H&M Personalized Fashion Recommendations

Yanjian Zhang
Facultad de Informática de Barcelona
Universitat Politècnica De Catalunya
yanjian.zhang@estudiantat.upc.edu

Yi Wu
Facultad de Informática de Barcelona
Universitat Politècnica De Catalunya
yi.wu@estudiantat.upc.edu

1 Task description

H&M online store offers an extensive selection of products to browse through. In the kaggle task (https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations/), we were asked to develop product recommendations based on data from previous transactions, as well as from customer and product meta data.

1.1 Dataset Overview

For tabular data, there are 3 tables.

- Transactions_train.csv for transactions with datetime and a span of two years, from Sep 20th, 2018 to September 22th, 2020. Duplicate rows correspond to multiple purchases of the same item.

	Transactions_train.csv
size	3.49GB
shape	(31788324,5)
columns	[’t_dat’, ’customer_id’, ’article_id’, ’price’, ’sales_channel_id’]

- Customers.csv for metadata for each customer_id in dataset.

	Customers.csv
size	207.1MB
shape	(1371980,7)
columns	[’customer_id’, ’FN’, ’Active’, ’club_member_status’, ’fashion_news_frequency’, ’age’, ’postal_code’]

- Articles.csv for detailed metadata for each article_id available for purchase.

	Articles.csv
size	36.1MB
shape	(105542,25)
columns	[’article_id’, ’product_code’, ’prod_name’, ’product_type_no’, ’product_type_name’, ’product_group_name’, ’graphical_appearance_no’, ’graphical_appearance_name’, ’colour_group_code’, ’colour_group_name’, ’perceived_colour_value_id’, ’perceived_colour_value_name’, ’perceived_colour_master_id’, ’perceived_colour_master_name’, ’department_no’, ’department_name’, ’index_code’, ’index_name’, ’index_group_no’, ’index_group_name’, ’section_no’, ’section_name’, ’garment_group_no’, ’garment_group_name’, ’detail_desc’]

For image data, it's used as a supplementary of products' meta data, to show how products looks like. Images are placed in subfolders starting with the first three digits of the article_id; note, not all article_id values have a corresponding image.

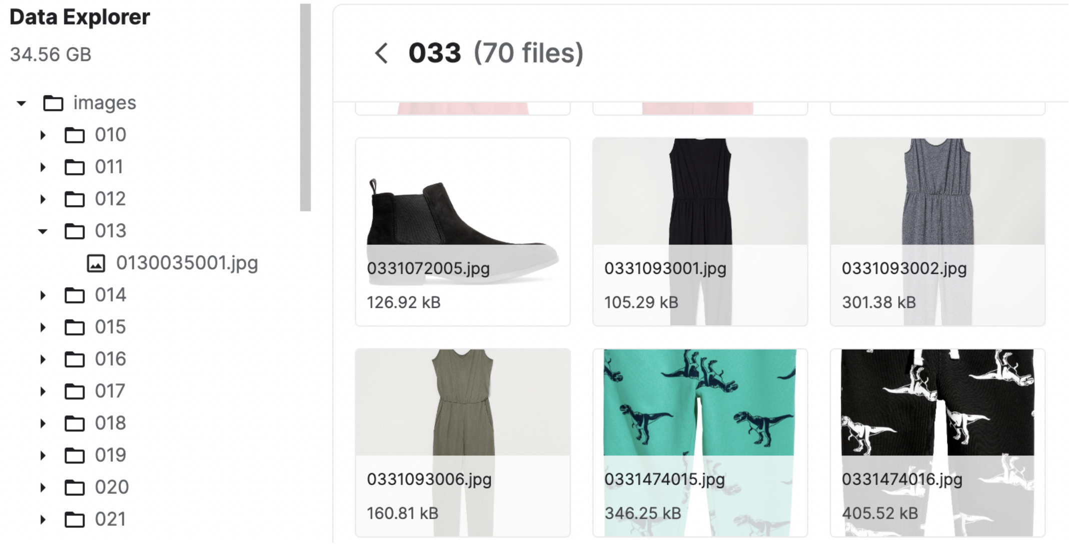


Figure 1: Image Data for Articles

1.2 Goals and Evaluations

The task is to predict up to 12 article_ids each customer_id will purchase during the 7-day period immediately after the training data period. Some customer_ids in test set never appears in the training set before.

