Generating your shopping list with AI: recommendations at Picnic

Thijs Sluijter · Follow
Published in Picnic Engineering
10 min read · Jan 9

Introduction

At Picnic, we're not just an online supermarket; we're the modern milkman. This means that we want to make the shopping experience of our customers as easy as possible while delivering the best personal service.

To do this we couldn't do without recommender systems. Recommender systems are used in many places within Picnic. From ranking customers' search results and previously bought items, to showing the most relevant recipes for each customer. Our goal is to personalise every aspect of the grocery shopping experience using machine learning. However, recommendations aren't just about algorithms; it's

about helping our customers save time, find the right things, and curate the shopping experience they deserve.

What do we have, what do we want?

We currently have multiple recommendation models in operation. One example is a model focussed on customers' repeat purchases, called Customer Article Rebuy Prediction (or CARP in short 🐟). CARP is used in many places in the Picnic app. Most notably in the previous purchases page where all of your previously bought articles are ranked by CARP to create your personal shopping list. CARP is a relatively simple machine learning model (*Under the hood CARP is powered by XGBoost)* with handcrafted features tailored to predicting repeat purchases.

These features include things like the average rebuying frequency of an item or the periodicity of a customer doing their shopping. Grocery shopping actually contains many of these patterns and they are often very clear! Indeed a large part of customers shopping is repeat behaviour. That makes repeat item recommendation not only quite straightforward but also very useful!

However, to realise our goal of personalising every aspect of the shopping experience, we also want to serve recommendations in general, i.e. for any article in our store, including the ones a customer hasn't bought before.

Beyond your weekly shopping list: repeat vs explore

Making the distinction between unbought, or explored articles, and repeat articles, is non-trivial. As noted in the literature [1, 2, 3, 4, 5], this element plays a vital role in grocery shopping recommendation performance.

As mentioned above, our CARP model is already running in production and serving millions of repeat recommendations on a daily basis. Developing a more general model that can additionally handle explore recommendations is however quite a difficult task.

The challenges of Explore Recommendations

Customers in grocery shopping have highly personal preferences that discriminate between sometimes very similar products (think of competing brands or subtle differences in flavours) while the assortment is huge and the distribution of customer-article interactions has a long tail. Ok, that's a bit of a mouthful. Let's break that down.

The long tail of the data is a large challenge in many recommendation systems.

The sparsity of the Supermarket assortment

Picnic has a large assortment, with over 30.000 articles. The reason why this makes explore recommendations so challenging is not the size of the assortment by itself though, it's the combination of a large assortment and the general behaviour that customers show when shopping for their groceries. Most people have a pretty fixed weekly shopping list containing about 30 to 60 general items. In fact the average Picnic customer has not bought more than 200 unique products out of the 30.000 that we sell. To make matters worse, the items on customers' grocery list tend to be largely the same, so while we have a lot of information on bananas and milk, we know little about more obscure articles.

Think of it this way, for almost every order placed at Picnic, the following shopping list probably accounts for quite a large portion of the total basket:

