成都理工大學中英合作办学
CDUT Sino-British Collaborative Education

OXFORD BROOKES UNIVERSITY

UNDERGRADUATE PROJECT REPORT

Project Title:	**Book Recommendation using Variational Autoencoders**
Surname:	**Wang**
First Name:	**Yutong**
Student Number:	**202018010315**
Supervisor Name:	**Dr. Joojo Walker**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**May 6, 2024**

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honours) Degree Project

Programme Name: **Computer Science**

Module No.: **CHC 6096**

Surname: …………………………………………Wang…………………………………………………………………………

First Name: ……………………………………Yutong……………………………………………………………………

Project Title: ………Book Recommendation using Variational Autoencoders………………………

Student No.: ……………………….202018010315………………………………………………………………...

Supervisor……………...…………….Dr. Joojo Walker……………………………………………………………………

2ND Supervisor (if applicable): **Not Applicable**

Date submitted: **May 6, 2024**

A report submitted as part of the requirements for the degree of BSc (Hons) in Computer Science

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at
https://www.brookes.ac.uk/students/sirt/student-conduct/

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.

Signature … Wang Yutong (Ryan) …… Date …May 6, 2024……………………

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.

Signature … Wang Yutong (Ryan) …… Date ………May 6, 2024…………………

Acknowledgment

The completion of this final report marks the successful conclusion of my undergraduate studies at Oxford Brookes University. These four years have been filled with countless learning experiences and growth. In this limited space, I wish to express my deepest gratitude to everyone who supported and helped me on this journey.

First and foremost, I must extend my thanks to Oxford Brookes University, which not only provided a platform for learning and exchange but also served as fertile ground for my development. I am grateful to every teacher, classmate, and staff member here; this experience has been unforgettable.

Special thanks to my supervisor, Dr. Joojo, who not only ensured the smooth progress of my project with his professional guidance and unwavering support but also inspired me to strive for excellence with his high standards. His distinctively intoned "Hello" always managed to capture our attention immediately. Although we only worked together for one year, the valuable knowledge and experience I gained during this short time will accompany me as I continue to advance in my future studies and career, pursuing high-quality outcomes.

I must also express my deepest gratitude to my family. Their selfless dedication, endless support, and understanding provided a strong backbone for me. Without them, I cannot imagine having completed this educational journey.

Additionally, I want to thank my classmates and friends. On the path of learning, we motivated each other and progressed together; beyond academics, we enjoyed traveling and gaming together. I feel incredibly fortunate to have such a group of friends by my side.

Once again, I thank everyone who offered me help and guidance; each bit of support was an indispensable part of this journey. I wish everyone well and a bright future.

As I face the road ahead, no matter how urgent or challenging, I will remember that "Haste makes waste." I will maintain patience and composure to ensure each step is taken solidly and effectively.

Table of Contents

Abstract

As e-commerce and online services continue to flourish and redefine market dynamics, recommender systems play a crucial role in improving user experience. Especially in the field of books, users are in urgent need of systems that can provide personalized recommendations in the face of massive book choices. The aim of this project is to build a platform that can provide personalized book recommendations using Variational Autoencoders (VAEs) technology. In this study, four book rating datasets are collected and preprocessed, and three variant VAE models, MultVAE, MacridVAE and RecVAE, are implemented and trained to explore their applicability and efficiency for book recommendation. The project comprehensively evaluated the performance of these models using various evaluation metrics such as normalized discount cumulative gain (NDCG), recall and precision. The results show that these models exhibit significant accuracy and efficiency in handling real user rating data. And the performance on unseen users verified the generalization ability of the models. In particular, the MacridVAE model performs well on multiple datasets, demonstrating its advantages in handling sparse data and improving the quality of long list recommendations. Additionally, the study found the importance of picking a good dataset for training the VAE model. GoodReads is demonstrated to be a good dataset to improve the recommendation quality of the recommender system effectively. This project demonstrates the potential application of VAEs in the field of book recommendation.

Keywords: Book recommendation, deep learning, collaborative filtering, variational autoencoders

Abbreviations

Abbreviations	Definition
RS	Recommender System
NDCG	Normalized Discounted Cumulative Gain
DCG	Discounted Cumulative Gain
VAE	Variational Autoencoder
GAN	Generative Adversarial Network
SVD	Singular Value Decomposition
DP	Differential Privacy
DPSGD	Differential Privacy Stochastic Gradient Descent
ELBO	Evidence Lower Bound
CF	Collaborative Filtering
ALS	Alternating Least Squares
ISBN	International Standard Book Number
CPU	Central Processing Unit
GPU	Graphics Processing Unit
Adam	Adaptive Moment Estimation
GDPR	General Data Protection Regulation
ACM	Association for Computing Machinery
BCS	British Computer Society
AutoML	Automated Machine Learning

