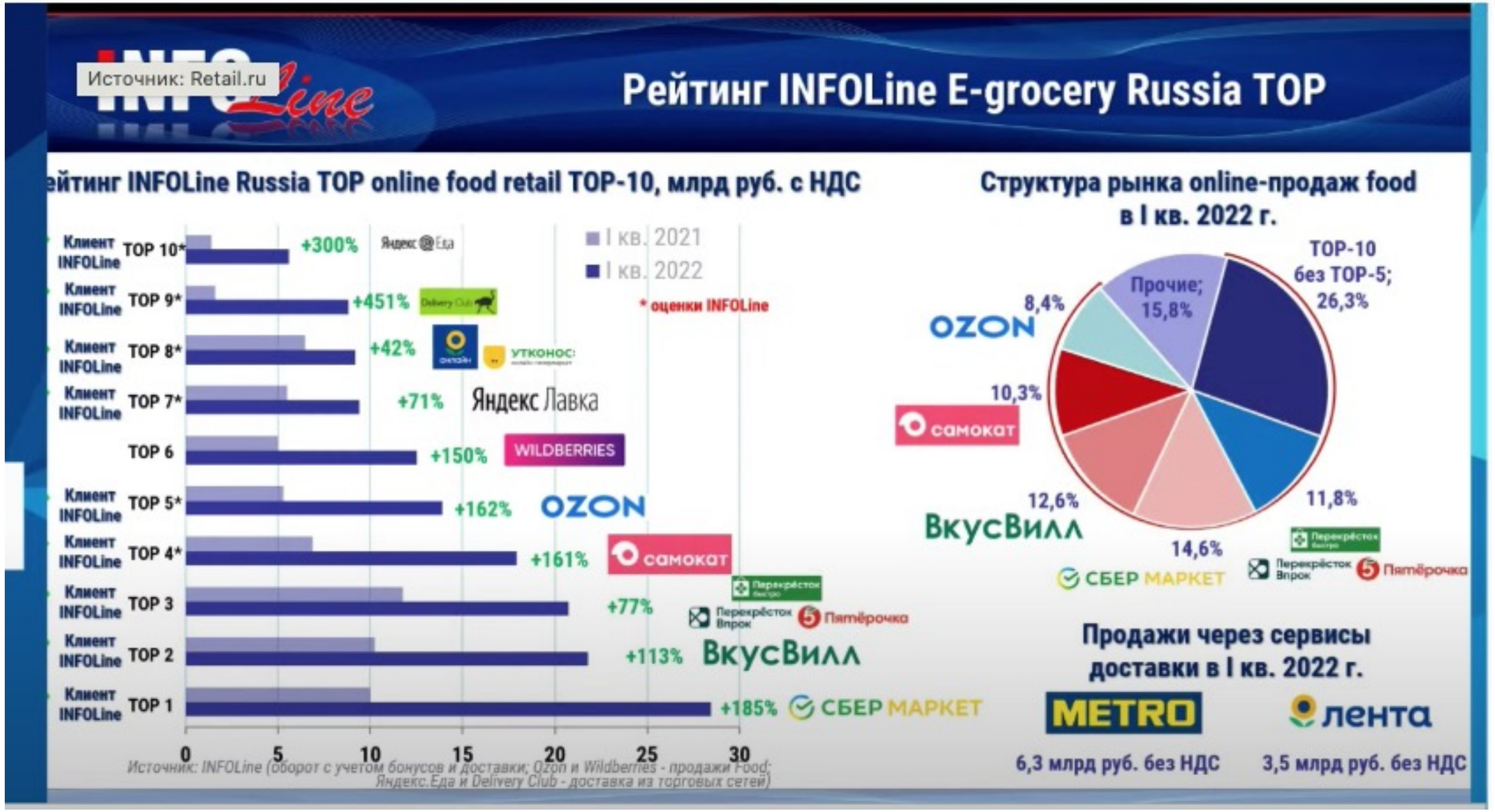
Рекомендации на корзинах

Лашинин Олег
09.04.24

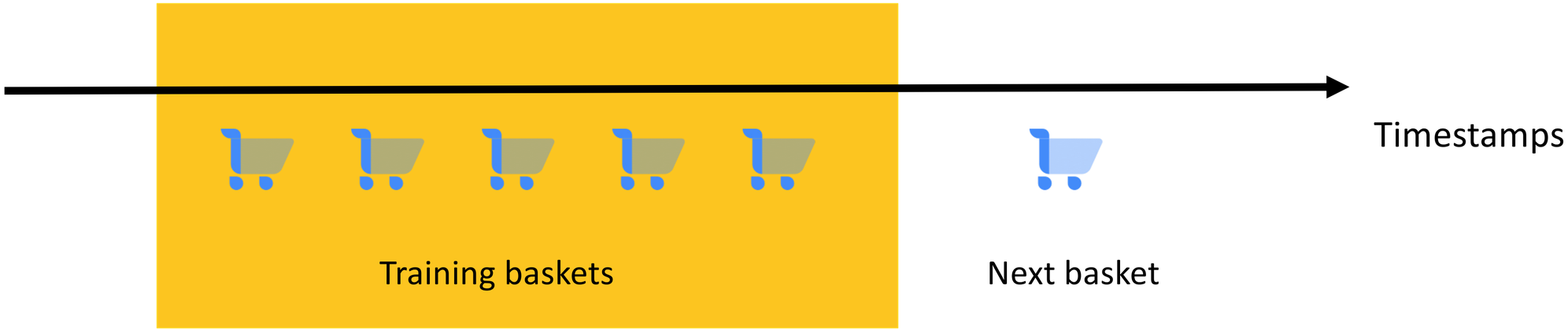
План занятия

- Рекомендации следующей корзины
- Up sell рекомендации на корзине
- Рекомендации на карточках товара

Почему сегодня эта тема?



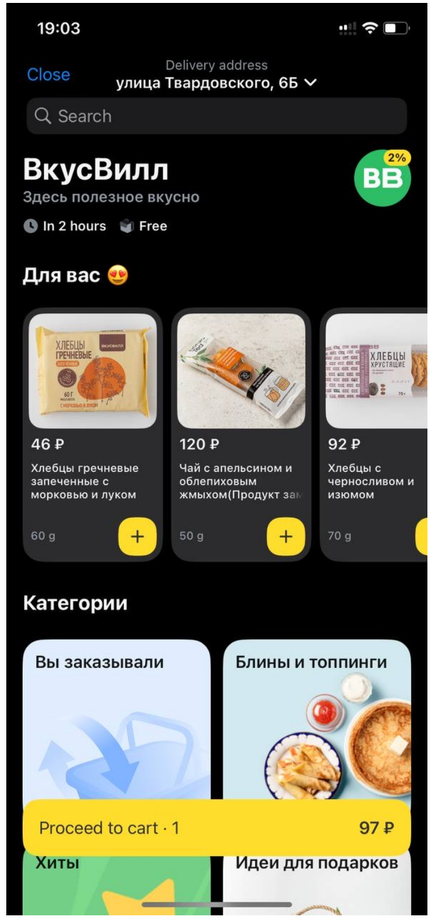
Next-Basket Recommendations



Особенности NBR

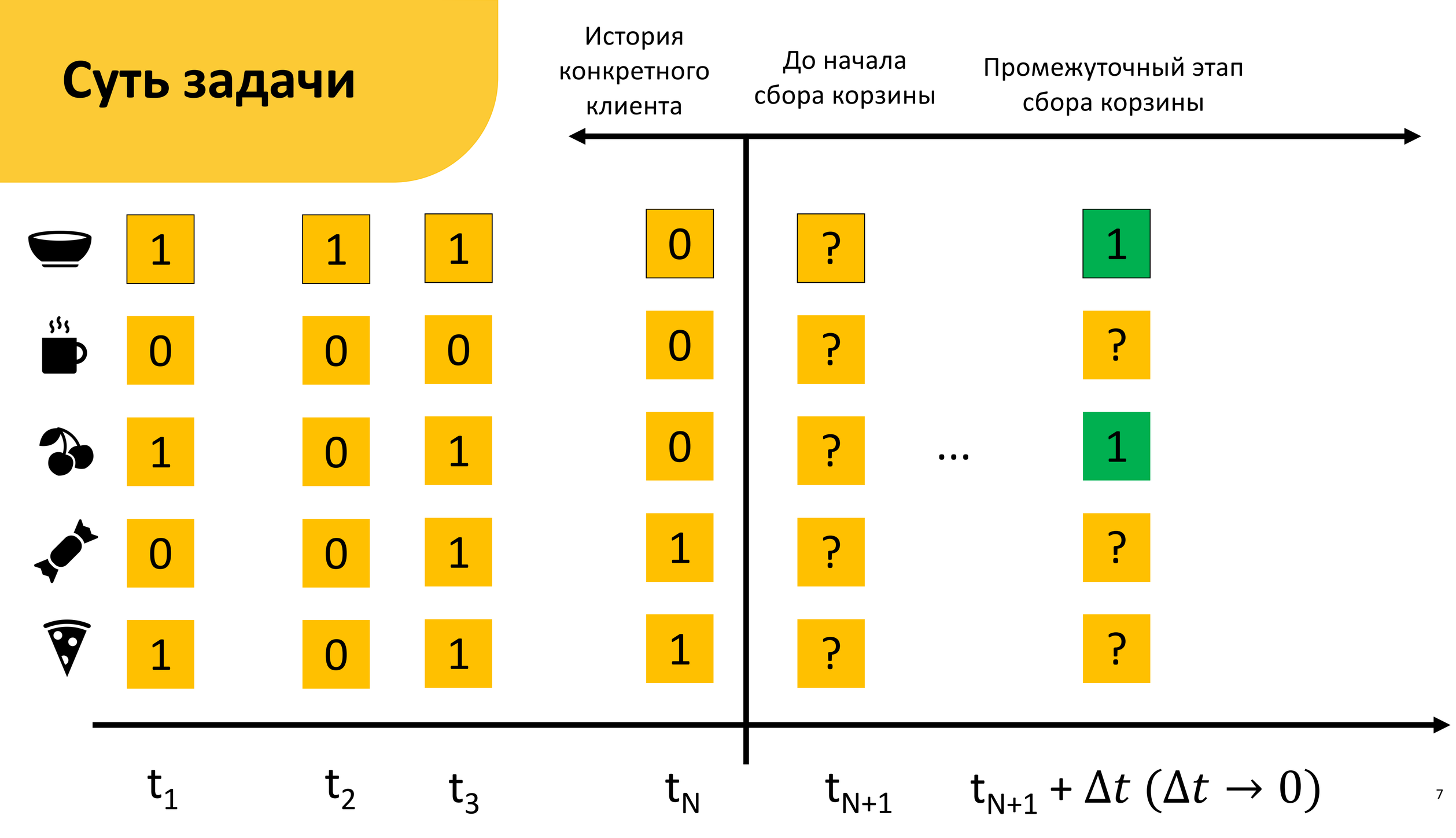
- Потребление товаров одномоментно
- Очень высокая степень повторов
- Товары покупаются «по списку», рекомендации бессильны

Интерфейсы: основное



- Рекомендации следующей корзины
- Рекомендации только повторов
- Рекомендации только новых
- Up sell рекомендации на корзине
- Рекомендации на карточках товара

Суть задачи


История конкретного клиента
До начала сбора корзины
Промежуточный этап сбора корзины
1 1 1 0 ? 1
0 0 0 0 ? ?
1 0 1 0 ? ... 1
0 0 1 1 ? ?
1 0 1 1 ? ?
t1 t2 t3 tN tN+1 tN+1 + Δt (Δt → 0)

Надо построить такую функцию/модель M,

что $\mathrm{Y}^{\mathrm{u}} = M(B_1^u, \dots, B_N^u) \to R^w$, и предложить

такой $M_{optimal}$, что $\mathrm{M}_{optimal} = max_M \sum_u Q(Y^u, B_{N+1}^u, K)$

$$Q(Y^u, B_{(N+1)}{}^u, K) = (|top_K(Y^{\mathrm{u}}) \cap B_{(N+1)}{}^u|)/|B_{(N+1)}{}^u|$$

Другими словами, надо предсказать как можно больше элементов $\mathrm{B}_{\mathrm{N+1}}$,
где предсказание есть K наибольших компонент вектора Y.

- W – размер каталога товаров
- K – длина списка рекомендаций
- $u \in U$ – множество пользователей в наборе данных
- B_i^u - корзина с порядковым номеров I пользователя u.
- Корзина - неупорядоченный набор товаров каталога I, $|I| = W$

Две эвристики

$S(u, i) = p(u, i).\ p(u, i)$ – популярность от (0, 1)

Две эвристики

S(u, i) = p(u, i). p(u, i) – популярность от (0, 1) (TopPopular, GPop)

S(u, i) = F(u, i) + p(u, i). F(u, i) – частота покупок товара I юзером u.

(TopPersonal, GP-pop)

Про популярность отдельно

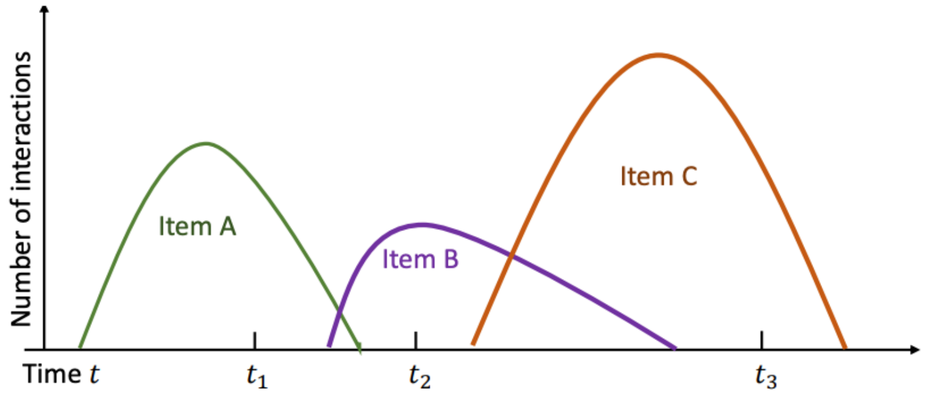


Figure 1: Popularity of items A, B, and C over time.

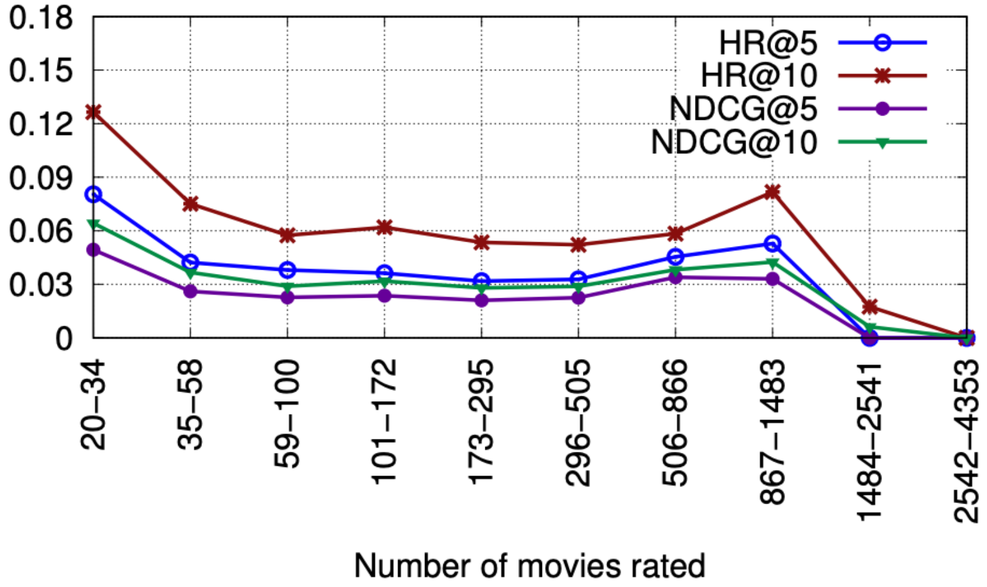


Table 1: Results of popularity methods; best results in bold

Popularity	HR@5	HR@10	NDCG@5	NDCG@10
MostPop	0.0304	0.0462	0.0198	0.0248
RecentPop	0.0530	**0.0845**	0.0338	0.0440
DecayPop	**0.0532**	0.0843	**0.0341**	**0.0441**

Три эвристики

S(u, i) = p(u, i). p(u, i) – популярность от (0, 1) (TopPopular, GPop)

S(u, i) = F(u, i) + p(u, i). F(u, i) – частота покупок товара I юзером u.