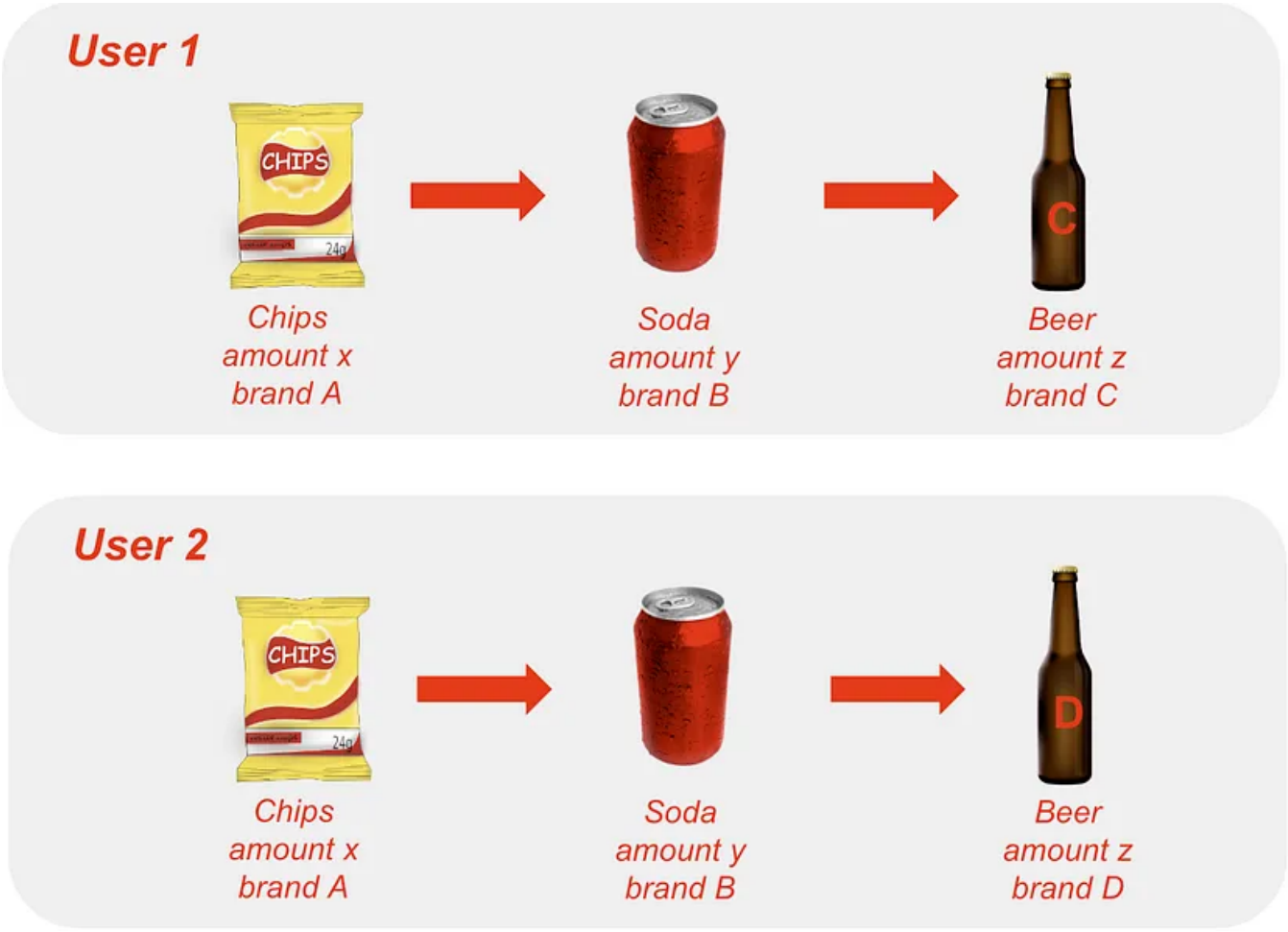
- Milk
- Yoghurt
- Bread

- Bananas
- Cucumber
- Cheese
- Tomatoes

That means that the distribution of buying behaviour is completely dominated by these products.

The subtlety of Customer Preferences

The second reason that makes explore recommendations difficult, is the fact that similar customers might prefer slightly different versions of similar products. For example, you and I might both like crips, but you like Lays and I like Tyrell's. You like Salt & Vinegar and I like Chili. Or maybe you usually buy a family bag and I don't. These types of subtle difference in preferences make recommending exactly the right article difficult (see [fig] for a graphic example). However, for repeat articles, we have a lot more information on buying patterns and subtle preferences than we do for explore articles. Especially if the products are in a category the customer has never bought before.



An example of the subtle differences between user preferences.

Tackling the problem

Because we have no existing approach for explore recommendations at Picnic we figured it best to benchmark a wide variety of models and try to find where they succeed and where they struggle. After reviewing the literature, we decided it was best to cast the problem as a sequential recommendation problem as the field is well studied for a variety of different domains (think of movie, music or news article recommendations). Moreover there exists a particular subfield of next-basket recommendations which is specifically tailored to grocery shopping. Next-basket recommendation models are however often quite complex in their architecture and did not seem like a good starting point for our experiments. We therefore selected the following range of general sequential recommendation models for our benchmarking:

1. Markov Chain models: FPMC, HRM

2. Recurrent Neural Networks (RNN): GRU4Rec

3. Transformer-based models: SASRec, BERT4Rec, LightSANS, RepeatNet

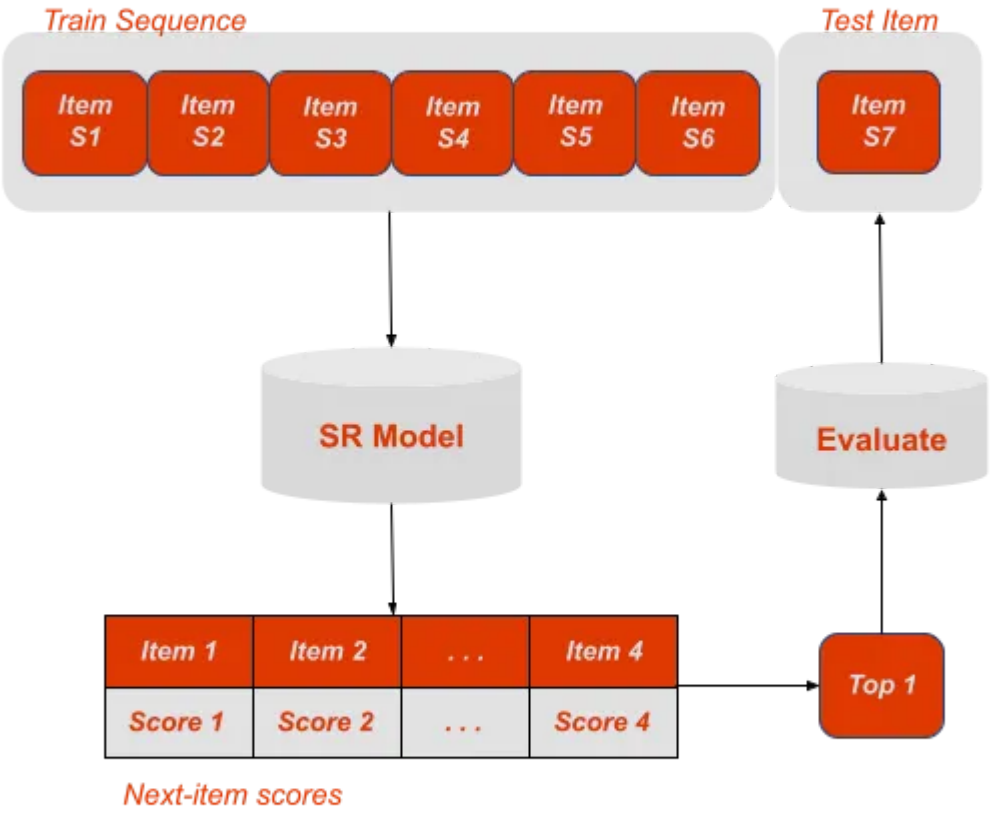
4. Convolutional Neural Networks (CNN): Caser

5. Graph Neural Networks (GNN): SRGNN

The choice for these particular models is mostly due to their state-of-the-art performance on a sequential recommendation task at their respective time of publishing. Additionally they are all implemented in the open-source library RecBole which is a really handy package for comparing recommendation models in Python in a structured and reproducible way.

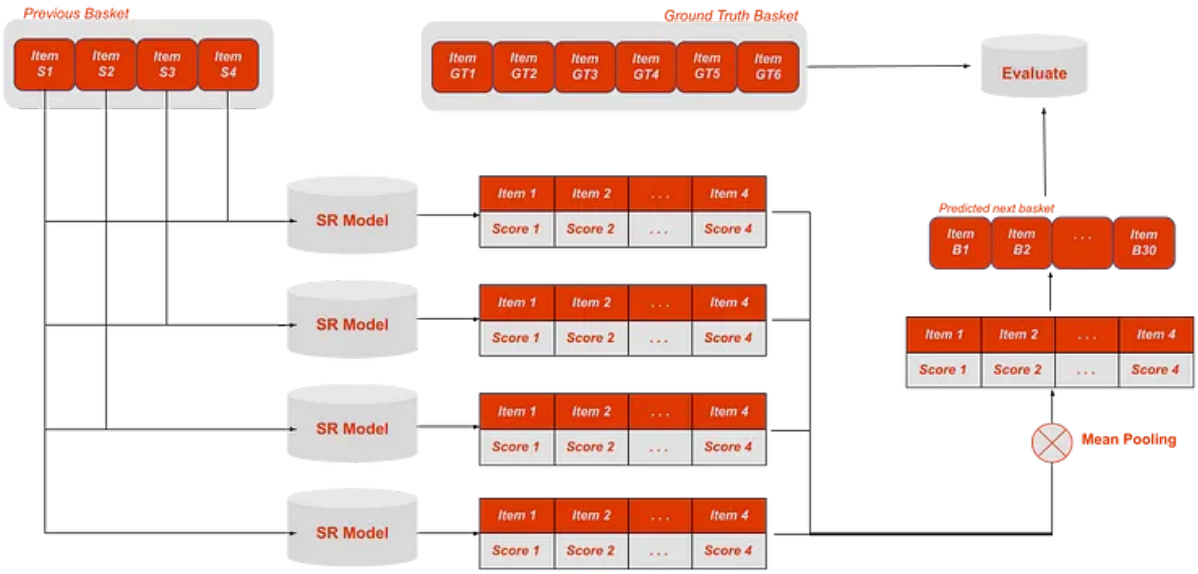
The original papers for all these models can be found linked on the RecBole website and we won't provide any detail in this blog as it is not too relevant.

