H&M Personalized Fashion Recommender System

A machine learning model

Team MINK (Marque, Intae, Nikhil, Kisha)

Spring 2022

Berkeley
UNIVERSITY OF CALIFORNIA

Overview

Background : H&M is a Swedish multinational clothing company with a revenue of 24.8 billion USD. Company's focus is on fast-fashion clothing for men, women, teenagers, and children. As any other retailers, H&M desires to predict customer shopping pattern, to provide users' with a personalized experience

Build a Fashion Recommendation System

"Predict what articles each customer will purchase in the 7-day period immediately after the training data time, leveraging customer sales data from 2018-2020"

Motivation :

- With the rise of Youtube, Amazon, Netflix and many other such web services, recommender systems have taken more and more place in our lives
- The study from University of Pennsylvania's Wharton School and scheduled to appear in the journal Information Systems Research, finds that recommendations increase the sales of recommended products by **9 percent**

Data

Data source: https://www.kaggle.com/c/h-and-m-personalized-fashion-recommendations/data

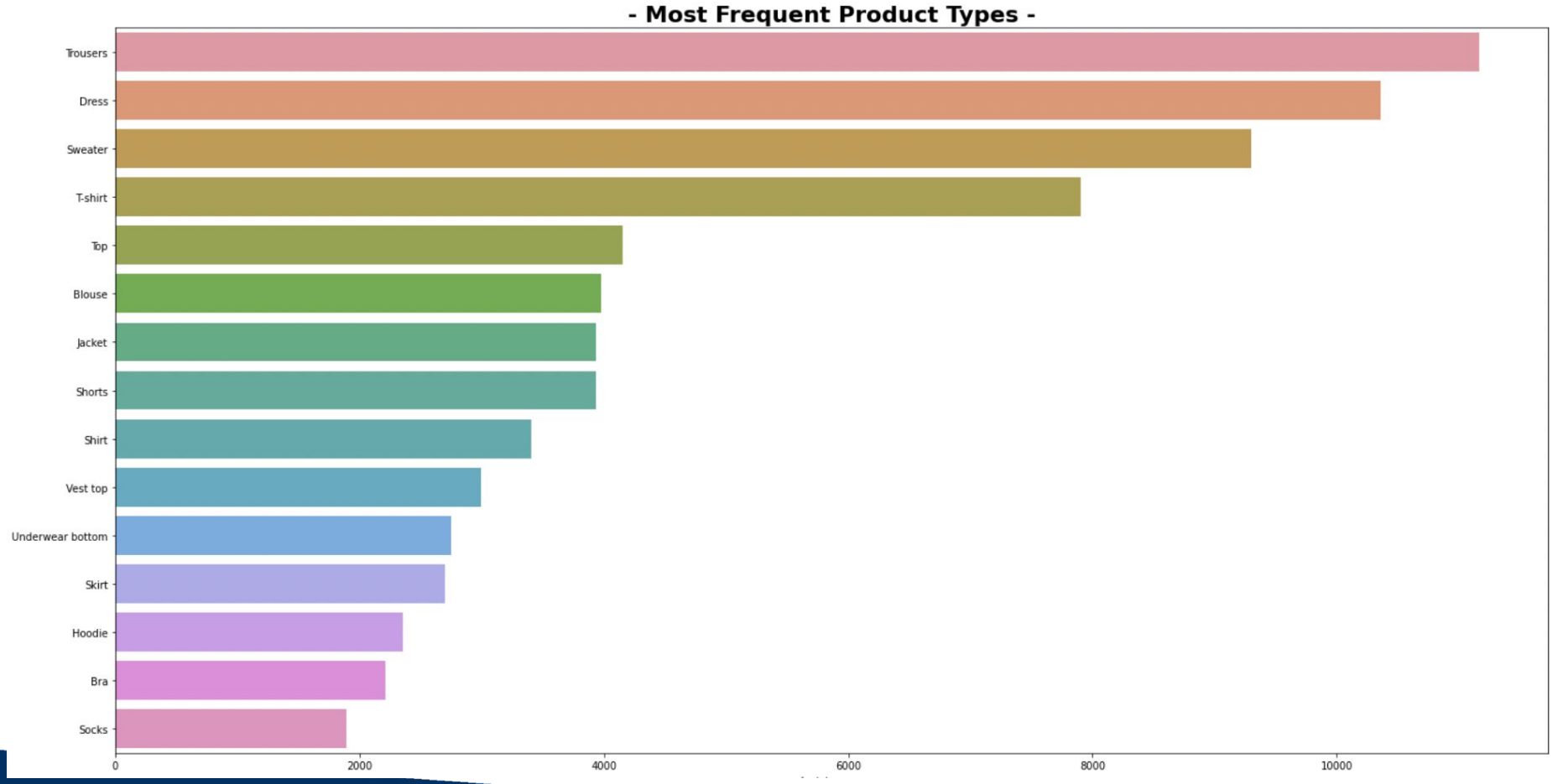
Data Set	Explanation
Articles	Detailed metadata for each article_id available for purchase
Customers	Metadata for each customer_id in the data set
Transactions_train	training data, consisting of the purchases each customer for each date, as well as additional information

