What is the impact of presenting frequent buyers with items similar to those that were outliers in their previous purchasing patterns?

Gauthier Le Compte
Data Science and Artificial Intelligence
University of Antwerp
gauthier.lecompte@student.uantwerpen.be

***Abstract*—This report delves into the different buying patterns of frequent buyers and explores the impact of presenting them with items similar to previous outlier purchases. The methodology includes defining a threshold for frequent buyers, identifying outlier articles based on price and buying frequency. The research also investigates generating similar item candidates for these outliers and splits data into two separate models for frequent and non-frequent buyers. The results show very slight improvements among frequent buyers, especially when combining similar articles based on price and article categories. The novelty scores indicate the system's effectiveness in introducing new items to users.**

1. Introduction

Not every buyer has the same buying pattern. Some buy more conservatively, while others purchase a lot more due to higher budgets or other factors. This seemed like an interesting topic to research: how do these buying patterns differ from one group of buyers to another? Initially, I wanted to analyse the impact of presenting frequent buyers with items they've never purchased before, in terms of colour or style. The thought process was based on the fact that these groups buy a lot and/or have a lot of money to spend. So, the chances that they would buy an item in which they have never shown interest seemed higher to me. However, this research question was quickly abandoned as the results weren't promising from the start, and in a setting like Kaggle, it was also doomed to fail. In a real-life dynamic environment, this could have been more interesting with A/B testing, or other more dynamic methods, but I was stuck with a fixed test set.

Still, the thought of these frequent buyers having some sort of buying pattern that differs was intriguing. Therefore, the final research question to be analysed in this paper delves deeper into the impact of presenting frequent buyers with items similar to outliers they have purchased before. This removes the complete randomness of the previous research question, while still maintaining the same concept: presenting a specific niche of frequent buyers with items they have shown interest in before, but was considered more a of rare case.

2. Methodology

2.1. Classifying Frequent Buyers

The first step in the process is, of course, defining a threshold for frequent buyers. It doesn't always have to be some state-of-the-art method; sometimes the simple tricks work just fine. In this case, the 90th quantile was primarily used as the threshold. This led to identifying about 36,999 frequent buyers out of a total of 412,745 customers, each with more than 12 transactions in the last 8 weeks. Later in the results, we will delve deeper into the effects of altering this threshold, as there were some interesting insights gained from changing this parameter. Bulk buys were also excluded, as we didn't want a scenario where someone placing an order for 100 small and cheap needles would be categorised as a frequent buyer. Another path we could have taken here was using a Machine Learning Method, like a tree-based model, to classify frequent buyers. Although this would have been an interesting approach, it was a bit out of scope for the research question and not what I wanted to focus on.

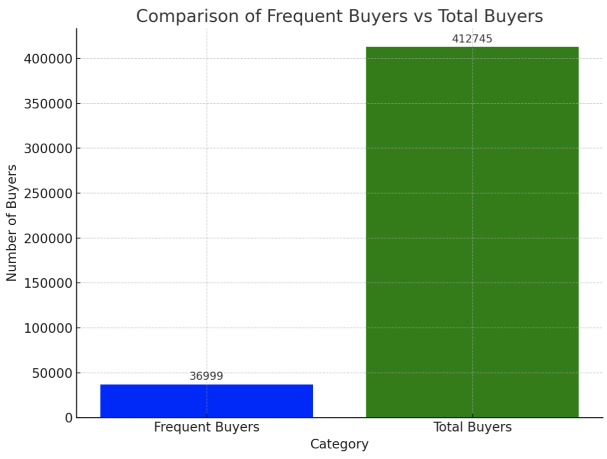

Figure 1: Frequent Buyers Chart

2.2. Defining Outliers

Defining outlier articles for a frequent buyer customer involved two different approaches. The first focused on outliers based on the price of an article. Working with numerical values was relatively straightforward: we calculated the lower and upper bounds using the IQR method. Any articles priced below the lower bound or above the upper bound were deemed outliers.