Glossary

Terminologies	**Definition**
Book Recommendation	A process that uses various algorithms to recommend books to users. It provides personalized recommendations based on users' reading habits, preferences, or the behavior patterns of other users.
Variational Autoencoder	A deep learning based generative model capable of generating new data points by learning potential representations of given data.
Collaborative Filtering	A technique in recommender systems that predicts user preferences for unknown items by analyzing similarities between users or between items.
Generative Adversarial Network	A deep learning framework consisting of a generator and a discriminator, where the generator tries to generate data that is as real as possible, while the discriminator tries to differentiate between real data and generated data.
Sparsity	In recommender systems, data sparsity refers to a large number of unknown entries in the user-item interaction matrix.
Cold Start Problem	A challenge for recommender systems refers to the difficulty of making effective recommendations when there is not enough user data or item data.
ELBO	A core concept in VAE for optimizing the parameters of a model in a variational Bayesian inference framework.

Chapter 1 Introduction

1.1 Background

In the past few decades, with the rise of YouTube, Amazon, Netflix, and many other web services, recommender systems have taken an increasingly important place in our lives. From e-commerce (recommending items to buyers that interest them) to online advertising (recommending content to users that matches their preferences) [1]. The development of e-commerce has facilitated shopping by providing a new way to shop. Rapid technological advancements have led to a substantial growth in the use of e-commerce websites. The percentage of total retail sales that come from e-commerce increased from 5.1% in 2007 to 16.0% in 2019. Books are also one of the commodity categories in shopping platforms, and the modern book industry is characterized by an extremely rich variety, which gives readers numerous choices when purchasing books on online platforms while also making it increasingly challenging to identify and select books that match their preferences and interests [2]. Therefore, an effective personalized book recommendation system (RS) is very important. At the same time, the act of reading is beneficial to both individuals and society, but studies have shown a decline in reading among young people. Recommender systems can also help stop this decline [3].

Book Recommendation Systems aim to personalize book recommendations by analyzing users' interests and behaviors to enhance the reading experience and boost book sales. Existing recommendation methods mainly include collaborative filtering (CF), content-based recommendations (CB), demographic-based recommendations, social recommendation systems, context-aware recommendations, and recommendations using association rules. Each method has its strengths; for example, collaborative filtering can recommend books based on the similarity between users, while content-based recommendations focus on analyzing the content features of books [3].Although these systems have achieved some success in helping users discover new books, they still face many challenges. Main issues include the cold start problem, data sparsity, and the difficulty in scaling up to handle a large number of users and books. For instance, new users or books may be difficult to recommend effectively due to a lack of sufficient historical ratings. These problems indicate possible directions for future research [2].

Variational Autoencoders (VAEs) are considered as potential candidates for application in book recommendation systems. They effectively perform inference by optimizing recognition models using the Stochastic Gradient Variational Bayes (SGVB) estimator. VAEs are widely used in various tasks such as data generation, denoising, data representation, and visualization [4]. Miguel et al. utilized a recurrent VAE architecture to enhance the prediction accuracy of vehicle trajectories, demonstrating how VAEs can leverage temporal data to generate realistic future paths [5]. Pooladzandi et al. explored the application of VAEs in speech re-synthesis, where they introduced novel decoder distributions such as the Gamma decoder to improve the quality of speech reconstruction significantly [6]. Meanwhile, Barreto et al. employed a multi-scale mean and covariance discrepancy VAE model to effectively extract scale-invariant facial features, showcasing VAE's capability in handling diverse image domains for enhanced face recognition accuracy [7]. These instances exemplify the adaptability of VAEs across different fields, leveraging their generative and representational capacities to address specific challenges in each domain.

Krishnathasan's movie recommendation system [8] provides a referable example for book recommendation systems. He proposed a hybrid recommendation system that uses two parallel variational autoencoders (VAE), separately capturing user preferences for movies and movie genres. In the following sections, we will focus on discussing some models based on VAE and their roles in recommendation systems, such as MultVAE [9], MacridVAE [10], and RecVAE [11]. We will also attempt to identify a model suitable for book recommendation systems through experiments.

1.2 Aim

The goal of this project is to establish a user-centric recommendation platform using VAEs. Optimizing the recommendation process using VAE's latent representation capabilities to improve the platform's ability to provide tailored and diverse book recommendations that are closely aligned with users' personal preferences and reading habits. By implementing different VAE-based models on the book rating dataset and conducting a comprehensive evaluation of their performance using evaluation metrics, we ultimately obtained a VAE model suitable for the book recommendation system.

