H&M Personalized Fashion Recommendation System (Kaggle Challenge)

Mahi Sai Srinivas Bobbili, Harsha Teja Gorijavolu
DS 5230: Unsupervised Machine Learning Coursework Project – Fall 22
Khoury College of Computer Sciences
Northeastern University

Abstract

Over the past few years, the world of machine learning has introduced new perspectives in each of the domains. Recommendations have taken a leap in enhancements since the time they were introduced. Developments have begun to stagnate, which is why most leading companies have started seeking assistance and exploring ways to improve their existing algorithms in the form of challenges, allowing enthusiasts to explore real-world data while the companies benefit from optimizations. Our objective here is to explore the H&M fashion dataset and build a personalized recommendation system using only customer, transactions, and article data which improves user experience. The given dataset included customer data, product data, transaction data, images, and metadata of the products. The challenge becomes interesting with data open to CV with images and NLP with metadata. We employed data exploration techniques and market basket analysis to understand customers and further implemented collaborative, content-based, and popularity-based filtering on the data. Our major emphasis would be on data processing as there is no satisfactory measure. For ease of computation with large datasets, we have used the library Turicreate. To evaluate the model, we are using a MAP@12 value of 0.895 on the subset data. For future works, we intend to explore the supervised methodologies, Image-based recommendations.

Introduction

By participating in this Kaggle challenge, we plan to explore new and innovative ways to improve recommendation systems. Since we have H&M publicly available, aim to identify a recommendation system that works well with that data. While we are exploring the possibilities of new recommender machines, we did like to uncover the trends in the dataset.

Since the competition was open for quite a long time, there were teams from all over the world, attempting more than one technique, combining better optimization techniques, and checking feasibility of the algorithms for this dataset.

What motivates us here by taking up this challenge is that, we are competing with an ample load of tools and techniques which have been used to complete this challenge, while most others have focused on many techniques involving the dataset and the data modelling aspect of this project, we here are experimenting the usage of new libraries, technologies and methods to complete this challenge, our idea is that if our thoughts/ process can improve the algorithm by 10% we did like to couple it with some of the other techniques and amplify the results. Our initial thoughts to working with this project is to only consider a small portion of the datasets, because of our limited computational power. Beyond the scope of computation, what we did like to discover are the trends and approaches that could make a difference with the recommendations. In this project, we are trying to find new ways to make the computation easier to enable processing of larger datasets, obtain better results by meaningful data preprocessing, uncovering trends within the data. These are some of the layers we planned to incorporate in our project. Even a small fraction like 10% can be more than a regular 10% fraction because we are considering 10% customer data to further consider their complete transactions which were nearly 300,000 transactions and the articles purchased by this lot were nearly 70,000 products.

Our step-by-step approach to achieving this goal would be as follows:

1. Subsetting the customers at random that could give us decent proportions in the transactions and articles.
2. Necessary EDA and market basket analysis to discover the hidden trends and possibilities of understanding them for a deeper understanding of a customer's preferences.
3. Generating 3 datasets to experiment the methods and ways which are further discussed in background.
4. Experimenting on popularity-based recommendations, content based recommendations, collaborative filtering techniques and different similarity measures.

    5. Identify the best performing model using RMSE, the dataset, some hidden trends, understand which models work and why.
6. Since there is no valid out of the box metric, we developed a custom industry standard metric based on the requirement to accurately deliver precision over recommendations based on the customers purchase history. This custom metric is MAP@K, Mean average precision @K where k=12, since we are recommending 12 products to the customer.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i$$

Mean Average Precision Formula

Figure 1: Mean Average Precision Formula.

We intend to follow this as our step-by-step guide with some tweaks along the way. However, the goal here is to build a recommender machine that is capable of recommending products that share close resemblance with the customers past transactions.

Background

Since the total dataset consists of over 1.3M transactions, we are only considering 10% of it and due to some technical difficulties and experimenting, we had to install a ton of libraries which seemed to be useful and while we were experimenting with libraries that could process large amounts of data while having support to recommendation libraries, we found only a handful of libraries that could do both, one such library was Turicreate, however, turicreate is not supported implicitly, for this very purpose, we have worked on Google colab from the data preprocessing part of the project. The reason being that google colab supported the use of turicreate module implicitly and which did not cause any technical failures with respect to other libraries.