(TopPersonal, GP-pop)

S(u, i, t) = F(u, i) + p(u, i, t) – тут вы накручиваете эвристики

Давайте сделаем персонально

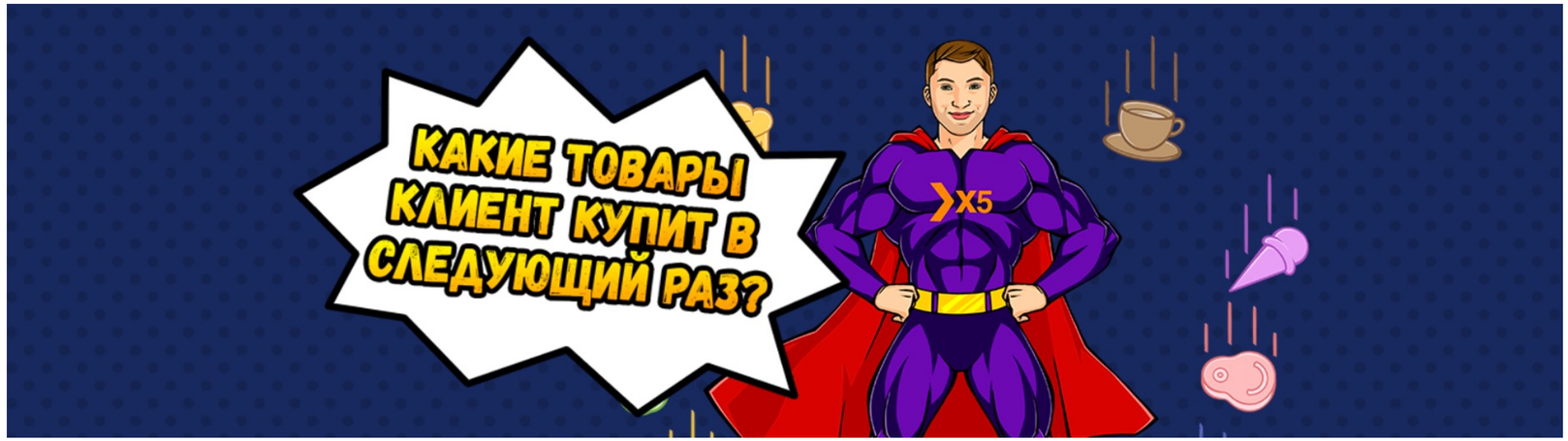
$$S(u, i, t) = F(u, i) + p(u, i, t)$$

Давайте сделаем персонально

$$S(u, i, t) = F(u, i) + s(u, i, t)$$

$s(u, i, t)$ – скор релевантности от 0 до 1

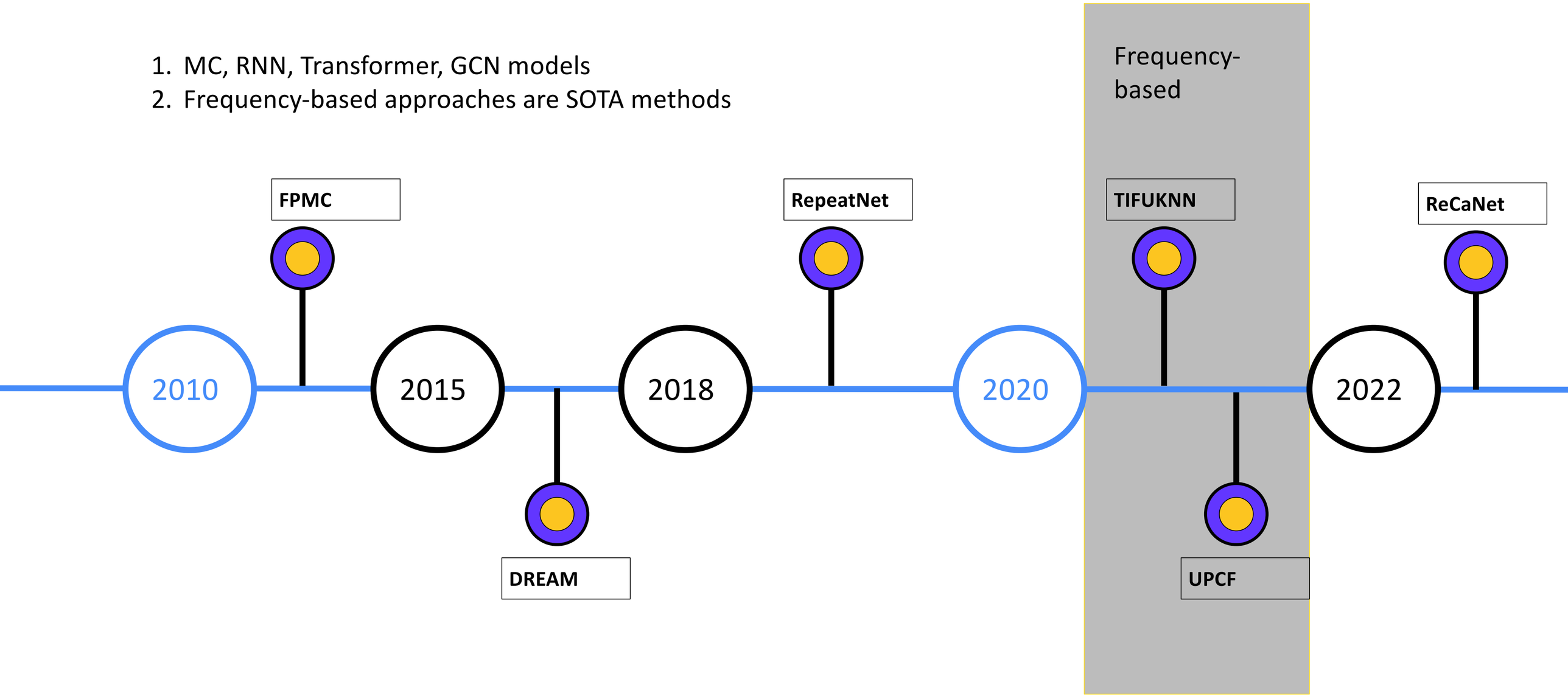
ML vs heuristics



#	Команды (83)		Score	Последний	Решений
1	aprotopopov		0,1483248823	3y	88
17	😎		0,133770888	3y	16

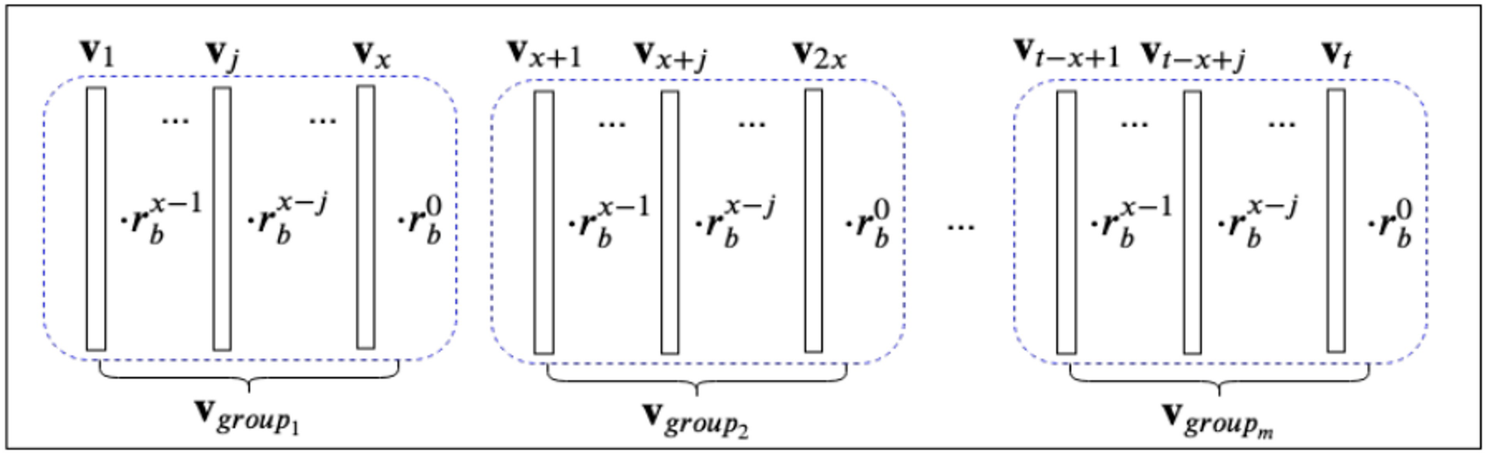
Related work

1. MC, RNN, Transformer, GCN models
2. Frequency-based approaches are SOTA methods

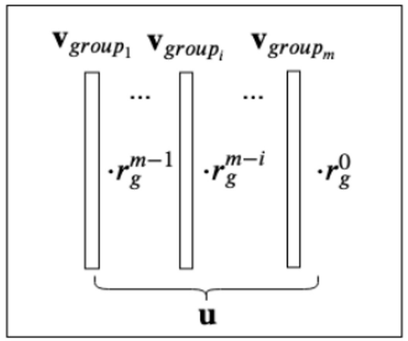


TIFU-KNN

Intra-group weights

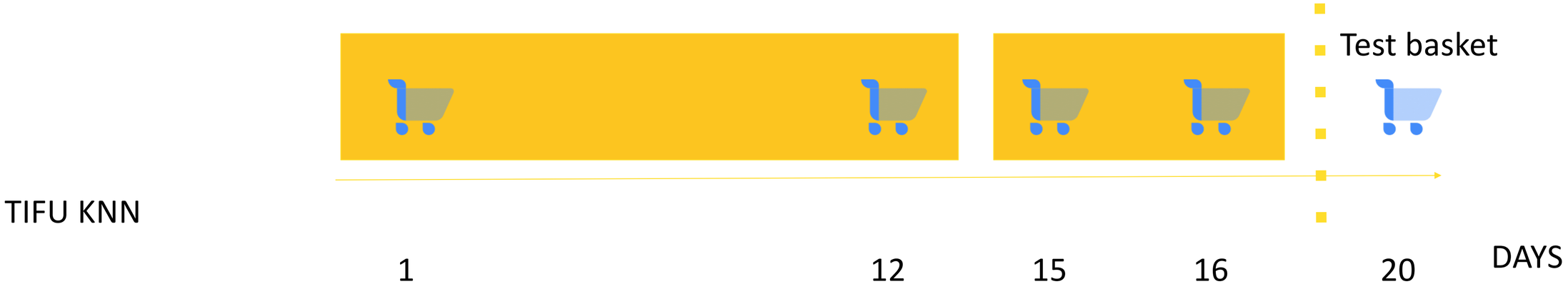


Inter-group weights

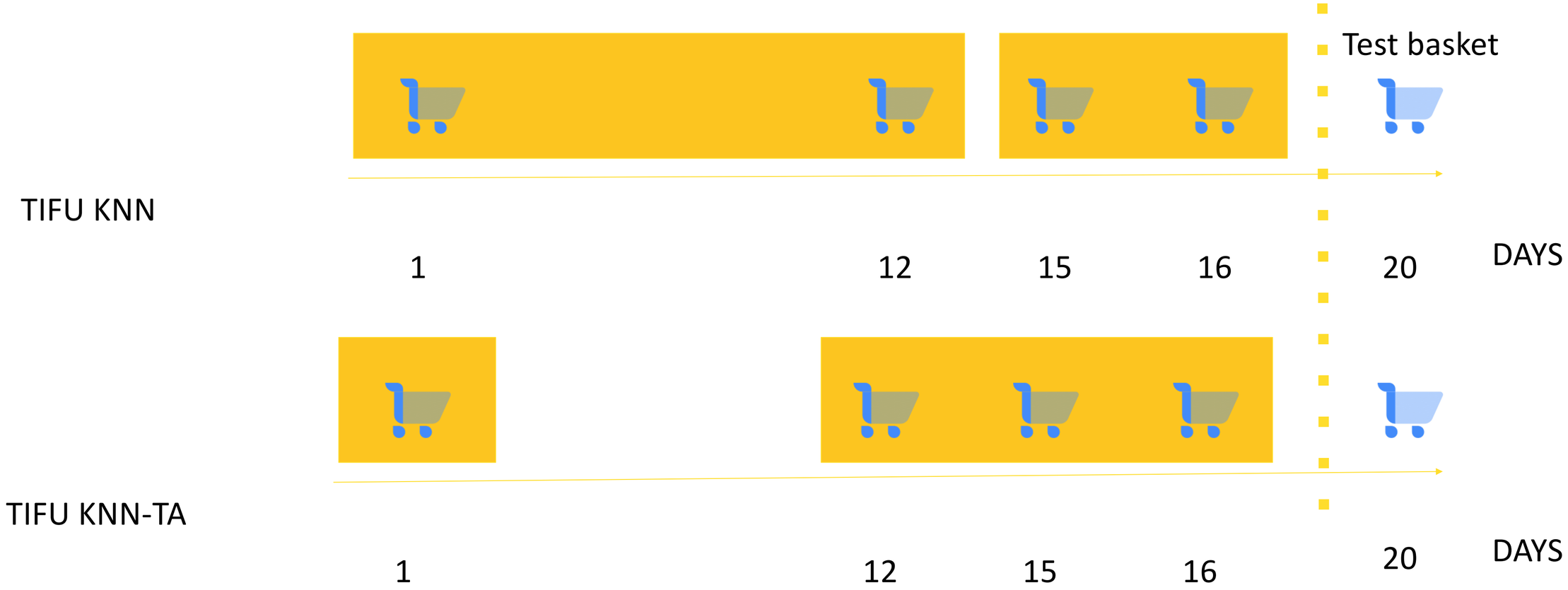


$$P(u) = \alpha \cdot v_u^{new} + (1 - \alpha) \cdot v_{nn}(v_u^{new})$$

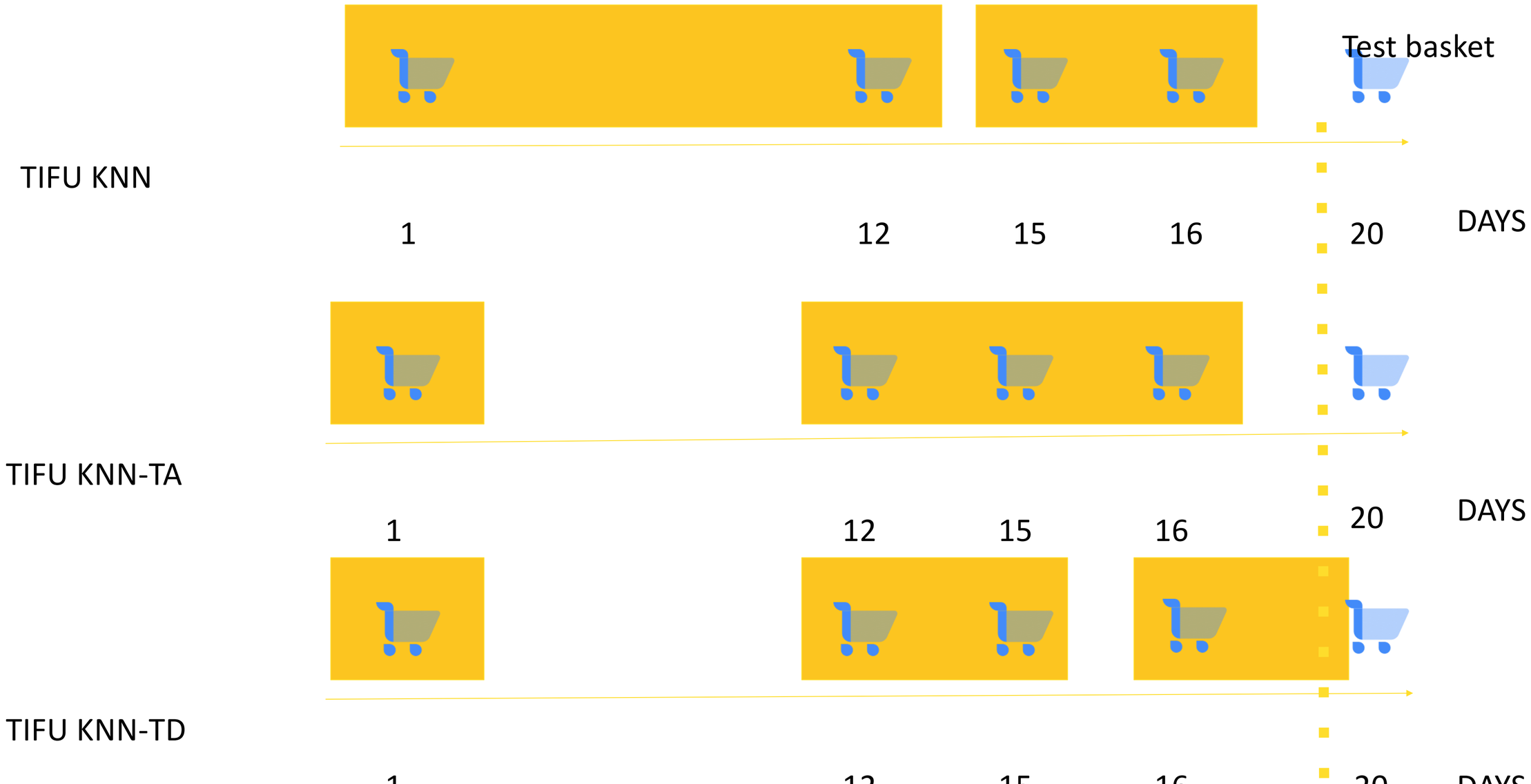
Limitations of TIFU-KNN



Limitations of TIFU-KNN



Limitations of TIFU-KNN



Experiments

Datasets:

Dataset	#users	#items	#baskets	#baskets per user	#items per basket	#items per user
Dunnhumby	2471	8644	251361	101.72	7.71	381.09
Tafeng	14006	13674	94274	6.73	6.34	37.61
Instacart	19999	26677	629067	31.45	9.94	100.22

Evaluation protocol: leave-one basket

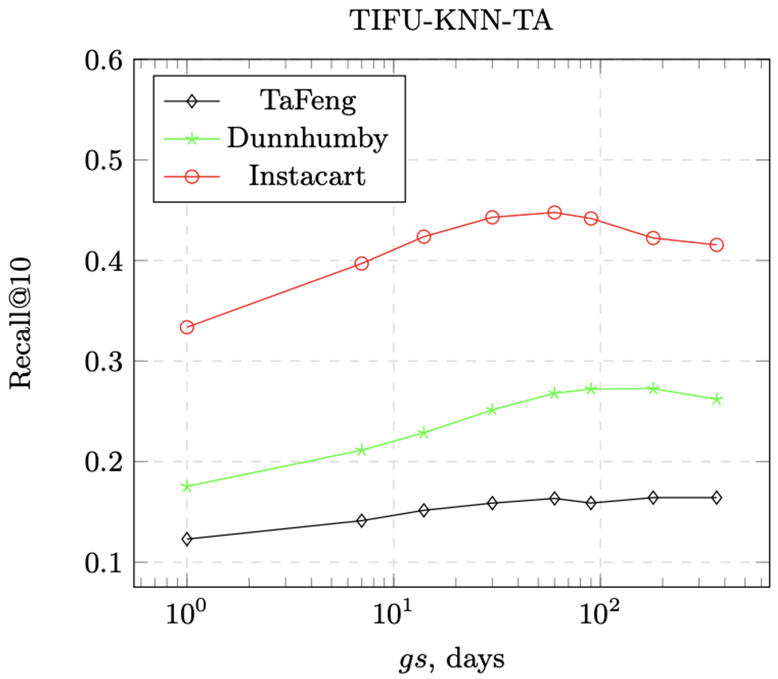
