Personalized Fashion Recommendations in Dynamic Fashion Trends

MSc Research Project
Data Analytics

Shafik Rahman Salim Badhusha
Student ID: x22123661

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shafik Rahman Salim Badhusha
Student ID:	x22123661
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Noel Cosgrave
Submission Due Date:	14/12/2023
Project Title:	Personalized Fashion Recommendations in Dynamic Fashion Trends
Word Count:	6942
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shafik Rahman Salim Badhusha
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Personalized Fashion Recommendations in Dynamic Fashion Trends

Shafik Rahman Salim Badhusha
x22123661

Abstract

This study examines the complexities in using RecBole sequential algorithms in suggesting fashion goods for H&M users, while acknowledging the distinct difficulties presented by the fashion industry's cyclical trends. The research thoroughly covers the crucial stages of data preparation, model development, training, and filtering, assuring compliance with industry norms and adhering to data integrity and ethics. An important addition is the establishment of a structured process for making recommendations in multiple stages, specifically designed to include the time-related aspects that are inherent in fashion data. By using a large dataset from H&M, thorough testing demonstrates the intricate performance of RecBole sequential algorithms. The study examines how well the algorithms perform in suggesting things in the ever-changing fashion environment of H&M. Different models using two main sequential algorithms, GRU4Rec and Bert4Rec were built and evaluated.

1 Introduction

The realm of fashion is characterized by its dynamic and ever-changing nature, where trends emerge and fade within short time frames. For fashion enthusiasts, staying abreast of the latest trends and selecting clothing that aligns with individual tastes and circumstances can be both challenging and time-consuming. Personalised fashion recommendations have emerged as a solution to this dilemma, tailoring suggestions based on a user's profile, behavior, and feedback. However, conventional fashion recommender systems often rely on simplistic, generic models that overlook the intricate nuances of fashion, such as style, fit, color, occasion, and season.

To address these limitations, this report introduces an innovative approach to user-specific fashion recommendations within the context of dynamic fashion trends, leveraging the RecBole framework. RecBole stands out as a comprehensive and adaptable library for recommender systems, accommodating various models, datasets, and assessment techniques. Specifically designed for dynamic fashion recommendations, RecBole integrates temporal data, visual features, and textual descriptions of clothing. The proposed research adopts a streamlined methodology encompassing data collection, preprocessing, algorithm integration, and evaluation, aiming to enhance personalized fashion suggestions through the application of the RecBole algorithm on the H&M dataset.

1.1 Research Background and Motivation

The ever-evolving landscape of fashion continually introduces novel trends, creating a challenge for individuals to stay abreast of the latest styles. This predicament is particularly evident in the fashion industry, prompting a growing reliance on personalized fashion recommendation systems. These systems play a pivotal role by suggesting tailored clothing and outfits that align with an individual's preferences and requirements. However, a significant drawback emerges as many of these systems hinge on static information about users and garments, which might not always capture the dynamic nature of evolving trends or changing personal tastes. Acquiring pertinent data to enhance these systems further exacerbates the challenge.

Recognizing the aforementioned gaps in existing personalized fashion recommendation systems, this study explores the utilization of the H&M dataset. This publicly accessible compilation encompasses fashion items and user interactions from the H&M online store. The primary objective is to investigate and address the intricacies of personalized fashion recommendations in the context of dynamic fashion trends. To achieve this, the study employs RecBole, a unified and comprehensive framework for recommender systems. By implementing and comparing a spectrum of cutting-edge recommendation models on the H&M dataset, the research aims to shed light on the efficacy of different approaches.

Integral to the recommendation process are distinctive techniques designed to incorporate temporal data and trend awareness. The study conducts in-depth evaluations to compare the performance of various models and approaches across a spectrum of tasks. These tasks encompass challenges such as cold-start suggestions, outfit selection, and next-item prediction. Through this comprehensive analysis, the research seeks to contribute insights into enhancing the adaptability and effectiveness of personalized fashion recommendation systems within the dynamic landscape of evolving fashion trends.

1.2 Research Question

Does the RecBole algorithm provide personalized fashion recommendations for H&M customers in the dynamic landscape of fashion trends? This research aims to evaluate the algorithm's performance utilizing the H&M dataset and, concurrently, to develop a recommender system for personalized fashion suggestions.

1.3 Contribution

The personalized fashion recommendation system offers u sers a d ynamic a nd tailored experience by incorporating temporal dynamics and leveraging the RecBole framework. This research contributes to the advancement of recommender systems by addressing the challenges posed by the ever-changing nature of the fashion industry. The adoption of RecBole provides a versatile platform for implementing and comparing state-of-the-art recommendation models. A multi-stage recommendation pipeline, encompassing data preparation, model training, and user-focused filtering, e nsures a comprehensive and structured approach. Evaluation metrics such as mean average precision, precision, recall provide a thorough assessment of system performance. Ethical consid-erations, including user privacy and responsible data usage, are emphasized throughout the research, which utilizes the H&M dataset for practical relevance. The main goal is to improve the accuracy and usefulness of recommendations by using RecBole, taking into account how time changes, and a multi-stage pipeline. This will allow for more precise and relevant fashion suggestions that are specific to each user's tastes. This research not only contributes to the immediate field but also sets a framework for future investigations in personalized fashion recommendations and recommender systems.

2 Related Work

The significance of literature reviews cannot be overstated in research papers, particularly those focused on personalized fashion advice. These papers delve into providing users with tailored clothing recommendations based on their preferences, style, and context. However, the realm of the fashion industry is complex and ever-changing, with trends evolving rapidly and consumer preferences fluctuating. Consequently, developing effective and dependable personalized fashion suggestion systems is a challenging task that requires innovative approaches and strategies.

2.1 Recommender Systems and Personalization Techniques

Recommender Systems and Personalization stand as significant realms within information retrieval and data mining research. As highlighted by Hong et al. (2012) and Koren (2009), temporal dynamics play a crucial role in recommendation systems. To offer context-aware and fitting fashion recommendations, these studies emphasize the necessity of considering user preferences and evolving fashion trends over time. Incorporating temporal dynamics has the potential to enhance the effectiveness and accuracy of personalized fashion recommendations by aligning them with user preferences and contemporary trends.