The second method to detect outliers was based on the articles' features. This was more challenging as categorical variables don't lend themselves to straightforward outlier calculation. After exploring several methods, such as the Chi-Square test, none completely met the needs I was looking for. Hence, we opted for a simpler approach: Buying Frequency. For each customer, we calculated their least purchased categories, product types, and colours. An article was considered an outlier if it met either of the following conditions:

- It matched the customer's least bought product group and colour group.
- It matched the customer's least bought product type and colour group.

It is also worth nothing that these features were not chosen randomly; they were picked using the feature importance measure in Python, which indicates which features most significantly impact model performance. We attempted other category matches, but these were not very successful

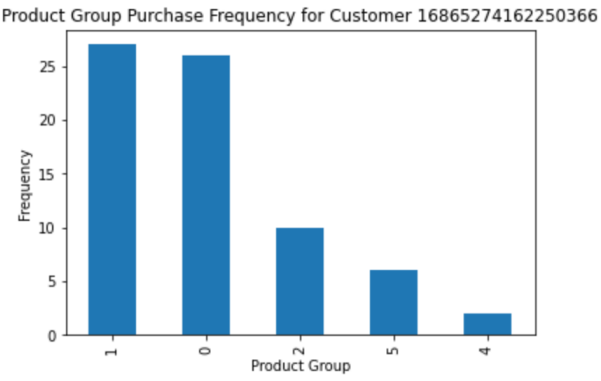

Figure 2: Least bought categories

Above you can see an example for a certain user, where it's clearly visible that this user shows a lot of interest in product group 1 and 0, product group 2 and 5 are more average, but product group 4 can be considered as an 'outlier' group compared to the other product groups.

3. Generating Similar Candidates

Building on the outliers identified in the previous step, the next task was to generate similar items for all these outliers. For price-related outliers, we established a price range. If an article's price was within 5% above or below this range, it was marked as similarly priced. For outliers based on article categories, we used the Cosine Similarity method to create a top-k similarity matrix for all outlier articles. The features considered for the cosine similarity were again based on our model's feature importance results, including product type, colour group, product group, and garment group. These features were hot encoded to ensure proper working with the cosine similarity.

With normalised features, we compute the cosine similarity matrix which represents the similarity between every pair of articles in our filtered dataset.

Similar to detecting outliers based on certain categories, finding the most optimal list of features could have potentially been more precise but would have required an extensive and computationally intensive hyperparameter tuning process. The features we used were a mix of trial and error and insights from the feature importance results of our model.

3.1. Model

Initially, Radek's code used a single model for all customers, which was logical in that context. However, with two distinct customer groups who we assume have different buying patterns, it made sense to split these into two separate models. Consequently, the models required different data sets.

The Non-Frequent Buyers continued to use the same data as before: the transactions, bestseller candidates, and last purchase candidates. The Frequent Buyers, however, were trained not on all transactions but solely on those made by frequent buyers. This approach was generally effective, but as the frequent buyer threshold increased, they had access to less (and potentially less representative) training data, which affected the model's performance. They also received the bestseller candidates and last purchase candidates, along with either the price outlier candidates or the article outlier candidates. I also experimented with a model that combined both price outlier and article outlier candidates, which, as you'll see in the next section, gave the best results.

The data for Non-Frequent Buyers was inputted into the standard model, the same as the Radek baseline. The Frequent Buyer data, on the other hand, was fed into a different model where the n_estimator value was adjusted for slightly better performance. In the end, the predictions from both models were combined for the final output.

