Personalized Category Frequency prediction for Buy It Again recommendations

Amit Pande
amit.pande@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

Kunal Ghosh
kunal.ghosh@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

Rankyung Park
rankyung.park@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

ABSTRACT

Buy It Again (BIA) recommendations are crucial to retailers to help improve user experience and site engagement by suggesting items that customers are likely to buy again based on their own repeat purchasing patterns. Most existing BIA studies analyze guests' personalized behaviour at item granularity. This finer level of granularity might be appropriate for small businesses or small datasets for search purposes. However, this approach can be infeasible for big retailers which have hundreds of millions of guests and tens of millions of items. For such data sets, it is more practical to have a coarse-grained model that captures customer behaviour at the item category level. In addition, customers commonly explore variants of items within the same categories, e.g., trying different brands or flavors of yogurt. A category-based model may be more appropriate in such scenarios. We propose a recommendation system called a *hierarchical PCIC model* that consists of a *personalized category model* (PC model) and a *personalized item model within categories* (IC model). PC model generates a personalized list of categories that customers are likely to purchase again. IC model ranks items within categories that guests are likely to reconsume within a category. The hierarchical PCIC model captures the general consumption rate of products using survival models. Trends in consumption are captured using time series models. Features derived from these models are used in training a category-grained neural network. We compare PCIC to twelve existing baselines on four standard open datasets. PCIC improves NDCG up to 16% while improving recall by around 2%. We were able to scale and train (over 8 hours) PCIC on a large dataset of 100M guests and 3M items where repeat categories of a guest outnumber repeat items. PCIC was deployed and A/B tested on the site of a major retailer, leading to significant gains in guest engagement.

KEYWORDS

Personalization, Recommender Systems, E-commerce, Repeat purchases, Buy it again, Survival Models, Time-Series Models, Neural Network

ACM Reference Format:
Amit Pande, Kunal Ghosh, and Rankyung Park. 2023. Personalized Category Frequency prediction for Buy It Again recommendations. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3604915.3608822

This work is licensed under a Creative Commons Attribution-Share Alike International 4.0 License.

RecSys '23, September 18–22, 2023, Singapore, Singapore
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3608822

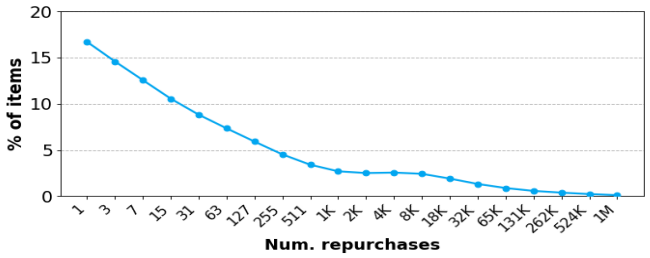
1 INTRODUCTION

With the advent of e-commerce, recommendation systems have become a hot topic for research. Digital grocery sales skyrocketed with the advent of Covid-19 as most shoppers switched to digital orders backed by digital fulfillment, order-pickup, drive-up, or personal shopper [6]. With this change in shoppers' behavior, a lot of attention went to both *next basket recommendation* (NBR) [10, 13, 15–18] that suggests items customers would like to purchase or consume next and to building personalized virtual aisles to aid the customer shopping experience.

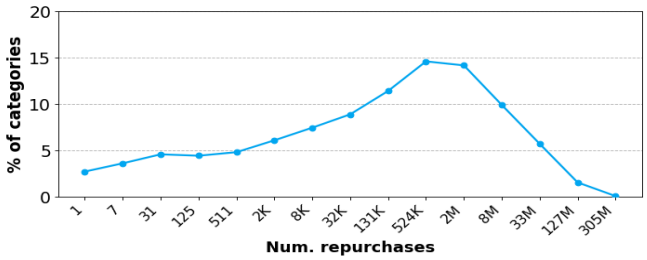
Given a sequence of baskets that a customer has purchased or consumed in the past, the goal of a NBR system is to generate the next basket of items that the customer would like to purchase or consume next. The NBR can be further divided into two similar but different problems. The first is repeat purchase recommendation, called the *Buy It Again* (BIA) problem, where the goal is to recommend items that customers have already purchased and do so at times when the customers might be running out of the item(s). The second is adjacent inspiration recommendation, or the *You might also like* problem, where the goal is to inspire customers to shop for items that may complement ones they have bought before or ones similar customers have purchased.