Effectively harnessing users' interests and content they generate is achievable on social resource-sharing platforms through the application of collaborative filtering methods, a subject explored by Huang et al. (2014) and Larrain et al. (2015). Leveraging user tags and social tagging data contributes to a deeper understanding of users' fashion preferences and tastes, ultimately resulting in fashion recommendations that are more pertinent and engaging.

As indicated in the study conducted by Lathia et al. (2010), temporal variation emerges as a crucial factor in the realm of personalized fashion recommendations. Recommender systems that consider the evolving fashion preferences of users can address this temporal variation, presenting a diverse array of recommendations over time. The integration of temporal diversity concerns might enhance the performance of the RecBole algorithm, enabling it to provide a more extensive selection of contextually relevant fashion choices.

Furthermore, the studies conducted by Liu, Rogers, Shiau, Kislyuk, Ma, Zhong, Liu and Jing (2017) and Medvedev et al. (2019) on practical applications of recommender systems at Pinterest and Instagram highlight the real-world effectiveness of these systems on large-scale platforms. These applications offer valuable insights into how the RecBole algorithm could be employed and improved to address evolving fashion trends and deliver customized recommendations on e-commerce platforms like H&M. The combination of deep learning approaches has demonstrated considerable potential in recommendation tasks, as evidenced by Liu, Gao, Feng and Li (2017) and Ke et al. (2019). By incorporating deep learning frameworks, RecBole may leverage rich visual and contextual data to provide more precise and engaging fashion recommendations in personalized settings.

2.2 Machine Learning Paradigms in Information Retrieval

The existing literature showcases advancements in recommendation systems within the domains of machine learning and information retrieval, especially in the context of personalized fashion recommendations using the H&M dataset, which encompasses dynamic fashion trends. In this regard, Ke et al. (2017) introduced "LightGBM," a powerful gradient-boosting tree framework designed to improve recommendation accuracy by capturing intricate patterns in user-item interactions. This framework also considers temporal dynamics, allowing it to provide contextually aware and timely suggestions.

Liu et al. (2010) proposed a multi-stage approach for addressing the research focus on tailored recommendations, aligning well with the H&M dataset and accommodating unique user preferences and behaviors in the dynamic realm of fashion. The utilization of Machine Learning algorithms in RecBole's multi-stage process enables it to furnish precise and nuanced personalized outfit selections. The integration of RecBole with deep convolutional neural networks, as demonstrated by Xiang et al. (2019), showcases its capability to offer compelling and relevant retrieval of fashion items, leveraging deep learning approaches in visual information retrieval for fashion-related visual content. These integrated advancements underscore the significance of machine learning in information retrieval and highlight RecBole's potential to provide customers with context-aware and aesthetically pleasing personalized fashion recommendations in the rapidly evolving landscape of online clothing platforms such as H&M.

2.3 RecBole Algorithm

The RecBole algorithm has proven to be a reliable framework for recommendation systems, tackling the research inquiry regarding its efficacy in terms of recommendation accuracy and relevance in the context of personalized fashion suggestions with the H&M dataset, encompassing dynamic fashion trends. Zhao et al. (2021) and Zhao et al. (2022) introduced RecBole as a comprehensive and unified framework for recommendation algorithms. Their contributions laid the foundation for RecBole's versatile toolkit, capable of accommodating various recommendation models and simplifying the implementation process. RecBole offers researchers the opportunity to explore various algorithms to enhance personalized fashion recommendations by aligning them with evolving fashion trends and user preferences. Xu et al. (2022) have extended the capabilities of RecBole, incorporating more realistic considerations in their enhancements, building upon previous work. This study underscores the continuous improvement and expansion of the RecBole framework in practical and meaningful ways. The recent advancements in RecBole may have bolstered its performance in tailoring fashion ideas to individual preferences by addressing practical challenges and incorporating real-world considerations. These improvements likely take into account factors such as scalability, effectiveness, and user-friendliness, making RecBole conducive to supporting extensive e-commerce platforms like H&M, where adapting to changing fashion trends is crucial for sustaining user engagement.

Figure 1 depicts the structure of a Recbole recommender system, which is segmented into three primary phases: Data preprocessing, Model training, and Model evaluation and deployment. During the Data preparation phase, the system undertakes the task of generating a dataset by gathering information from different sources, subsequently cleaning and modifying it, and finally dividing it into training, validation, and test sets. The Model training stage involves training recommender models that take into account both sequential and non-sequential user interactions with items, while considering the order in which these interactions occur. The Model evaluation and deployment phase include a Case study component that assesses the effectiveness of the trained models. In addition, there are modules for training and assessing models, configuring random seeds, and initialising loggers.

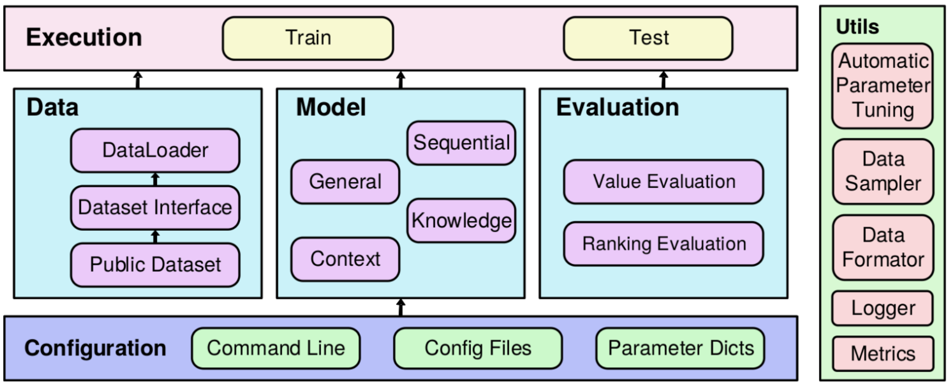


Figure 1: RecBole - Overall Architecture

The research specialization revolves around leveraging the H&M dataset and the RecBole algorithm to enhance personalized fashion recommendations, with a primary focus on elevating recommendation accuracy and relevance. The proposed study aims to refine the RecBole model by incorporating temporal dynamics, utilizing the structured CSV data within the H&M dataset. This enhancement seeks to deliver fashion recommendations that are more contextually relevant, aligning with the evolving fashion tastes and dynamic trends of consumers. The investigation will also evaluate RecBole's collaborative filtering methods, considering user-tag-based preferences and social tagging data from the dataset to provide more customized and socially conscious fashion suggestions. The anticipated contribution of the research is to showcase how RecBole enhances personalized fashion recommendations using structured data from the H&M dataset. This, in turn, aims to offer valuable insights for e-commerce platforms like H&M to enhance their recommendation systems and deliver more precise and captivating fashion suggestions tailored to individual user preferences.

