H&M Personalized Fashion Recommendations
Midterm Report

Data

The original data was provided by Kaggle, which consists of the transaction history from 2018/09/20 to 2020/09/21; the product with price and simple descriptions; images that are corresponding to each article_id; metadata for each customer_id and article_id.
The original data has more than one million transaction histories and user data. For more effective data analysis, the following two processes were performed.

1. Drop everything except the last few(up for experimentation) days. The info from previous months is not coming of much use. We only kept 4 weeks as train and the last week as validation.

```
data = pd.read_csv("./hmData/transactions_train.csv",
                   dtype={'article_id':str}
                  )
data.head()
data["t_dat"] = pd.to_datetime(data["t_dat"])
train1 = data.loc[(data["t_dat"] >= datetime.datetime(2020,9,8)) & (data['t_dat']
< datetime.datetime(2020,9,16))]
train2 = data.loc[(data["t_dat"] >= datetime.datetime(2020,9,1)) & (data['t_dat']
< datetime.datetime(2020,9,8))]
train3 = data.loc[(data["t_dat"] >= datetime.datetime(2020,8,23)) &
(data['t_dat'] < datetime.datetime(2020,9,1))]
train4 = data.loc[(data["t_dat"] >= datetime.datetime(2020,8,15)) &
(data['t_dat'] < datetime.datetime(2020,8,23))]
train5 = data.loc[(data["t_dat"] >= datetime.datetime(2020,8,7)) & (data['t_dat']
< datetime.datetime(2020,8,15))]

val = data.loc[data["t_dat"] >= datetime.datetime(2020,9,16)]
```

2. There are three types of users.
    - not_cold_users: Those who purchased within 3 months and have purchased a total of 5 or more so far.
    - cold_inactive_users: Those who have purchased less than 5 items and have not purchased in the last 3 months
    - cold_active_users: Those who have purchased less than 5 and have purchased in the last 3 months

Because the characteristics of the three types of users are completely different, a smaller dataset was created according to the user ratio.(data-process.ipynb)

```
import random

import pandas as pd

from typing import Tuple


def generate_dataset(

    rate # dataset size rate

) -> Tuple[pd.DataFrame, pd.DataFrame]:

    new_not_cold_users = random.sample(not_cold_users.to_list(),
round(len(not_cold_users) * rate))

    new_cold_inactive_users = random.sample(cold_inactive_users.to_list(),
round(len(cold_inactive_users) * rate))

    new_cold_active_users = random.sample(cold_active_users.to_list(),
round(len(cold_active_users) * rate))


    new_users = new_not_cold_users + new_cold_inactive_users +
new_cold_active_users


    new_train = train[train.customer_id.isin(new_users)]

    new_customer = customers[customers.customer_id.isin(new_users)]


    return new_train, new_customer

tiny_train, tiny_customer = generate_dataset(0.002)

tiny_train.to_csv('data-sample/transactions_train.csv', index=False)

tiny_customer.to_csv('data-sample/customers.csv', index=False)

articles.to_csv('data-sample/articles.csv', index=False)

sample_submission.to_csv('data-sample/sample_submission.csv', index=False)
```

Raw Data

a. images/

a folder of images corresponding to each article_id; 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

b. articles.csv

detailed metadata for each article_id available for purchase

Columns Name	Type	Meaning	Example
article_id	string	Unique ID of a commodity	108775015; 108775044; 108775051; 110065001; 110065002; 110065011;
product_code	string	ID of a group a commodity, each group may have various parameters, e.g. color and size	108775; 108775; 108775; 110065; 110065; 110065;
prod_name	string	Name of a group product	Strap top; OP T-shirt; 20 den 1p Stockings; Shape Up 30 den 1p Tights; Support 40 den 1p Tights; BANDEAU 2-p;
product_type_no	string	ID of a group product type	253; 306; 304; 302;

			273;
product_type_name	string	Name of group product type	Vest top; Bra; Underwear Tights; Socks; Leggings/Tights; Sweater; Bra; Top; Trousers;
product_group_name	string	Name of group product	Garment Lower body; Garment Upper body; Underwear; Socks & Tights;
graphical_appearance_no	string	Type of product's image	1010017; 1010016; 1010001; 1010010;
graphical_appearance_name	string	Name type of product's image	Solid; All over pattern; Stripe; Melange;
colour_group_code	string	Code of product's color	9; 8; 7;
colour_group_name	string	Name of product's color	Black; White; Off White; Light Beige; Beige; Grey; Light Blue; Dark Blue; Dark Grey;
perceived_colour	string	ID of product's perceived	3;

