H&M Personalized Fashion Recommendation System

1 Introduction

This work focuses on developing product recommendations for customers based on data from previous transactions, as well as from customer and product meta data. The available meta data spans from simple data, such as garment type and customer age, to text data from product descriptions, to image data from garment images. The data set includes transactions of previous customers, descriptions of the articles bought by the customers and demographics of the customers. The images of the articles are also present in the data set. The goal of the project is to recommend products to the user. The model will be evaluated by predicting the products that the user will buy in the week after the training data ends. The evaluation metric that will be used is Mean Average Precision for the twelve items that are to be predicted.

2 General Idea

To predict articles for each customer, we can either look at general purchase behaviors across all customers and use that to recommend items or look at customer's behaviors and make specific predictions by looking at their buying trends. In general, an amalgamation of these two methods gave us the best results. Default predictions for those having no prior purchase history were made and user's buying interests were considered if they had a purchase history.

There were several observations from the data set and intuitions from people's buying trends that led us to finalize the models for making predictions. For example, overall, sales for most items sold in each category was predominantly from the ladies item group (Figure 1). So, if our model's predictions would recommend more of ladies apparel, it would be predicting a more likely purchase according to the purchasing trends.

Similarly, we observed that the sales of sweaters and jackets increased more in the autumn and winter season (Figure 3). Since the prediction period was in the autumn season, predicting seasonal popular items should ideally have given a high score. However, this test prediction time span coincided with peak Covid lockdown time. Hence, people's buying trends varied and they tended towards buying other items like shirts and cardigans (Figure 4) as they had to work from home.

Another observation was that most people purchased black colored clothes (Figure 2) and so, giving higher weight-age to black colored clothing item predictions for each customer turned out to have a significant impact on the prediction scores.

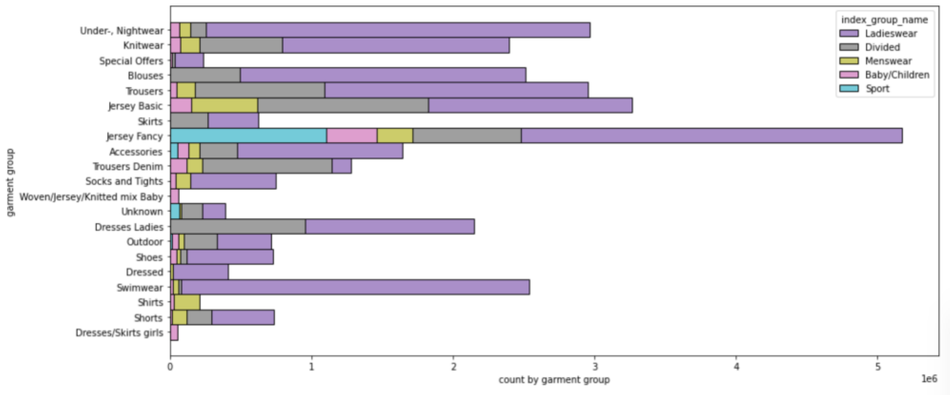

Figure 1: Distribution of Sales across different Customer Groups

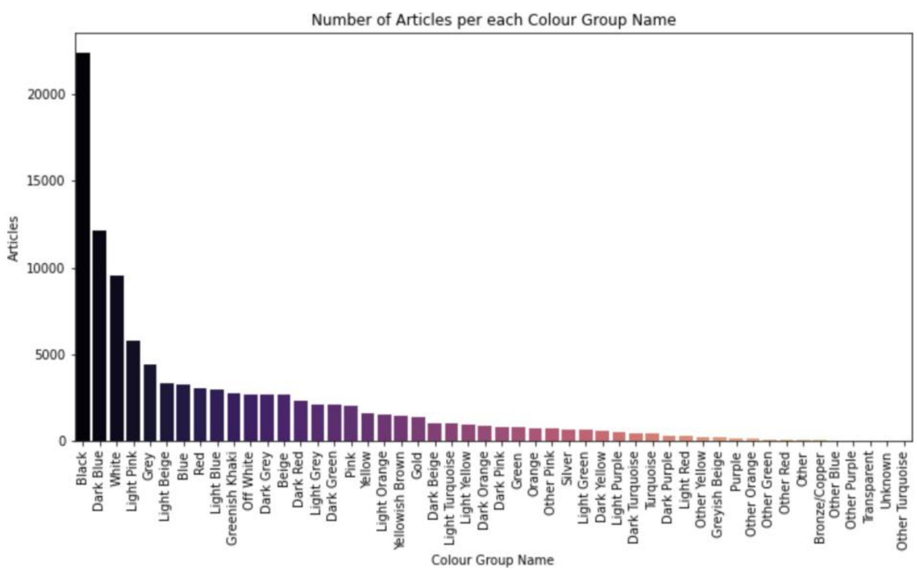

Figure 2: Color Distribution over Articles Bought

3 Data Pre-Processing and Cleaning

The following steps were performed for pre-processing and cleaning of the data:

- Load and keep only useful columns for the three data frames in the data set provided.
- Decrease the size of the data being used by filtering it using factors like top customers, most frequent articles, most recent purchases, et cetera.
- Drop Null values.
- Normalize data being used for collaborative filtering.
- Create a feature called purchase count which counts the total number of items that the user has purchased.
- Create a dummy binary feature that indicates if the user has made any purchases or not.
- Convert indexes such as customer id, article id to int data type since it requires lesser memory as compared to the original data format.
- Load pandas data frame using cudf to process it on GPU for faster processing.
- Since machines have limited RAM, delete the variables and data frames using garbage collector, when not required.

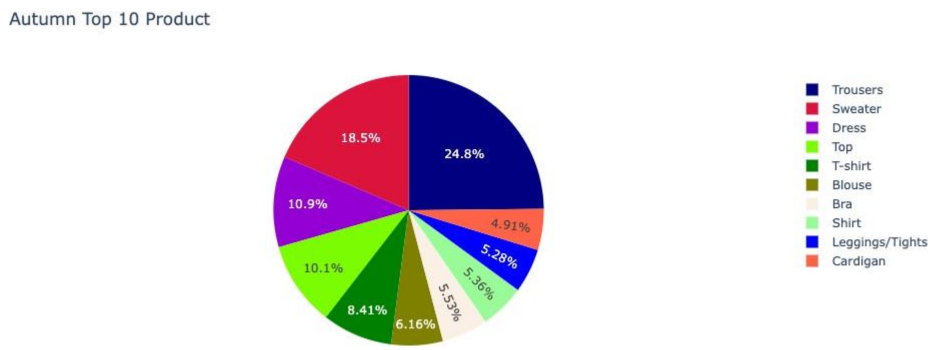

Figure 3: Article Type Buying Trends during Autumn 2020 (Peak Covid lockdown)

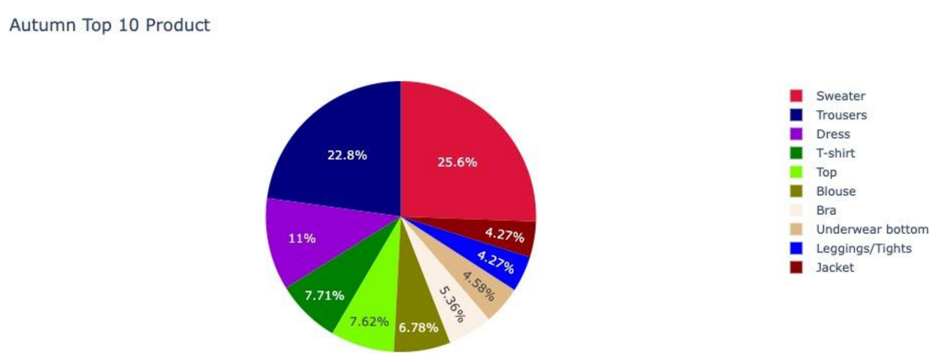

Figure 4: Article Type Buying Trends in Autumn 2018-2020 (General)

4 Algorithms and Models

4.1 Collaborative Filtering

In this algorithm, implicit user to user patterns and user to article patterns were exploited to provide recommendations to the users. A new data frame was created containing users as rows and articles as column ids. The cell values had binary values indicating if a user had purchased that particular item or not. This algorithm used transaction history to understand user patterns and so, it was important to have user's history to predict the items. This is why recommendation for all the users could not be predicted using this model and for such users, predictions from other algorithms were used. The important parameters for this algorithm were the total number of articles and the total number of users. The top articles were the ones which were most frequently purchased and the top users were the most frequent users.