Submissions are evaluated according to the Mean Average Precision @ 12 (MAP@12):

$$MAP@12 = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{min(m,12)}\sum_{k=1}^{min(n,12)} P(k) \times rel(k)$$

where U is the number of customers, *P(k)* is the precision at cutoff k, n is the number predictions per customer, m is the number of ground truth values per customer, and *rel(k)* is an indicator function equaling 1 if the item at rank k is a relevant (correct) label, zero otherwise.

2 Work Flow

Our work is about the recommendation system. The data feature embraces both time series feature and tabular plus images data set. Apart from the traditional recommendation system, we decided to try out RecBole recommendation library, which contains the most state-of-art Sequential Recommendation models. However, RecBole supports only atomic files instead of csv, and it offers 28 atomic file example data sets. In order to use our own data, we need to modify the RecBole/data source code to create atomic files for the kaggle data. Then the data can be trained via RecBole system.

Our workflow is as blow.

- Download customers, articles and transactions csv from Kaggle, along with articles image set.
- Preprocess all tabular data, including statistical description, filling null values, normalizing numerical columns, change id representation from encrypted long strings

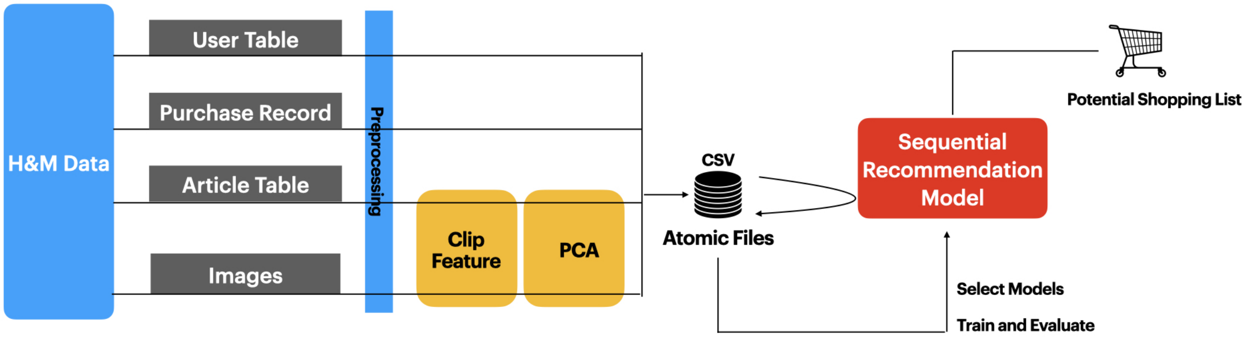


Figure 2: Work FLow

to simplified numbers, etc. Note that there is no need to conduct one-hot encoding, since the RecBole will do that internally in the last step.

- Merge the articles images with article text description via Clip(1) to generate new features, and get the final preprocessed data.
- Moidify Recole/data source code to transform the final preprocessed data into required atomic file formats.
- Utilize the state-of-art Sequential Recommendation models from Recbole to predict the potential shopping list.

3 Feature Engineering

Since the data set contains both tabular data set and images data, the feature engineering contains two part. First part is for tabular data only. Second part is to implement Clip(1) on articles text description and articles image set.

The step-by-step illustration can be found in the attached file *FeatureEngineering.py*. In this chapter, main operations will be addressed.

3.1 Customers.csv

Drop Unnecessary Columns

The primary key contains 1371980 unique values, while the column postal_code contains 352899 unique values, nearly one third of the previous; and the postal_code content is encrypted text. Thus it can be inferred that postal_code contains very few meaningful information. We decided to delete this column.

After a close observing on the data, it's found that the columns FN and Active are almost synchronous with the column fashion_news_frequency. After checking the Pearson correlation, their correlations reach above 95 percent of each other. We decided to reserve colunmn fashion_news_frequency only and delete the other two, since the colunmn fashion_news_frequency has already filled the nulls with string 'NONE'.

