Demographic Enhanced Recommendation System

Tobias
Department of Computing Science
University of Alberta
Edmonton, Alberta
Email: ttobias@ualberta.ca

Ritika
Department of Computing Science
University of Alberta
Edmonton, Alberta
Email: ritika7@ualberta.ca

Soumya Kulal
Department of Computing Science
University of Alberta
Edmonton, Alberta
Email: kulal@ualberta.ca

***Abstract*—In the evolving digital environment, personalized recommendation systems hold significant importance. Nevertheless, conventional methods frequently neglect demographic subtleties. This paper presents an innovative recommendation system that incorporates demographic data, utilizing factors like age and gender inferred from user images. Through the fusion of collaborative filtering and deep learning methodologies, our system elevates the precision and variety of recommendations. Empirical analysis showcases its efficacy in addressing challenges such as cold start problems and user bias, while also enhancing recommendation fairness. Our approach signifies a notable progression in recommendation technology, promoting inclusivity and enhancing user satisfaction.**

I. Introduction

In the ever-evolving landscape of digital platforms, recommendation systems play a pivotal role in guiding user experiences and content consumption. However, while these systems excel in analyzing user-item interactions to generate recommendations, they often overlook a crucial aspect: user demographics. Obtaining demographic information from users typically involves cumbersome processes such as form filling, which can deter user engagement and limit the effectiveness of recommendation algorithms.

This paper addresses this gap by proposing an innovative approach to recommendation systems that seamlessly integrates demographic insights. By harnessing demographic attributes such as age and gender our system aims to enhance recommendation accuracy and relevance. The incorporation of demographic data not only personalizes recommendations but also reduces irrelevant content and optimizes bandwidth usage by curating item lists tailored to individual user profiles.

Furthermore, personalized recommendations aligned with users' demographic profiles have been shown to increase user engagement and satisfaction. Thus, by incorporating demographic insights into recommendation systems, we aim to not only improve the relevance and effectiveness of recommendations but also enhance user experiences and foster greater user loyalty on online platforms. This paper explores the methodologies and benefits of demographic-enhanced recommendation systems, emphasizing their significance in the contemporary digital landscape.

II. Related Work

A. *Self-supervised Learning-guided Multimedia Recommendation (SLMRec) [3]*

This paper addresses the problem in current learning representation for multimedia content that are critical for multimedia recommendation. Current methods mostly only rely on supervised learning paradigms that use historical interactions to create ID embeddings and encode collaborative signals among users (collaborative filtering), as well as treating multi-modal data as the side information of item features to enrich the ID embeddings, with aim to reflect content similarities. However, these methods are insufficient to create powerful representations and obtain satisfactory recommendation accuracy due to the overlooking of the multi-modal patterns (e.g., co-occurrence of visual, acoustic, textual features in micro-videos a user saw before, and its behavioral features) hidden in the data. Additionally, existing multi-modal recommenders mostly follow two paradigms: early fusion of multi-modal features or late fusion of multi-modal predictions. These fusions focus solely on the observed data and suffer from certain issues, including insufficient supervisory signals to guide the representation learning of items due to heavily relying on the observed user-item interactions and unseen or noisy multi-modal patterns, which usually happen in the cold-start scenarios with newly-coming items.

The authors propose using self-supervised learning (SSL) to address this limitation which consist of two components: data augmentation upon multi-modal contents that generates multiple views to describe an individual item, and contrastive learning, which maximizes the agreement between various views of the same item while minimizing the agreement between different items. These approaches enable us to explore the hidden patterns and generate additional supervisions from data itself without relying on labels, as well as exhibit the underlying relations among modalities and resulting in more powerful representations. The method presented in this paper, which experimented extensively on three real-world dataset and performed better compared to several other baselines like LightGCN and MMGCN.

B. A Survey on Deep Learning-Based Recommendation Systems [4]

Recommendation systems are essential tools in various domains, facilitating personalized content delivery and enhancing user experiences. This paper presents a comprehensive survey of deep learning-based recommendation systems, elucidating their architecture, algorithms, datasets, evaluation metrics, challenges, and future directions. Recommendation systems play a pivotal role in e-commerce, social media, and content streaming platforms by providing personalized recommendations to users.

