CIS 5220 – Final Project – Technical Report

Fashion Net

April 2023

Team Members:

- Xi Cao; caoxi; Email: `caoxi@seas.upenn.edu`
- Meiru Han; meiru; Email: `meiru@seas.upenn.edu`
- Xinrui Yu; yuxinrui; Email: `yuxinrui@sas.upenn.edu`

Abstract

Fashion shopping is getting increasingly more data-oriented these days. Recommendation systems allow customers to discover new products that match their unique and diversified tastes while making it easier for them to navigate through a crowded cyber marketplace. By using data-driven algorithms and machine learning techniques, fashion recommendation systems can provide personalized recommendations based on factors such as item information, past purchases, browsing history, and search queries. In this project, we explore three distinct methods for building a recommendation system for the *H&M* dataset: Content-based KNN, Multilayer Perceptron, and Graph Neural Network. By incorporating customer and transaction data into the models, we are able to generate personalized recommendations that are more accurate and diverse.

1 Introduction

Online shopping has grown tremendously over the last few years, recommending proper outfits to customers has been a popular topic in fashion recommendation systems. Recommendation system help users to navigate large collections of products to find items relevant to their interests leveraging large amount of product information and user signals like product views followed or ignored items, purchases, or web-page visits to determine how, when and what to recommend to their customers. Due to market dynamics and customer preferences, there is a large vocabulary of distinct fashion products, as well as high turnover. This leads to sparse purchase data, which challenges the usage of traditional recommendation systems.

In this project, we focus on *H&M* data set to recommend proper outfits for customers. The dataset contains customer information, item information and transaction information. How to properly incorporate these three data information to do recommendation remains to be an open and challenging problem. To evaluate our model, hit rate is defined in section (3) and calculated in each section for different models. In this study, we evaluated three different methods for building a recommendation system: Content-based K Nearest Neighbor, Neural Network, and Graph Neural Network. In section (5.1), we discussed the performance of the content-based K Nearest Neighbor method, while section (5.2) focused on the Neural Network approach. To incorporate customer, item, and transaction data, we utilized innovative representation techniques that yielded satisfying results. The most advanced deep learning method we evaluated was the Graph Neural Network (GNN), which we discussed in section (5.3). Both the machine learning and deep learning approaches produced promising results, but the GNN method was particularly effective in providing variable and up-to-date recommendations for fashion outfits. Our study highlights the importance of incorporating customer and transaction data into recommendation models to achieve more personalized and accurate results. Furthermore, our findings demonstrate the effectiveness of innovative representation techniques and the benefits of utilizing advanced deep learning methods, such as GNN, in fashion recommendation systems.

2 Related Work

In the online internet era, the idea of Recommendation technology was initially introduced in the mid-90s. Traditional recommendation systems include Collaborative Filtering and Content-Based Filtering. Collaborative Filtering is a widely-used recommendation approach that involves analyzing user ratings for fashion items in order to learn their preferences and predict missing ratings for other products. This is done by creating a rating matrix where each entry represents a user's rating for an item. The system then recommends items with the highest predicted ratings (i.e., missing entries in the matrix) to the user. On the other hand, Content-based recommendation techniques rely on analyzing the content of fashion items, such as their textual description or visual features, to generate recommendations based on their content features. While the above mentioned methods have difficulties in the fashion domain due to the sparsity of purchase data or the insufficient detail about the visual appearance of the product in category names [SWZ+16]. Instead, more recent literature has leveraged models that capture a rich representation of fashion items through product images [HM16], text descriptions or customer reviews which are often learned through surrogate tasks like classification or product retrieval. However, learning product representations from such input data requires large datasets to generalize well across different image (or text) styles, attribute variations, etc. There are also methods focus on users social circle to do fashion recommendation. According to different studies, e-commerce retailers, such as Amazon, eBay, and Shop- style, and social networking sites, such as Pinterest, Snapchat, Instagram, Facebook, are now regarded as the most popular media for fashion advice and recommendations . Research on textual content, such as posts and comments emotion and information diffusion has attracted the attention of modern researchers, as it can help to predict fashion trends and facilitate the development of effective recommendation systems [JPB+14].