Our model is trained under the assumption that the real-world dataset that the model is given transactions details, articles information and the customer information. As mentioned before the dataset did not have ratings information and we are only considering the transactions part of the data. For this reason, we have taken 3 datasets, as we are experimenting with the data using purchase count in place of purchase count.

The dataset we are considering for this experiment will have the customer id, article id, and purchase counts of that article with respect to that user. In the second dataset, we are considering a dummy variable purchase scale with a value of 1, this acts as a bias parameter that all users tend to buy all the products in addition to the purchase frequency. The third dataset is similar to the first one, the major difference is when the purchase counts are scaled using a normalizer. The normalized dataset is computed using the formula.

Scaled target column = current_value – min / (max-min)

The reason we are using 3 datasets is because of we are planning to experiment with the way the recommendations work, this is not a vanilla approach where we use ratings to recommend the products. Since this is a Kaggle challenge, we wish to see how new ways and methods can be used to implemented to recommend products when there is no satisfactory measure especially.

We wish to explore the existing algorithms and seek new methods and ways in which we can improvise the recommendations, the thought behind our dataset approach is that we wish to see if we can obtain better results from a normalized dataset, or a dataset with bias (inspired from PageRank algorithm), or a general dataset with no scaling or bias. If there exists an algorithm that works best with the dataset.

An important mention about the type of data we are working with, since most of the customer information, product information is encrypted, it does not make sense implicitly for someone who is not actually working with the dataset, but the most effective way to get the most out of what we can do would be to understand the EDA process of the project, because the initial data driven analysis by itself does not contribute much towards the recommendations, however, will give us a good understanding of the data, what kind of data it is etc.

In addition to these task oriented challenges, we also have to deal with the regular problems like the imbalanced data, it is always possible that the 10% fraction of the data from the parent data, can have some users with most transactions and some users with minority of the transactions.

To address the challenges of turicreate, we are partially changing our environment from local to cloud based google colab environment, we have saved our subset dataset and then exported it to google colab to make sure there is synchronicity between both the versions.

Related Work

There is a lot of scope and ongoing research that is going on in the field of recommender systems. On a higher level, all the recommendation system techniques ideally fall into one of the following categories. 1. Content based filtering, 2. Collaborative filtering and 3. Popularity based recommendation.

Content based filtering mainly aims at recommending products based on the customer’s transaction history or their profile and is the best way to recommend personalized products. Only drawback with content-based filtering is that it is not helpful for new users and for it work efficiently it expects that there is some sort of profile information to recommend this problem is referred to as cold start problem. Collaborative filtering is based on the idea that people who like stuff or have similar purchase history and satisfaction with them will lead to a trend in purchases. These similarities can be calculated either by item-item or user-user. Popularity based recommendation systems are always incorporated in multiple e-commerce websites or applications to provide the recommendations based on the total number of purchases over a certain period.

In addition to these 3 parent libraries, there are plenty of libraries that support different variants of recommendation systems. While there are numerous libraries that support the functioning of recommendation systems and their implementations, there are not more than a handful of libraries that propagate efficiency into the process of recommendation systems. One such library that can also assist with the working of larger datasets is Turicreate, this library has options that can support the functionality of recommendation systems while also working on optimizing the computation for larger datasets.

The most used measures required to compute collaborative filtering are item-item similarities and user-user similarities are Cosine similarity measure, Pearson correlation measure and spatial distance measure. Cosine similarity is based on the angle between two product vectors while the Pearson correlation is computed by overall product similarity with respect to the mean averages and spatial distance measures are computed by calculating pair wise distances between all the points to form a distance matrix grid and thereby may result in a bit of over computation when compared to Pearson or cosine similarity. Where we compete against the world for this challenge is the implementation of optimized datasets and our technique to see if this would enhance the optimization and in general provide much better recommendations and the validation over MAP@K, earlier known challengers have not used MAP@K or in fact any other dedicated metric to evaluate their model.

From paper 1 (Hyunwoo.H et al. 2018), our thoughts were driven that we would have the best results for either item based Collaborative Filtering or Content Based Filtering, which is when we then proceeded to explore more research papers and planned on exploring the underlying unsupervised aspect for this challenge. Our reference paper 2(Maria.A et al.2019) inspired us to consider popularity-based recommendations, which propelled us to work on trends based on popularity and their recommendations.