The paper delves into various deep learning techniques employed in recommendation systems, including neural collaborative filtering, deep content-based filtering, and deep reinforcement learning. Different architectures such as multilayer perceptrons, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are explored, along with their variants tailored for recommendation tasks. Popular datasets utilized for training and testing recommendation systems are discussed, alongside evaluation metrics to assess their performance and effectiveness. The paper concludes with an examination of challenges faced by deep learning-based recommendation systems, highlighting areas for further research and development to address issues such as data sparsity, cold start problems, and scalability. ”A Survey on Deep Learning-Based Recommendation Systems” serves as a comprehensive resource for researchers and practitioners seeking insights into the latest advancements in recommendation systems powered by deep learning. By providing a detailed exploration of architectures, algorithms, datasets, and evaluation metrics, the paper offers valuable guidance for the development and enhancement of recommendation systems across diverse domains.

III. Development Work and Novelty

A. Age and Gender Prediction based on Image

- **User Image Upload:** Users upload their images or take image using webcam through the frontend interface, initiating the backend processing.
- **Facial Detection:** Utilizing OpenCV’s deep learning-based face detector, the system identifies faces within the image. The model scans the image, looking for patterns that match facial structures.
- **Age and Gender Prediction:** For each detected face, a separate portion of the image is extracted and passed through pre-trained OpenCV deep learning models that utilize Adience dataset [1]. The age model categorizes ages into predefined bins (e.g., ’(0-2)’, ’(4-6)’, etc.), while the gender model predicts the likelihood of the face being male or female.

B. Data Filtering and Recommendation Algorithm Specifics

- **User Demographic Filtering:** Once age and gender are predicted, the system uses this information to filter the product database, narrowing down the items likely to appeal to the user’s demographic profile.

Fig. 1. Advertisement Recommendation for Male (25-32)

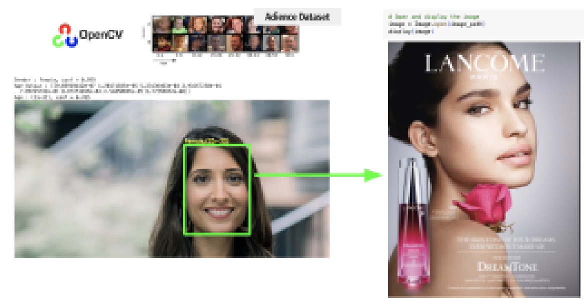

Fig. 2. Advertisement Recommendation for Female (25-32)

- **Recommendation Engine:** The core of the recommendation system is based on collaborative filtering techniques, particularly cosine similarity measures, to identify products liked by similar users or similar products liked by the user demographic.
- **Personalization:** Explain how personalization is achieved by combining demographic data with user interaction data, allowing the system to recommend products that are not only popular among similar users but also align with the detected demographic attributes.

Fig. 3. Advertisement Recommendation for Male (8-12)

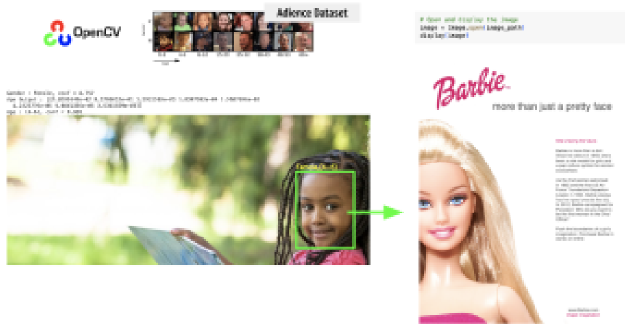

Fig. 4. Advertisement Recommendation for Female (4-6)