Existing work in BIA recommendations has focused on modeling item repurchase probabilities by using variants of recurrent neural networks [8–11, 14, 16–18] or statistical models [1, 3–5, 7]. Large retailers handle hundreds of millions of items and guests, but the majority of repurchase transactions are on a small subset of items and guests. This can lead to underfitting for item-grained models, as the data ends up being represented sparsely in a very high dimensional space. In the worst case, training itself may become infeasible due to computational resource limitations.

In this work, we emphasize the effectiveness of personalized category frequency modeling on BIA predictions. Customers will often explore variants of an item or new items within a category for reasons such as the desire to try different brands, the need to satisfy varying taste preferences in the customer's family, or the presence of discounts on alternative items. Category-based repurchase modeling can effectively capture higher abstraction information on these item repurchase dynamics. As shown in Figure 1, the percentage of items that have high numbers of repurchases is small (Figure 1a), but most categories demonstrate high levels of repurchases (Figure 1b). The discrepancy means that models geared toward category repurchases may be more effective at satisfying guest preferences. Furthermore, due to the aforementioned sparsity,



(a) Percentage of customers repurchasing the same item



(b) Percentage of customers repurchasing from the same category

Figure 1: Percentage of items and categories against number of repurchases in 1.5 years. (a) Most items have small number of repurchasing transactions. (b) Most categories have large number of repurchasing transactions. Categories have more sufficient amount of data for modeling than items.

it is far more difficult to train performant BIA recommendation models on item repurchases than it is on category repurchases.

The main contributions of this work as summarized below:

(1) In this paper, we propose a 2-tier *PCIC model* for BIA recommendations. The *personalized category model* (PC model) predicts which categories customers will buy again on their next visit, and the *personalized item within categories model* (IC model) provides personalized ranks of items in categories. Final BIA recommendations for individual customers are generated by combining both predictions. We show how the model supports our insights that customers tend to explore brands, sizes, flavors, etc. similar to a given item within a cateogy.
(2) We demonstrate that the proposed PCIC model outperforms existing baselines of public datasets. We also show that PCIC scales to large datasets.
(3) We deploy PCIC in a commercial setting to provide BIA recommendations for millions of customers. We demonstrate improved guest experience on the site as evidenced by multiple A/B tests. We discuss our experiences deploying and scaling PCIC.

2 MODEL

2.1 Category level repurchase modeling

We use category level features to predict the customers' likelihood to repurchase items. Each customer has their own features crafted by their purchase history, and the last m days of customer purchase data is used to generate labels to train a category level model. All purchase history before this m days is used to generate the features. Any category in which customers repurchased an item in this time period is considered label 1 while the other categories are assigned label 0. The main features considered to train the model are enumerated in subsequent subsections.

2.1.1 Survival Analysis. Survival analysis [12] focuses on the expected duration of time until occurrence of an event of interest. It differs from traditional regression by the fact that parts of the training data can only be partially observed, which is stated as being censored. For these censored observations, we only know that the event time is greater than the time at the point of censoring. In the retail scenario, we consider the purchase of an item within a category as an event. For each category, repeat purchase data can then be used to construct a life table across customers for each category, which will allow us to predict repeat purchase risk as a function of time. A life table summarizes the events and censored cases across time. At time 0, all observations (reference purchases) are still at risk, which meants that they have not yet repeated the purchase (event) or been censored. As events and censored cases occur, observations fall out of the risk set.