2.4 RecBole Sequential Recommendations

Recbole has different sequential recommendation models. The Github[1] link displays all the recbole models which are tested on three different datasets namely ml-1m, Amazon-Books and yelp2022 and the time and memory cost taken by individual models. The paper "Improved Recurrent Neural Networks for Session-based Recommendations" Tan et al. (2016) specifically examines the application of recurrent neural networks (RNNs) in the field of deep learning for recommender systems, with a focus on session-based recommendations. Session-based recommendations are a specific form of recommendation system that seeks to anticipate the next-click items for an unidentified user by analysing the pattern of short-term sequences. Recurrent Neural Networks (RNNs), specifically

[1]https://github.com/RUCAIBox/RecBole/blob/master/asset/time_test_result/Sequential_recommendation.md

long short-term memory (LSTM) and gated recurrent unit (GRU) networks have demonstrated exceptional performance in modeling sequential data, especially data supplied by users in a session-based fashion. RNNs offer significant performance advantages over traditional approaches when used in session-based recommendations. GRU-based models, like Gru4Rec, are good at managing sequential user data, which makes them even more useful for session-based recommendation tasks.

There is a strong case for using BERT4Rec along with Gru4Rec for sequential recommendation problems in the paper "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer" Sun et al. (2019). BERT4Rec uses Bidirectional Encoder Representations from Transformers (BERT) to solve the problems that come up with session-based recommendations (SBR). It is very important to correctly guess what the next item in a sequence will be, especially when there is not a lot of data from previous users. BERT4Rec differs from Gru4Rec by utilizing bidirectional architectures instead of unidirectional ones. This enables BERT4Rec to gather contextual information from both the preceding and succeeding parts of a user's behaviour sequence. Bidirectional modeling improves the depiction of things, resulting in more resilient and contextually aware sequential recommendations. BERT4Rec has a notable benefit in its capacity to escape the constraints of strictly ordered sequences and fully utilise the potential of concealed representations in user behaviour sequences. The research emphasises the greater ability of BERT-based models to capture complex sequential patterns, making them a more advanced and effective option for sequential recommendation tasks compared to classic unidirectional models such as Gru4Rec. Referencing algorithms like GRU4Rec, Caser, SRGNN, SASRec, BERT4Rec, and RepeatNet are used in that paper.

Li et al. (2023) Table 3 shows the novelty scores for repeat-next users (RNU) and explore-next users (ENU). The novelty measures, such as Novelty@1 and Novelty@3, measure the degree of uniqueness in suggestions for users who engage in repetitive interactions and those who seek out new goods. GRU4Rec has higher scores than BERT4Rec in terms of Novelty@1 and Novelty@3 for repeat-next users (RNU), with scores of 0.567 and 0.725, compared to BERT4Rec's scores of 0.385 and 0.530. Nevertheless, when it comes to explore-next users (ENU), BERT4Rec outperforms GRU4Rec, demonstrating its superiority in delivering innovative recommendations for users who are seeking out new goods. The findings indicate that GRU4Rec is very effective in capturing recurring patterns in user behaviour, whereas BERT4Rec exhibits proficiency in providing innovative recommendations for users who desire various suggestions. The recommendation system's objectives determine the decision between GRU4Rec and BERT4Rec, which trades repetition and exploration. If recommendations should match users' past interactions, GRU4Rec may be best. However, BERT4Rec may be better for novelty and exploration. For the recommendation system's purposes and user preferences, the two models' repetition-exploration balance determines the choice.

3 Methodology

The GRU4Rec model and a KDD (Knowledge Discovery in Databases) strategy are the foundations of this project's methodology. To provide tailored fashion recommendations in the always-changing world of dynamic fashion trends. Utilising information from the well-known clothing store H&M, the method starts with a thorough data pre-processing stage [2] . The integration of customer, article, and transaction data involves intricate steps, including date handling, grouping, and aggregation. Exploratory data analysis sheds light on product group dynamics, customer age distribution, and time-series patterns in the context of weekly sales. The model can make sophisticated recommendations because several parameters were carefully considered during configuration and training. The research then assesses the model's performance using metrics like recall, precision, and MAP to determine how well-customized fashion recommendations work in the ever-evolving fashion ecosystem that is moulded by H&M's wide-ranging and constantly-evolving product catalogue.