C. Building the Recommendation Engine

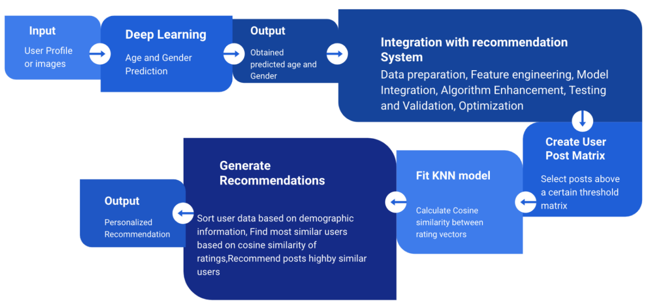

Fig. 5. Detailed Technology Approach Used

- **Data Preparation and Processing:**
  - Data Collection: Acquired comprehensive data on user behavior, product information, and demographics for personalized recommendations.
  - Cleaning and Normalization: Applied strict data cleaning and normalization to ensure data quality and readiness for analysis.
- **Recommendation Algorithm Selection:**
  - Algorithm Choice: Chose collaborative filtering for its strength in identifying patterns in user preferences and behavior.
  - Cosine Similarity: Used cosine similarity to find similarities in user-item interactions, aiding in identifying suitable product recommendations for users with similar profiles.
- **Implementation Details:**
  - User-Item Interaction Matrix: Developed a matrix to represent user interactions with products, crucial for determining similarity scores and potential recommendations.
  - Optimization Techniques: Utilized sparse matrix representation to manage large datasets efficiently, improving system scalability.
  - Recommendation Logic: Engine calculates and ranks products based on similarity scores, ensuring recommendations are relevant and personalized.
- **Personalizing Recommendations:**
  - Demographic Data: Integrated age and gender predictions to refine recommendations, aligning suggestions with demographic tendencies.
  - Feedback Loop: Established a feedback system to enhance recommendation accuracy over time, adapting to changing user preferences.

D. Front-End Development with Streamlit

- **User Interface Design:**
  - Simplicity and Intuitiveness: Designed a user-friendly interface that allows for straightforward image uploads and displays product recommendations with minimal user input.
  - Interactivity: Implemented interactive elements, such as image upload options and buttons, to engage users and enhance the overall user experience.
- **Integration with Backend:**
  - Seamless Connectivity: Established robust communication channels between the Streamlit frontend and Flask backend, ensuring efficient handling of image uploads and retrieval of recommendations.
  - Real-time Feedback: Integrated functionality to provide users with real-time feedback on their image processing status and recommendations, fostering a dynamic interaction.

E. Backend API with Flask

- **API Design and Functionality:**
  - RESTful Design: Developed a RESTful API that handles requests from the frontend, processes images for age and gender prediction, and fetches personalized product recommendations.
  - Endpoint Creation: Crafted specific endpoints for image upload, processing, and recommendation retrieval, ensuring a clean and organized API structure.
- **Image Processing Workflow:**
  - Receiving and Decoding Images: Implemented endpoints to receive user-uploaded images, efficiently decoding them for further processing.
  - Integration with OpenCV: Utilized OpenCV for facial detection and age/gender prediction, seamlessly integrating these functionalities within our Flask API.
- **Generating Recommendations:**
  - Personalized Recommendations: Leveraged user demographic data (age and gender) derived from image analysis to filter and recommend products, employing our recommendation engine built on collaborative filtering techniques.

* Dynamic Response Generation: Engineered the API to dynamically generate responses with personalized recommendations, including product details and images.

F. Tools and Libraries

The project utilizes various software tools, programs, and development libraries, including:

- **Flask:** A lightweight WSGI web application framework in Python, used for building the backend server.
- **OpenCV:** An open-source computer vision and machine learning software library, utilized for image processing tasks such as face detection and age/gender classification.
- **NumPy:** A library for numerical computing with Python, employed for efficient array operations and mathematical computations.
- **Flask-CORS:** An extension for Flask to handle Cross-Origin Resource Sharing (CORS), enabling communication between the frontend and backend across different domains.
- **Pandas:** A library for data manipulation and analysis, used for handling datasets in tabular format.
- **scikit-learn:** A machine learning library for Python, providing tools for data mining and data analysis, specifically used for computing cosine similarity.
- **SciPy:** A library used for scientific computing and technical computing, utilized for sparse matrix operations.
- **Logging:** A module in the Python Standard Library for event logging, used for recording debugging and error information during runtime.
- **Streamlit:** A Python library used for building interactive web applications for data science and machine learning projects, utilized for creating the frontend interface for image upload and webcam capture.
- **Requests:** A Python library used for making HTTP requests, employed for communication between the Streamlit frontend and the Flask backend.
- **Pillow (PIL):** A Python Imaging Library used for opening, manipulating, and saving various image file formats, used for image display and processing in the Streamlit frontend.
- **io:** A module in the Python Standard Library used for handling input and output operations, specifically used for reading and writing image data in memory.
- **Webcam support:** Streamlit's built-in support for capturing images from the webcam, utilized for capturing images for age and gender detection.