Repeat purchase data can be used to compute a few useful features:

1. `hazard` (eq. 1) is the probability of event occurring at kth day, conditional on the event not occurring before day k. It denotes an approximate probability that an event (repurchase) occurs in a given time interval, under the condition that an user would remain event-free up to that time (no purchase).

$$\texttt{hazard}_k = \texttt{n\_event}_k / \texttt{n\_risk}_k \tag{1}$$

2. `cum_hazard` (eq. 2) is cumulative sum of hazard over time.

$$\texttt{cum\_hazard}_k = \sum_{kk=0}^{k} \texttt{hazard}_{kk} \tag{2}$$

3. `survival` (eq. 3) is probability of the event occurring after day k or equivalently, the proportion that have not yet experienced the event by time t.

$$\texttt{survival}_k = \exp(-1 * \texttt{cum\_hazard}_k) \tag{3}$$

4. `cum_survival` (eq. 4) as probability of event occuring in ±3 days to today. We additionally define this feature since many grocery customers shop once a week.

$$\texttt{cum\_survival}_k = \texttt{survival}_{k+3} - \texttt{survival}_{k-3} \tag{4}$$

5. `normalized_risk` (eq. 5) is defined as risk associated with the user category today as a fraction of risk on the day of purchase.

$$\texttt{norm\_risk}_k = \texttt{n\_risk}_k / \texttt{n\_risk}_0 \tag{5}$$

6. `normalized_event` (eq. 6) is defined as the event probability on the given day normalized by event plus censor population.

$$\texttt{norm\_event}_k = \texttt{n\_event}_k / \texttt{n\_event\_\&\_censor}_k \tag{6}$$

Building this model gives a population level overview of the item repurchase rate.

	Num Items	Num Users	Basket Size	Baskets/ User	Items/ user
tafeng	12062	13949	6.27	5.69	6.397
dunhumby	4997	36241	7.33	7.99	22.56
shoppers	7907	10000	8.71	56.85	24.934
instacart	8000	19935	8.97	7.97	33.271
Internal	~3M	~100M	~10	~25	~200

Table 1: Some characteristics of datasets considered for evaluation

2.1.2 ARIMA models. Autoregressive Integrated Moving Average or ARIMA models are useful for short term forecasts on non-stationary time series problem. For each customer and category, we try to characterize their purchase pattern using ARIMA and predict the next day of purchase. ARIMA models have three parameters (p, d, q) where p is the order of the autoregressive model, d is the degree of differencing, and q is the order of the moving-average model. We build one ARIMA model that observes the past dates of purchases within a category to predict the next one and a second model to consider the quantity of item purchased and predict the current rate of consumption by the customer (say X uses 2 oz of shampoo daily). This is then used to predict the date when the customer will likely run out of the item. For each customer-category pair, we train these models and use their forecasts `ARIMA(date)` and `ARIMA(rate)` as features.

2.1.3 Other features. We consider three more behavioral category level features: `NumPurchases` - Number of times a given customer has purchased from the category, `tripsSinceLastPurchased` - the number of purchases in other categories customer has made since purchasing in this category, `daysSincelastPurchased` - the time difference between today and last date the customer made a purchase in this category.

2.1.4 Model training. We take the past 1.5 years of user shopping data to train the model to ensure we capture a yearly cadence. The last m days of data is held out to generate labels. For example - we may take Jan 2021- July 24 2022 dataset to generate features for all guests. For those guests who shopped during July 25 - 31 (m = 7), we generate labels 0 and 1 for categories not shopped and shopped respectively. The 6 features from survival model, 2 predictions from two ARIMA models and the 3 other features mentioned earlier are generated for each user and category pair.

We trained a 2 layer neural network on the category level guest purchase dataset. We wanted to keep it light because the number of input features is small (11), and we wanted it to scale well for the large number of users. The most performant neural net was composed of 2 fully connected layers (10 and 5 neurons) with sigmoid activations. The output layer is run through a softmax and the logistic loss function is used for optimization.