Data: Articles

	article_id	product_code	prod_name	product_type_no	product_type_name	product_group_name	graphical_appearance_no	graphical_appearance_name	co
0	108775015	108775	Strap top	253	Vest top	Garment Upper body	1010016	Solid	
1	108775044	108775	Strap top	253	Vest top	Garment Upper body	1010016	Solid	
2	108775051	108775	Strap top (1)	253	Vest top	Garment Upper body	1010017	Stripe	

- Total number of unique article ids: 105,542
- Total Number of unique Product Types: 131 (e.g. Trousers, Blouse, Jacket)
- Total Number of unique Product Group: 19 (e.g. Garment Upper body, Accessories, Shoes, Swimwear, Bags, Furniture)

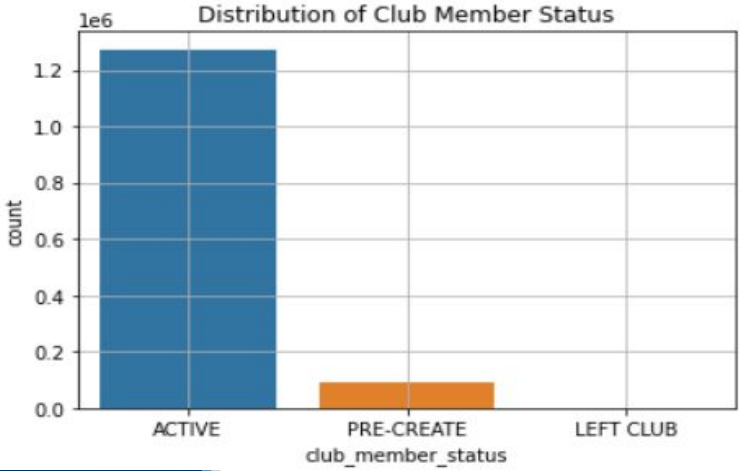
Data: Articles

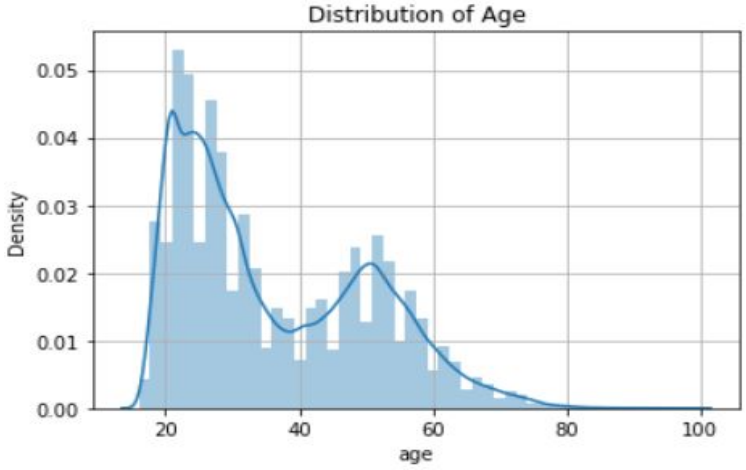


Data: Customer

This is metadata for each customer_id in the data set

- Total number of unique customers: 1,371,980
- 35% of customers receive the fashion newsletter
- 92% of customers are active club members, while 6.8% are in the pre-create stage. (Indicator if the customer is active to receive communications from H&M)
- 64% of customers don't receive the fashion news, while 35% receive it regularly.
- The mean customer age for this dataset is 36.4