Project Description

We have a dataset consisting of 5 files, customers data, transactions data, articles data, images of over 10000 products, and metadata information. Due to computational overhead time constraints, we have taken only 10% customers from the customers data. And we have considered only the transactions made by these customers, and articles involved in these transactions. We have obtained around 288K transactions and 68K products and 28k customers. This is the subset of the data which we will be working on, and this dataset is hereby referred to as the subset data. We save this subset and use this as dataset whenever we change the working environment due to the technical difficulties and computational limitations.

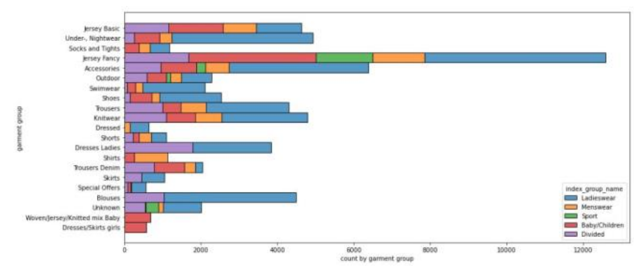

Figure 2: Distribution of subcategories to categories with respect to the total counts.

From the subsets, we visualized the distribution of products over its subcategories and across the dataset. We were curious about what age group of customers made most purchases and decided to visualize the customers data. We noticed that most of the customers purchased were from the age range 18 to 32. We can deduce that the best age for the online purchases at H&M is 18 to 32 and beyond this point, it is again between the ages of 45 to 55. These are the two primes for the online H&M online purchases. This will form a basis for the future work for us. At this phase of the project this distribution is not much relevant to us.

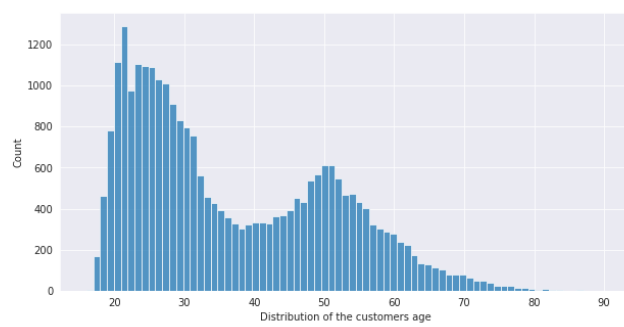

Figure 3: Distribution of customers from transactions.

We perform analysis on the data to understand how distributed the products are and the categories they belong to. The goal of this analysis was to understand the given subset on a deeper level. Our approach is backed by the thought that the better we understand our target audience, the better we can recommend products to them. With this thought in mind, we have visualized the distribution of customers by age, outliers by price, five most expensive items, five least expensive items in the subset. Doing this sort of analysis will allow us to understand and communicate with the data much more easily.

After our initial EDA, we found out that most of the images involved in our subset are not part of provided dataset, and beyond this point, we decided not to operate with images as the images involved in our subset are too low and cannot fully meet scope of this project, hence we only deal with the customers, transactions, and their articles data.

Figure 4: 5 Most expensive items (Top 5) & 5 Least expensive items (Bottom 5).

We further performed market basket analysis to uncover the frequent patterns in transactions. This as an independent phase of the project would still make sense but considering the nature of the dataset most of the customers information and the transaction information. Hence, we have only used this to better understand the customers in general and not as a deep dive into understanding the patterns. This phase is successful if the custom-built function can identify what are the frequent purchases made by a customer.

Once we were done with the EDA, we worked on generating the 3 datasets discussed in the background and the reasons behind why we needed 3 datasets. One major challenge we incurred at this point was that the installation of the turireate module, the module is not supported by most of the local jupyter notebook environments due to the way it operates, and the installation of this module in the google colab notebook which was our secondary environment.

We implemented popularity based recommendations to obtain the most bought products over the pool of transactions. This was our baseline model; we then implemented the content based filtering and collaborative filtering. All the three algorithms were run against the 3 datasets and the 3 models were run against the test dataset to find the best recommendations. We used RMSE metric to evaluate our models.