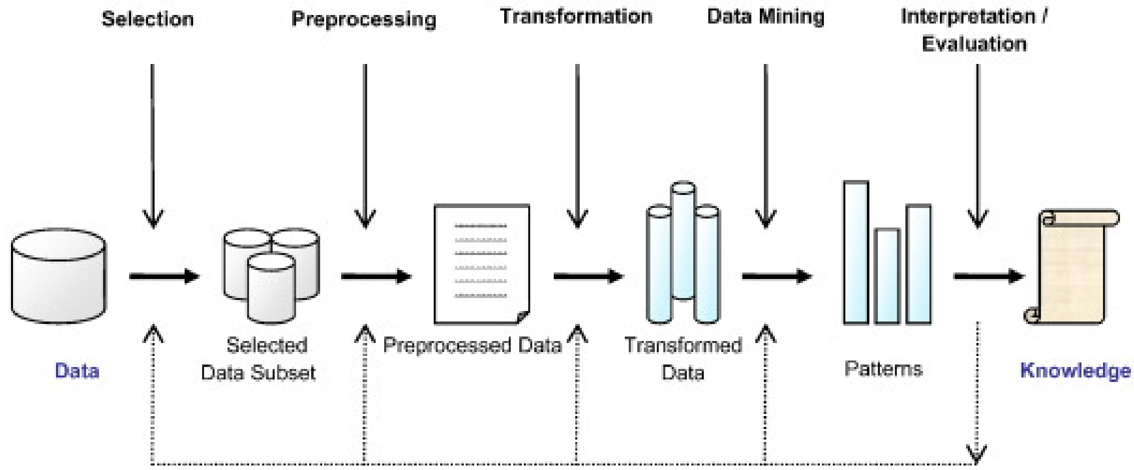


Figure 2: KDD Process

3.1 Data Collection

Data for this research was gathered using Kaggle, notably from the H&M hosted "Personalized Fashion Recommendations" competition. This dataset includes information on a variety of user interactions with fashion-related products, such as past purchases, browsing habits, ratings, and feedback. In addition, user demographics, and temporal data documenting the timeline of interactions, brand, category, color, material, and style elements of fashion items will be included. To safeguard user privacy and maintain data anonymization throughout the research, ethical considerations will be taken into account. For this purpose, we are using the Kaggle dataset which was published by H&M [3]. This dataset offers a rich and realistic depiction of consumer behaviors and preferences by providing an authentic representation of user interactions. This real-world dataset of user interactions serves as the basis for building a recommender system that precisely captures every detail of consumers' interactions with H&M's online store. The use of actual user data improves the system's robustness and usefulness by ensuring that the developed recommendation models are in line with the complexity of customer interactions in a real-time fashion retail environment.

[2] KDD Methodology:https://www.csbj.org/article/S2001-0370(16)30073-3/fulltext
[3] https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations

3.2 Data Preprocessing

Data preprocessing serves as an essential step in this project's methodology, playing a crucial role in preparing the raw data for subsequent analysis and modeling. The complexities of managing various data formats, maintaining consistency, and extracting valuable features are covered in this part, which lays the foundation for the GRU4Rec model's integration into the RecBole framework. Preprocessing includes all of the integration of transaction, article, and customer data, including the finer points of grouping, aggregation, and data handling. By examining this crucial phase, one may gain an understanding of the actions made to improve the dataset's relevance, quality, and structure. This prepares the ground for a thorough analysis and the creation of an advanced recommender system that is suited to the ever-changing world of fashion trends.

3.3 Data Transformation

The transformation step in this project's methodology is an important part of making the preprocessed data better so that it can be used most effectively in the GRU4Rec and Bert4rec model built on the RecBole framework. Examining the high-level techniques used to transform the preprocessed data into a format that satisfies the particular needs of the recommendation system. The transformation includes several different tasks, including creating new features and transforming the data into atomic files that can be used for modeling. This conversion enhances the efficiency of the GRU4Rec model by streamlining access to the necessary information. This stage also involves filtering the dataset to the top N popular goods, which is essential to guarantee that the model can make suggestions that are in line with the prevailing trends and tastes in the H&M product catalog. This stage serves as a link between the preprocessed data and the other phases of analysis and model building, establishing the foundation for an advanced and customized recommender system in light of the ever-changing fashion trends influenced by the H&M dataset.

3.4 Model Selection

RecBole, a comprehensive and state-of-the-art framework for recommendation algorithms, will be utilized to integrate the preprocessed dataset. In this context, the model-building phase of data mining encompasses training, optimisation, feature engineering, and method selection. Using the GRU4Rec and Bert4Rec models, this step aims to get deep insights from the expanded dataset. This will allow the system to spot small patterns in how customers behave and what they like. The Gated Recurrent Unit for Recommendation, or GRU4Rec, represents a recurrent neural network (RNN) architecture specifically designed for sequential recommendation applications.[4] Additionally, the Bert4Rec model, which is an extension of the BERT architecture tailored for recommendation tasks, will contribute to enhancing the system's understanding of user-item interactions in a more contextualized manner. Both GRU4Rec and Bert4Rec belong to the neural collaborative filtering family, providing advanced capabilities for capturing sequential patterns and contextual information in recommendation scenarios.

To evaluate the model's performance, a training set comprising 90% of the dataset is utilized, with the remaining portion constituting the test set.

[4] https://recbole.io/docs/user_guide/model/sequential/gru4rec.html

3.5 Evaluation Metrics

In the evaluation phase of the recommendation system, a collection of robust metrics is employed to thoroughly assess its performance. Mean Average Precision at K (MAP@K) is a crucial indicator that evaluates the precision of the top-K recommended items by calculating the average precision across all users. This statistic is especially relevant in situations where the sequence of recommendations is significant, such as in the context of fashion recommendations. The calculation of MAP@K entails determining the precision at each place within the recommendation list and then taking the average of these values across all users.

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

where U is the number of customers, P(k) is the precision at cutoff k, n is the number of 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.

Precision, another crucial metric, focuses on the accuracy of the recommended items. Precision@K quantifies the proportion of relevant items among the top-K recommendations. This indicator offers valuable insights into the system's ability to accurately choose things that match the user's tastes, based on the set number of recommendations.

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive(FP)}$$

The third parameter in our toolbox is recall. Recall measures how well the system can record all pertinent elements, which is a complement to accuracy. In our case, Recall@K assesses the proportion of pertinent things to total relevant items discovered in the top-K suggestions. This measure is essential for figuring out how well the system covers the whole range of relevant items, making sure it doesn't miss any potentially interesting offerings.

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

These measures together provide a detailed insight of our recommendation system's effectiveness in the dynamic and visually-driven field of fashion recommendations, where user satisfaction depends on personalised and trend-right choices. While precision and recall give particular insights into the accuracy and coverage of the system, respectively, MAP@K offers an aggregate metric that takes into account both the precision and order of suggestions. Through these perspectives, we can assess our system and adjust its settings and algorithms to improve user experience and engagement with the always changing fashion scene.

4 Design Specification

The overall architecture of the data is shown in Figure 4. The sources of the datasets used in this research are various and include a wide spectrum of fashion-related interactions. The data comes mostly from H&M's online store and includes comprehensive transaction records, in-depth article information, and rich consumer profiles as shown in the below table.