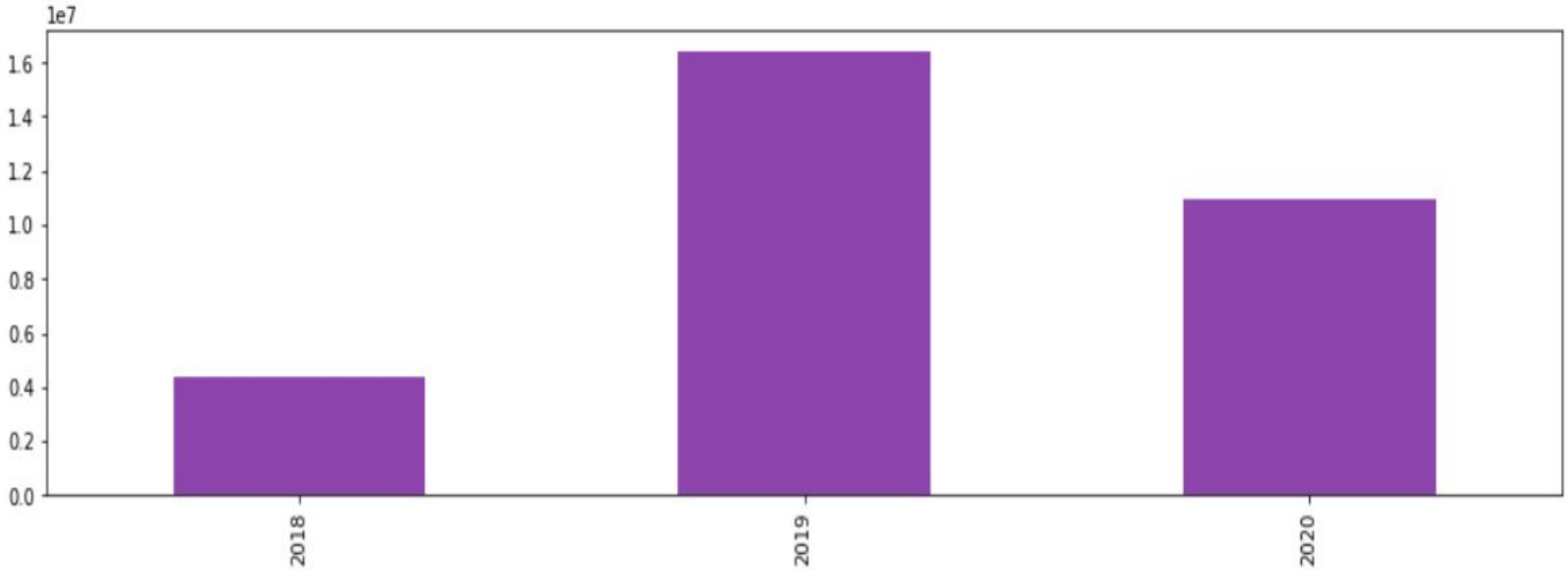
Data: Transactions

This is metadata for each customer_id and article_id transaction interaction in dataset

Total Interactions : 31,788,324

Transactions through years:

	2018	2019	2020
0	4411262	16396930	10980132



Data: Transactions

This is metadata for each customer_id and article_id transaction interaction in dataset