Our project focuses on the development of hybrid recommendation systems that leverage information from multiple sources, including user data, item data, and transaction data. By combining these different sources of information, our methods aim to provide more accurate and personalized recommendations to users. One of the key advantages of our approach is that it is computationally light, which makes it more efficient for real-world applications. Additionally, our methods are shown to produce diverse results, which is important for ensuring that users are presented with a range of options to choose from.

3 Evaluation Metric: Customer Hit Rate

For KNN and advanced Deep learning techniques GNN, we chose to use a customer-based evaluation metric. For the KNN baseline and GNN, for each customer we make 50 recommendations based on all the articles they have bought except for one, which we will check the 50 recommendation against. We repeat the process and leave out each article from the list of all articles bought by the customer, if any of the left-out item is in the set of 50 recommended articles, the customer is considered "**hit**", otherwise '**miss**". The customer hit rate is calculated by **hit/(hit+miss)**.

For DL baseline, we first sample 100 customers who are not included in the training set. For each customer, we recommend 50 items and check whether the set of these 50 items have non-empty intersection with the set of items actually bought by the customer. If yes, this customer is considered a "**hit**", otherwise "**miss**". We then calculate the customer **hit rate: hit/(hit+miss)**.

4 Dataset and Features

4.1 description

Our datasets contain articles information, customer information and transaction information.

Articles information are briefly shown in the below table, number in the parentheses stands for the number of type in this category. We list three for each for the readers to get a sense about the information.

Customer information includes

- Club member status (3): Active, leave club, pre create,

Article information			
Product type(56)	Bag	Ballerinas	Belt
Product group (10)	Garment upper body	Socks Tights	Garment Lower body
color group(37)	Black	Blue	White
Department name(45)	Jersey Basic	Tights Basic	Knitwear
index name (3)	Ladieswear	Lingeries/Tights	Ladies Accessories
section name (16)	Women Everyday Basics	Women Nightwear Skicks and Tigh	Mama

- fashion news frequency(2): None and Regular,
- Customer age.

Transaction data include the price, date, of each customer for each product.

4.2 Pre-processing

To narrow down our recommendation range, we only focus on the 'ladieswear' articles. We clean the data by the following procedures to limit the amount of data we finally feed in the neural network.

- Sample 5000 customers from the most active customers (made more than 50 transactions)
- Choose the top 2000 articles bought the most frequently.
- Drop rows with null values (for three datasets).

At each step, we adjusted the transaction accordingly to keep only the transactions made by the filtered customers and on the filtered articles. The final cleaned dataset contains **4980** customers, **1988** ladieswear articles and a total of **146499** transactions.

5 Methodology

5.1 ML based: Content Based Methods

There are 1988 articles resulting from our data cleaning. For each customer c , we get a list of all articles $a_1, ...a_n$ bought buy this customer. For each article from the complete 1988 articles, we calculate the summation of Euclidean distances between this article to each $a_i, i = 1, ..., n$ to get an overall distance score between this article to all purchases of customer c. The distance score is based on article features and the information on which article is bought from the transaction records. No customer feature is used here. We then rank the distances and select the 50 articles (which have not already been bought by c) with lowest distances as our final recommendation.

5.2 Deep learning based:

For the deep learning baseline, we transformed the recommendation problem into a binary classification problem and took use of customers' features, articles' features and the transaction records. We partially adopted the method Neural Collaborative Filtering introduced by [HLZ+17].

We constructed our train and test set by taking the cross product of the (one-hot encoded) article table and the (one-hot encoded) customer table. Each row in our dataset represents an (article, customer, label) triplet, where the first part of the row contains article features(product type, color, secion, etc.), the second part of the row contains the features of a customer (age, fashion news frequency, etc.), and the final entry is a binary label that we aim to predict, 0 indicating that this customer has never bought this article, and 1 indicating that this (article, customer) pair has appeared in the transaction record. We provide the figure (1) information to get the visual sense of our training data.

Except for a 70%-30% split, we also implemented the train-test split to ensure that customers in the training set do not overlap with those in the testing set.Since there are much more class 0 instances

	product_type_name_Bag	product_type_name_Ballerinas	product_type_name_Belt
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

(a) Article

...

club_member_status_PRE-CREATE	fashion_news_frequency_None	fashion_news_frequency_Regularly	label
0	1	0	0
0	0	1	0
0	1	0	1
0	0	1	0
0	1	0	0

(b) Customer and label

Figure 1: The features for both customer and items are one hot encoded. The predicted label is whether this customer has bought this article.

than class 1 instances, we resampled the dataset to reduce the number of 0 instances to be the same as 1's to balance out the labels.