_value_id		color (nickname)	1; 4;
perceived_colour_value_name	string	Name of product's perceived color(nickname)	Dusty Light; Dark; Light; Medium Dusty;
perceived_colour_master_id	string	ID of product's master color(official color id)	5; 9; 11;
perceived_colour_master_name	string	Name of product's master name(official color name)	Black; White; Beige;
department_no	string	ID of product's department	1676; 1339; 3608;
department_name	string	Name of product's name	Jersey Basic; BasicClean; Lingerie; Clean Lingerie; Tights basic;
index_code	string	ID of product's index	A; B;
index_name	string	Name of product's index	Ladieswear; Lingeries/Tights;
index_group_no	string	ID of product's index group	1; 4;
index_group_name	string	Name of product's index group	Ladieswear; Baby/Children; Menswear; Divided;
section_no	string	ID of product section	16; 61;
section_name	string	Name of product section	Women's Everyday Basics; Womens Lingerie; Womens

			Nightwear, Socks & Tigh;d
garment_group_no	string	ID of product's garment group	1002; 1017;
garment_group_name	string	Name sof product's garment group	Jersey Basic; Under Nightwear; Socks and Tights;
detail_desc	string	The description of product	Jersey top with narrow shoulder straps.

c. customers.csv

metadata for each `customer_id` in dataset

Columns Name	Type	Meaning	Example
customer_id	string	Unique ID of a customer	00000dbacae5abe5e23885899a1fa44253a17956c6d1c3d25f88aa139fdfc657
FN	int	Unknown	1; null;
Active	int	Status of customer	1; null;
club_member_status	string	Current status of customer	ACTIVE; PRE-CREATE;
fashion_news_frequency	string	Purchase frequency of customer	NONE; Regularly;
age	int	Age of customer	25; 49;
postal_code	string	Unknown	52043ee2162cf5a

			a7ee7997428164 1c6f11a68d27642 9a91f8ca0d4b6ef a8100;

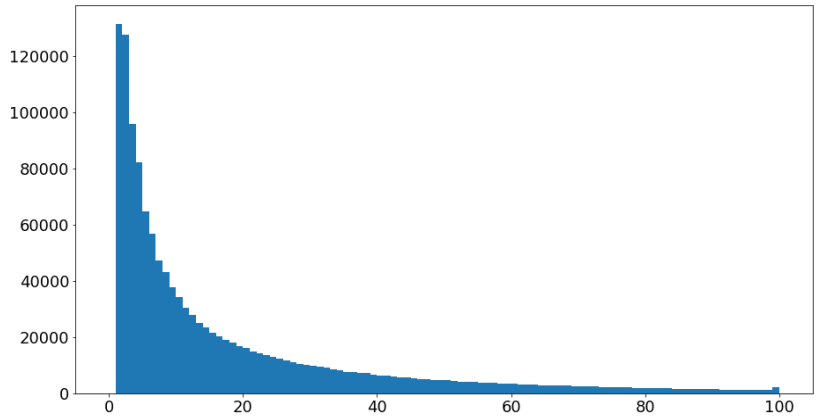
d. transactions_train.csv

the training data, consisting of the purchases each customer for each date, as well as additional information. Duplicate rows correspond to multiple purchases of the same item. Your task is to predict the article_ids each customer will purchase during the 7-day period immediately after the training data period.

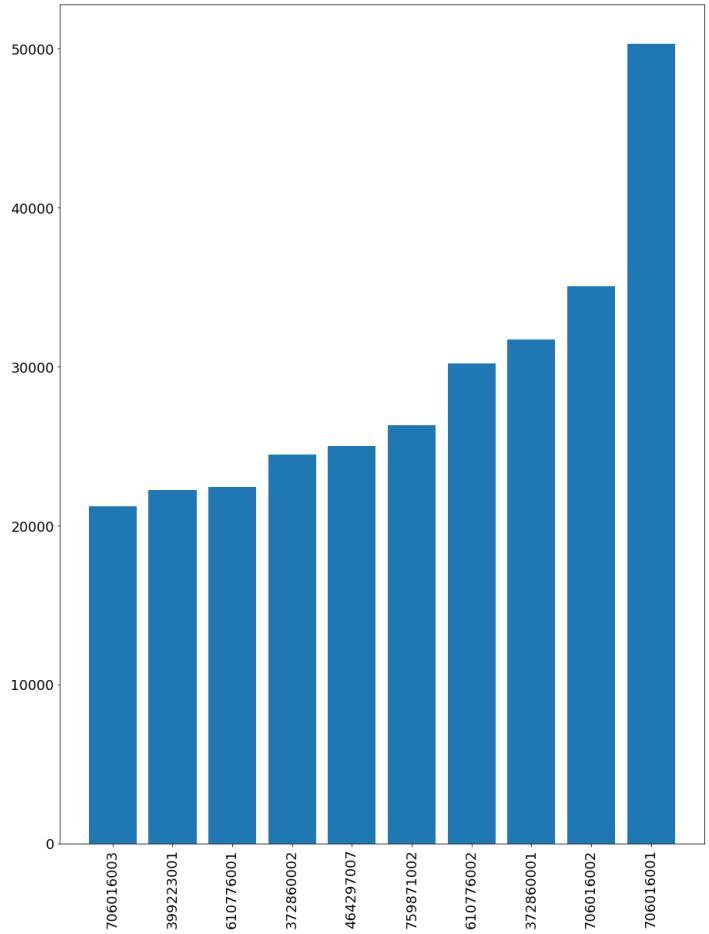
Columns Name	Type	Meaning	Example
t-dat	string	Date of transactions made	2018--09-20
customer_id	string	ID of customer	000058a12d5b43 e67d225668fa1f8 d618c13dc232df 0cad8ffe7ad4a10 91e318
article_id	string	ID of commodity	0663713001
price	float	Price of commodity	0.050830508474 576264
sales_channel_id	int	Number of the purchase	1; 2;