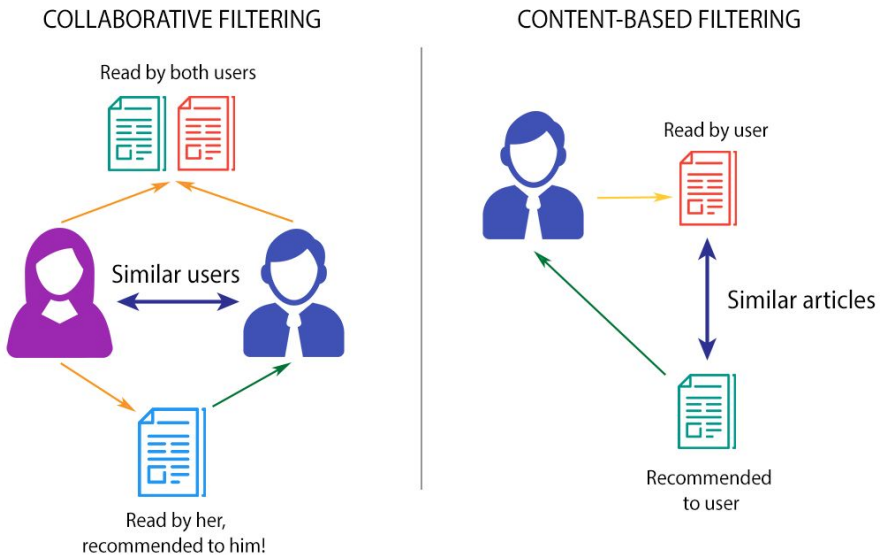
	t_dat	customer_id	article_id	price	sales_channel_id
0	2018-09-20	000058a12d5b43e67d225668fa1f8d618c13dc232df0ca...	663713001	0.050831	2
1	2018-09-20	000058a12d5b43e67d225668fa1f8d618c13dc232df0ca...	541518023	0.030492	2
2	2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699...	505221004	0.015237	2
3	2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699...	685687003	0.016932	2
4	2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699...	685687004	0.016932	2

Transformed Data:

	customer_id	article_id	cust_art_int
0	d00063b94dcb1342869d4994844a2742b5d62927f36843...	678342001	570
1	94665b46e194622ccdbcadc0170f13a2f8ede1ff6d057d...	629420001	199
2	61da44a2758206d5701771f4315637b40c8321b5111916...	507909001	188
3	ef38ec0f0cb29ee8bbb87efc82fd16f4b99127e3eeefe6...	570002001	170
4	5cba04ed9a3759bc02a8a9e01efccc07ce76c35c1a70dc...	688558002	166

Recommender System

The purpose of a recommender system is to suggest relevant items to users. To achieve this task, there exist two major categories of methods :



- Collaborative filtering methods : based solely on the past interactions recorded between users and items
- Content based methods : use additional information about users and/or items

Algorithm

Collaborative filtering methods : k-Nearest Neighbor

- Leveraged nearest customer to find recommended article

Hybrid: **LightFM,** is capable of hybrid method using collaborative and content-based

	Precision score	Recall Score	Comments
K-NN	0.00001424	0.0000211	Prediction accuracy for 12 nearest neighbors: 0.000124
LightFM	0.000045	5.149 e-05	

Model Evaluation

Evaluation methods for recommender systems can mainly be divided in two sets: evaluation based on well defined metrics and evaluation mainly based on human judgment and satisfaction estimation. The team will be using metric based evaluation.

Performance metrics:

- **Precision_at_k :** the fraction of known positives in the first k positions of the ranked list of results. A perfect score is 1.0
  - H&M recommendation system used precision at 12. The score is out of 12 recommendations made, how many did the customer purchase recommended item

- **Recall_at_k :** the number of positive items in the first k positions of the ranked list of results divided by the number of positive items in the test period. A perfect score is 1.0.

- **AUC Score :** the probability that a randomly chosen positive example has a higher score than a randomly chosen negative example. A perfect score is 1.0.
  - If there are no interactions for a given user the returned AUC will be 0.5.
  - excellent for AUC values between 0.9-1, good for AUC values between **0.8-0.9**, fair for AUC values between 0.7-0.8, poor for AUC values between 0.6-0.7 and failed for AUC values between 0.5-0.6

Baseline Model

LightFM with below variables

```
model = LightFM(loss='warp',
                random_state=0,
                learning_rate=0.90,
                no_components=150,
                user_alpha=0.000005)
```

```
model = model.fit(train_csr,
                  epochs=100,
                  num_threads=16, verbose=False)
```

	Loss Function	Learning Method	Epoch	Sample train size	Precision	Recall score	AUC score	Interpretation
Baseline - LightFM at k=12	'warp'	adagrad	100	10k	0.000045	5.149052459884249e-05	0.501689	Per the evaluation criteria, the model is barely hitting the recommendations.