Table 1: Details of Datasets

Dataset	Count	No. of Features
Transactions	31.8 M	3 context features
Articles	106 K	24 item features
Customers	1.3 M	6 customer features

Transactions have customer_id and article_id which are foreign keys for the customer and articles tables. Besides this, transaction also contains sales_channel_id. To handle the volume of the data, a targeted dataset covering one week was assembled for the study. The design specification for the data architecture is a comprehensive plan that outlines the various stages involved in the data processing and recommendation generation process. The raw data is undergone preprocessing process to ensure data quality and consistency. This includes Data cleaning, normalization and transformation to address any inconsistent data. The preprocessed data is then visualized and analyzed to gain insights into the dataset's structure and patterns, which helps in the identification of potential features for the recommendation system.

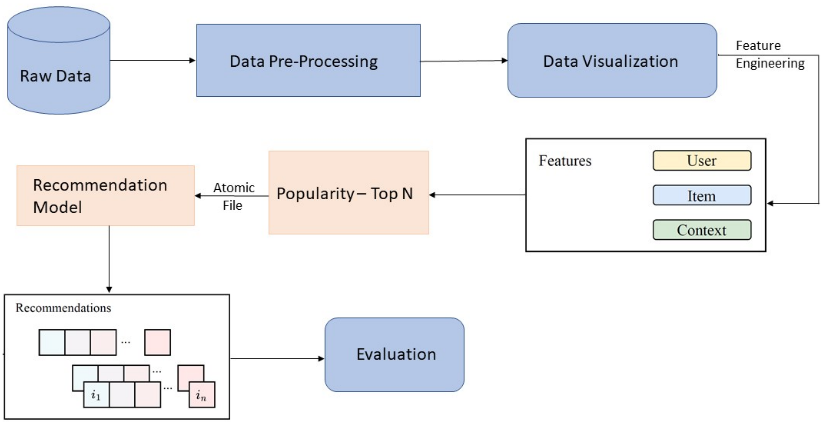


Figure 3: Overall Architecture

The data architecture also includes a feature engineering stage, where new features are created or transformed existing ones to better represent user preferences and item characteristics. This stage involves selecting appropriate features, feature scaling, and feature engineering techniques to improve the performance of the recommendation system. A key component of the architecture is the Popularity Top N module, which identifies the most popular items within the dataset. This module helps in understanding the overall preferences of the users and the most popular items in the dataset, providing valuable insights for further analysis and recommendation tuning. The recommendation model is another crucial component of the architecture. It leverages the processed data to generate personalized recommendations to each user. The model is trained on preprocessed and feature-engineered data, taking into account the identified user preferences and item characteristics. The model's performance is evaluated using appropriate metrics, such as precision, recall, and mean absolute error, to ensure its effectiveness in generating accurate and personalized recommendations. The final stage of the data architecture involves the generation of recommendations based on the trained recommendation model. These recommendations are personalized and tailored to each user, taking into account their preferences and the model's predictions. The recommendations are then evaluated to assess their quality and effectiveness in meeting the user's needs and expectations."

5 Implementation

In the implementation phase of this project, a systematic approach to leverage the provided dataset for predicting customers' future article purchases was undertaken. Python language was utilized as the primary programming language leveraging its rich ecosystem of libraries and frameworks for data analysis, preprocessing, and machine learning. Jupyter Notebooks was used as an interactive development environment, providing a seamless platform for code exploration, experimentation, and visualization. In this part, the things discussed in methodology will be explained in detail.

5.1 Cloud-Based Development Environment and Computational Infrastructure

The development environment for the models encompassed a seamless integration of Google Cloud Platform (GCP) services to leverage robust computational resources and efficient data management. Cloud data storage is used to store the Data from H&M ensuring scalable and secure data access. A virtual machine (VM) was provisioned on GCP, specifically configured with TensorFlow 2.13, a pivotal framework for deep learning applications. The VM boasted a compute-optimized configuration, equipped with 8 vCPUs and 32 GB RAM, providing substantial computational power for model training and analysis. Running Jupyter Notebooks further improved this environment's usefulness by fostering an interactive and collaborative development experience. This mix of cloud-based infrastructure, custom machine specs, and flexible development tools created a flexible setting that was good for testing, exploring, and training recommendation models like GRU4Rec and Bert4Rec within the RecBole framework.

5.2 Data Preprocessing

In the data preprocessing phase, a systematic approach was employed to refine and structure the dataset for subsequent analysis and modeling. The dataset resides in the Google Cloud storage bucket in a CSV format, as shown in the below figure. The first step

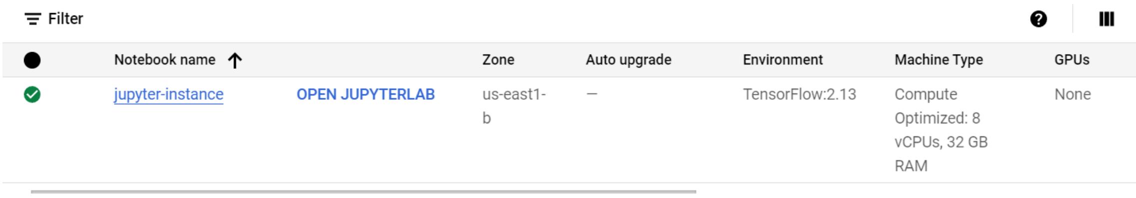


Figure 4: GCP Environment

was to load the "articles," "customers," and "transactions" datasets from CSV files. To combine transactional data with customer and article metadata, a sequence of inner joins was executed on the 'custome_id' and 'article_id' variables. 'FN' and 'postal_code,' two unnecessary columns, were removed to make the dataset more manageable. The 't_dat' column was converted to DateTime format to manage temporal characteristics, and new features like 'year_month' and 'year' were added to capture temporal patterns. Furthermore, the data is filtered to last one week data to handle the voluminous nature of the dataset. To guarantee the accuracy and consistency of the dataset, unique values, and missing data were also checked.