3.2. Results

3.2.1. Combined Results. Figure 3 below shows the combined results for both non-frequent and frequent buyers.

	MAP@12	Recall@12	Precision@12
Radek Base	0,02312	0,04609	0,00981
Price Outliers	0,0233	0,0463	0,0099
Article Outliers	0,02335	0,045989	0,00985
Price & Article Outliers	0,02340	0,04625	0,00986

	ΔMAP	ΔRecall	ΔPrecision
Radek Base	0	0	0
Price Outliers	0,00018	**0,00021**	**0,00009**
Article Outliers	0,00023	-0,000101	0,00004
Price & Article Outliers	**0,00028**	0,00016	0,00005

Figure 3: Combined Results

Initially, these results might seem underwhelming. However, considering the distribution of frequent versus non-frequent buyers, as illustrated in figure 1, the limited impact on overall results is understandable. Significant improvements among frequent buyers alone would only minimally affect the combined outcome due to their smaller proportion. Our focus is on a specific niche within frequent buyers, particularly the outliers, so a big change in results wasn't expected and actually would have been surprising. This got me wondering, what if we only look at how the frequent buyers' results changed?

3.2.2. Frequent Buyer Results. Figure 4 shows the results for only the frequent buyers.

	MAP@12	Recall@12	Precision@12
Radek Base	0,02312	0,04609	0,00981
Price Outliers	0,02331	0,04631	0,00991
Article Outliers	0,02335	0,045989	0,00985
Price & Article Outliers	0,02340	0,04625	0,00986

	ΔMAP	ΔRecall	ΔPrecision
Radek Base	0	0	0
Price Outliers	0,00019	**0,00022**	0,0001
Article Outliers	0,00023	-0,000101	0,00004
Price & Article Outliers	**0,00028**	0,00016	**0,00005**

Figure 4: Frequent Buyer Results

Interestingly, both recall and precision improved for this group. Initially unexpected, this outcome is a pleasant surprise. The combination of similar articles based on price and article categories showed the most improvement in both MAPK and Recall, with an increase of 0.00224 over the baseline model for MAPK and 0,00127 for Recall. The methods using price and article categories separately also improved over the baseline but were less effective than the combined approach.

I experimented with the frequent buyer threshold, adjusting the quantile to include more or fewer frequent buyers. Following the assumption that frequent buyers have different buying patterns, decreasing the range should lower MAPK and Recall scores, whereas increasing it should improve them. Figure 5 and 6 show the MAPK and Precision across different quantile values.

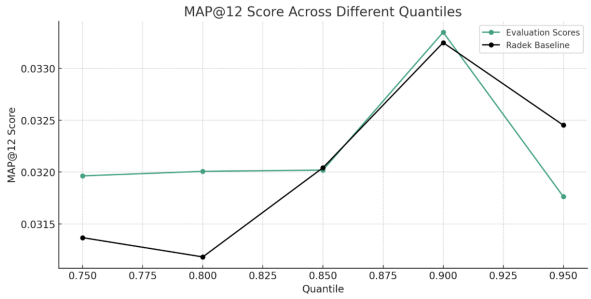

Figure 5: Quantiles MAPK

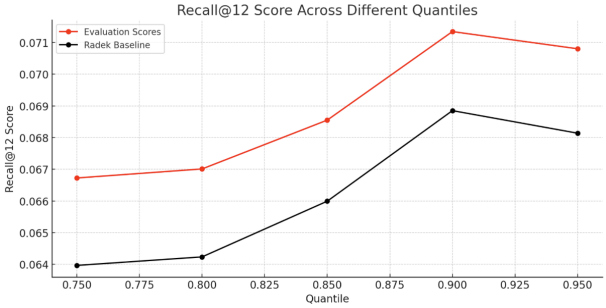

Figure 6: Quantiles Recall

And as you can see, lower quantile values led to lower scores, with an increase observed up to the 90th quantile. Beyond this point, scores began to decrease, which contradicts our assumption. This drop could be due to providing only frequent transactions to a smaller group of frequent buyers, resulting in less relevant training data and lower scores.

I also played with the n_estimator rank as this was initially set to 1 which seems suboptimal for such a large dataset. To enhance this, experiments were conducted with the following values [1, 10, 25, 50, 100]. This allows the model to learn more complex patterns by using a larger number of decision trees, which then leads to more accurate predictions. Overall n_estimator with a value of 50 performed the best. There were loads of articles online which suggested trying 500 or even 1000 for larger datasets, but unfortunately due to limited computational resources, this wasn't a viable option.