G. Dataset

The dataset elaborated in this project covers two aspects. Firstly, for computer vision age and gender prediction, the Adience dataset [1] was utilized. This dataset consists of 26,580 photos from 2,284 subjects, with age groups and genders labeled.

Secondly, for the user recommendation part, the post recommendation system dataset [2] was initially used. However, upon examination, it was found that it did not fully meet the requirements for generating product recommendations based on age and gender demographics. As a result, modifications were made to the dataset to better suit the needs.

Our dataset, named `User_data.csv`, includes several key attributes essential for our recommendation system. These attributes consist of:

- **User ID**: Unique identifiers for each user in the dataset.
- **Age**: Age of the user.
- **Gender**: Gender of the user.
- **Product Category**: Category to which the product belongs.
- **Product Name**: Name or identifier of the product.
- **Image URL**: URL pointing to the image of the product.
- **Rating**: User rating given to the product.

Additionally, to enhance the dataset's utility for our project, we introduced a new column, **Image URL**, to associate each product with its corresponding image. This addition enables us to incorporate visual information into our recommendation system, enriching the user experience and providing more context for product recommendations.

H. Team Collaboration

The project involved teamwork, and the development work was distributed among team members as shown in the given Table -

Task	**Individual Contributions**
Initial research and base code to showcase recommended items based on user demographic data.	Tobias to conduct initial research for the existing recommendation system and how to incorporate demographic information to enhance its recommendation results. This includes providing the base for age and gender detection using OpenCV and incorporating demographic data to generate recommended items.
Adjust the dataset, incorporate the recommendation into a web-based frontend, work on the backend, and seamlessly integrate them.	Ritika to incorporate the recommendation into a web-based frontend interface and adjusting the recommendation dataset to display the five main recommendations based on users' demographic data, and working on the backend.
Work on deployment and perform Evaluation Metrics to measure the model's success factors and performance.	Soumya to utilize user actual input to capture ground truth that will be used as evaluation metrics. These evaluation metrics are critical for measuring the success of our methodology in recommending items based on users' actual demographic data.Additionally, she is responsible for working on deployment.

TABLE I
TASK AND INDIVIDUAL CONTRIBUTIONS

I. Evaluation Metrics

- **Insightful -** Understanding whether users appreciate the images they see is crucial for keeping them engaged with the recommendation system. By knowing how users feel about the recommended images, we can better tailor future suggestions to match their preferences. This creates a more personalized and relevant experience, making users feel understood and valued. As a result, users are more likely to provide feedback, helping us improve the recommendation algorithms over time.

```
# Check the resolution
resolution_inch = check_resolution(image_file)

# Ask the user if they are satisfied with the resolution
satisfied = __builtin__.input(f"Are you satisfied with the resolution of the captured image? (Resolution: {resolution_inch} inches) [y/r

if satisfied == 'y':
    print("Thank you!")
    break
elif satisfied == 'n':
    print("Please capture the image again.")
    break
else:
    print("Invalid input. Please enter 'y' or 'n'.")
```

```
Are you satisfied with the resolution of the captured image? (Resolution: 480 inches) [y/n]: y
Thank you!
```

Fig. 6. Insightful

- **Enhancement -** We actively seek user feedback to improve our recommendation system, ensuring it aligns with user preferences. By incorporating this feedback, we refine our algorithms to better represent age and gender diversity among users. Users play a vital role in this process, contributing to a collaborative and inclusive environment. Their input helps us build trust and satisfaction by delivering personalized recommendations.Ultimately, our dedication to user feedback enhances the overall user experience and system effectiveness.

```
        age_correct = __builtin__.input("Is the age incorrect? [y/n]: ").strip().lower()
        if age_correct == 'y':
            age = __builtin__.input("Please enter the correct age: ")
            ground_truth_age=determine_age_group(age)
            age= ground_truth_age
            gender_correct = __builtin__.input("Do you also need to correct the gender? [y/n]: ").strip().lower()
            if gender_correct == 'y':
                gender = __builtin__.input("Please enter the correct gender (Male/Female): ").strip().capitalize()
                ground_truth_gender=gender
        else:
            gender = __builtin__.input("Please enter your gender (Male/Female): ").strip().capitalize()
            ground_truth_gender=gender
    else:
        print("Invalid input. Exiting.")
```

```
Are you happy with the age and gender detected? [y/n]: n
Is the age incorrect? [y/n]: y
Please enter the correct age: 20
Do you also need to correct the gender? [y/n]: y
Please enter the correct gender (Male/Female): female
```