Metrics: Recall@k and NDCG@K

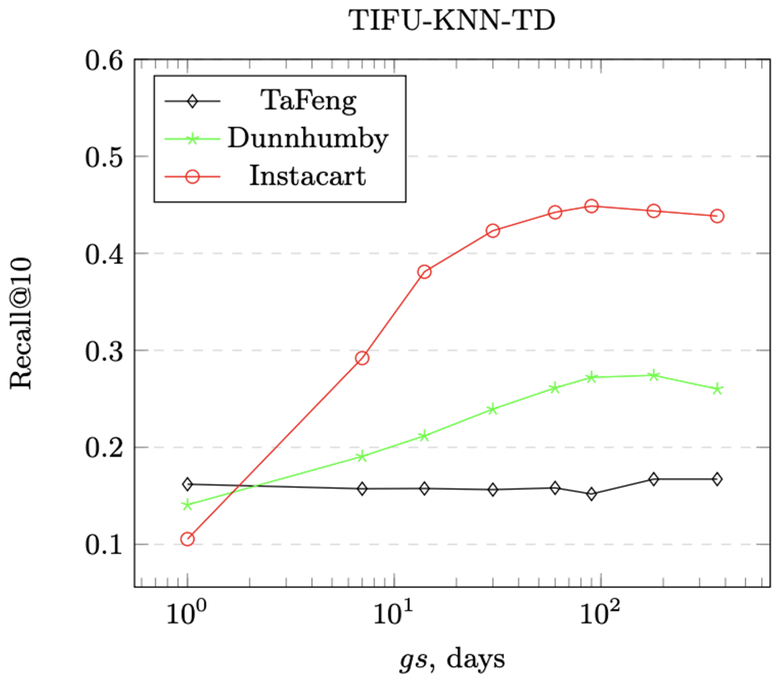
Hyperparameters tuning: Optuna with 300 trials

Experiments

Dataset	Metric	Baselines				Ours	
		G-Pop	GP-Pop	UP-CF@r	TIFU-KNN	TIFU-KNN--TA (▲%)	TIFU-KNN--TD (▲%)
DHB	Recall@5	0.1379	0.2326	0.2397	0.2491	**0.2572** (3.3)	0.2570 (3.2)
	nDCG@5	0.1229	0.2222	0.2294	0.2355	**0.2433** (3.3)	0.2422 (2.8)
	Recall@10	0.1359	0.2473	0.2611	0.2709	**0.2760** (1.9)	0.2743 (1.3)
	nDCG@10	0.1158	0.2188	0.2298	0.2384	**0.2439** (2.3)	0.2425 (1.7)
TaFeng	Recall@5	0.0815	0.1026	0.1244	0.1403	0.1415 (0.9)	**0.1448** (3.2)
	nDCG@5	0.0895	0.0979	0.1121	0.1347	0.1341 (-0.4)	**0.1393** (3.4)
	Recall@10	0.0841	0.1260	0.1537	0.1632	0.1642 (0.6)	**0.1673** (2.5)
	nDCG@10	0.0877	0.1047	0.1227	0.1406	0.1401 (-0.4)	**0.1453** (3.3)
Instacart	Recall@5	0.1092	0.4070	0.4371	0.4524	0.4541 (0.4)	**0.4559** (0.8)
	nDCG@5	0.1183	0.4238	0.4527	0.4668	0.4691 (0.5)	**0.4725** (1.2)
	Recall@10	0.0969	0.4000	0.4276	0.4476	0.4469 (-0.2)	**0.4496** (0.4)
	nDCG@10	0.1051	0.4039	0.4320	0.4484	0.4493 (0.2)	**0.4526** (0.9)

Experiments





gs = 4 days

1 12 15 16 20 DAYS

Очень много повторов

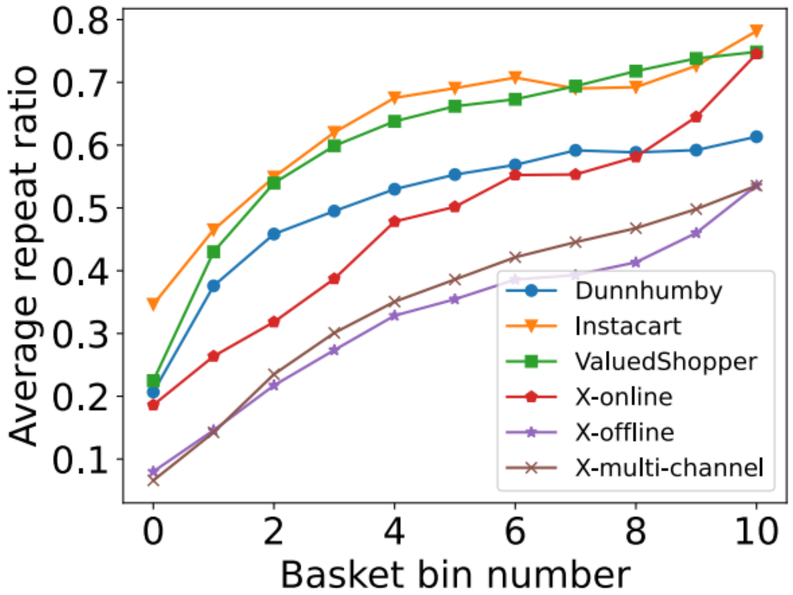


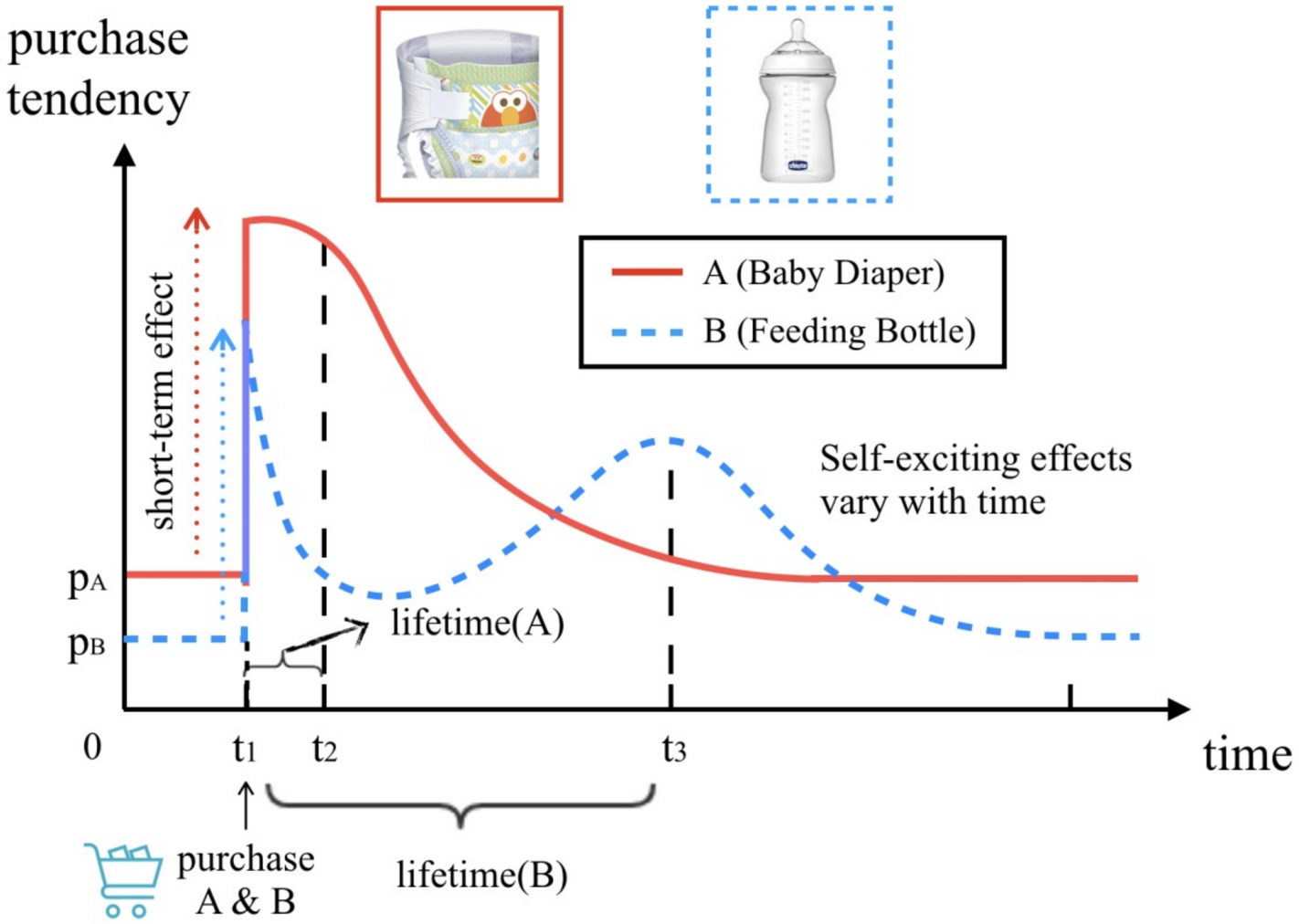
Figure 1: Distribution of repeat ratio in the baskets of users for all datasets.