However, we plan to evaluate our best model using MAP@K at K=12, as we are recommending 12 products, in addition to that, we are taking up the 8 most bought products from the market basket analysis we did for that user, use those products to compare against our predicts from the best model identified. This way we were able to obtain a MAP@12 value of 0.895, which gave us good predictions.

Emprirical Results

Our initial market basket analysis provided frequent purchases made by every user in our customers subset. However, with the encryption in the dataset, most of it was in numbers, we can derive meaning from it upon converting the predictions from their id to the general naming conventions. Some users might have a single frequent purchase, while there could be users with more than 2 frequently bought items, we are considering the products that have appeared at least 50% to all the users, else that product is more towards the personalized purchase category. The results of the market basket analysis were as follows.

	customer_id	preds
0	0006d3ff0caf0cb4d4e0615ee5cb7d268622364d483335...	624486001 658298001
1	000ee39f35322b568db9d7f39500ad1b58a6f8a3fc2abf...	768912001
2	0011657961496b514982fce1c6150ff32816e26a866027...	562245050

Figure 5: Market Basket Analysis on the customers subset w.r.t transactions subset.

We can notice that the predictions in Figure 5 depicts that the user 0 has 2 most frequently bought items while the users 2 and 3 has only single most frequently bought item.

The reason for the three datasets, and the information has been explained clearly in the background. We have taken a total of 3 datasets of the following schema. We believe that one of these schemas of the dataset would work the best for the recommendation of the models we have selected. Since we are not having the information about any satisfactory measure, we are using these following purchase count, purchase_dummy acting as a bias, scaled purchase frequency to obtain the best results.

	customer_id	article_id	purchase_count
0	0006d3ff0caf0cb4d4e0615ee5cb7d268622364d483335...	624486001	1
1	0006d3ff0caf0cb4d4e0615ee5cb7d268622364d483335...	658298001	1

	customer_id	article_id	purchase_count	purchase_dummy
0	0006d3ff0caf0cb4d4e0615ee5cb7d268622364d483335...	624486001	1	1
1	0006d3ff0caf0cb4d4e0615ee5cb7d268622364d483335...	658298001	1	1

	customer_id	article_id	scaled_purchase_freq
0	014624037815b4e8da7d1b1254ed2cc97d47b452a921bd...	108775015	0.0000
1	0162c0eacae80811a6b926cb3432888bc3f5fffb963b66...	108775015	0.0000

Fig 6: Schema of the 3 generated datasets 1. Regular dataset (Top), 2. Biased dataset (Middle) and 3. Scaled dataset (Bottom).

We obtained the following popularity based recommendations for our dataset based on the transactions subset.

	customer_id	article_id	purchase_count	rank
8439	3d6374418244159f2519637404f3220345a3a29a90130a...	156231001	15	1.0000
12636	5cb90633fb3bd8b9ebf2a19abfda4aed39ed9cb07f2f9a...	156231001	13	2.0000
31674	eb3bc08d3045a7fd0c140e3c0e5ff01fb768649681649b...	179123001	13	3.0000
15314	71774c5396e1eeff814425a531149c8e899885cf2c4b53...	200182001	8	4.0000
21450	9c8d3d5381630ec8e017676a81ff21dfa37e8d36e5a215...	228257001	8	5.0000
22702	a65e8e3aad74a6af2df8854cad5039cfd2d76183f6d236...	572797002	8	6.0000
18024	85e4c12212722b1c5a552bd3a976f37243acf521606ec7...	685814001	8	7.0000
15869	754e9a2dcdf1a128242eb9792abccad5f0221623077c20...	111586001	7	8.0000
20304	94f7375f6c59d30b255b44fe85de1a827e81840f649bef...	156231001	7	9.0000
30352	e084a4e6236281a396fc413ae767fc039af09b58a56276...	179123001	7	10.0000
21546	9d87a05cdf6979f0b111002927780686d458996769e36c...	228257001	7	11.0000
12827	5e1e16427836c47ed39b603f7863d40eecb70303d89290...	228257001	7	12.0000

Figure 7: Illustrated popularity based recommendations results, which we would be using as a baseline model.

Purpose of the popularity based model is that the results obtained here with this recommendation can be used to recommend products to users who does not have a transaction history yet or to address the cold start problem, along with content based and collaborative based filtering, in addition these recommendations can be used to recommend products to the user during specific times to showcase more of the trending recommendations to make it more personalized, we can incorporate this to the personalized aspect by taking in products from the market basket analysis. This however is future work.