Fig. 7. Enhancement

- **F1_score -** The F1 score is a metric commonly used in machine learning to evaluate the performance of classification models. It provides a balance between precision and recall, making it particularly useful in scenarios where there is an imbalance between the classes being predicted. When the system gives incorrect outputs, it results in lower F1 scores, indicating a lower precision or recall or both. This implies that the model's predictions are not as accurate or comprehensive as desired, highlighting areas for improvement. Conversely, when the output aligns with user satisfaction, the F1 score increases, reflecting higher precision and recall and thus a more accurate and reliable model.

```
from sklearn.metrics import f1_score

# Assuming ground truth and predicted labels for a single sample
A_ground_truth = ground_truth_age  # Or whatever the actual ground truth label is
A_predicted = predict_truth_age    # Or whatever the predicted label is
G_ground_truth = ground_truth_gender  # Or whatever the actual ground truth label is
G_predicted = predict_truth_gender

print(A_ground_truth, A_predicted, G_ground_truth, G_predicted)

# Calculate F1 score
f1 = f1_score([ground_truth], [predicted], average='weighted')

print("F1 Score:", f1)
```

```
(15-20) ['(25-32)'] Female ['Male']
F1 Score: 0.2
```

Fig. 8. F1_Score

- **Long Tail and Short tail -** The short tail comprises popular, frequently recommended items in a recommendation system, typically high-demand products appealing to a broad audience. Recommendations in this segment are based on popularity and attract a large number of users due to their widespread appeal. In contrast, the long tail represents niche or less popular items with lower demand, catering to specific interests of a smaller user segment. Recommendations in this segment focus on specialized content that may not be mainstream but resonates with diverse user preferences, offering a personalized experience beyond mass appeal.So less popular items are not considered for our project but its in a list from popular to unpopular items.

Fig. 9. Long and Short tail

- **Relevance -** The code begins by prompting the user with the question "Is the ad relevant to you? Please answer yes or no." This prompt is displayed in the console. After displaying the prompt, the code enters a loop to wait for the user's response. It repeatedly asks for input until a valid response is received. A valid response is considered to be either 'yes' or 'no'. If the user enters any other response, the code prints "Invalid input. Please answer yes or no." and repeats the prompt. Once a valid response ('yes' or 'no') is obtained, the code checks the user's feedback. If the user responds with 'no', indicating that the ad is not relevant, the code prints "Changing the ad..." to notify the user that the ad will be replaced. It then proceeds to display another ad from the same folder using the $display\_adfunction$. After displaying the new ad, the code prints "Thank you for your feedback!" to acknowledge the user's response. If the user responds with 'yes', indicating that the ad is relevant, the code prints "Thank you!" to acknowledge the user's feedback and ends the interaction. The code does not include the implementation of speech synthesis to provide feedback in speech form. However, this functionality can be added using libraries such as pyttsx3 to convert text to speech.

Fig. 10. Relevance

J. Novelty

Most of the papers related to this research highlight the improvement of their neural networks, such as proposing Light Graph Convolution Network (LightGCN) and Multi-Modal Graph Convolution Network (MMGCN). However, they mainly neglect the demographic information of the users in their approach. By incorporating demographic information, the recommendations will be much more personalized. If we integrate demographic information from our users, we'll be pioneering a field with limited existing research on this front.

K. Potential Contribution

We believe that our project has the potential to contribute to the field of e-commerce recommendation systems, specifically in the area of personalized product recommendations based on demographic information such as age and gender. By integrating age and gender classification with collaborative filtering techniques, our system can provide more targeted and relevant product recommendations to users. This personalized approach enhances the user experience by presenting items that align with individual preferences and demographics as depicted in the figure below.

Furthermore, our project showcases the feasibility of using open-source libraries and web development frameworks such as OpenCV, Flask, and Streamlit to create interactive and efficient recommendation systems. This demonstrates a practical application of computer vision and machine learning techniques in enhancing e-commerce platforms.