Частотные методы лучшие

Dataset		Instacart						
K	Approaches	Recall	Precision	F1-score	PHR	MAP	MRR	NDCG
5	TOP	0.0487	0.0956	0.0581	0.3668	0.2270	0.2321	0.0666
	FPMC	0.0481	0.0948	0.0620	0.3600	0.2211	0.2264	0.0651
	DREAM	0.0763	0.0486	0.0594	0.2176	0.1543	0.1481	0.0754
	Beacon	0.0539	0.1049	0.0638	0.3930	0.2181	0.2250	0.0695
	Sets2Sets	0.1266	0.1652	0.1209	0.5503	0.3141	0.3260	0.1375
	UP-CF@r	0.2512	0.3580	0.2512	0.7946	0.5818	0.6138	0.2970
	TIFUKNN	**0.2616**	**0.3733**	**0.2619**	**0.8052**	**0.5992**	**0.6312**	**0.3094**
	CLEA	0.1850	0.3025	0.1973	0.7136	0.5623	0.5865	0.2450

Dataset		Dunnhumby						
K	Approaches	Recall	Precision	F1-score	PHR	MAP	MRR	NDCG
5	TOP	0.0966	0.0763	0.0606	0.3294	0.2190	0.2282	0.0861
	FPMC	0.0746	0.1033	0.0866	0.3928	0.2566	0.2674	0.0903
	DREAM	0.0756	0.1049	0.0879	0.3984	0.2616	0.2726	0.0918
	Beacon	0.0795	0.1038	0.0737	0.3948	0.2517	0.2682	0.0932
	Sets2Sets	0.1040	0.1249	0.0904	0.4353	0.2515	0.2867	0.1112
	UP-CF@r	**0.1794**	**0.2417**	**0.1693**	**0.6005**	**0.4367**	**0.4602**	**0.2105**
	TIFUKNN	0.1725	0.2329	0.1630	0.5947	0.4260	0.4465	0.2027
	CLEA	0.1193	0.1720	0.1165	0.5042	0.3512	0.3663	0.1422

Копнем глубже в поведение пользователей



Копнем глубже в поведение пользователей

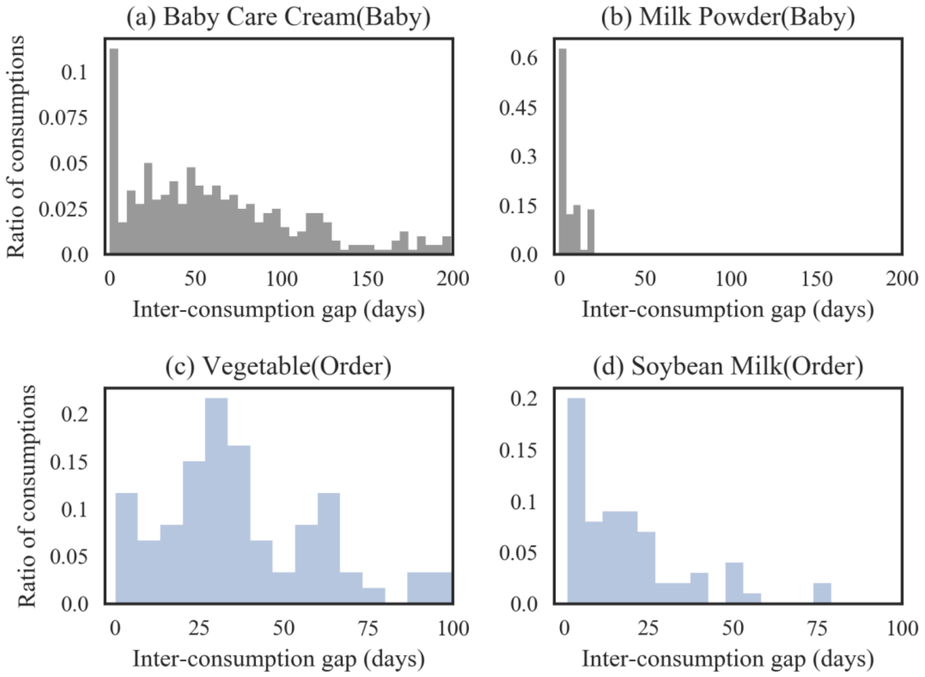
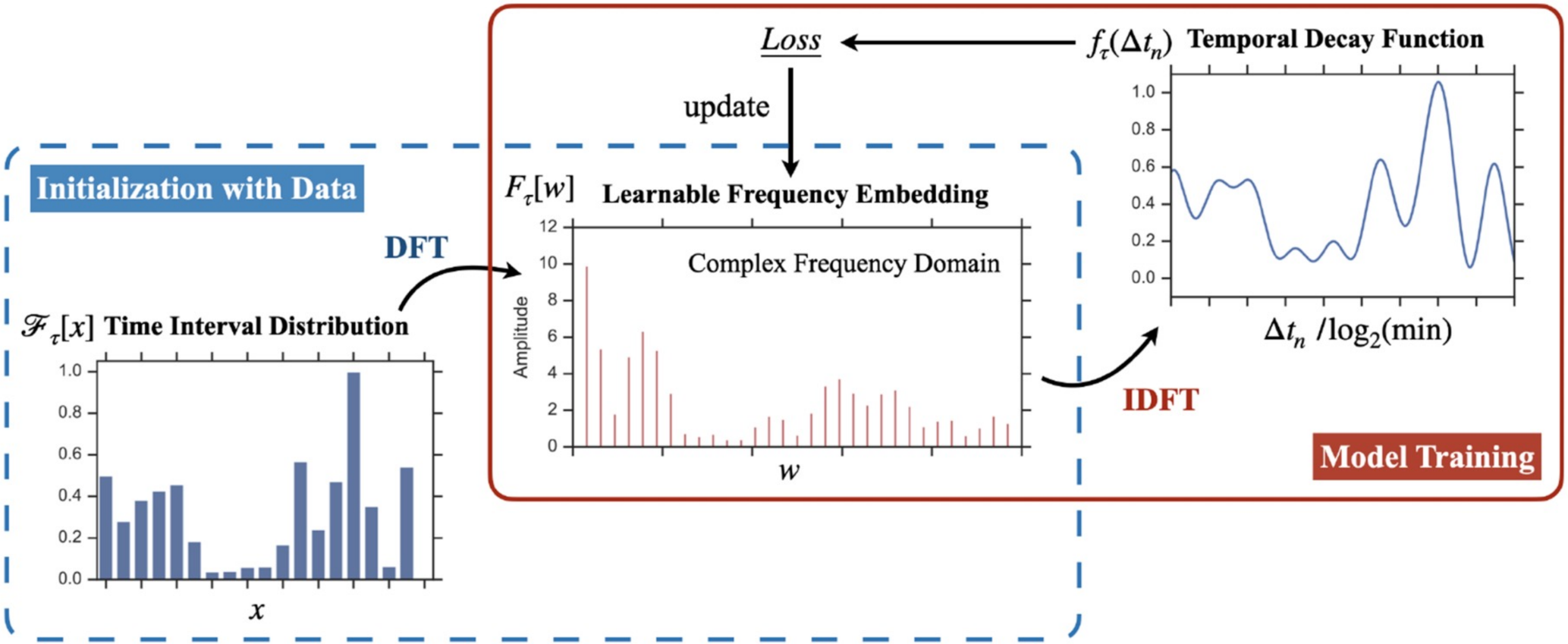


Figure 3: Inter-consumption gap distributions of different items. Various types of items inherently have distinct temporal dynamics of repeat consumption.

$$\lambda(t) = \lambda_0 + \alpha \sum_{t_j < t} \gamma(t - t_j) \,,$$

$$\gamma_i(\Delta t) = \overbrace{\pi_0^i E(\Delta t|\, 1/\beta^i)}^{short-term} + \overbrace{\sum_{z \in [1,Z]} \pi_z^i N(\Delta t|\, \mu_z^i, \sigma_z^i)}^{life-time} \,.$$

Копнем глубже в поведение пользователей



Что мы можем делать

$$\gamma_i(\Delta t) = \pi^i E(\Delta t|\ \beta^i) + (1 - \pi^i)\mathcal{N}(\Delta t|\ \mu^i, \sigma^i)$$

Что мы можем делать

$$\gamma_i(\Delta t) = \pi^i E(\Delta t|\ \beta^i) + (1 - \pi^i)\mathcal{N}(\Delta t|\ \mu^i, \sigma^i)$$

$$\lambda_{u,i}(t^u_{|B^u|+1}) = \lambda^0_{u,i} + \alpha_i \sum_{(b^u_j, t^u_j) \in B^u:\ i \in b^u_j,\ t^u_j < t^u_{|B^u|+1}} \gamma_i(t^u_{|B^u|+1} - t^u_j)$$

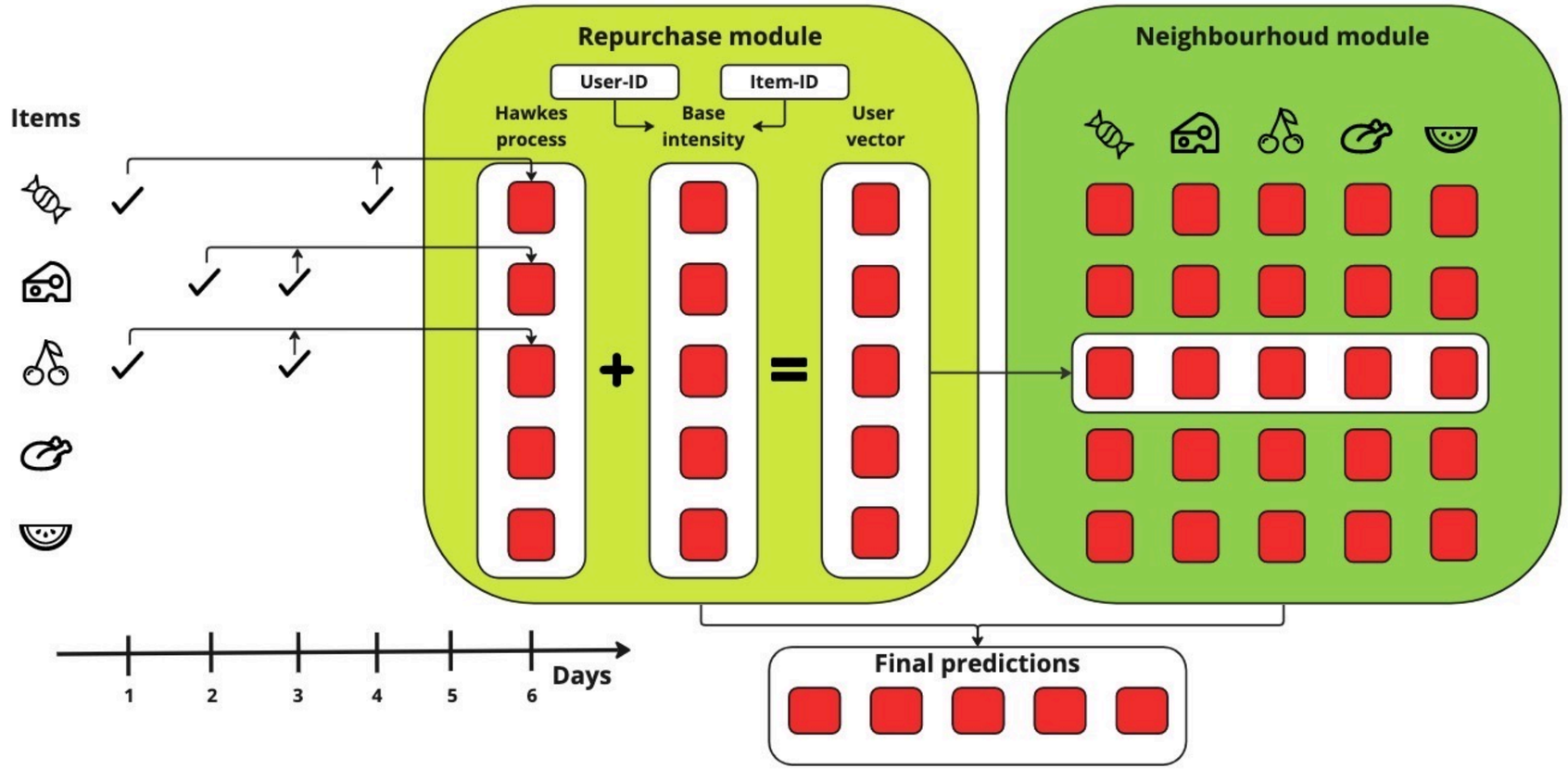
Что мы можем делать

$$\gamma_i(\Delta t) = \pi^i E(\Delta t|\ \beta^i) + (1 - \pi^i)\mathcal{N}(\Delta t|\ \mu^i, \sigma^i)$$

$$\lambda_{u,i}(t^u_{|B^u|+1}) = \lambda^0_{u,i} + \alpha_i \sum_{(b^u_j, t^u_j) \in B^u:\ i \in b^u_j,\ t^u_j < t^u_{|B^u|+1}} \gamma_i(t^u_{|B^u|+1} - t^u_j)$$

$$f(u, t^u_{|B^u|+1}) = (\lambda_{u,1}(t^u_{|B^u|+1}), \ldots, \lambda_{u,|V|}(t^u_{|B^u|+1}))$$

Что мы можем делать



$$P_u = \alpha * f(u, t^u_{|B^u|+1}) + (1-\alpha) * \frac{1}{k} \sum_{\hat{u} \in kNN(u)} f(\hat{u}, t^{\hat{u}}_{|B^{\hat{u}}|})$$

Что мы можем делать

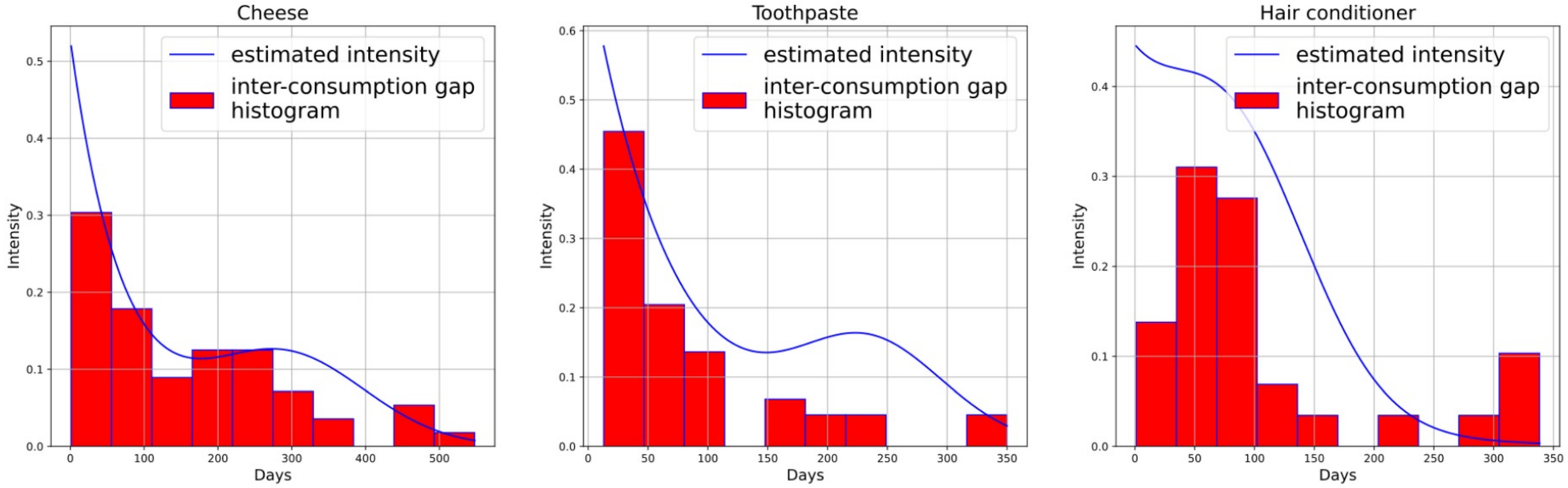
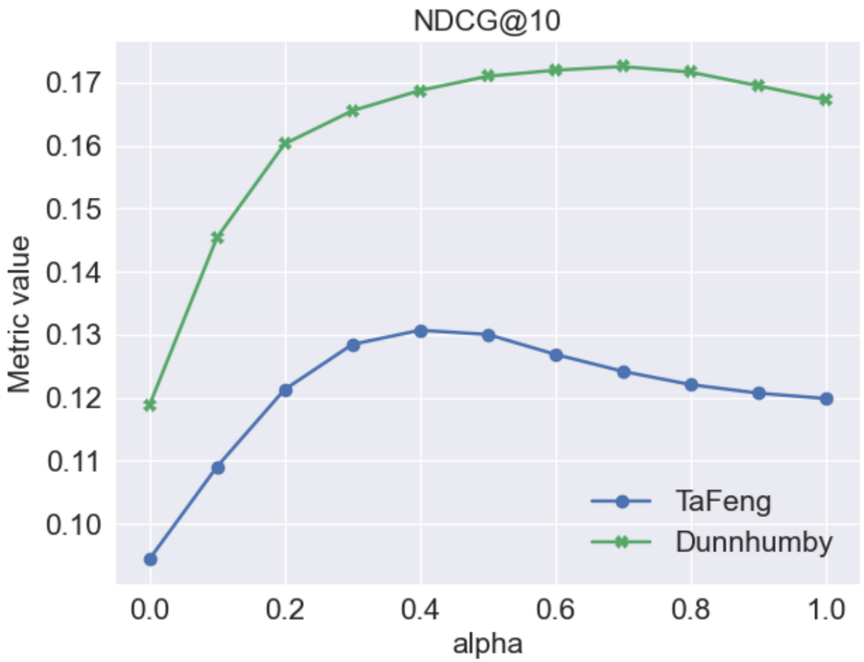


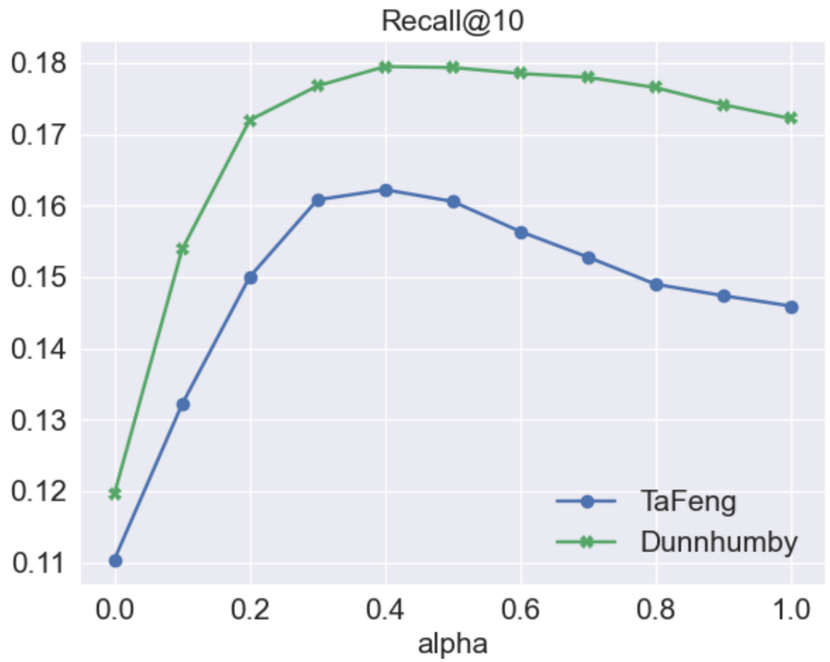
Figure 5: Inter-consumption gap distribution and estimated intensity functions for specific Dunnhumby items.