2.2 Inter-category Product Ranking

In general, we observed that a customer is most likely to repurchase their most frequently or most recently bought items. The two main features used to rank products within a category are frequency (Freq) and recency (Rec) of purchase. We wanted to combine them both to arrive at optimal ranks, however, recency is measured in days and frequency is a count. To come to a common ground, we convert both into ranks. Item Frequency Rank (IFR) and Item Recency Rank (IRR) are obtained by ranking the frequency counts and days (respectively) since the last purchase of an item (DaysSincePurchase). $\texttt{IFR} = Rk(Freq)$, $\texttt{IRR} = Rk(DaysSincePurchase)$. We combine the ranks using a weighted average, rank again, then divide the rank by number of times the item is bought (NIB). This insight was based on user feedback and will be discussed in later sections. The equation 7 shows how final Item Rank (IR) is calculated.

$$\texttt{IR} = ceil(\frac{1}{NIB} \times Rk(\alpha \times IRR + \beta \times IFR)) \tag{7}$$

where the parameters α and β were obtained using exhaustive grid search in the range [0,1].

2.3 Model output

We combine the outputs of PC and IC models to get an aggregated single list of items for recommendations. Let Rk_{PC} and Rk_{IC} represent the PC rank for an item's category and IC rank of the item respectively. The PCIC model outputs in a round robin manner i.e. $Rk = Rk(sortByAscending(Rk_{PC}, Rk_{IC}))$

3 EXPERIMENTS

In this section, we conduct experiments to answer the following questions: Q1: What is the effectiveness of the proposed method? Does it outperform state-of-the-art NBR/ BIA methods? Q2: How well does this method scale up to generate recommendations for millions of users? Q3: How is model performance impacted by the input features? Q4: How do training and testing date ranges change the performance of the model?

3.1 Experimental Settings

3.1.1 Datasets & Metrics. We use four publicly available datasets shown in Table 1 to compare the performance of the proposed method with existing methods in literature: ValuedShopper[1], Instacart[2], Dunnhumby[3], and TaFeng[4]. We also evaluate using an internal dataset consisting of the sales history of users at a large retailer. There are around 100M users and 3M products in this dataset. We use recall (@K) and NDCG (@K) metrics to evaluate and compare our methods.

3.1.2 Baselines. **TopSell**: It uses the most frequent items that are purchased by users as the recommendations to all users. **FBought**: It uses the most frequent items that are purchased by a user as the recommendationto him. **userKNN** [13]: It uses classical collaborative filtering based on kNN on purchase baskets. **RepeatNet** [15]:

[1]https://www.kaggle.com/c/acquire-valued-shoppers-challenge/overview
[2]https://www.kaggle.com/c/instacart-market-basket-analysis
[3]https://www.dunnhumby.com/careers/engineering/sourcefiles
[4]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset

	Recall @10				NDCG @10			
Dataset	V Shopper	Instacart	Dunhumby	TaFeng	V Shopper	Instacart	Dunhumby	TaFeng
TopSell	0.0982	0.0724	0.0819	0.0773	0.0779	0.0641	0.0601	0.0519
FBought	*0.2109*	*0.3426*	*0.1853*	0.0704	*0.2128*	*0.3618*	0.1771	0.0766
userKNN	0.0988	0.0720	0.1135	0.1089	0.1415	0.1020	0.1707	0.0832
RepeatNet	0.1031	0.2107	0.1324	0.0645	0.1439	0.2285	0.1545	0.0592
FPMC	0.0951	0.0763	0.0919	0.0868	0.1188	0.0946	0.1025	0.0667
DREAM	0.0991	0.0866	0.0915	0.0902	0.1231	0.1063	0.1009	0.0763
SHAN	0.0847	0.0902	0.1007	0.0878	0.1032	0.1152	0.1149	0.0813
Sets2Sets	0.1259	*0.3021*	0.2068	0.1190	*0.1626*	0.3487	*0.2134*	0.0844
TIFUKNN (NBR)	**0.3578**	**0.3952**	**0.2087**	0.1301	0.3060	*0.3825*	*0.1983*	0.1011
TIFUKNN(BIA)	*0.3500*	*0.3700*	*0.1940*	0.0990	*0.3000*	*0.3800*	*0.1860*	0.0860
RCP	0.0416	0.1090	0.0635	**0.3860**	0.0591	0.1175	0.0634	**0.2363**
ATD	0.0350	0.1600	0.0468	*0.3100*	0.0605	0.1264	0.0350	*0.2310*
PG	0.1694	0.2375	*0.1332*	*0.3100*	0.0684	0.1331	0.0351	*0.2336*
MPG	0.1762	0.2183	0.0820	*0.3200*	0.0680	0.1240	0.0450	*0.1600*
PCIC model	*0.3528*	0.2548	0.1540	0.1427	**0.3531**	**0.5700**	**0.2321**	0.1180