Assumptions: baseline and tuning was done with 10k customer's transaction data due to computing power limitations, but the model will be retrained with 100k customer.

Hypertuning

Hyperparameters

- **Loss function**
  - **logistic**: useful when both positive and negative interactions are present.
  - **BPR**: Bayesian Personalised Ranking 1 pairwise loss. Maximises the prediction difference between a positive example and a randomly chosen negative example. Useful when only positive interactions are present and optimising ROC AUC is desired.
  - **WARP**: Weighted Approximate-Rank Pairwise 2 loss. Maximises the rank of positive examples by repeatedly sampling negative examples until rank violating one is found. Useful when only positive interactions are present and optimising the top of the recommendation list (precision@k) is desired.
  - **k-OS WARP**: k-th order statistic loss 3. A modification of WARP that uses the k-th positive example for any given user as a basis for pairwise updates.

Interpretation : Logistic loss function outperformed.

1. **Control variable : Loss Function**

	Loss Method	Precision at 12 score	AUC Score
0	logistic	0.003976	0.695463
1	bpr	0.002537	0.686377
2	warp	0.000079	0.503086
3	warp-kos	0.000045	0.500517

Hypertuning

Hyperparameters

- **Learning schedule** : Stochastic gradient descent
  - **Adagrad vs Adadelta**:
    - **Adagrad**: Adaptive Gradient Algorithm
      - Decay the learning rate for parameters in proportion to their update history (more updates means more decay).

	Loss Method with adagrad learning schedule	Precision at 12 score	AUC Score
0	logistic	0.003964	0.695383
1	bpr	0.002311	0.685909
2	warp	0.000079	0.500998
3	warp-kos	0.000079	0.503825

    - **Adadelta**: Adadelta is a more robust extension of Adagrad that adapts learning rates based on a moving window of gradient updates, instead of accumulating all past gradients.

	Loss Method with adadelta learning schedule	Precision at 12 score	AUC Score
0	logistic	0.003715	0.708839
1	bpr	0.004146	0.631282
2	warp	0.001529	0.62222
3	warp-kos	0.001518	0.598134

Hypertuning

Hyperparameters

- **Rho value with 'adadelta' learning schedule**: moving average coefficient for the adadelta learning schedule.
  - Grid Search - np.linspace(0.1,0.99,10)

	Rho adadelta learning schedule	Precision at 12 score	AUC Score
0	0.100000	0.003976	0.690764
1	0.198889	0.003874	0.690580
2	0.297778	0.003885	0.692039
3	0.396667	0.003727	0.694126
4	0.495556	0.003885	0.696941
5	0.594444	0.003930	0.699571
6	0.693333	0.003851	0.702235
7	0.792222	0.003749	0.704554
8	0.891111	0.003761	0.707426
9	0.990000	0.003817	0.707956

Rho = 0.99 seems to be the best possible value

Hypertuning

Hyperparameters

- **Epoch:** Epoch is a term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed.
  - Grid Search - [1,10,25,50,75,100]

	epoch adadelta learning schedule	Precision at 12 score	AUC Score
0	1.0	0.004384	0.710546
1	10.0	0.003523	0.711514
2	25.0	0.003727	0.708586
3	50.0	0.003727	0.708024
4	75.0	0.004010	0.708749
5	100.0	0.003874	0.707830

Epoch = 10 seems to be the best possible value

Hypertuning

Hyperparameters

- **L2 Regularization:**
  - Applied for user_alpha and item_alpha
  - adds an L2 penalty which is equal to the square of the magnitude of coefficients.
  - Grid Search - [0.05, 0.005, 0.0005, 0.00005, 0.000005]

	alpha adadelta learning schedule	Precision at 12 score	AUC Score
0	0.050000	0.001371	0.108889
1	0.005000	0.000159	0.370070
2	0.000500	0.004134	0.683314
3	0.000050	0.004134	0.690546
4	0.000005	0.003647	0.711197