Что мы можем делать

Dataset	Baseline	Metrics Precision@10	Recall@10	NDCG@10
TaFeng	GP-Pop	0.0572	0.1308	0.1084
	TIFU-KNN	0.0574	0.1360	0.1179
	DNNTSP	0.0502	0.1306	0.1120
	SLRC	0.0621	0.1458	0.1194
	TIFU-KNN-TA	0.0615	0.1503	0.1248
	UPCF	0.0576	0.1365	0.1154
	TAIW	0.0644 (▲3.7%)	0.1565 (▲4.1%)	**0.1267** (▲1.5%)
	TAIWI	**0.0671** (▲8.1%)	**0.1642** (▲9.2%)	0.1261 (▲1.0%)
TaoBao	GP-Pop	0.0115	0.1116	0.0741
	TIFU-KNN	0.0077	0.0749	0.0524
	DNNTSP	0.0002	0.0016	0.0012
	SLRC	0.0116	0.1118	0.0801
	TIFU-KNN-TA	0.0078	0.0763	0.0543
	UPCF	0.0085	0.0824	0.0553
	TAIW	0.0122 (▲5.2%)	0.1177 (▲5.3%)	**0.0815** (▲1.7%)
	TAIWI	**0.0123** (▲6.0%)	**0.1190** (▲6.4%)	0.0810 (▲1.1%)
Dunnhumby	GP-Pop	0.1091	0.1577	0.1490
	TIFU-KNN	0.1157	0.1653	0.1613
	DNNTSP	0.0613	0.0929	0.0931
	SLRC	0.1192	0.1727	0.1675
	TIFU-KNN-TA	0.1162	0.1705	0.1593
	UPCF	0.1167	0.1663	0.1600
	TAIW	**0.1214** (▲1.8%)	**0.1791** (▲3.7%)	0.1706 (▲1.9%)
	TAIWI	0.1211 (▲1.6%)	0.1773 (▲2.7%)	**0.1713** (▲2.3%)

Что мы можем делать





$$P_u = \alpha * f(u, t^u_{|B^u|+1}) + (1-\alpha) * \frac{1}{k} \sum_{\hat{u} \in kNN(u)} f(\hat{u}, t^{\hat{u}}_{|B^{\hat{u}}|})$$

Предсказание только повторных покупок

Table 2: Performance of the oracle personal NBR model.

Type	Dataset	Recall@$\|B\|$	nDCG@$\|B\|$
Online	Instacart	0.6087	0.6945
	X-online	0.6214	0.7236
Offline	Dunnhumby	0.4488	0.5331
	ValuedShopper	0.7168	0.7815
	X-offline	0.3811	0.4843
Multi-channel	X-multi-channel	0.4959	0.6055

Предсказание только повторных покупок

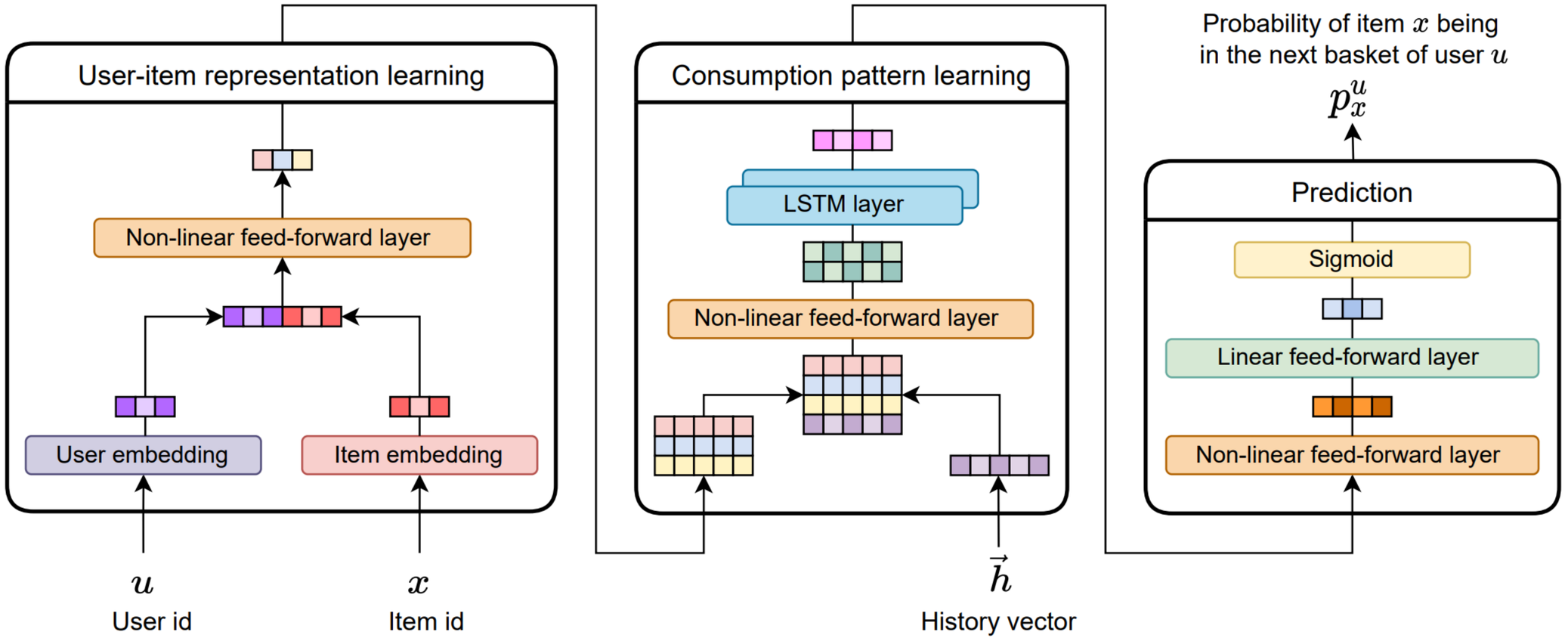


Figure 3: Overview of ReCANet's architecture.

Предсказание только повторных покупок

Dataset	Metric	Baselines					ReCANet (▲%)	
		P-Pop	GP-Pop	TIFU-KNN	UP-CF@r	DNNTSP		
Instacart	Recall@10	0.3222	0.3230	<u>0.3434</u>	0.3400	0.3427	**0.3592**† (4.6)	
	nDCG@10	0.3131	0.3134	0.3332	0.3319	<u>0.3336</u>	**0.3502**† (5.0)	
	Recall@|B|	0.2901	0.2904	<u>0.3115</u>	0.3077	0.3079	**0.3295**† (5.8)	
	nDCG@|B|	0.3299	0.3301	<u>0.3521</u>	0.3479	0.3480	**0.3717**† (5.6)	
X-online	Recall@10	0.1603	0.1603	0.1443	<u>0.1619</u>	0.1601	**0.1645**† (1.6)	
	nDCG@10	0.6283	0.6283	0.5771	<u>0.6361</u>	0.6334	**0.6462**† (1.6)	
	Recall@|B|	0.3613	0.3613	0.3009	<u>0.3680</u>	0.3679	**0.3735**† (1.5)	
	nDCG@|B|	0.4323	0.4323	0.3731	<u>0.4396</u>	0.4391	**0.4467**† (1.6)	
Dunnhumby	Recall@10	0.1315	0.1315	<u>0.1485</u>	0.1421	0.0892	**0.1621**† (9.2)	
	nDCG@10	0.1267	0.1267	<u>0.1379</u>	0.1364	0.0825	**0.1489**† (8.0)	
	Recall@|B|	0.1048	0.1048	<u>0.1244</u>	0.1173	0.0576	**0.1310**† (5.3)	
	nDCG@|B|	0.1216	0.1216	<u>0.1423</u>	0.1346	0.0685	**0.1503**† (5.6)	

Предсказание только новых покупок

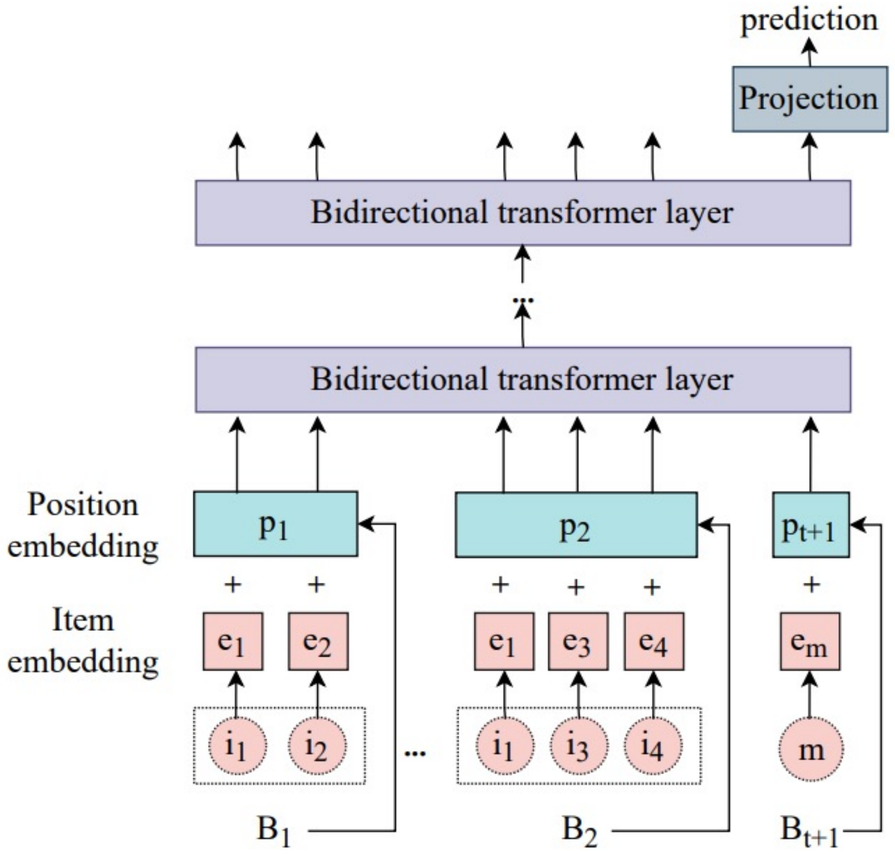


Figure 1: The overall architecture of the BTBR model.

Masked and Swapped Sequence Modeling for Next Novel Basket Recommendation in Grocery Shopping

Table 4: Results of methods training for finding novel items, i.e., Train-explore, compared against the methods training for finding all items, i.e., Train-all. Boldface and underline indicate the best and the second best performing performance w.r.t. the NNBR task, respectively. Significant improvements and deteriorations of Train-explore over the corresponding Train-all baseline results are marked with ↑ and ↓, respectively. (paired t-test, $p < 0.05$).

Dataset	Metric	Train	G-Pop	TIFUKNN	Dream	Beacon	CLEA	DNNTSP
Tafeng	Recall@10	all	0.0587	0.0714	0.0960	0.0926	0.0870	**0.1024**
		explore	=	0.0911↑	<u>0.1021</u>↑	0.0967↑	0.1010↑	0.0940↓
	nDCG@10	all	0.0603	0.0662	0.0823	0.0789	0.0755	<u>0.0855</u>
		explore	=	0.0783↑	**0.0859**↑	0.0819↑	0.0857↑	0.0767↓
	Recall@20	all	0.0874	0.0926	0.1244	0.1252	0.1150	0.1245
		explore	=	0.1157↑	0.1244	**0.1257**	<u>0.1253</u>↑	0.1168↓
	nDCG@20	all	0.0703	0.0738	0.0928	0.0909	0.0861	<u>0.0943</u>
		explore	=	0.0876↑	0.0939	0.0929	**0.0952**↑	0.0858↓
		all	0.0468	0.0497	0.0494	0.0499	0.0499	0.0514

Коррекция списков рекомендаций

(a) ( ,  , )

Shoe 1 Shoe 2 Shoe 3

Predicting top-K ranked items via a user represenation.

(b) ( →  → )

Shoe Jersey Sports drink

Generating the items one by one considering item correlations.

Generative NBR

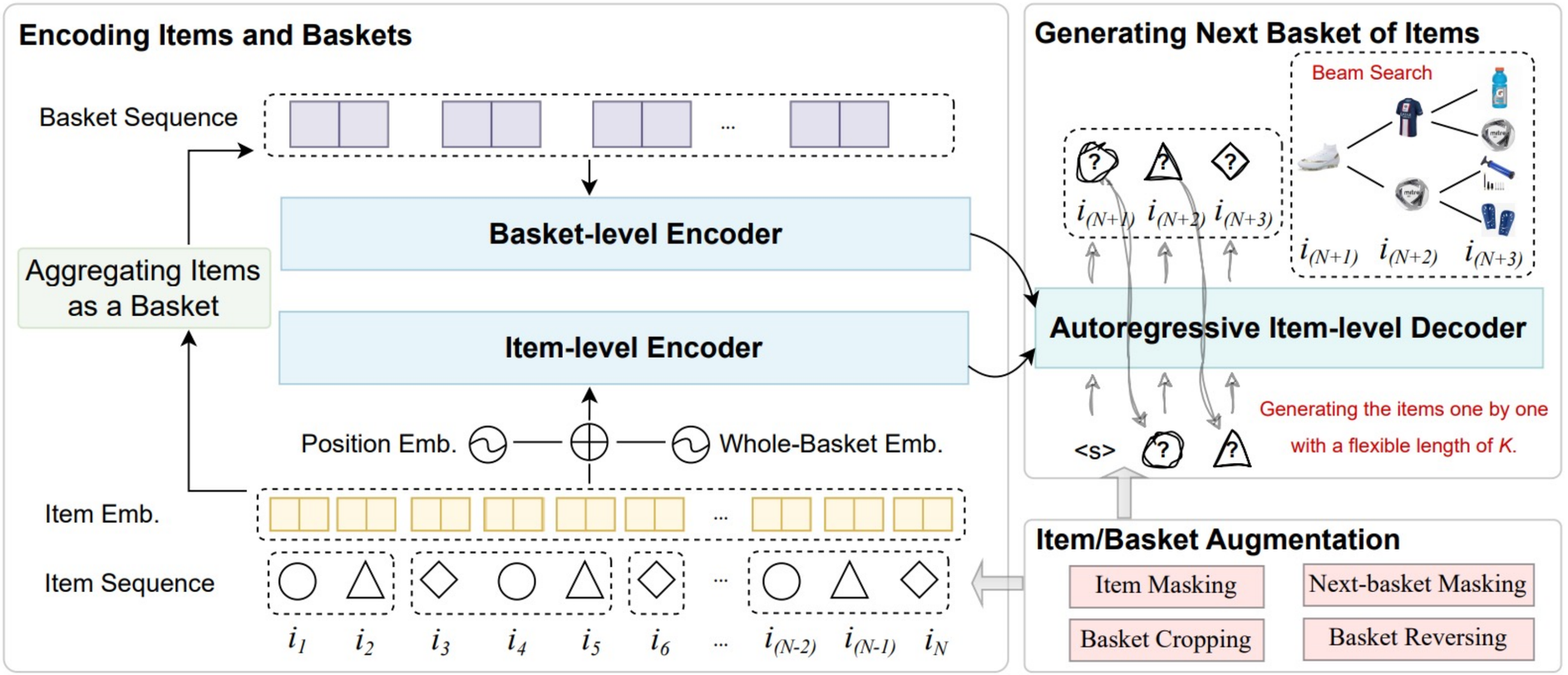


Figure 2: The overall framework of the proposed GenRec.