Table 2: Performance comparison with existing baselines. The top performing algo in a dataset are in bold. The three runner ups are in italics.

RNN-based model for session-based recommendation which captures the repeated purchase behavior of users. **FPMC** [16]: Matrix Factorization uses all data to learn the general taste of the user whereas Markov Chains can capture sequence effects in time. **DREAM** [18]: Dynamic REcurrent bAsket Model (DREAM) learns a dynamic representation of a user but also captures global sequential features among baskets. **SHAN** [17]: A deep model based on hierarchical attention networks. It partitions the historical baskets into longterm and short-term parts.**Sets2Sets** [10]: The state-of-the-art end-to-end method for following multiple baskets prediction based on RNN. **RCP** [2]: Repeat Customer Probability (RCP) finds repeat probably of an item & repeat items based on that. **ATD** [2]: Aggregate Time Distribution Model fits a time distribution to model probablity distribution and time characteristics of repeat items. **PG** [2]: Poisson Gamma distribution fitted to predictaggregate purchasing behavior. **MPG** [2]: A modified PG distribution to make the results time dependent and intergate repeat customer probability.

We use grid search to tune the hyper-parameters in compared methods. For userKNN, the number of nearest neighbors is searched from range(100, 1300). For FPMC, the dimension of factor is searched from the set of values [16, 32, 64, 128]. For RepeatNet, DREAM, SHAN, and Sets2Sets, the embedding size is searched from the set of values [16, 32, 64, 128]. For PCIC model, ARIMA model was autofitted in range (3, 3, 0).

3.2 Performance Comparson (Q1)

Table 2 gives the performance comparison of PCIC model with existing baselines. Several observations can be made from the table. First, we observe that the PCIC model has highest (or near highest) recall and NDCG values in most cases on Valued Shopper, instacard and Dunhumby datasets. Surprisingly, RCP model performs well on tafeng dataset. Just like our model captures personalized category frequency, TIFUKNN model tries to explicitly capture personalized item frequency. TIFUKNN model uses nearest neighbor approach to collaborative filtering to learn repurchasing pattern from other users. We modified the code and ran it to run on the BIA task only TIFUKNN(BIA). In PCIC model, the survival analysis features use user repurchasing pattern at category level. Sets2Sets captures personalized item frequency explicitly but subsequently learns coeffients for RNN. RCP, ATD, PG and MPG models try to model repeat purchase pattern using a Poisson Gamma or modified Poisson Gamma distribution. Hence, we can see that these methods perform better than any existing methods which do not capture item or category frequency such as RepeatNet, userKNN,FPMC and DREAM. FBought is a pretty simple baseline in that it simply ranks the most frequently bought items of a user in that order. It surprisingly performs better than many baselines here. It is a simple to implement baseline and performs pretty well.