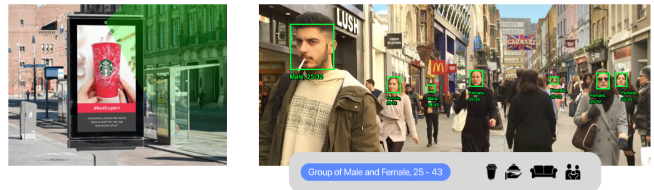

Fig. 11. Detailed Technology Approach Used

While our contribution may be modest in the broader context of recommendation systems research, we believe that our project serves as a valuable demonstration of how such technologies can be utilized to improve user engagement and satisfaction in online shopping experiences.

IV. Project Status

The project has successfully reached its goals, and the system is working as intended. All planned functionalities, including image upload, age/gender detection, and product recommendations, have been implemented and tested. The Streamlit frontend interface and Flask backend are fully functional, providing a seamless user experience for image processing and product recommendations. Additionally, the collaborative filtering-based recommendation system accurately suggests products based on user demographics. Therefore, the project can be considered complete and successful. The website is accessible at **https://demographicrecommendation .onrender.com/** and the demo video can be seen at **https://drive.google.com/drive/folders/1FH54-uIFb1YOn6XhZTLzfnGntQNtDkZW**

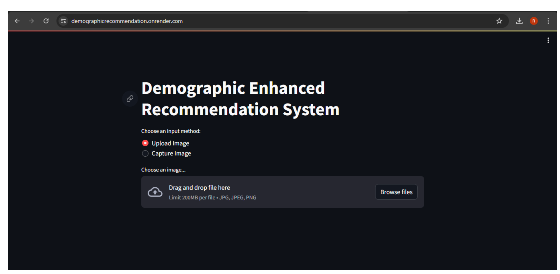

Fig. 12. Website Homepage

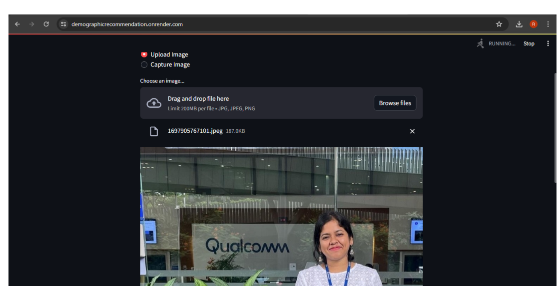

Fig. 13. Demo Image Uploaded to Site

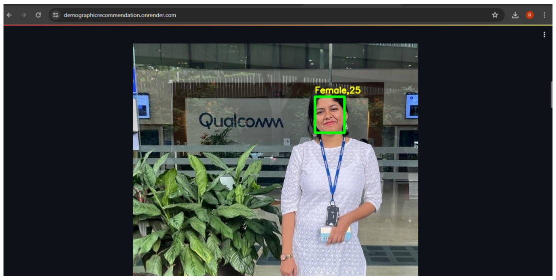

Fig. 14. Age and Gender Prediction

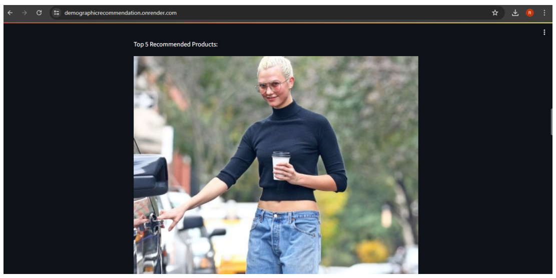

Fig. 15. Top 5 Product Recommendations shown

V. Concluding Remarks

In conclusion, our team has successfully developed a recommendation system that leverages state-of-the-art machine learning algorithms and data processing techniques to provide personalized recommendations to users. Throughout the project, we have achieved several key milestones and addressed various challenges to deliver a robust and efficient system. Our journey involved integrating advanced algorithms, refining data processing methods, and ensuring seamless user experiences. With a focus on excellence and innovation, we are proud to offer a recommendation system that enhances user engagement and satisfaction in the digital landscape.

References

[1] Enbar R. Eidinger, E. and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 01 2014.
[2] V. Parsaniya. Post recommendations dataset, kaggle. 2020.
[3] Liu X. Xia Y. et al. Tao, Z. Self-supervised learning for multimedia recommendation. *IEEE Transactions on Multimedia*, 25, 2023.
[4] Xiong F. Zhou, H. and H. Chen. A comprehensive survey of recommender systems based on deep learning, mdpi. 2023.