Коррекция списков рекомендаций (Beam Search)

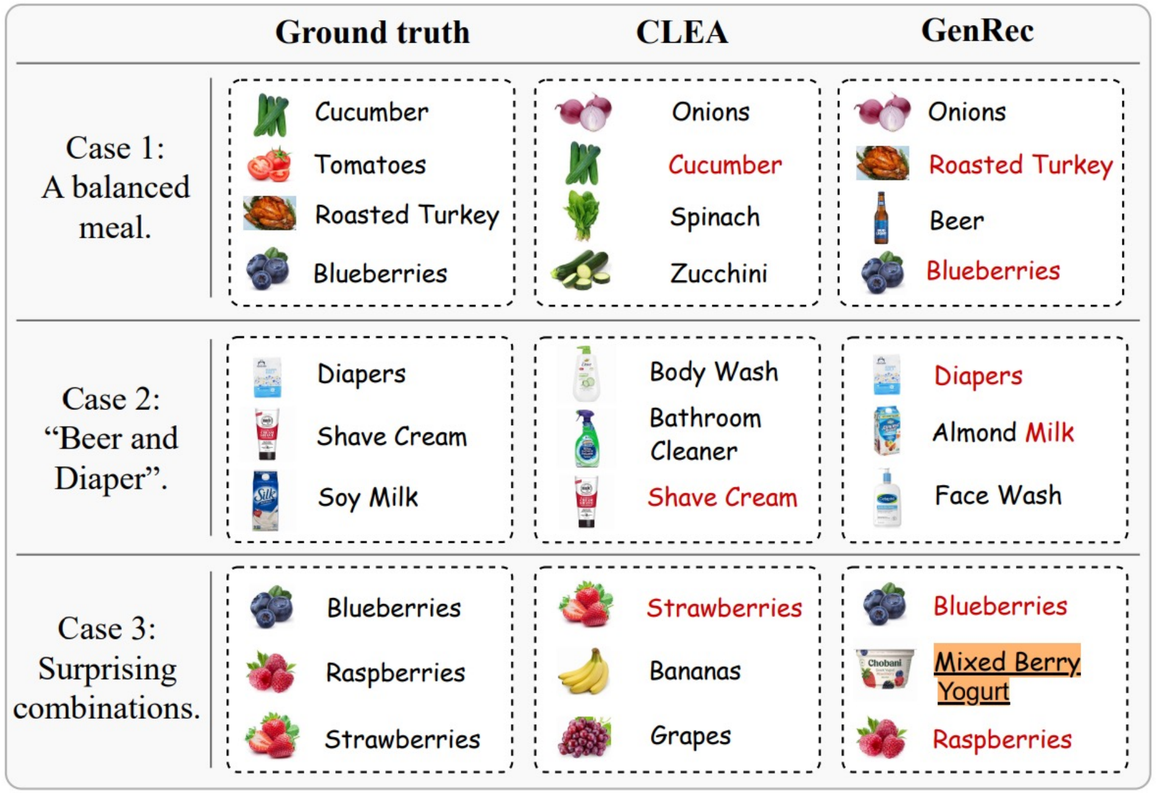
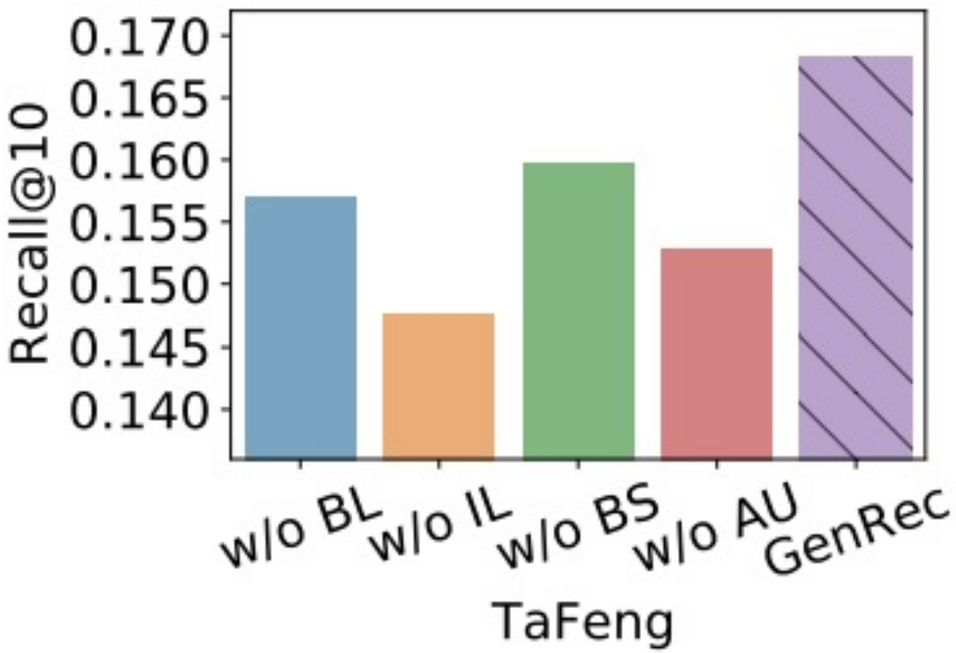
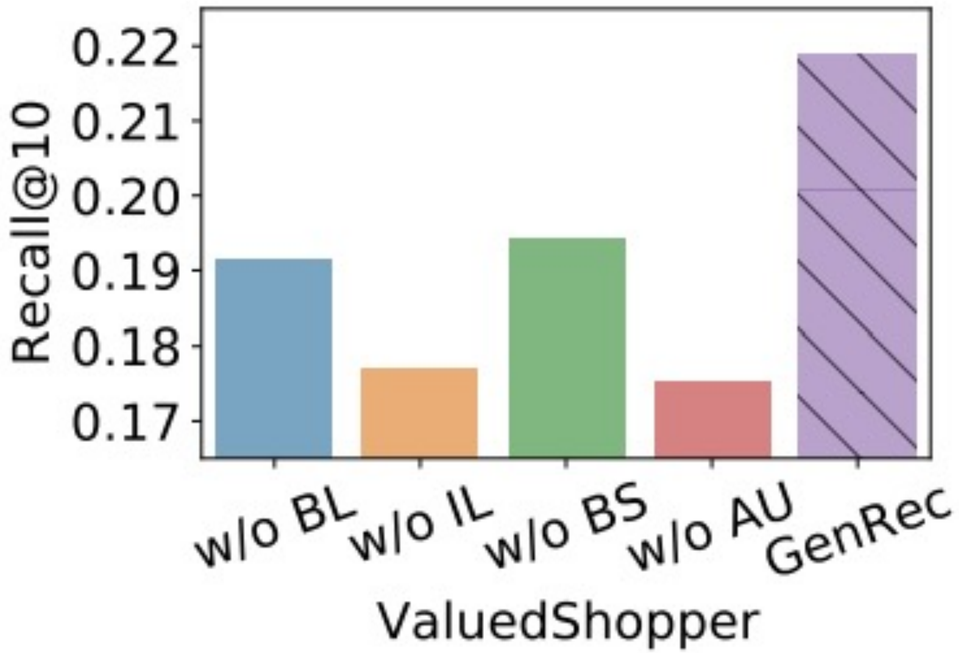


Figure 5: Examples of NBR cases. GenRec could provide more diversified, correlated, and reasonable item combinations.

Коррекция списков рекомендаций (Beam Search)





Коррекция списков рекомендаций (Beam Search)

Dataset	Metric	FPMC	TIFU-KNN	BERT4Rec	DREAM	Beacon	Sets2Sets	CLEA	BART4Rec*	GenRec	Improv.
TaFeng	Recall@5	0.0639	0.0851	0.0876	0.0945	0.0837	0.0953	0.1269	0.1184	**0.1372***	+8.12%
	NDCG@5	0.0582	0.0774	0.0807	0.0834	0.0764	0.0897	0.1248	0.1132	**0.1297***	+3.93%
	Recall@10	0.0761	0.1262	0.1129	0.1182	0.1068	0.1362	0.1564	0.1421	**0.1682***	+7.54%
	NDCG@10	0.0621	0.0972	0.1064	0.1013	0.0941	0.1084	0.1457	0.1367	**0.1593***	+9.33%
ValuedShopper	Recall@5	0.0584	0.1369	0.0724	0.0667	0.0569	0.0814	0.1065	0.1354	**0.1591***	+16.21%
	NDCG@5	0.0633	0.1402	0.0893	0.0843	0.0751	0.1021	0.0979	0.1337	**0.1624***	+15.83%
	Recall@10	0.0947	0.2014	0.1082	0.0981	0.0849	0.1237	0.1621	0.2029	**0.2190***	+7.94%
	NDCG@10	0.1164	0.2136	0.1307	0.1235	0.1095	0.1564	0.1608	0.2059	**0.2287***	+7.07%
Electronics	Recall@5	0.0174	0.0295	0.0353	0.0324	0.0309	0.0291	0.0346	0.0342	**0.0361***	+2.27%
	NDCG@5	0.0098	0.0172	**0.0206**	0.0185	0.0174	0.0182	0.0197	0.0193	0.0204	-0.97%
	Recall@10	0.0209	0.0347	0.0438	0.0410	0.0398	0.0339	0.0442	0.0428	**0.0449**	+1.58%
	NDCG@10	0.0127	0.0224	0.0245	0.0229	0.0217	0.0217	0.0251	0.0244	**0.0256**	+1.99%

Correlation-Sensitive Next-Basket Recommendation

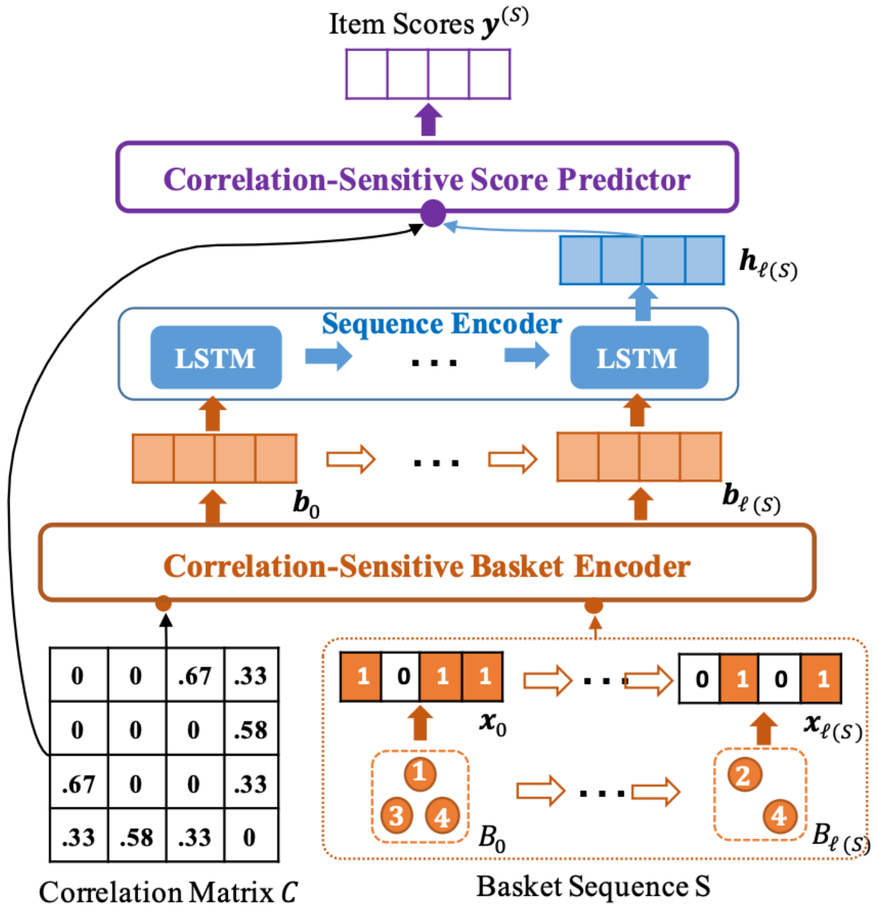


Figure 2: Architecture of the proposed framework *Beacon*.