Fill Nulls for Age

To fill the overall null values of the Customers.csv, if it's a categorical column, fill with string 'NONE' as a category. The reasoning is that, for column fashion_news_frequency, null values cover more than 50 percent of the data, so it makes sense to treat 'NONE' as a category. For column club_member_status, it contains no more than 1 percent of null values, we could either delete them, or fill with the mode, or fill with 'NONE' as a category. If it's numerical column, for example, column age, the regression models are implemented to fill the nulls.

Three regression models are considered at first to predict null values of age, which are SVM regression, linear regression and decision tree regression. However, the data set size is too big for SVM regression to run quickly, even we tried the linear kernal. So finally, only linear regression and decision tree regression are tested with results.

The results are evaluated by both MSE and MAE as below. Typically the MSE is more sensitive to observations that are further from the mean, which means that account for outliers more in the loss. Decision Tree Regression is slightly better than the Linear Regression in this case, so we use the former to predict age.

	MSE	MAE
Linear Regression	203.02547291078378	12.387169015046158
Decision Tree Regression	198.55994977163093	12.223805367042202

But none of them has gained good performance. Thus we decided to fill the 'age' nulls via mean value.

Simplify the Customer_id Format

Another feature engineering we conduct on Customers.csv is that, in order to save the storage space and lessen the computation cost, the encrypted string-format customer_id has been rewritten into numerical format, through a dictionary indexes. In the Transaction_train.csv, the column customer_id is also transformed.

Finally the preprocessed result is output as Customers_preprocessed.csv.

3.2 Articles.csv

3.2.1 Clip-based featuring on text and images

We utilized Clip(1), which is a model Connecting Text and Images, to transform the detail_desc columns in article table and the images into 512-dim vectors. For the article without corresponding images, we artificially create a pure black(all 0 images) to fetch their features.

3.2.2 Principal Component Analysis

Since the description feature and images feature, which are both 512 dimension, are too large for the model, we use Principal Component Analysis to project the 512 dimensions to 20 dimensions.

3.2.3 Drop Unnecessary Columns

The feature engineering on Articles.csv mainly focuses on dropping the unnecessary columns.

Some columns are the sub class of the other columns, for example, in table Articles.csv, column 'product_type_name' with 131 unique values is the subclass of column 'product_group_name' with only 19 unique group values. In this case, information is duplicated, we delete the superclass columns to kill duplication and keep detailed information. In same case, columns 'index_group_name', 'garment_group_name' are also deleted.

Some columns are the similar representation of other columns, in table Articles.csv, 'product_type_name' has similar meaning as 'department_name', thus 'department_name' is deleted.

Also, out of 105542 unique article_id, there are 47224 'product_code' and 45875 'product_name', thus these two columns are deleted. It's too expensive to get them one-hot encoded.

3.2.4 Merge original features with clipped features

Finally, both image features and description text features are merged with original features, and output as Articles_preprocessed.csv.

3.2.5 Transaction_train.csv

Simplify the Customer_id Format

Towards column customer_id, it's again transformed from the encrypted string-format into numerical format.

Transform the Timestamp Format

In order to fulfill the data format requirements of the RecBole, the column t_dat is transformed from date type to epoch type. The column sales_channel_id is left unchanged since the RecBole will do one-hot encoding internally for categorical columns.

Scale the Price

Column price is scaled via Min-max scaler.

Finally the preprocessed result is output as Transaction_train_preprocessed.csv.

4 Prediction via Sequential Models

We run three sequential prediction models in our dataset, including GRU4Rec(2), SINE(3) and LightSANs(4)

4.1 Transform Data into Atomic Files

RecBole is an efficient framework t0 reproduce and develop recommendation models. In the lastest release, it includes 77 recommendation algorithms and provides the support for 28 benchmark recommendation datasets. In order to use RecBole, we need to convert original datasets to the atomic file which is a kind of data format defined by RecBole via its conversion_tools source code(5).

A new class called HMData was added into script *../conversion_tools/src/extended_dataset.py.* Also the struct of the data schema was claimed in it, where the data types shall be token, token_seq, float, and float_seq etc., just another naming of integer, string, paragraph, float, etc. Lastly the HMData is appended into the data set list in script *../conversion_tools/src/utils.py.*

Finally the atomic files *hm.inter, hm.user, hm.item*, which correspond to transactions, costumers and articles data were generated to be feeded into Recbole.