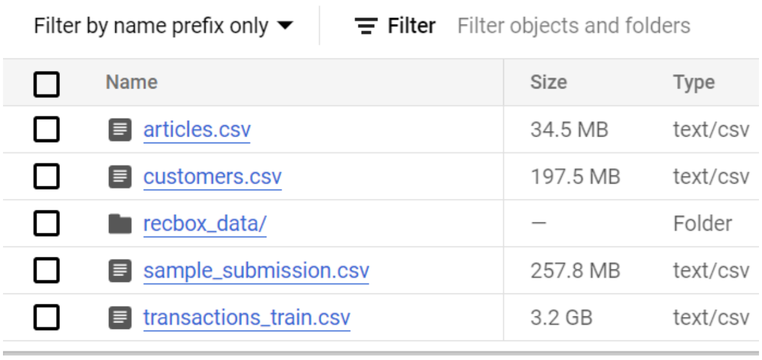


Figure 5: Google Cloud Storage bucket

5.3 Data Visualization

Several visualizations were used in the visualization research to highlight different aspects of the dataset and provide important information for the creation of a customized fashion recommendation system. The distribution of product types among various product groups was depicted in the first set of bar plots, which highlighted the diversity within each category as shown in Figure 6. Following that, the analysis focused on index groups and index names, uncovering the popularity of particular indices and supporting the identification of potential trends.

The age distribution histogram displayed the age (as shown in figure 7) demographics of clients, which contributed to the demographic profiling element of the analysis. Moreover, by visualizing the quantity of consumers about the frequency of fashion news, we gained insight into the level of user involvement with fashion-related material. Furthermore, the display of recognized color master names demonstrated the dominance of certain color categories within the dataset. The last series of visualizations concentrated on sales transactions over a period, offering valuable insights into the temporal fluctuations of sales channels. These visualizations were essential in revealing patterns, trends,

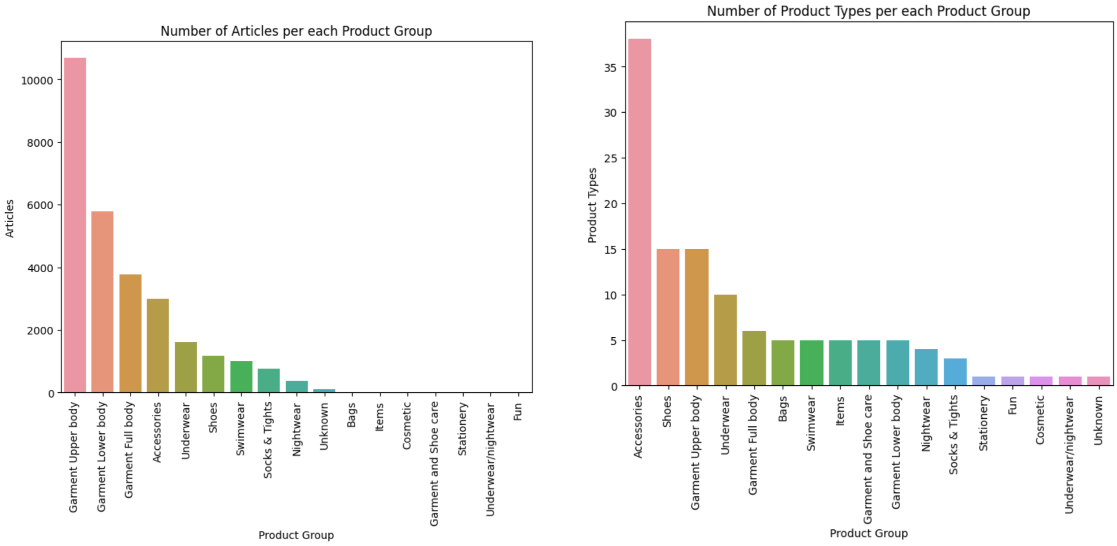
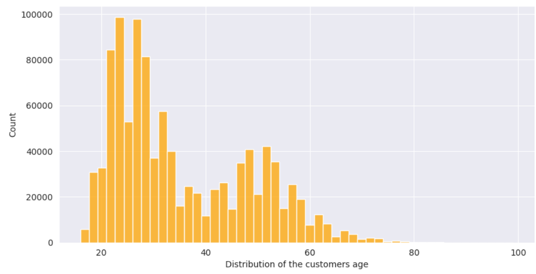
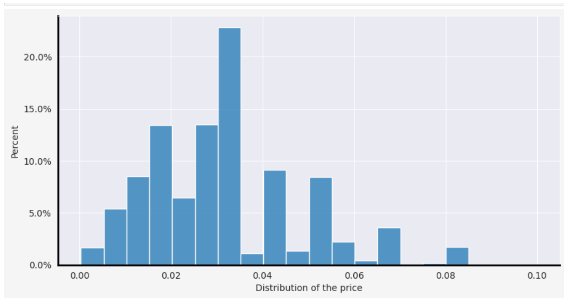


Figure 6: Bar graph of Product Group



(a) Age Distribution



(b) Price Distribution

Figure 7: Age and Price Distribution

and anomalies in the data, which inspired further analysis and improved the decision-making process. The use of visualizations aided in gaining a thorough comprehension of the dataset's attributes, hence improving the overall caliber of subsequent studies and model development procedures.

5.4 Feature Engineering

In the intricate process of developing our recommendation system, we embarked on a multifaceted journey aimed at predicting customer purchases through a harmonious fusion of historical sales data, nuanced insights into customer interactions, and an in-depth analysis of product popularity. The initial phase involves the selection of crucial columns. To find the popularity of N products, the fundamental idea behind this computation is the sales rate quotient, a statistic that was specifically created to capture an article's popularity while taking into consideration the ever-changing nature of product trends. The most recent transaction moment is captured by the variable last_ts, which holds the last timestamp in the dataset. Understanding the time interval between transactions and computing temporal aspects depends heavily on this timestamp. In the given code, the latest point in time inside the dataset is identified by determining the maximum timestamp in the 't_dat' column of the DataFrame, or last_ts.

