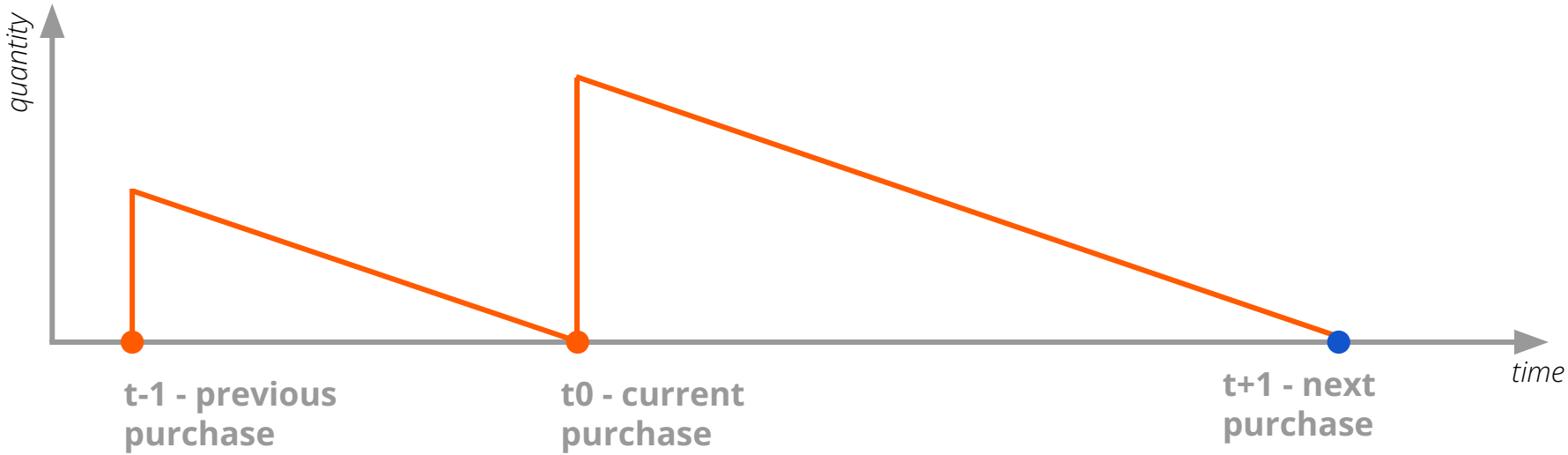
Day/Customer	123	456	789	...
1	~~1.000~~	~~1.000~~	~~1.000~~	...
2	0.980	0.982	0.983	...
3	0.980	0.981	0.982	...
4	0.980	0.981	0.982	...
5	0.980	0.981	0.982	...
6	0.947	0.949	0.982	...
...	...	...	...	...

X customers
*
Y days
*
Z categories

Regression models fand LSTM model



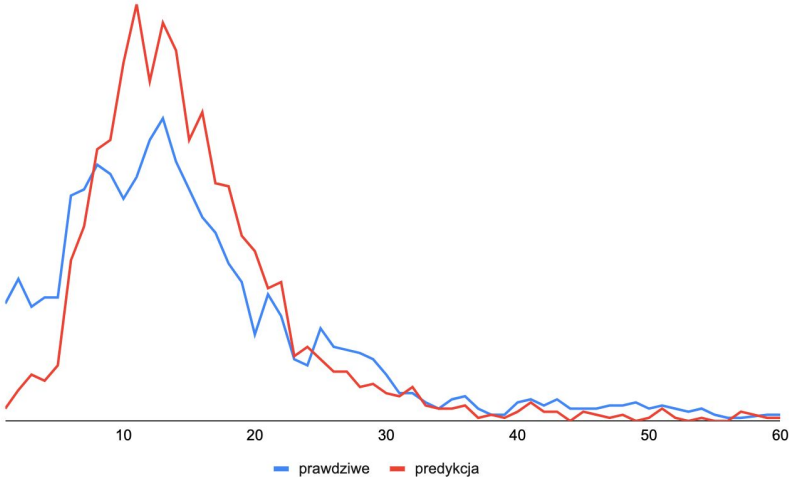
Data for models



	Previous purchase			Current purchase	Customer data	Target
customer	**time to next p**	**quantity**	**usage speed**	**quantity**	**Geo, demographic, transactions, ...**	**Time to next purchase**
123	17 dni	2 kg	2/17	3 kg	...	20 dni

Challenges - distribution of predictions vs real data

- Difference in predictions distribution is noticeable
- Tested methods:
  - survival analysis
  - different loss functions
  - data transformations
  - models for subgroups



Plans:

- more data per category
- advanced feature engineering
- seasonal features

Tech Stack

Technical implementation

Data Processing

BigQuery

+ serverless
+ fast and well optimized
+ cost
- maintenance of SQLs

Modeling

Python

VertexAI

- one category = one model
- hyper parameters tuning
- tested several different models
- AutoML tests

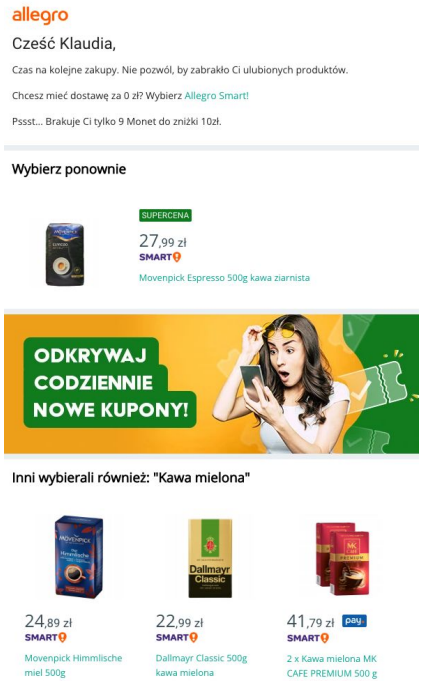
Deployment and Reporting

Cloud composer (Airflow)

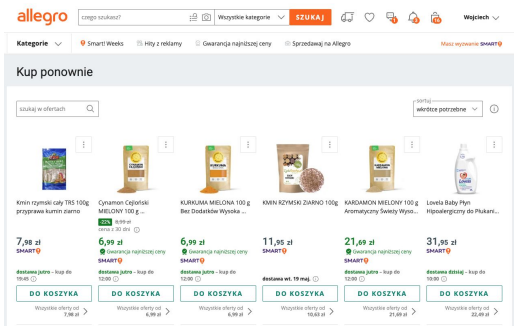
- scheduled daily
- alerts on slack
- models quality checks
- campaign metrics

Application of the solution

Business application



- email and push notifications sent to users in more than 400 different categories
- dedicated recurring purchases space on the platform



Room for improvements

- feature engineering + new models architectures for verification
- NER model for robust parameters retrieval from the data