The one thing all models share among them is their representation of customer-item interactions as sequences in time, where given a customer U_j and their sequence of purchases S_u_j = {i_1, i_2, …, i_n}, the probability of the customer buying a certain new product is defined as: P(i_x|S_u_j) = P({S_u_j + i_x}). The modelling of these patterns ranges from reasonably simple Markov Chains to more complicated approaches such as attention, convolutions or graphs. As always with more complex modelling also comes higher cost in compute and memory so there is a trade-off to keep in mind here.



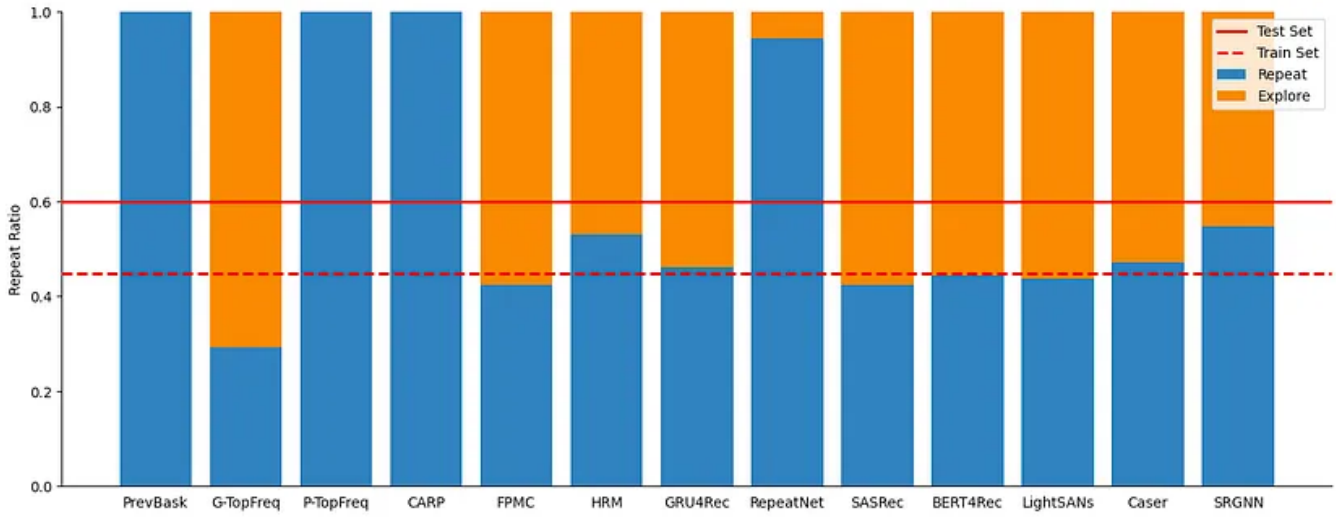
Standard sequential recommendation modelling approach. The input sequence is processed to produce a single “next-item” prediction, which is evaluated against the ground-truth test item.

The general sequential modelling described above was developed in the context of things like movie, music and news-article recommendations. Therefore it makes sense that it only ever predicts a single “next-item” as shown in [fig]. However, grocery shopping tends to work a little different, because customers don’t watch a single movie or listen to one song, but instead buy an entire “basket” worth of products. To better suit this use-case we adapted the basic sequential modelling to a to next-basket recommendation model by representing the previous baskets of customers as interleaving sequences. We can then mean-pool the “next-item” scores outputted by all these sequences to obtain our final prediction [other fig].



Our next-basket adaptation of the sequential model. Each subsequence of the input sequence produces a "next-item" prediction. These rankings are mean-pooled and the top-n of the resulting ranking is outputted as the "next-basket" prediction.

This approach sits somewhere between 'vanilla' sequential recommendation and full next-basket recommendation and to the best of our knowledge it is a novel approach. In our experiments we saw that our method generalises well and is better able to match repeat-explore patterns in the data than previous methods from the field of next-basket recommendation. As can be seen in the graph below most models neatly match the repeat ratio of the train set.



Repeat and explore ratios of the predictions of all benchmarked models compared to the actual ratios present in our Picnic train and test sets.

Key Findings

To test the models we ran experiments on a historic dataset consisting of customer item interactions within Picnic, as well as on the publicly available TaFeng dataset. Evaluation was done by looking at the HitRate@1, Precision@5, Precision@10 and MAP@30 of the predicted next basket. The ground truth was the final basket in the dataset for each customer.