One of the pivotal parts is sorting the popular products. By sorting the popular products, the model will perform better. At the core of this calculation is the concept of the sales rate quotient, a metric to encapsulate the popularity of an article while accounting for the dynamic nature of product trends over time. The sales rate quotient is computed for each article by evaluating the ratio of its weekly sales count on the last day of the week (count_targ) to its overall weekly sales count (count). This ratio provides a normalized measure of popularity, effectively adjusting for variations in product popularity over time. Selecting the Top N products comes next after all articles' sales rate quotients have been determined. These articles are the most prominent in the present market landscape, as indicated by their greatest sales rate quotients. This carefully selected list of the Top N products is arranged into a DataFrame, which functions as a useful subset that highlights the most well-liked products. The recommendation system not only finds popular articles but also gives priority to those that are in line with the changing and dynamic preferences of customers in the fashion industry. This is made possible by the complex interplay of metrics and considerations that go into calculating the popularity of the Top N products. The timestamps are converted to Unix timestamps and stored in a column timestamp. The final dataset with the user interaction data is stored in an atomic file.[5]. Storing data in atomic files ensures atomicity, consistency, and reliability, preventing data corruption and incomplete records, and supporting error handling in data management processes.

5.5 Model Building

In the model-building phase, the recommendation system is configured with a set of parameters encapsulated in the parameter_dict dictionary as per the official Recbole documentation.[6] These parameters span crucial aspects such as data paths, user and item identification fields, timestamp details, and training specifications, including the selection of loss functions and evaluation metrics to gauge the system's effectiveness. The GRU4Rec model was chosen for its efficacy in sequence-based recommendations and Bert4Rec for its ability to effectively capture extensive sequential relationships in user-item interactions. The models are then instantiated and trained using the specified configuration.Tan et al. (2016) The dataset is split into 30% test and 70% train data using the data_preparation function of recbole library. The user interval is taken as 10 while the item interval is 20. The dataset within the test train is converted to a tensor datatype and stored in it. Throughout the training process, the model's performance is tracked, and the best results on the validation set are recorded. After training, a case study is presented that explores the process of creating the top 12 recommendations for every user. To determine which items each user prefers based on the model's learnings, full_sort_topk function is made use of. Transforming internal item IDs into their external counterparts is a unique aspect of this approach that improves the recommendations' interpretability.

[5]https://recbole.io/docs/user_guide/data/atomic_files.html
[6]https://recbole.io/docs/user_guide/config/parameters_configuration.html

6 Evaluation

In the evaluation phase, a robust analysis of the recommendation system's performance was conducted, focusing on key metrics such as Mean Average Precision (MAP) at 12, precision, and Recall. To fully evaluate the effectiveness of the recommendation system, our code uses the full_sort_topk function from the RecBole utility module during the assessment stage. The code starts by obtaining external user IDs using the id2token method. It then proceeds to cycle over each user, providing a set of top-12 suggestions using the training model and test data. The recommendations are subsequently transformed into external item IDs and structured into a DataFrame called 'result', which includes user IDs and their respective top-k forecasts. The 'result' DataFrame is useful for assessing the system's efficacy by providing insights into the correspondence between recommended items and user interactions during the testing phase. The result dataframe contains customer_id and prediction fields where the prediction has 12 article IDs separated by spaces as shown in below figure.

	customer_id	prediction
0	f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	900267001 765743007 706016002 900279001 905957...
1	9d5b2ad4d39a93106a9995b6d20292c168e178c133ca45...	905957001 933989001 867969003 885951001 863595...
2	03c3771e117f921c472552497e243607243fd7de0c57fd...	905957007 923037001 900267001 900279001 867969...
3	7b07c1844804a1b435f3a613560bb1c0132c63455a9b89...	863583002 923758001 905957001 914441004 863595...
4	e266bff45e8d2418205e8bd42b64bdd4a03c7b1ef72b72...	751471043 924243001 898713001 898694001 850917...

Figure 8: Results Dataframe

6.1 GRU4Rec Algorithm on User Interaction Data:

The primary investigation was around the use of the GRU4Rec algorithm on user interaction data, highlighting its effectiveness in revealing sequential patterns in user-item interactions. The model was trained to anticipate users' forthcoming interactions by utilizing the temporal dependencies included in the data, which are derived from their past behavior. Significantly, limitations on the frequencies of user and item interactions were set using the parameters 10 and 20 on 'user_inter_num_interval' and 'item_inter_num_interval' correspondingly. To calculate the loss during training, the 'loss_type' parameter was set to 'CE' (Cross-Entropy). Negative sampling was not used ('neg_sampling' set to None), and the 'train_neg_sample_args' parameter was set to null. The model training was conducted using an epoch count of 5. The sample recommendation for one user ID is shown in the below image. The GRU4Rec algorithm demonstrated performance with a MAP@12 score of 0.0117, in suggesting relevant items within the top 12 predictions. Additionally, the computed average precision and average recall scores were approximately 0.0117 and 0.0165.

customer_id	prediction	prod_name
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	889652001	ED Space Jeggings
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	900279001	HM+ Tina dress
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	928171001	MC Cypress dress
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	934793001	Aussie Tim Tam AO jacq Sweat
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	918200001	ED Lucien lace blouse
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	765743007	SPORT Sulima jacket
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	900267001	ED Georgia leggings
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	740519006	SORRENTO RW trs
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	858147005	Venice SP bralette
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	902507001	Banana split sweatshirt
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	858460002	Ocean skirt(1)
f08a89f42c35c1154b89d5c0e20428a6233e1881071ad2...	925372001	LOGG Spinel

Figure 9: Top 12 Recommendations for a user

6.2 BERT4Rec Algorithm on User Interaction Data:

The second case study was the implementation of the BERT4Rec algorithm on user interaction data. This was done by utilizing a specific set of parameters that were modified to suit the algorithm's unique properties. The parameters used here are similar to those used in the GRU4Rec case study. The BERT4Rec model achieved a slightly lower MAP@12 score of 0.0102 and the average precision and average recall scores for BERT4Rec were approximately 0.0102 and 0.0144 respectively.