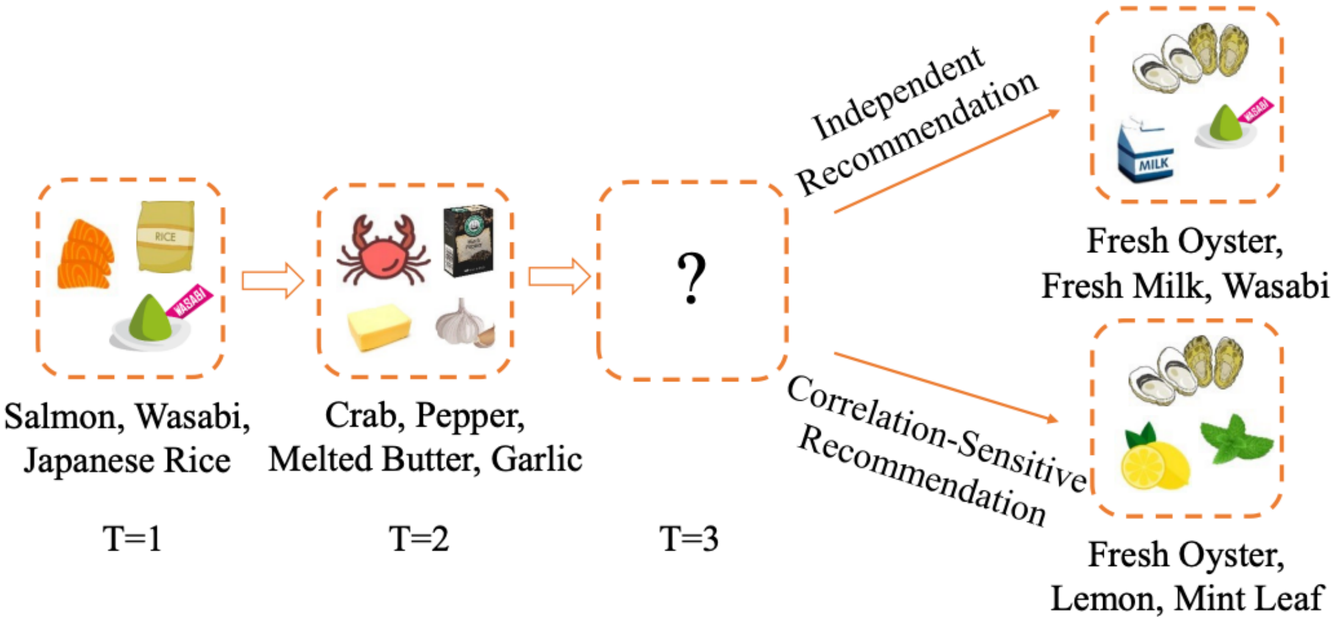
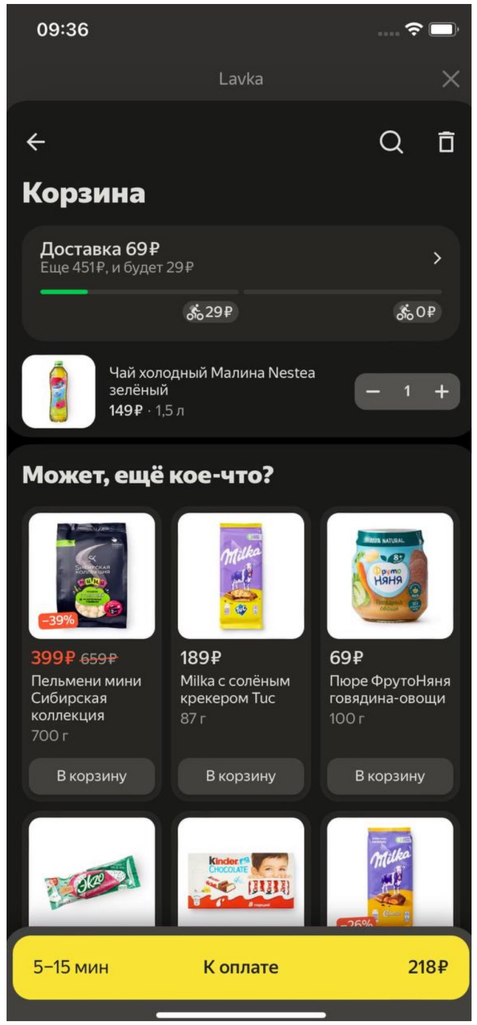
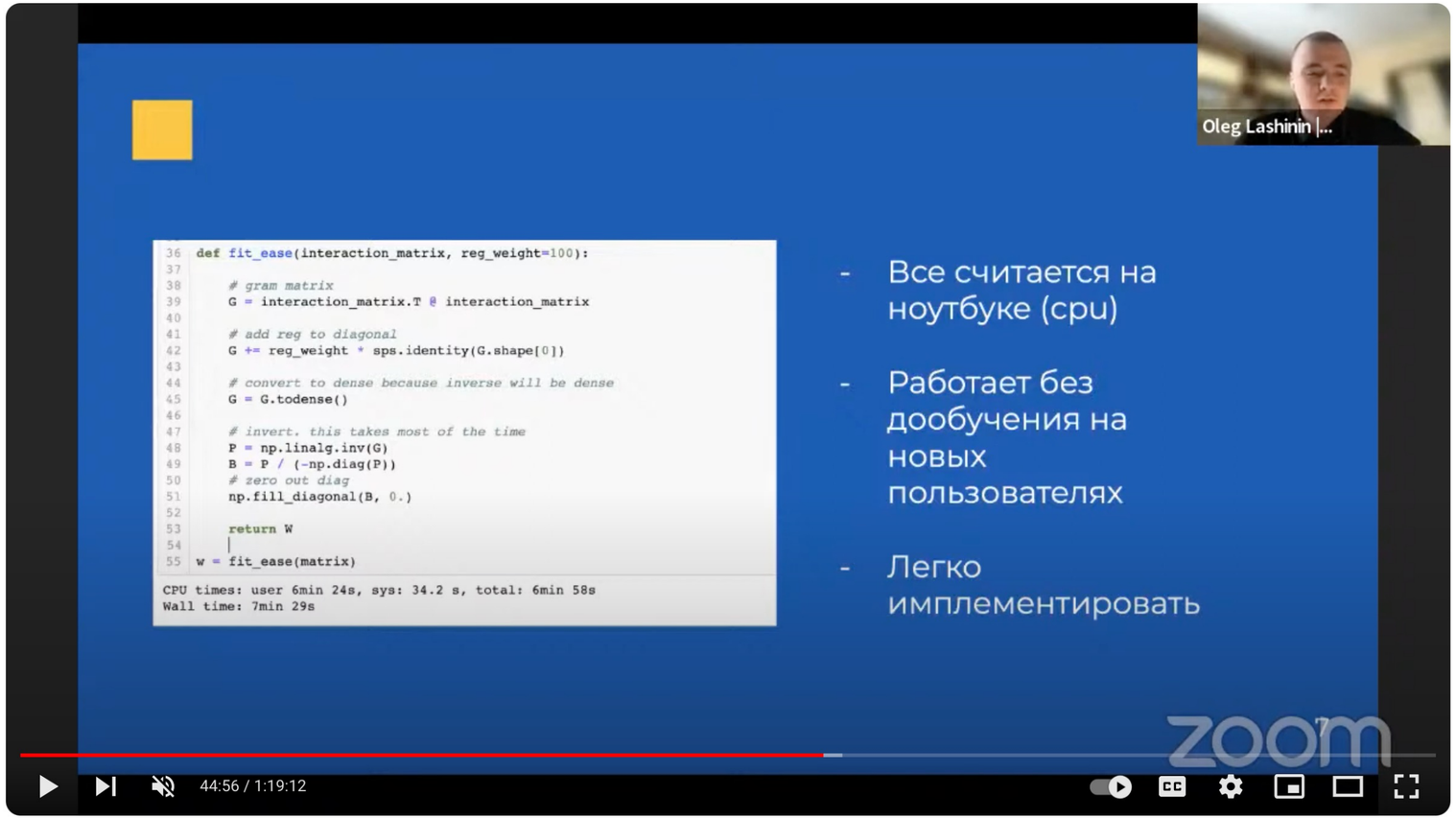


Figure 1: Motivating example for correlation-sensitive next-basket recommendation.

Up-sale на корзине – это другое





Hack The Cart // Финальные питчи

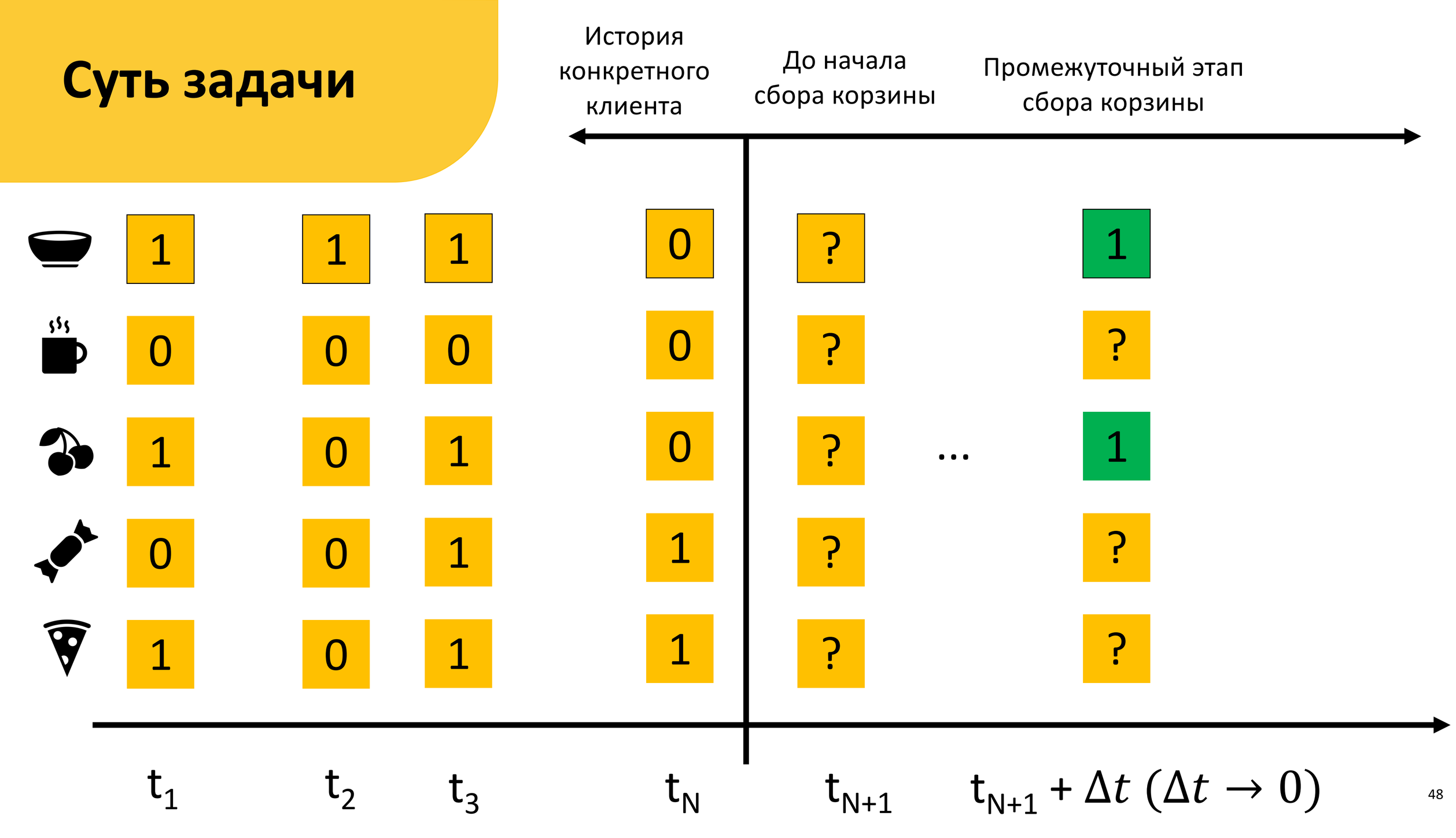
Up-sale на корзине – это другое

Консалтинговая компания **«Диджитал Консалтинг Солюшнс»** совместно с ИТ-компанией **«ЭВОТОР»** представляли кейс по созданию универсальной рекомендательной системы для предложения товаров клиентам. Разработанные решения участников помогут увеличить эффективность продаж и удовлетворенность ритейл игроков.

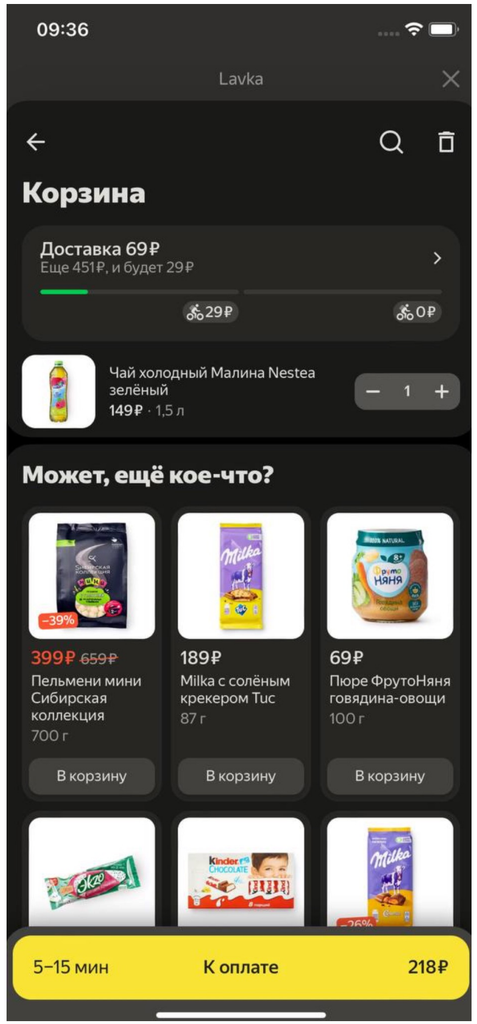
Призовые места в кейсе от компаний «Диджитал Консалтинг Солюшнс» и «ЭВОТОР» заняли команды:

1. Нару Тое Токру То (Санкт-Петербург, Ульяновская область, Москва)
2. Ling Bizkit (Санкт-Петербург, Москва, Московская область, Смоленская область)
3. JETFORK (Свердловская область)

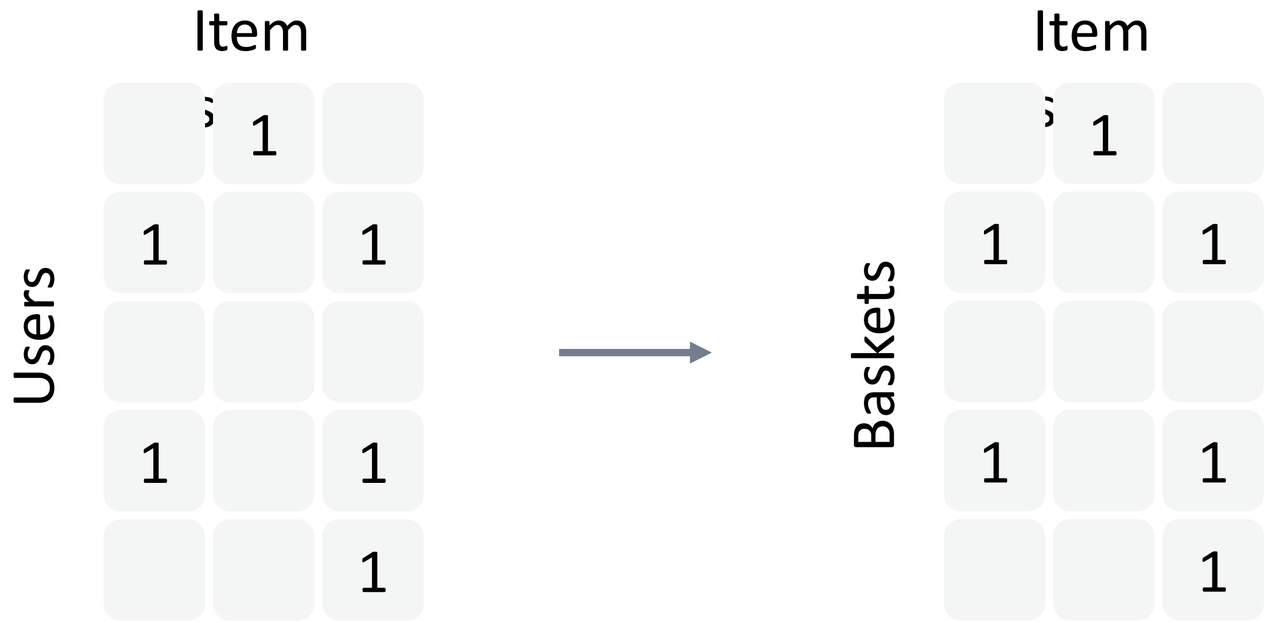
Суть задачи


История конкретного клиента
До начала сбора корзины
Промежуточный этап сбора корзины
1 1 1 0 ? 1
0 0 0 0 ? ?
1 0 1 0 ? ... 1
0 0 1 1 ? ?
1 0 1 1 ? ?
t1 t2 t3 tN tN+1 tN+1 + Δt (Δt → 0)