CARP wins overall

Looking at the overall performance (where ground truth baskets contain both repeat and explore items) CARP is the clear winner. It outperforms all benchmarked models by large margins, often double digits. It is important to note here that it is reaching this performance while for part of the basket, i.e. the explore items, it cannot possibly predict them. Which leads us to the repeat performance, where frankly, CARP is a beast.

Repeat is doable

When looking at repeat items only the performance gap becomes even more clear. Here CARP is consistently outperforming the others by double digits. Interestingly the benchmarked models do achieve much better performance on the repeat side than they do on the explore side, which might be indicative of them modelling (with varying success) the sort of patterns that we hard-wire into CARP.

These findings definitely show that in the repeat case, where we have all the data about rebuy patterns and customer preferences available, a simple model with well crafted features is all that you need. And remember over half the shopping done at Picnic is repeat behaviour, so there is a large and relatively simple gain to be had here if you don't have a model like CARP. Additionally (as often within machine learning) these findings make a strong case for being careful in blindly adopting complex models such as transformers and just throwing more and more compute at them.

Explore is really difficult

On the explore side of things the story is rather different however. First of all CARP cannot be used in this setting as the necessary data (which all consists of repeat article information) is not available for explore items, and thus CARP cannot make any predictions. Moreover, we see the performance of the benchmarked models is much lower than it was for repeat items, even the very complex ones. Actually the gap between repeat and explore performance is huge, and repeat performance is often 10x better than explore performance. Moreover in many cases simply using the baseline of always predicting the 30 most popular articles in the entire store

(think back to those bananas and cucumbers) is better in terms of our metrics than using a couple million parameter transformer.

That shows that getting explore recommendations in grocery shopping right is indeed very difficult. And the problems we mentioned in the introduction might be even worse than we feared. However, there is also a different way of looking at explore recommendations, which tries to solve a bit of an easier problem yet still potentially produces a lot of value.

Explore categories are easier!

Remember the main reasons why explore recommendations are so difficult? Well they all boil down to sparsity. Sparsity is often a problem within recommendation systems, not just on the explore side, but it is extra pronounced within this particular setting. However we do have a task which is very similar to explore item recommendation but a lot less sparse: explore category recommendation. Within Picnic we have several levels of categorisation ordered in a tree. With about 20 high level categories such as “Fruit” or “Bread” and hundreds of subcategories such as “Glutenfree Pastry”. This means we can aggregate our recommendation task and make the data less sparse resulting in much better performance. These category recommendations can then be used either as is, or in some clever combination to compute an article level recommendation.

We can make the problem even easier...

Maybe we’ve been a bit too strict..? Evaluating recommendation algorithms purely on whether a recommended article was actually bought in the next delivery is quite a narrow definition of relevance. Maybe a customer loved the products in our recommendations but just didn’t buy them… yet. As is often the case in data science, missing data might be the most interesting data here.

If we look a couple more orders into the future we see that for most models the performance does go up. On one side this is trivial, we are making the ground truth set larger, thus the problem easier. However, it does make intuitive sense that sometimes it takes a week or two for a preference to materialise into a purchase. When testing this hypothesis we do indeed see performance go up in the four baskets after the initial next-basket, which equates to on average about a month of extra time before the purchase is made.

Concluding remarks

In conclusion, delving into the world of recommending unbought articles in grocery shopping poses a fascinating and relevant challenge, with numerous complexities to unravel. At Picnic, we're dedicated to all ongoing development in this and many other fields, focusing on solutions and building stuff that works. Some related initiatives that we are currently working on include "maybe you forgot these articles"- and recipe-recommendations.

If you're passionate about tackling technological frontiers and building cool stuff for a better planet, consider joining the Picnic on our journey of innovation and reshaping the landscape of grocery shopping. We are currently looking for Search&Ranking ML Engineer and many other roles!

References

[1] Ashton Anderson, Ravi Kumar, Andrew Tomkins, Sergei Vassilvitskii. The dynamics of repeat consumption. In Proceedings of the 23rd international conference on World Wide Web, pages 419–430, 2014.

[2] Jun Chen, Chaokun Wang, and Jianmin Wang. Will you "reconsume" the near past? Fast prediction on short-term reconsumption behaviors. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, 2015.

[3] Jun Chen, Chaokun Wang, Jianmin Wang, and S Yu Philip. Recommendation for repeat consumption from user implicit feedback. IEEE transactions on knowledge and data engineering, 28(11):3083–3097, 2016.

[4] Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. A next basket recommendation reality check. 9 2021

[5] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4806–4813, 2019.

AI | Data Science | Recommendation System | Grocery Shopping | Machine Learning

Follow

Written by Thijs Sluijter

35 Followers · Writer for Picnic Engineering