3.3 Scaling up (Q2)

We attempted to train the top performing models above on a much larger (100M user) data set. TIFUKNN uses a user embedding the size of the entire product catalog, which made it impossible to scale up to this data set. As a result, we subsampled the larger data set, creating a representative sample with 1M users. We compared TIFUKNN and Sets2Sets to PCIC using this subsampled data. We observed a 30-35% reduction in NDCG and recall metrics in TIFUKNN and Sets2Sets against PCIC. As a result, we did not put effort into scaling either algorithm.

PCIC was implemented in a distributed hadoop cluster using Apache Spark and takes around 6-8 hours of time to train and test the model for 100M users. We also implemented MPG model in distributed cluster using the maths described in the paper. Table 3 shows the performance comparison of FBought, MPG and PCIC models. Although PCIC performs well in terms of NDCG, the recall is slightly lower than MPG . Next, we calculated MPG parameters at category level instead of original item level and input it as part of features to PC. The performance of integrated PCIC(+MPG) outperforms both PCIC and MPG.

3.4 Feature Importance (Q3)

To obtain the feature importance, we replaced the original neural layer with a Gradient Boosting Tree clasifier. The values are plotted

	Recall@3	NDCG@3	Recall@5	NDCG@5
FBought	0.2020	0.0832	0.0305	0.1212
MPG	**0.0307**	0.1036	**0.0433**	0.1328
PCIC	0.0267	**0.1071**	0.0377	**0.1368**
PCIC(+MPG)	0.0317	**0.1091**	0.0447	**0.1408**

Table 3: Performance comparison on internal dataset

in figure 2. We can observe that the ARIMA forecasts have a very high impact on the output of the model, particularly the model that tries to predict the next purchase based on rate of individual consumption of item by the user. The survival features have smaller impact on the prediction quality meaning other user's purchases play a small role in user's repurchase than his own characteristics. This can be one of the reason why approaches like itemKNN or TIFUKNN which focus on collaborative user behavior don't perform as well as PCIC. MPG does capture rate of consumption with a statistical model and it comes close to PCIC. The features such as number of days since past purchase and explicit category frequency (num purchase) also have high feature importance. if we were to collect the top 3 features, we can say that we can predict whether a user will purchase an item today based on how many times he has purchased before, how many days since his last purchase with us, how much did he purchase last time and how long will it last.

To obtain the feature importance, we replaced the original neural layer with a Gradient Boosting Tree clasifier. The values are plotted in figure 2. We can observe that the ARIMA forecasts have a very high impact on the output of the model, particularly the model that tries to predict the next purchase based on rate of individual consumption of item by the user. The survival features have smaller impact on the prediction quality meaning other user's purchases play a small role in user's repurchase than his own characteristics. This can be one of the reason why approaches like itemKNN or TIFUKNN which focus on collaborative user behavior don't perform as well as PCIC.

3.5 Impact of train and test data selection (Q4)

We held out one week of the most recent customer purchases from this dataset for testing and used one year of purchases made prior to that week for training. A customer and their product purchase were considered as a repeat purchase in the test period only if the customer purchased a product in the training period (y years before the test period, y =1.5) and also purchased the same product sometime in the test period. The (user, category) pairs purchased in this duration are labeled 1 and the categories the user did not purchase in this duration was labeled as 0.

As the pandemic caused increased adoption of the app and website, users started shopping online more frequently particularly. Based on the initial feedback, we observed that the BIA list was not updating particularly for the highly engaged users. We hypothesized that this can be because of the following reasons: (1) the model being trained on all users may not be able to exactly capture the signals and behavior of highly engaged user. (2) The labels are captured based on last 1 week of purchases. But highly engaged users shop much more often, hence their labels are not very accurate. We experimented with scoring the model daily on 1 day of user purchases. We also experimented on training the model only on the most engaged users, defined as users who have made purchases in more than 25 categories.