L2 = 0.000005 seems to be the best possible value

Collaborative Filtering Model Finalization

LightFM with below variables

```
final_model = LightFM(loss='logistic',
                      learning_schedule='adadelta',
                      random_state=0,
                      rho=0.99,
                      no_components=150,
                      user_alpha=0.000005,
                      item_alpha=0.000005)

final_model = final_model.fit(train_csr,
                        epochs=10,
                        num_threads=16, verbose=False)
```

	Loss Function	Learning Method	Epoch	Sample train size	Precision	Recall score	AUC score	Interpretation
Final-LightFM at k=12	logistic	adadelta	10	100k	0.003557	0.011028	0.711213	Significant improvement

Hybrid Model Finalization

LightFM with below variables

```
final_model = LightFM(loss='logistic',
                      learning_schedule='adadelta',
                      random_state=0,
                      rho=0.99,
                      no_components=150,
                      user_alpha=0.000005,
                      item_alpha=0.000005)

final_model = final_model.fit(train_csr,
                        epochs=10,
                        num_threads=16, verbose=False)
```

	Loss Function	Learning Method	Epoch	Sample train size	Precision	Recall score	AUC score	Interpretation
Final-LightFM at k=12	logistic	adadelta	10	10k	0.001764			Shows potential

Conclusions & Impact

- Hyperparameter tuning and inclusion of more training data drastically improved results
- Optimization of code is crucial in improving time complexity of model training
- LightFM hybrid model shows potential, but likely requires more advanced techniques than we were able to explore

Contributions of Team Members

	Intae	Nikhil	Marque	Kisha
Preliminary Research	✔	✔	✔	✔
Exploratory Data Analysis	✔	✔	✔	✔
Data Transformation(Cleansing/Splitting/Processing)	✔	✔	✔	
Model Evaluation	✔	✔	✔	✔
Hypertuning Parameters		✔		✔
Presentation Slides	✔	✔	✔	✔

Thank you!

Appendix

Assumptions:

- Customer who did not make any purchase during that time are excluded from the scoring

Appendix: Data: Articles

This is a detailed metadata for each `article_id` available for purchase

- Initial dataset shape: (105542, 25)
- Main features to be used:
  - article_id: unique identifier for each product
  - prod_name: description of product
  - product_type_name: more general description of product
  - product_group_name: most general description of product group
    - ex. Garment upper body, accessories, etc.
  - graphical_appearance_name: description of product appearance
    - ex. Solid, striped, transparent, etc.
  - colour_group_name: description of product color
  - perceived_colour_value_name: description of product brightness
  - perceived_colour_master_name: more general description of product color
  - department_name: relevant H&M department of product
  - index_name: more general description of relevant department
  - index_group_name: most general description of relevant department
  - section_name: relevant H&M section of product
  - garment_group_name: type of product group
    - ex. Shirts, trousers, special offers, etc.

```
features_short = ['article_id',
            'prod_name','product_type_name','product_group_name','graphical_appearance_name', ## product groups info
            'colour_group_name','perceived_colour_value_name','perceived_colour_master_name',## color groups
            'department_name', ##departments
            'index_name','index_group_name','section_name', ##sections
            'garment_group_name' ##garment groups
           ]
```

Appendix: Data: Customer

This is metadata for each customer_id in the data set

- Initial dataset shape(1371980, 7)
- Main features to be used:
  - FN: Indicator if the customer is signed up to receive a fashion newsletter
  - Active: Indicator if the customer is active to receive communications from H&M
  - Club Member Status
  - Fashion news frequency: How often you are receiving fashion newsletter
    - Ex. regularly/monthly
  - Age: Age of the customer

Appendix: Data: Transactions

This is metadata for each customer_id and article_id transaction interaction in dataset

- Initial dataset shape (31788324, 5)
- Main features to be used:
  - Customer_id : Unique ID of each customer
  - Article_id : Unique ID of each article
  - T_dat : Transaction Date (Quantile cut transformation)
  - Article_count : New feature extracted based on number of articles bought within T_dat quantile cut period