Under this problem formulation, we trained Multilayer Perceptron to predict the labels for new (article, customer) pairs in the test set. We used a sequence of linear layers followed by batchnorm and relu, as shown below. We then applied a sigmoid layer to convert output into a probability score.

```
Net(
(layers): ModuleList(
(0): Linear(in_features=173, out_features=64, bias=True)
(1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU()
(3): Linear(in_features=64, out_features=128, bias=True)
(4): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): ReLU()
(6): Linear(in_features=128, out_features=256, bias=True)
(7): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(8): ReLU()
(9): Linear(in_features=256, out_features=128, bias=True)
(10): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(11): ReLU()
(12): Linear(in_features=128, out_features=64, bias=True)
(13): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(14): ReLU()
)
(out): Linear(in_features=64, out_features=1, bias=True)
)
```

We trained the MLP model using mini-batch gradient descent with a Binary Cross Entropy loss to capture the distance.

As for the recommendation itself, given a specific customer, we first create a tensor of the (article, customer) pairs, matching the customer with all possible articles in our article dataset. This tensor is inputed into the MLP model to output a score for each article. We then rank the articles by their scores and pick the 50 items with the highest scores (highest probability to be bought) as the final recommendation for this customer.

5.3 Advanced DL method

The advanced methods we adopt is Graph Neural Network, which incorporates user, item, and transaction data for recommendation. GNN is a neural network approach for graph-structured data, which was first introduced by [GMS05]. In our project, we represent each item as a node in the graph, where the node's features are the item's characteristics, and the label is its category. The edges of the graph represent whether two items have been purchased by the same customer. To further enhance our model's accuracy, we add age as an edge feature, considering that people of different ages may have different purchasing behaviors.

To construct the graph and train the network, we utilize the GraphSAGE [HYL17] package. Our GNN model's output is the link prediction based on the current purchase information, which recommends the most relevant items to a customer based on their previous purchases.

$$\hat{y}_{u\sim v} = f(h_u, h_v) \tag{1}$$

6 Results

In this section, we initially show the training results and testing results for each method.

6.1 Content Based

The hit rate we get for Content Based method is **0.53**. To visualize the result we got, we initially show the purchases for a chosen customer in figure (2). The purchases predicted by our KNN model is shown in figure (3). We could see from figure (3) that the articles recommended show similar style in color, shape as the original purchases.

Figure 2: For a chosen customer, the known purchased Articles.

6.2 Deep Learning

To get correct label about the buy information, we get the training process as below. After 1400 Batch training process, we get the final hit rate **0.56**. The corresponding recommended items by our deep learning model is the same as figure (5).

6.3 GNN

We tune the hyper parameter number of layers, number of features each layer for the GNN model. The results we get in the table is the link prediction results for the graph. The best model we get is the GNN with **2** layers, each layer with **64** features. We also tune the recommended items' number, with results as the following figure (7). The recommended outfit by our GNN model is in figure (8).

7 Discussion

Our project focuses on building a recommendation system using three different methods: Content-based KNN, Neural Network, and Graph Neural Network. Each of these methods has its own unique approach to constructing the model and utilizing item, customer, and transaction information. By using these different methods, we are able to obtain varying recommended results, which provides valuable insight into the importance of considering different types of information during the model selection process.

Additionally, our project provides guidance on how to incorporate different types of data when building a recommendation system. For example, we explored the use of pairwise customer-article