Table 4(a) shows the improvement in NDCG metric for the PC model with the changes in test time frame and with training on only the most engaged users. Reducing the test time frame significantly improved the performance of the model. The most engaged users had a lower NDCG performance than all users when the test dates were 7 days. We also observed that training the model only on the most engaged users improves NDCG for all users too although it leads in savings on training time. The time taken to train the generate the features and train the model on all users is 2.5x the time taken for highly engaged users

4 DEPLOYMENT JOURNEY

In this section, we discuss several user-facing questions we addressed as well as our experience in deploying PCIC.

4.1 Deployment and Online Experience

We deployed PCIC to a production environment where recommendations are generated daily in our compute cluster on an Apache Spark ecosystem and exported to the cloud for real-time serving. When a user visits the site, these recommendations are then served to them, filtered on the item availability based on inventory and available shipment options selected by the user.

4.2 Human-in-the-loop feedback

We first rolled out the results to a pool of internal team members for testing. This gave us some feedback as to having an exclusion list of some categories which users may not be very comfortable looking at, in their App (with friends and family or otherwise). Based on the feedback, we built an exclusion list of categories which are applied on top of recommendations as filters.

We found that users were sometimes recommended an item they'd recently purchased (e.g., a new flavor of yogurt) from a category where they repurchase, but not one they'd like to repurchase. We used a two step approach filter out such items from recommendations. Apart from the category being a repurchase category, we tried to ensure that the item was bought by the guest at least twice in the past n months (n=6). This helps the customer to identify the items in buy it again list as an item they have repeat purchased. Second, we identified items with low repurchase rates (similar to repurchase rate threshold in RCP [2]) and removed them.

In initial testing, test users noted they typically buy more than one item from a specfic category (e.g., two or more flavors of yogurt) in a single trip. To resolve this, we calculate a variable *NIB* which denotes the number of times the item was purchased by the user per trip. We tweaked the math used to combine the two lists by dividing item rank by NIB and then taking a ceil function to create new item ranks.

4.3 Metrics

We performed A/B tests against existing online baselines. Each test was run for more than two weeks and stopped after ensuring that the samples are statistically significant. The metrics considered for tests are defined as follows:

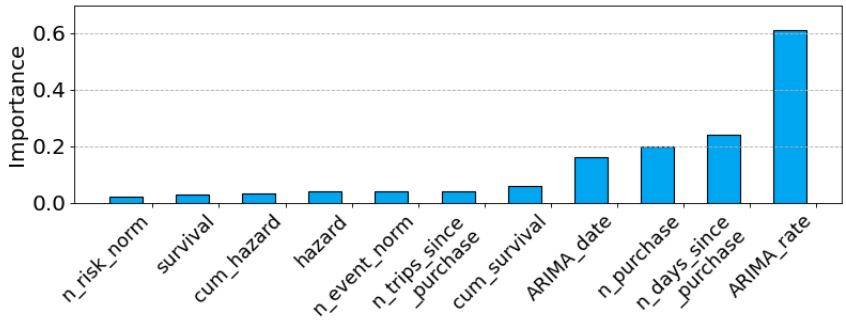


Figure 2: Relative importance of input features to PC model

Trained on	Test Timeframe	NDCG (Test)	
		Most Engaged	All
All	7 days	0.2009	0.2325
All	1 day	0.3501	0.3583
Most Engaged	1 day	0.3602	0.3589

	Lift (%)
CTR	6
Conversion	8.5
Units	27.5

Table 4: Test results (a) Modifications in performance of PC model with changes in training data selection and testing timeframe. (b) Measuring impact of BIA against FBought in online A/B test.

- CTR or Click Through Rate : Percentage of recommendation displays which were clicked by the guest.
- Conversion Rate: Percentage of clicked recommendations which were purchased by the guest same day.
- Units: The total number of units purchased by the users who were part of the treatment.

4.4 A/B testing results

When we introduced Buy It Again recommendation lists to the guest shopping experience, we A/B tested PCIC against a baseline of FBought. The results are given in Table 4(b). We can see that there is significant lift across all three metrics - 6% in CTR, 9% in Conversion and 27% in units purchased.