Table 1: Collaborative Filtering

Total Users	Total Articles	Score
800,000	20,000	0.0055
600,000	20,000	0.0092
400,000	10,000	0.0223

Table 2: Customer Age + Trending Products Weekly + Who Bought This Frequently Buy This

Recommendations from (Customer Age + Trending Products Weekly)	Recommendations from (Who Bought This Frequently Buy This)	Total Predictions	Score
6	6	6+6 = 12	0.0222
9	3	9+3 = 12	0.0225

4.2 Customer Age + Trending Products Weekly + Who Bought This Frequently Buy This

For this idea, customers were grouped into distinct age bins, determined by EDA. Then, the latest popular articles in each ages between the dates of 2020-09-01 and 2020-09-21 (last transaction date in training data) were checked. The top predictions from this algorithm were taken as part of the final predictions. The other predictions were made on the basis of the 'Who Bought This Frequently Buy This' idea. Here, items that were purchased together with the top predictions that we got earlier, were found. The top predictions and these predictions combined together made the final twelve predictions. (Table 2)

4.3 Grouping Customers by Average Item Price

The general intuition behind this algorithm was that people generally have a budget for the items that they buy. If someone usually bought items in the price range of 10 to 20 dollars, we could collect data of trending items across all the training data to give predictions of similar price range items.

For this model, we calculated the average price that a customer pays for an item. Then, we grouped all the customers in the data set into avg-cost bins. Predictions for each customer were made based on the most popular items in each bin + customer purchase history trends (most frequently bought).

4.4 Ensembling

In this approach, six main ideas were merged namely 'Last Purchase and Other Colours', 'Trending', 'Trending Products Weekly', 'Customer Age + Who Bought This Frequently Buy This', 'Collaborative Filtering' and 'Price'. Each of these algorithm's submissions were given different weights and a list of final predictions was obtained.

In the 'Last Purchase and Other Colours' algorithm, first, the last purchased items were found and then, other colours of the purchased items were found. Also, popular items were found and the predictions from these three methods were blended. The 'Trending' and ' Trending Products Weekly' methods dealt with the most trending products in a recent transaction period. The remaining three algorithms were mainly dependent on customer age, collaborative filtering and price.

5 Results

On trying several different algorithms, we found that there was no one particular method that worked best, but an ensembling of several methods. Around 6 different algorithms were combined together and ensembled with different weights to produce a score of 0.0243 on the public leader board. Several ideas like last purchase and other colours, trending products weekly, customer age, collaborative filtering, price, et cetera, were used and the weighted predictions from all of these ideas were combined together to give the final twelve predictions. (Table 3)

6 Other Insights

These are some methods that we tried that did not yield good results:

- Purely Seasonal Based Approach- This is because we predicted the most popular items across all transactions spanning the months of August to October. This method did not

Table 3: Results

Sr. No.	Algorithm	Individual Score	Weight
1	Last Purchase and Other Colours	0.0232	1.2
2	Trending	0.0231	0.88
3	Trending Products Weekly	0.02232	0.08
4	Customer Age + Who Bought This Frequently Buy This	0.0225	0.07
5	Collaborative Filtering	0.0223	0.05
6	Customer Average Price	0.0221	0.05

involve any customer specific predictions and the test period for the predictions fell under the Covid lockdown period, therefore customer purchase trends changed a bit due to these reasons.

- Neural Network Based Approaches like LSTMs- These models took a lot of time to train and the results were not satisfactory as compared to the non-neural network based approaches.
- Collaborative Filtering on Large Data Set (approximately 800K customers)- Scores were comparatively lower when considering a large number of customers. Reducing the data set to consider only the top 400K customers with the highest purchase history gave the best results.

In conclusion, working with lesser data proved to be more beneficial, i.e. by choosing only the top customer data or by considering transactions closer to the prediction dates. Ensembling and combining results from different models also proved to be more beneficial rather than making predictions using a single model.

References

[1] Hsieh, C. K., Campo, M., Taliyan, A., Nickens, M., Pandya, M., Espinoza, J. J. (2018). Convolutional Collaborative Filter Network for Video Based Recommendation Systems. arXiv preprint arXiv:1810.08189.

[2] Ames, Jewel. “The Secret Sauce Behind Search Personalisation.” Gojek, 3 Dec. 2019, www.gojek.io/blog/the-secret-sauce-behind-search-personalisation.

[3] Yan An, Tong Zhao, (2022). Personalized complementary product recommendation. Amazon Science. https://www.amazon.science/publications/personalized-complementary-product-recommendation

[4] Karpov, Dannil. “EDA First Look.” Kaggle, 2022, www.kaggle.com/code/vanguarde/h-m-eda-first-look.

[5] Borisova, Mirena. “HM BY CUSTOMERS AGE.” Kaggle, 2022, www.kaggle.com/code/mirenaborisova/h-m-by-customers-age-lb-0-0227.

[6] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18). Association for Computing Machinery, New York, NY, USA, 31–39. https://doi.org/10.1145/3240323.3240354

[7] Harald Steck. 2018. Calibrated recommendations. In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18). Association for Computing Machinery, New York, NY, USA, 154–162. https://doi.org/10.1145/3240323.3240372