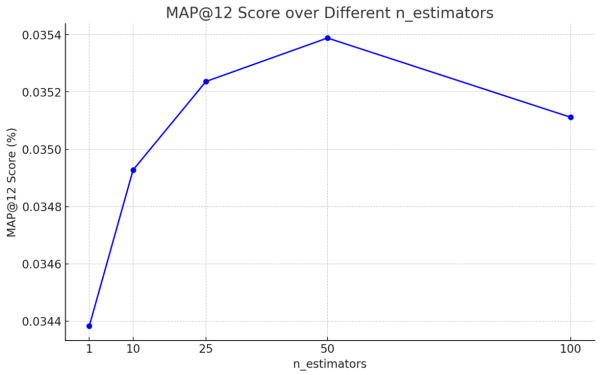

Figure 7: MAPK over different n_estimators

3.2.3. Novelty. Novelty measures how unfamiliar recommended items are to a customer. A higher score means that the model is suggesting items that a user has not seen or interacted with before. Results for the novelty scores are:

- **Baseline**: 0.9995
- **Own Implementation**: 9997

These high values suggest that both systems are effective in introducing new or less familiar items to users. There is a slight improvement in my system over the baseline, which is a positive sign.

The interaction frequencies with a lot of items in my dataset are quite low, indicating that most items are relatively rare and appear only a few times. This could mean that the dataset has a long-tail distribution, common in many retail datasets, where a few items are very popular (high frequency), while a large portion are rarely or less commonly interacted with.

4. Discussion

My initial method, where I used a certain quantile as the frequent buyer threshold, seems to work just fine. However, there most likely is a better optimal method for this, as my method doesn't take into account all factors like how often they buy on average, how much they spend, etc. So, a decision tree-based model would be interesting to see how it classifies buyers into different groups and how this affects the results.

A few other things that I tried were removing the outliers from the data. I attempted this for only the frequent buyers, only the non-frequent buyers, and for all users, but this did not improve results. Generating too many additional candidates had a negative impact as our users would get flooded with too many "unpopular" items. When I tried removing the bestseller rank, my score decreased significantly, once again indicating the importance of popular and best selling items in the e-commerce setting.

Another approach that could potentially have been used, and perhaps was even more suitable, is using cosine similarity for outlier detection. This could provide more personalised recommendations since it considers the overall profile of customers' preferences and can capture better relationships between articles. It also works well in complex datasets with many attributes. However, this method is more costly, especially in such large datasets, as it can become computationally very expensive.

As previously mentioned, the data for the frequent buyers was different, where they only received the transactions for frequent buyer transactions. The other buyers still received all transactions. When giving the non-frequent buyers only the non-frequent transactions, I noticed there wasn't any big change, so I just left this as it was.

The features considered for outlier detection and cosine similarity were based on the feature importance results from my model. However, additional tests could be done to perhaps find an even better set of parameter combinations. In general, I feel like there are a lot of parameters in my model, each having quite an influence on its performance. Hypertuning all these parameters would have been computationally and time-wise too big of a task, but there certainly is a parameter setting/combination that would provide even better results.

Finally, I also experimented with one model versus two separate models. As expected, the two separate models, where the output was combined, performed the best.

5. Conclusion

This study's research into the buying patterns of frequent buyers revealed some interesting insights. The global results, although not ground breaking, become more meaningful when focusing on only on the relevant portgion of this research, the frequent buyers.

For this group, we observed notable improvements in recall and precision and mapk, especially when items were presented based on combined price and article categories. This suggests that targeted recommendations for specific buyer segments can be quite effective. Experimenting with the frequent buyer threshold and n_estimator values helped fine-tune the model, showing the importance of adjusting parameters for better results. The novelty scores also indicate that the system is successfully introducing new items to users, a promising sign for e-commerce platforms. Overall, the study highlights the value of understanding and catering to the distinct preferences of frequent buyers.

References

[1] R. Osmulski, [LB 0.19] LGBM Starter Pack, in *H&M Personalized Fashion Recommendations*. https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations/discussion/309220.