Exploratory data analysis

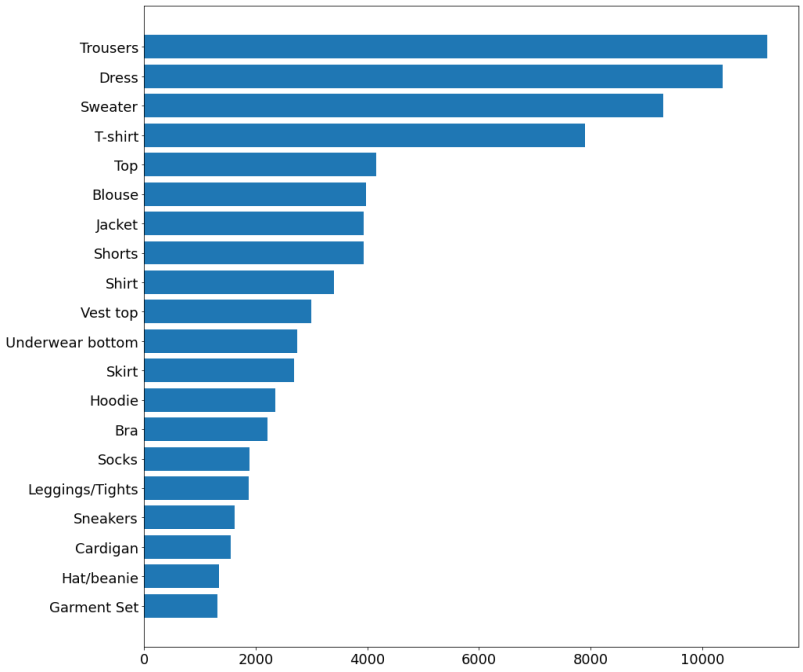
1. How many times customer has shopped in two years



2. Best selling items



3. Best selling category


Trousers
Dress
Sweater
T-shirt
Top
Blouse
Jacket
Shorts
Shirt
Vest top
Underwear bottom
Skirt
Hoodie
Bra
Socks
Leggings/Tights
Sneakers
Cardigan
Hat/beanie
Garment Set
0
2000
4000
6000
8000
10000

Methodology

Performance metrics

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

```python
def apk(actual, predicted, k=12):
    if len(predicted)>k:
        predicted = predicted[:k]

    score = 0.0
    num_hits = 0.0

    for i,p in enumerate(predicted):
        if p in actual and p not in predicted[:i]:
            num_hits += 1.0
            score += num_hits / (i+1.0)

    if not actual:
        return 0.0

    return score / min(len(actual), k)

def mapk(actual, predicted, k=12):
    return np.mean([apk(a,p,k) for a,p in zip(actual, predicted)])
```

Theme

The topic selected by our group is “H&M Personalized Fashion Recommendations”. For this topic, we are given the purchase history of customers across time, along with supporting metadata. What we have to do is establish a model to predict what articles each customer will purchase in the 7-day period right after training time.

Brainstorming & Discussion

At the beginning of the project, we first analyzed what we knew: the purchase history of customers. After brainstorming and group meetings, we realized that there are multiple data

metrics that can affect the accuracy of the forecast, such as the gender of the customer, the timeline of the data, the repeat purchase rate of an item and so on.

Baseline Models

Baseline models were developed based on popularity. (hnm-exponential-decay.ipynb)

1. Popularity exponential decay with alternate items
2. Trending products weekly(trending-product-weekly.ipynb)

Ensemble the baseline models for better performance

The baseline models were ensembled based on weight, individual score and appearance in each result.(ensemble.ipynb)

A final score of 0.0236 was achieved in comparison of 0.0217 and 0.0225 for each individual model.

Plans for the next phase

So far, we have used the basic data analysis models, and the results obtained are quite satisfactory. Our next goal is to introduce deep learning models and see if there will be any improvement.

Reference:

https://www.kaggle.com/code/lunapandachan/h-m-eda?scriptVersionId=92456185
https://www.kaggle.com/code/adldotori/all-in-one-sample-dataset/notebook
https://www.kaggle.com/lunapandachan/h-m-trending-products-weekly-add-test/notebook