Table 2: Evaluation Metrics for Different Models

Model	**MAP@12**	**Precision**	**Recall**
GRU4Rec	0.0117	0.0117	0.0165
BERT4Rec	0.0102	0.0102	0.0144
GRU4Rec with Additional Attributes	0.0078	0.0078	0.0106
BERT4Rec with Additional Attributes	0.0075	0.0075	0.0103

6.3 GRU4Rec Algorithm with Additional Attributes:

In this experiment where the GRU4Rec algorithm was improved with more attributes, the model's parameter configuration grew to include a wider range of features related to items, such as the item's code, type, name, appearance, colour group code, perceived colour information, department details, index information, and specifics about the garment group. The enhanced set of features was designed to provide the model with a more refined comprehension of item attributes, hence enabling better-personalised recommendations. Nevertheless, the evaluation measures demonstrated a minor decline in the model's performance when compared to the baseline GRU4Rec model. This was evident from the MAP@12 score of 0.0078, average precision of 0.0078, and average recall of 0.0106. The incorporation of more features increased the granularity of the model, requiring a meticulous evaluation of the potential trade-offs between improved feature representation and computational performance.

6.4 BERT4Rec Algorithm with Additional Attributes:

The BERT4Rec method was improved in this experimental iteration by integrating supplementary attributes into its parameter setting. The expanded range of functionalities encompassed a variety of item-specific particulars, including item code, category, designation, visual representation, colour grouping code, perceived colour data, department specifications, index information, and garment group details. The purpose of this augmentation was to enhance the model's comprehension of item attributes, thereby enabling more refined personalised recommendations. Nevertheless, the evaluation metrics indicated a slight decrease in the model's performance in comparison to the baseline BERT4Rec model. The MAP@12 score, average precision, and average recall were 0.0075, 0.0075, and 0.0103, respectively. The inclusion of supplementary features increased the complexity of the model, requiring a meticulous assessment of the trade-off between improved feature representation and computational performance.

6.5 Discussion

The review of three separate recommendation system studies, utilizing GRU4Rec, BERT4Rec, and a GRU4Rec with additional features, yielded useful insights into their performances. GRU4Rec exhibited strong sequential pattern recognition, yielding competitive metrics (MAP@12: 0.0117, Precision: 0.0117, Recall: 0.0165). BERT4Rec leveraged contextual information effectively, showcasing comparable scores (MAP@12: 0.0102, Precision: 0.0102, Recall: 0.0144). Although adding a variety of features to GRU4Rec resulted in a modest decrease in performance (MAP@12: 0.0078, Precision: 0.0078, Recall: 0.0106), it highlighted the complex link that exists between interpretability and feature richness. Analysing the tests critically revealed areas that may be improved. By using a multistage recommender system Higley et al. (2022), targeted feature engineering and ensembling can improve the recommendation precision.

7 Conclusion and Future Work

The models' performance, which was characterised by relatively low scores, prompted the investigation of strategic pathways to raise the calibre of recommendations. As per Li et al. (2023) the sequential recommendation models such as GRU4Rec and Bert4Rec perform better, which reflects the same in our research, and Gru4rec is much more efficient in terms of execution, while Bert4rec took doube the time to execute one epoch. By including a multi-stage recommender pipeline, motivated by results from the fashion domain literature Higley et al. (2022) precision of the recommendation can be improved. The pipeline needs to apply a variety of tactics, drawing from the study's findings about the transient nature of fashion trends and the need to take into account recent user transactions. In this research, only the top 100 popular products were filtered, similarly implementing Repurchase-TopN, utilising graph embeddings for TopN recommendations, and recommending items of the same products the recommendation can be improved.

References

Higley, K., Oldridge, E., Ak, R., Rabhi, S. and de Souza Pereira Moreira, G. (2022). Building and deploying a multi-stage recommender system with merlin, *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 632–635.

Hong, W., Li, L. and Li, T. (2012). Product recommendation with temporal dynamics, *Expert Systems with Applications* **39**(16): 12398–12406.

Huang, C.-L., Yeh, P.-H., Lin, C.-W. and Wu, D.-C. (2014). Utilizing user tag-based interests in recommender systems for social resource sharing websites, *Knowledge-Based Systems* **56**: 86–96.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree, *Advances in Neural Information Processing Systems 30*.

Ke, G., Xu, Z., Zhang, J., Bian, J. and Liu, T.-Y. (2019). Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 384–394.

Koren, Y. (2009). Collaborative filtering with temporal dynamics, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 447–456.

Larrain, S., Trattner, C., Parra, D., Graells-Garrido, E. and Nørvåg, K. (2015). Good times bad times: A study on recency effects in collaborative filtering for social tagging, *Proceedings of the 9th ACM Conference on Recommender Systems*.

Lathia, N., Hailes, S., Capra, L. and Amatriain, X. (2010). Temporal diversity in recommender systems, pp. 210–217.

Li, M., Vardasbi, A., Yates, A. and de Rijke, M. (2023). Repetition and exploration in sequential recommendation.

Liu, D. C., Rogers, S., Shiau, R., Kislyuk, D., Ma, K. C., Zhong, Z., Liu, J. and Jing, Y. (2017). Related pins at pinterest: The evolution of a real-world recommender system, pp. 583–592.

Liu, L., Lecue, F., Mehandjiev, N. and Xu, L. (2010). Using context similarity for service recommendation, pp. 277–284.

Liu, Y., Gao, Y., Feng, S. and Li, Z. (2017). Weather-to-garment: Weather-oriented clothing recommendation, *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, pp. 181–186.

Medvedev, I., Wu, H. and Gordon, T. (2019). Powered by ai: Instagram's explore recommender system, Retrieved June 17, 2022.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W. and Jiang, P. (2019). Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450.

Tan, Y. K., Xu, X. and Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations, *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 17–22.

Xiang, J., Zhang, N., Pan, R. and Gao, W. (2019). Fabric image retrieval system using hierarchical search based on deep convolutional neural network, *IEEE Access* **7**: 35405–35417.

Xu, L., Tian, Z., Zhang, G., Wang, L., Zhang, J., Zheng, B., Li, Y., Hou, Y., Pan, X., Chen, Y. et al. (2022). Recent advances in recbole: Extensions with more practical considerations, *arXiv preprint arXiv:2211.15148* .

Zhao, W. X., Hou, Y., Pan, X., Yang, C., Zhang, Z., Lin, Z., Zhang, J., Bian, S., Tang, J., Sun, W. et al. (2022). Recbole 2.0: Towards a more up-to-date recommendation library, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4722–4726.

Zhao, W. X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., Li, K., Lu, Y., Wang, H., Tian, C. et al. (2021). Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms, pp. 4653–4664.