To identify which of the models performed well, we have calculated RMSE score to test our models. Out of all the 9 combinations, the collaborative based item-item filtering with cosine similarity on the original data gave us the best result of ~ 0.42 RMSE overall. This was tested against the test dataset and custom inputs of users. The results of this model can be observed in Figure 8. It contains the RMSE for the recommendation of 12 products for a target customer and their support count. The incorporation of this model with turicreate has clearly shown the results in a better format and also computation speed increased which otherwise would have been a lot more difficult than usual. All these combinations of models were implemented using turicreate library.

The results of this project are stored in a text format saved in the 3 files, eval_count, eval_dummy, eval_normal to determine the results of the 3 algorithms with the combinations. We have decided that the content based filtering, popularity based filtering worked fine independently, for the collaborative filtering we decided to use similarity between item to item along with their respective purchase counts.

```
Data:
+------------+---------------------+-------+
| article_id |         rmse        | count |
+------------+---------------------+-------+
| 689109001  | 0.3509555303444006З |   33  |
| 554598001  | 0.2846071104797731б |   34  |
| 611415001  | 0.21880452125121927 |   41  |
| 658298001  | 0.12225333296898121 |   27  |
| 730683001  | 0.21408289782187323 |   54  |
| 372860001  |  0.4345708065918535 |  174  |
| 677930023  | 0.21396936650079434 |   42  |
| 610776028  |  0.2904414786993288 |   43  |
| 688537011  | 0.21850047215124904 |   64  |
| 684209013  | 0.22364862523752724 |   63  |
+------------+---------------------+-------+
[200 rows x 3 columns]
Note: Only the head of the SFrame is printed.
You can use print_rows(num_rows=m, num_columns=n)
to print more rows and columns., 'rmse_overall':
0.4293266283512445}]
```

Figure 8: Results of the Collaborative Item-Item filtering, with cosine similarity, RMSE values recorded at the bottom of the image, image demonstrates the recommendations for a target customer along with individual RMSE value computed against his past transactions and support for each recommendation alongside the recommendation.

Conclusion & Future Work

We conclude that we have developed 3 different recommendation models content-based recommendation system, collaborative filtering based on purchase counts, popularity based recommendation system. We faced a few challenges while developing a machine learning model for H&M. Although content based filtering was the star of the 3 recommendation systems in our case given that the data is without a satisfactory measure. Performed market basket analysis allowed us to understand the trends of the customer purchases and transactions.

Our first challenge when developing this model was the lack of satisfactory measure and generating a new feature purchase count and purchase frequency. The basis of our recommendation model is that we replaced the regular rating to a completely different feature.

The secondary challenge is the size of the dataset itself, even the subset of the data was completely larger than what we have anticipated which led to more research and the solution we ended up was Turicreate library. Another challenge here is the usage of this library, this is not supported by Anaconda software, so we had to separately run this on google colab and Kaggle notebooks to utilize their environment.

The popularity based models were predicting the top 12 products that a customer could order based on the purchase counts alone. This works well to address the cold start problem and to recommend products to customers with less than 12 recommendations and was our baseline model.

Content based filtering and collaborative filtering on the other hand was able to predict 12 recommendations for the target audience. The combination of cosine similarity on the generated dataset gave us the best Root Mean Squared Error (RMSE) value and we decided to reaffirm our evaluation by using Mean Average Precision@K (MAP@K) against our model which gave a score of 0.895 with decent recommendations. For the future students of DS 5230, I wouldn't recommend anyone to work with turicreate unless they have their base environment setup separately.

We would like to explore the possibilities of improvised usage of Turicreate to its fullest capabilities, the library has a lot more to offer than what we have implemented. We believe that the usage of this library and revisiting this problem statement with few tweaks to our approach which we might not have noticed at this point would help in bringing out the best result possible.

We also plan on developing a User Interface (UI) to make it more interactive to the target audience. We can explore the possibilities of using images, text metadata to further narrow down on the solution at hand and enhance the recommendation capabilities of the model.

In addition to the previously mentioned enhancements, we might investigate the possibilities of exploring the transactions on the given magnitude and utilize cloud services or some open-source GPU availability to train the model and our approach for feedback and to optimize our resolution.