We also tested adding a Buy It Again recommendation list to the search results of all users. For this, we filtered the Buy It Again results using the search query context.We found that the user interaction with this recommendation list was significantly higher than existing search results (by over 20%). It was observed that the add-to-carts, average order values, and units per order went up by 0-2% (including all guest visits where guests looked for new items).

4.5 Building virtual aisles

We then rolled out BIA to guests by filtering recommendations by categories (Milk, Yogurt, Beauty, etc) to create a virtual aisles experience for online users in a dedicated space in App/site. We use the personalized list of categories for each guest using PC model. For each category, we present a list of recommended items from IC model to form a virtual aisle. In each aisle, we first showed the BIA items of the guest followed by other relevant items. Users who interacted with these recommendations had a significant increase in units per order (25-50%), and average order value (7-35%). Since the buy it again essentials are lower ticket items, they have a smaller dollar impact in order value than units per order. We saw higher guest engagement with virtual aisles experience in the App than in the site.

5 FUTURE DIRECTIONS

Buy It Again recommendations help users to quickly complete their shopping missions. Traditional approaches tend to model guest personalized behavior at item granularity. In this paper, we present the case for a coarse grained model which can capture the customer behavior at item category level. The proposed Personalized Category (PC) model combined with Items-within-Category (IC) model outperform existing BIA and NBR models on standard public datasets. The PCIC model also scales well for large retailers with millions sized product catalogs and millions of active guests. The A/B tests on the site show a significant improvement in guest shopping experience and guest spends.

In the future, we would recommend that retailers explore models that combine the insights from Personalized Category features with Personalized Item features. Moreover, we would recommend considering mutual excitation among items and categories as simultaneous consumption has some inherent relationship with repeat consumption.

REFERENCES

[1] 1959. The Pattern of Consumer Purchases. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 8, 1 (1959), 26–41. http://www.jstor.org/stable/2985810
[2] Rahul Bhagat, Srevatsan Muralidharan, Alex Lobzhanidze, and Shankar Vishwanath. 2018. Buy it again: Modeling repeat purchase recommendations. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 62–70.
[3] Chris Chatfield and Gerald Goodhardt. 1973. A Consumer Purchasing Model with Erlang Inter-Purchase Times. *J. Amer. Statist. Assoc.* 68 (1973), 828–835.
[4] Suvodip Dey, Pabitra Mitra, and Kratika Gupta. 2016. Recommending Repeat Purchases Using Product Segment Statistics. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/2959100.2959145
[5] P S Fader, B G Hardie, and K Lee. 2009. Probability Models for Customer-Base Analysis. *Journal of Interactive Marketing* 23 (2009).
[6] Sofia Gomes and João M Lopes. 2022. Evolution of the online grocery shopping experience during the COVID-19 Pandemic: Empiric study from Portugal. *Journal of Theoretical and Applied Electronic Commerce Research* 17, 3 (2022), 909–923.

[7] Gary L. Grahn. 1969. NBD Model of Repeat-Purchase Loyalty: An Empirical Investigation. *Journal of Marketing Research* 6 (1969), 72 – 78.
[8] Ruining He, Wang-Cheng Kang, Julian J McAuley, et al. 2018. Translation-based Recommendation: A Scalable Method for Modeling Sequential Behavior.. In *IJCAI*. 5264–5268.
[9] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.
[10] Haoji Hu and Xiangnan He. 2019. Sets2sets: Learning from sequential sets with neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1491–1499.
[11] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1071–1080.
[12] Komal Kapoor, Mingxuan Sun, Jaideep Srivastava, and Tao Ye. 2014. A hazard based approach to user return time prediction. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (08 2014). https://doi.org/10.1145/2623330.2623348
[13] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM* 40, 3 (1997), 77–87.
[14] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.
[15] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.
[16] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
[17] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention network. In *IJCAI International Joint Conference on Artificial Intelligence*.
[18] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 729–732.