4.2 Configuration & Environment

We randomly split our dataset into training set, validation set and test set with the proportion of 18:1:1.

We use four GPUs of RTX 3090 for model training. We train 50 iterations for all the models.

We choose the best performance model in validation set and evaluate their performance in test set based on metrics of MRR@10(Mean Reciprocal Rank), NDCG@10(Normalized Discounted Cumulative Gain), Hit@10, Precision@10 and Recall@10.

4.3 Result and Evaluations

For validaiton set, the best model result as follows:

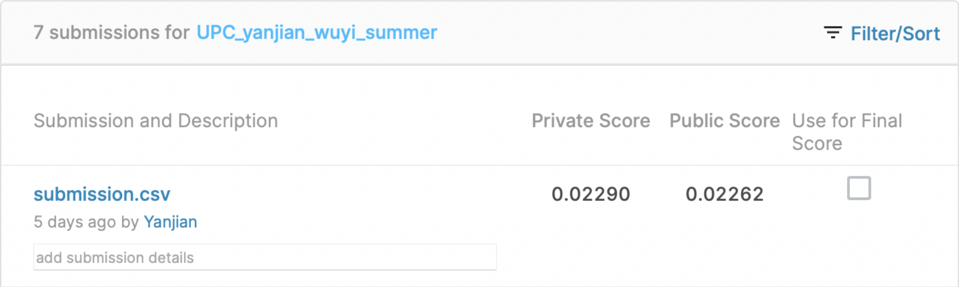
Model	Recall@10	MRR@10	NDCG@10	Hit@10	Precision@10
GRU4Rec	0.0281	0.0137	0.0172	0.0281	0.0028
SINE	0.092	0.0609	0.0685	0.092	0.0092
LightSANs	**0.1093**	**0.0977**	**0.1005**	**0.1093**	**0.0109**

For test set, the best model result as follows:

Model	Recall@10	MRR@10	NDCG@10	Hit@10	Precision@10
GRU4Rec	0.0262	0.0122	0.0155	0.0262	0.0026
SINE	0.0792	0.0521	0.0586	0.0792	0.0079
LightSANs	**0.1704**	**0.1556**	**0.1591**	**0.1704**	**0.017**

We choose the best model(LightSANs) and generate the predictions for privates test dataset evaluation in H&M competition.

For summiting to H&M Competition, we create the based submission with HM: Faster Trending Products Weekly and replace the prediction that LightSANs predict. We achieve the MAP@12 score of 0.02290 in the contest, ranking top 41% within all the enrolled teams.



7 submissions for UPC_yanjian_wuyi_summer

Submission and Description	Private Score	Public Score	Use for Final Score
submission.csv 5 days ago by Yanjian add submission details	0.02290	0.02262	□

Figure 3: Kaggle Submission

5 Conclusions

In this project, we explore the data processing and model training in large-scale recommendation dataset with 31,788,324 transactions records, 1,371,980 customers and 105,542 articles.

To handle complicate feature, like descritions and image feature, we use pre-trained model to transform them into usable features.

We trained three most-used and State-of-the-art sequential recommention models for comparing their performance on H&M dataset.

References

[1] Donald E. Knuth (1986) *The TEX Book*, Addison-Wesley Professional. Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International Conference on Machine Learning. PMLR, 2021. 2, 3, 3.2.1

[2] Tan, Yong Kiam, Xinxing Xu, and Yong Liu. "Improved recurrent neural networks for session-based recommendations." Proceedings of the 1st workshop on deep learning for recommender systems. 2016. 4

[3] Tan, Qiaoyu, et al. "Sparse-interest network for sequential recommendation." Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021. 4

[4] Fan, Xinyan, et al. "Lighter and better: low-rank decomposed self-attention networks for next-item recommendation." Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.2021. 4

[5] Zhao, Wayne Xin, et al. "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms." Proceedings of the 30th ACM International Conference on Information Knowledge Management. 2021. https://github.com/RUCAIBox/RecSysDatasets 4.1