References

Hwangbo, H., Kim, Y. S., & Cha, K. J. (2018). Recommendation system development for fashion retail e-commerce. Electronic Commerce Research and Applications, 28, 94-101.

M. A. Stefani, V. Stefanis and J. Garofalakis, "CFRS: A Trends-Driven Collaborative Fashion Recommendation System," 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), 2019, pp. 1-4, doi: 10.1109/IISA.2019.8900681.

K. Suekane et al., "Personalized Fashion Sequential Recommendation with Visual Feature Based on Conditional Hierarchical VAE," 2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR), 2022, pp. 362-365, doi: 10.1109/MIPR54900.2022.00071.

Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay: "Deep Learning based Recommender System: A Survey and New Perspectives", Arxiv, 2017

W. Li and B. Xu, "Aspect-Based Fashion Recommendation With Attention Mechanism," in IEEE Access, vol. 8, pp. 141814 - 141823, 2020, doi: 10.1109/ ACCESS. 2020. 3013639.

Y. Lu, R. Dong and B. Smyth, "Coevolutionary recommendation model: Mutual learning between ratings and reviews", *Proc. World Wide Web Conf.*, pp. 773-782, 2018.

Y. Ding, Y. Ma, W. K. Wong and T. -S. Chua, "Modeling Instant User Intent and Content-Level Transition for Sequential Fashion Recommendation," in IEEE Transactions on Multimedia, vol. 24, pp. 2687-2700, 2022, doi: 10.1109/TMM.2021.3088281.

C. Yan, Y. Chen and L. Zhou, "Differentiated Fashion Recommendation Using Knowledge Graph and Data Augmentation," in IEEE Access, vol. 7, pp. 102239-102248, 2019, doi: 10.1109/ACCESS.2019.2928848.

M. Mameli, M. Paolanti, R. Pietrini, G. Pazzaglia, E. Frontoni and P. Zingaretti, "Deep Learning Approaches for Fashion Knowledge Extraction From Social Media: A Review," in IEEE Access, vol. 10, pp. 1545-1576, 2022, doi: 10.1109/ACCESS.2021.3137893.

U. Sharma, G. P. Sajeev and S. S. Rani, "Personalized Fashion Recommendation Using Nearest Neighbor PageRank Algorithm," 2022 International Conference on Connected Systems & Intelligence (CSI), 2022, pp. 1-6, doi: 10.1109/CSI54720.2022.9924114.

Z. Yang, Z. Su, Y. Yang and G. Lin, "From Recommendation to Generation: A Novel Fashion Clothing Advising Framework," 2018 7th International Conference on Digital Home (ICDH), 2018, pp. 180-186, doi: 10.1109/ICDH.2018.00040.

H. Zhan, B. Shi, J. Chen, Q. Zheng, L. -Y. Duan and A. C. Kot, "Fashion Recommendation on Street Images," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 280-284, doi: 10.1109/ICIP.2019.8802939.

P. Su and H. Ye, "An Item Based Collaborative Filtering Recommendation Algorithm Using Rough Set Prediction," 2009 International Joint Conference on Artificial Intelligence, 2009, pp. 308-311, doi: 10.1109/ JCAI. 2009.155.

L. De Divitiis, F. Becattini, C. Baecchi and A. Del Bimbo, "Style-Based Outfit Recommendation," 2021 International Conference on Content-Based Multimedia Indexing (CBMI), 2021, pp. 1-4, doi: 10.1109/CBMI50038 .2021 .9461912.

H. Zhan and J. Lin, "PAN: Personalized Attention Network For Outfit Recommendation," 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 2663-2667, doi: 10.1109/ICIP42928.2021.9506344.

Z. Lu, Y. Hu, C. Yu, Y. Jiang, Y. Chen and B. Zeng, "Personalized Fashion Recommendation with Discrete Content-based Tensor Factorization," in IEEE Transactions on Multimedia, 2022, doi: 10.1109/TMM.2022.3186744.

H. Zhan, B. Shi, J. Chen, Q. Zheng, L. -Y. Duan and A. C. Kot, "Fashion Recommendation on Street Images," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 280-284, doi: 10.1109/ICIP.2019.8802939.