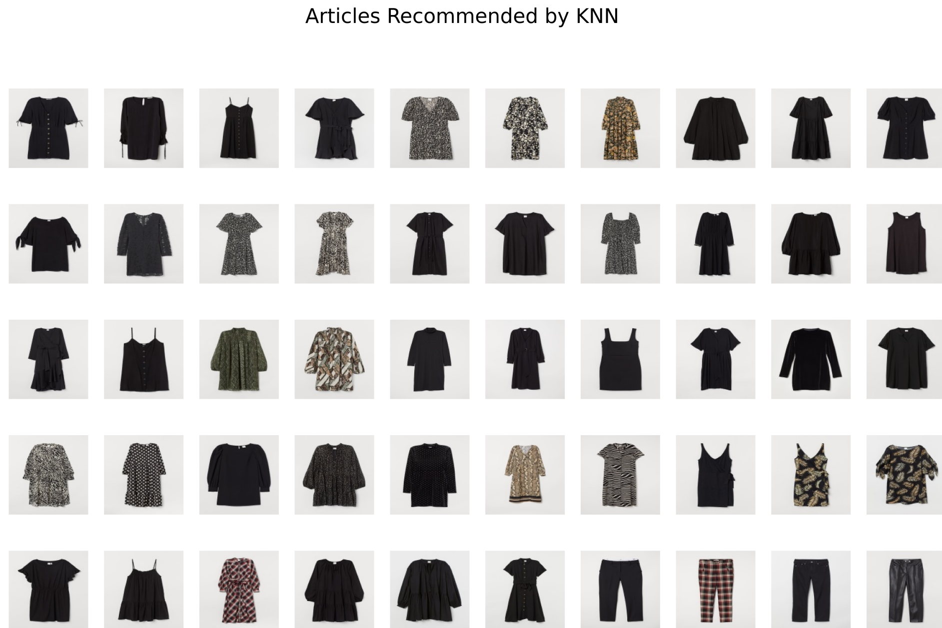

Figure 3: The Articles recommended by KNN model.

information in the deep learning section, as well as the use of simultaneously buying conditions in the GNN section. Both of these approaches involve concatenating data and making predictions about transactions between two data sets.

Overall, our project emphasizes the importance of considering multiple factors when building a recommendation system, and provides valuable insights into how different types of data can be used to improve the accuracy and effectiveness of the model.

7.1 Findings

The final hit rate comparison is shown in the table below.

Baseline KNN	MLP	GNN
0.53	0.56	0.95

Table 1: Hit Rate comparison for the three methods.

Based on our findings, the Graph Neural Network (GNN) model outperformed the other two methods, with the deep learning method coming in second place. The content-based approach, which only utilizes item features, had the lowest performance among the three methods. We also conducted a visual analysis of the recommended results and found that the GNN and deep learning models generated more diverse and flexible recommendations compared to the content-based approach. This is a valuable insight for businesses looking to improve their recommendation systems, as it suggests that incorporating more information about customers and their behaviors can lead to more personalized and satisfying recommendations.

There are other research directions like recommend one out of a fit based on the previous whole fit [CLW+19].Research on textual content, such as posts and comments [LA11], emotion and information diffusion [SDX13], and images has attracted the attention of modern day researchers, as it can help to predict fashion trends and facilitate the development of effective recommendation systems.

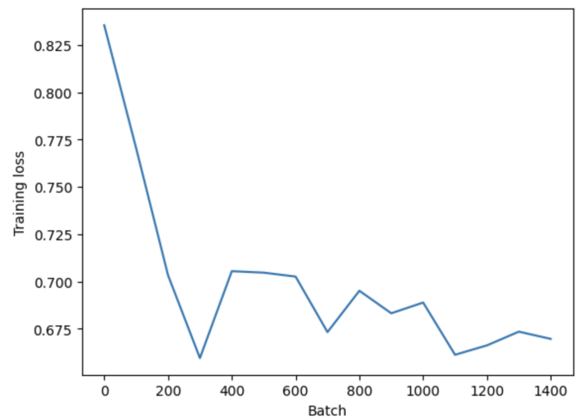

Figure 4: The training process of Base Deep Learning Model

Figure 5: The Articles recommended by Deep learning model.

7.2 Limitations and Ethical Considerations

The accuracy of our recommendation models heavily depends on the quality and quantity of the data used for training. In our study, we only focus on ladies-wear dataset in $H\&M$, which may not accurately reflect the full range of customer behaviors and preferences. This could limit the generalizability of our findings to other datasets. The content-based approach, which relies solely on item features, may struggle with new items that do not have enough data to be properly characterized. This is known as the cold start problem and could lead to less accurate recommendations for new or less popular items. While the GNN and deep learning models performed well, they are often seen as "black boxes" that are difficult to interpret. This means that it may be challenging to understand how the models are making their recommendations, which could limit the ability to fine-tune or improve the models. Moreover, the GNN model requires significant computational resources and can be challenging to scale for large datasets or high traffic websites. This could limit its practical applications in some scenarios.