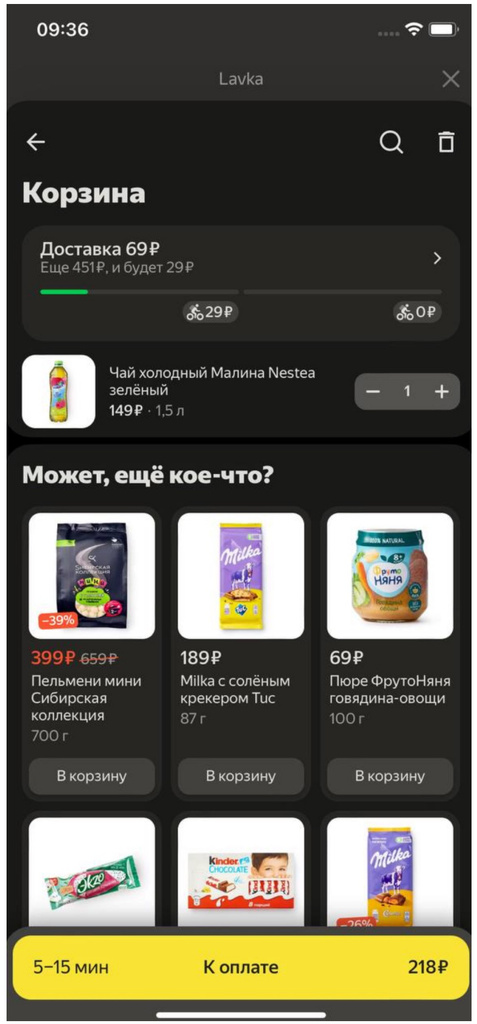
Up-sale на корзине – это другое



$$\min_B \quad ||X - XB||_F^2 + \lambda \cdot ||B||_F^2$$



Up-sale на корзине – это другое

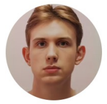


- Имеет оптимальное решение в явном виде
- Имеет всего 1 гиперпараметр и устойчив к его варьированию
- Растет качество при добавлении редких айтемов
- Можно считать и обновлять рекомендации на лету

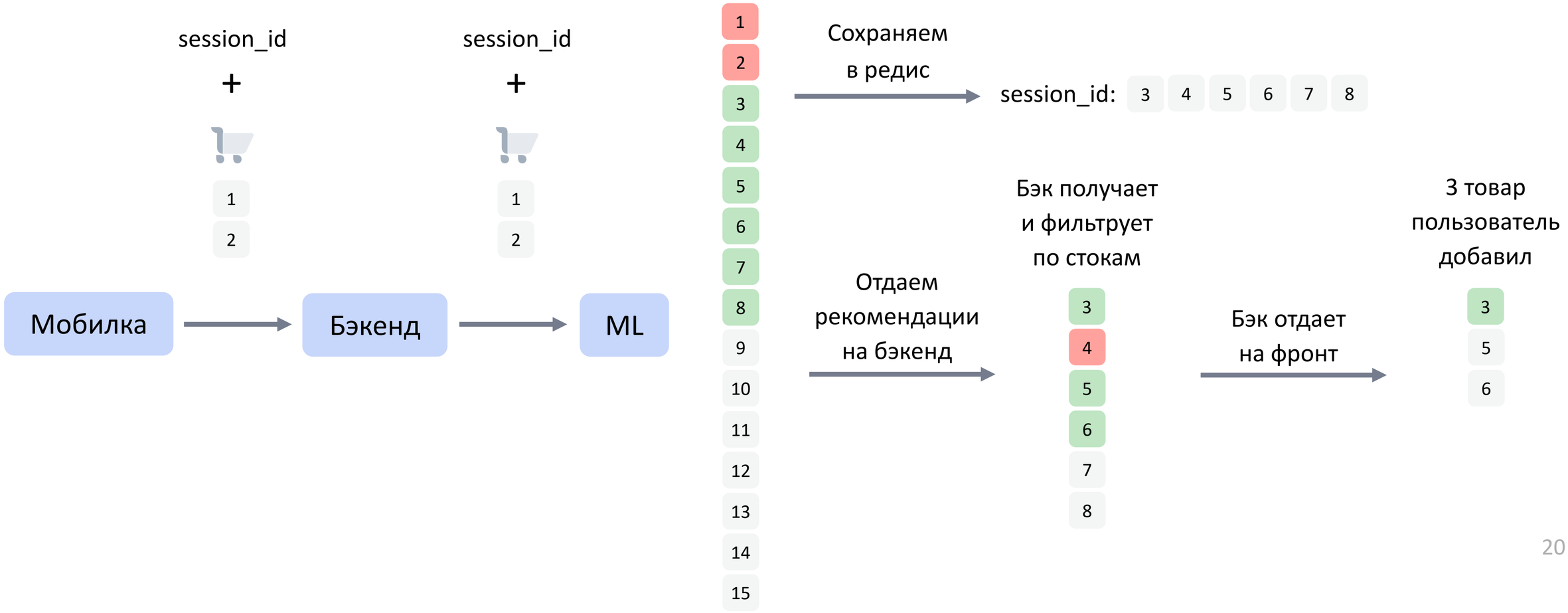
- Плохо масштабируется на большие каталоги
- Нужно переучивать, чтобы добавить новые айтемы
- Не учитывает последовательность интеракций
- Не работает с негативным фидбеком

Онлайн up sell рекомендации (не ALS и не бустинг)

Можно ли строить рекомендации онлайн, не используя при этом ALS или бустинг? Андрей поделится, как у команды получилось это сделать для up sell рекомендаций в сервисе «Продукты» в Тинькофф Городе, с какими проблемами столкнулись и как их решили.

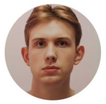


Андрей Бабкин
Тинькофф



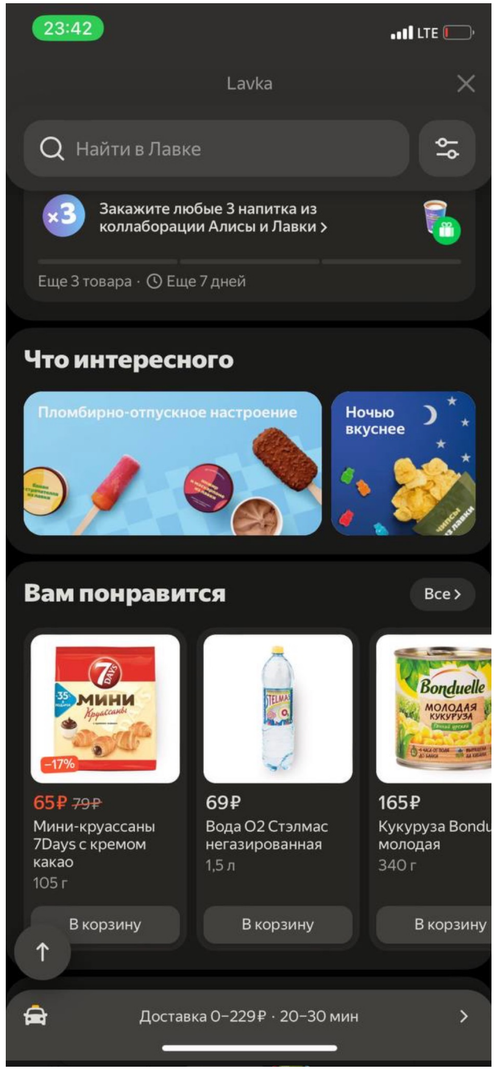
Онлайн up sell рекомендации (не ALS и не бустинг)

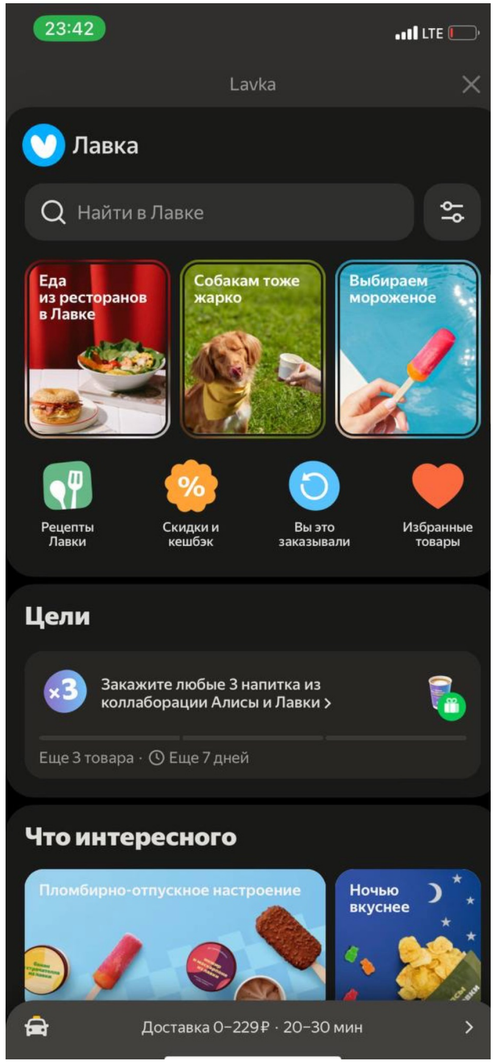
Можно ли строить рекомендации онлайн, не используя при этом ALS или бустинг? Андрей поделится, как у команды получилось это сделать для up sell рекомендаций в сервисе «Продукты» в Тинькофф Городе, с какими проблемами столкнулись и как их решили.



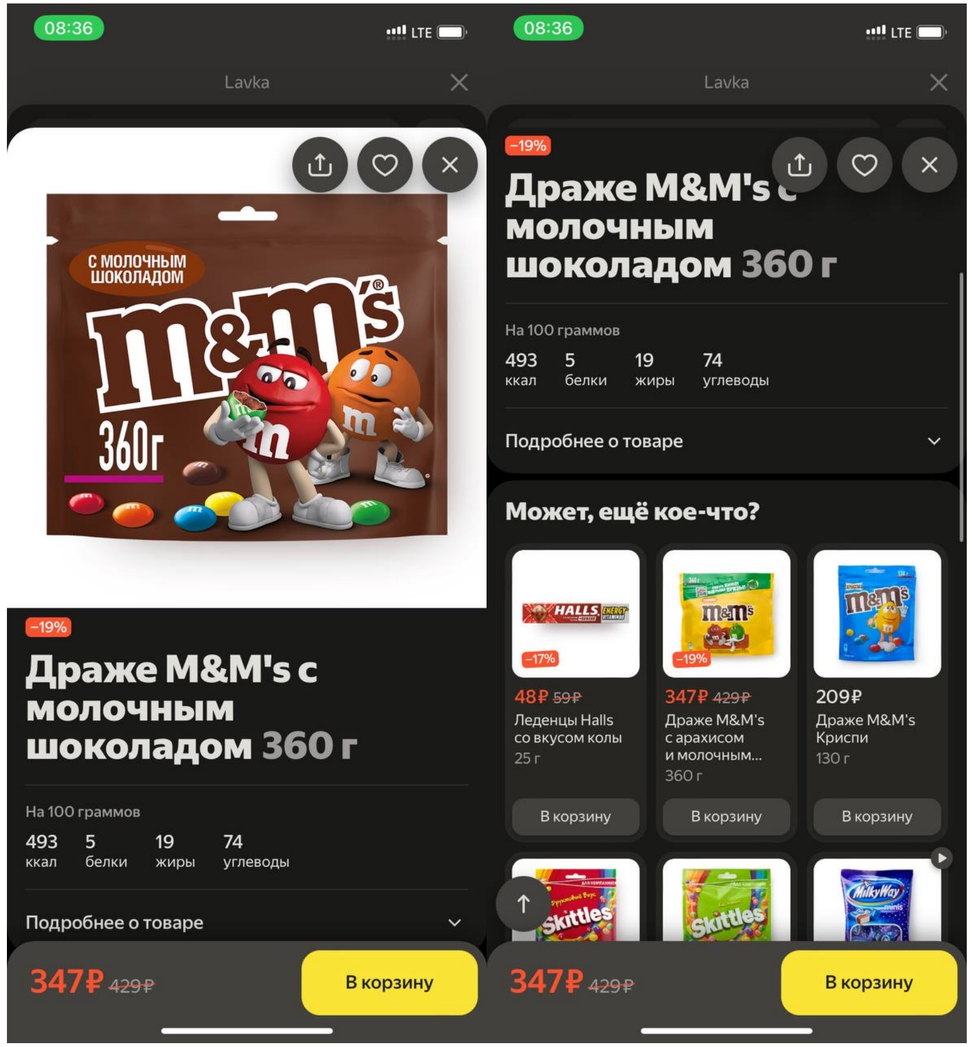
Андрей Бабкин
Тинькофф

Можно разделить старые айтемы и новые в интерфейсе





Рекомендации на карточке



Embarrassingly Shallow Autoencoders for Sparse Data*

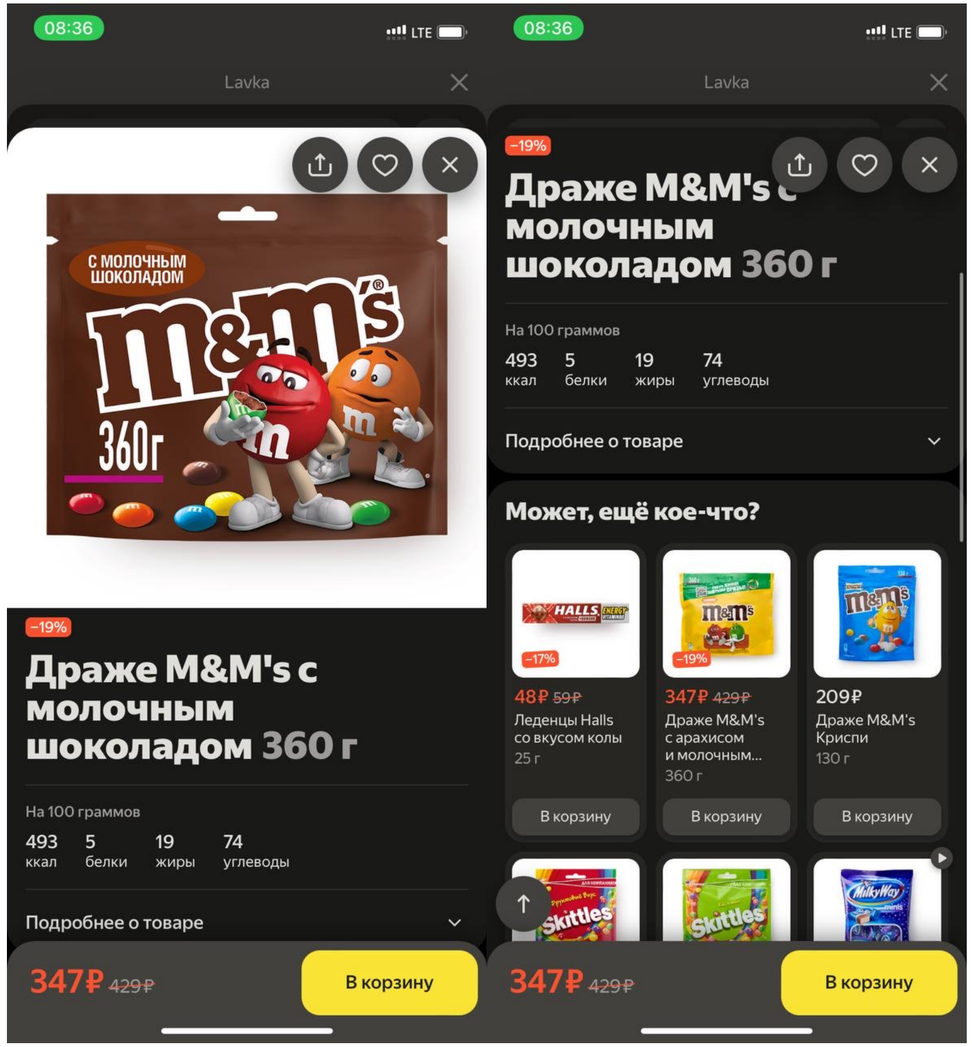
Harald Steck
Netflix
Los Gatos, California
hsteck@netflix.com

The predicted score $S_{u,j}$ for an item $j \in \mathcal{I}$ given a user $u \in \mathcal{U}$ is defined by the dot product

$$S_{uj} = X_{u,.} \cdot B_{.,j}, \tag{1}$$

where $X_{u,.}$ refers to row u, and $B_{.,j}$ to column j.

Рекомендации на карточке



Hugging Face