7.3 Future Research Directions

Because of limitations in time and computational resources, we only incorporated the age feature in our GNN model. However, in future work, we plan to include other customer features such as social status and fashion news frequency. Additionally, we aim to integrate the most recent Hierarchical Fashion

num_features/ layers	16	32	64	128
2	0.7152	0.7165	0.7196	0.7188
3	0.7148	0.7159	0.7157	0.7154
4	0.7099	0.7152	0.7166	0.7134

Figure 6: Hyper parameter tuning for the GNN model, number in the block stands for the link prediction accuracy.

# of rec articles	20	50	100	150
Hit Rate	0.7940	0.9518	0.9867	0.9938

Figure 7: Number of recommended articles tuning for the best GNN model.

Graph Network from [LWH+20] into our Personalized Outfit Recommendation system to enhance its performance.

8 Conclusions

Our study investigated three distinct approaches to building a recommendation system: Content-based KNN, Neural Network, and Graph Neural Network. Our results indicate that the Graph Neural Network approach was the most effective, followed by the deep learning method. However, it is worth noting that the effectiveness of these methods can be influenced by the quality and quantity of data used for training. Our findings underscore the importance of incorporating customer and transaction information into recommendation models, as these factors can greatly impact the accuracy and diversity of the recommendations provided.

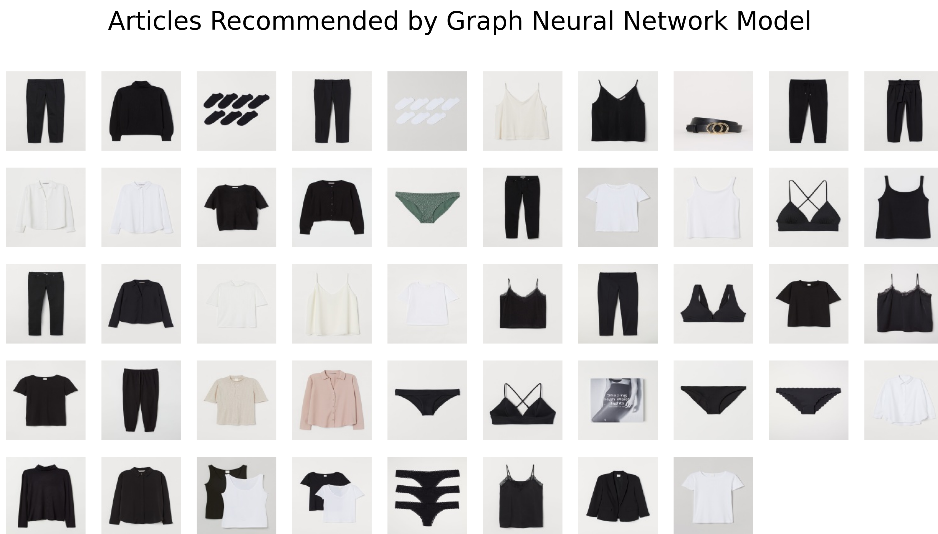

Figure 8: Articles recommended by GNN model.

9 References

References

[CLW+19] Zeyu Cui, Zekun Li, Shu Wu, Xiao-Yu Zhang, and Liang Wang. Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In *The world wide web conference*, pages 307–317, 2019.

[GMS05] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2, 2005.

[HLZ+17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[HM16] Ruining He and Julian McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[JPB+14] Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan. Large scale visual recommendations from street fashion images. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1925–1934, New York, NY, USA, 2014. Association for Computing Machinery.

[LA11] Himabindu Lakkaraju and Jitendra Ajmera. Attention prediction on social media brand pages. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 2157–2160, New York, NY, USA, 2011. Association for Computing Machinery.

[LWH+20] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. Hierarchical fashion graph network for personalized outfit recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 159–168, 2020.

[SDX13] Stefan Stieglitz and Linh Dang-Xuan. Emotions and information diffusion in social media—sentiment of microblogs and sharing behavior. *Journal of Management Information Systems*, 29(4):217–248, 2013.

[SWZ+16] Dandan Sha, Daling Wang, Xiangmin Zhou, Shi Feng, Yifei Zhang, and Ge Yu. An approach for clothing recommendation based on multiple image attributes. In *Web-Age Information Management: 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part I 17*, pages 272–285. Springer, 2016.