1.3 Objectives

i. Conduct a comprehensive literature review of existing studies related to the book recommendation.

ii. Collect book rating datasets and perform thorough data cleaning and processing.

iii. Implement different VAE models and train, valid, and test these VAE models using different datasets.

iv. Employ a set of metrics including NDCG, recall, and precision at various thresholds to comprehensively evaluate the performance of each VAE model and compare their effectiveness in book recommendation.
v. Analyze and discuss the results and findings obtained from the research to draw meaningful conclusions and implications.

1.4 Project Overview

1.4.1 Scope

This research aims to explore the efficacy of variational autoencoders (VAEs) in enhancing book recommendation systems with a focus on providing highly personalized and diverse book recommendations to users. By testing and evaluating models based on VAE, this project seeks to find the model which is suitable for integration into a recommendation platform that can provide customized book recommendations according to readers' unique reading preferences and habits. This can contribute to improving the reading experience of users.

1.4.2 Audience

The outcomes of this project will benefit several key stakeholders. These stakeholders include:

i. According to individual preferences, **readers (users)** can enhance their book exploration experience by exploring a wider range of literature.
ii. **Publishing professionals (Authors, editors, and publishers)** can use recommender systems to gain a deep understanding of reader preferences and trends. Adjust content creation and marketing strategies based on user preferences to increase book sales and readers' feeling of engagement.
iii. **E-commerce platform** can utilize the recommendation algorithms developed in this project to increase user engagement and conversion rates (the ratio of user purchases to views).
iv. The findings of this project can contribute to **existing research** on personalized book recommendation systems.

Chapter 2 Background Review

A recommender system (RS) predicts the likelihood that users will be interested in items that they may not yet know about. To make recommendations, RS typically require user data and user commentary on these elements. User's feedback on items can be obtained either explicitly or implicitly following a proposal is put forward. The system preserves the feedback in a database and utilizes it for future suggestions [3]. Deploying recommendations in e-commerce has many benefits for both sellers and consumers [3]. However, RS still faces challenges. The most well-known issue is rating scarcity, which can lead to cold starts for users or items.

Kingma and Welling [4] showed how the reparameterization of variational lower bounds produces simple differentiable unbiased estimators of the lower bounds; such SGVB (Stochastic Gradient Variational Bayesian) estimators can be used for efficient approximate a posteriori inference in virtually any model with continuous latent variables and/or parameters and can be optimized directly using standard stochastic gradient ascent techniques. For the case of continuous latent variables in the dataset and at each data point, they propose the auto-encoding VB (AEVB) algorithm. In the AEVB algorithm, optimize the recognition model to make inference and learning particularly efficient by using an SGVB estimator that allows us to perform very efficient approximate a posteriori inference using simple ancestor sampling. The approximate a posteriori inference model, which has been learned, can be applied to a multitude of tasks, such as image recognition, noise reduction, data representation, and visualization [4]. When using neural networks for recognition models, we obtain variational autoencoders.

Additionally, Krishnathasan [8] explored the use of Variable Auto-Encoder in movie recommendation systems. This research presented an innovative approach using two VAEs running in parallel to learn and predict user preferences for movies and genres, respectively. This approach effectively addressed some limitations of traditional collaborative filtering in large sparse datasets. In order to handle large sparse datasets, the paper also highlighted the importance of using neural networks and machine learning methods like matrix decomposition and singular value decomposition (SVD). By compared with the AutoRec model [12], the study demonstrated the Concurrent Hybrid Variational Autoencoders approach in predicting user ratings of movies. This study demonstrates the potential application of VAE in movie recommendation systems and also provides a reference for book recommendation systems.

Collaborative filtering is a prevalent approach employed within recommender systems. It facilitates prediction of user preferences by identifying and utilizing similarities between users and items [3]. The VAE extends to CF with implicit feedback and generalizes the linear latent factor model, allowing us to explore nonlinear probabilistic latent variable models supported by neural networks on large-scale recommendation datasets. The article presented a polynomial conditional likelihood model of neurodegeneration [9]. Recommender systems are often evaluated using ranking-based metrics. Previous research has shown that direct optimization of Top-N ranking loss is challenging, and previous work on this subject has resorted to employing relaxations and approximations [13],[14]. Thus, it is evident that the multinomial likelihoods are well-equipped to simulate implicit feedback data.

Hybrid Variational Auto-Encoder (Hy-VAE) model, also used in collaborative filtering recommender systems. The model combines traditional VAE and deterministic self-encoder (DeepAE), which can better balance the relationship between encoder and decoder during the training process, thus alleviating the a posteriori collapse problem in VAE [15].

Zhang et al. [16] introduced a recommender system model named Adversarial Variational Autoencoder (AVAE) that integrates a VAE and a GAN. This model utilizes users' historical interaction data to predict items they may be interested in, especially for implicit feedback data. AVAE increased the model's ability to handle nonlinear features by applying GAN on the hidden layer of VAE. It used VAE to capture potential features of users and items at first, and then further optimized the representations of these features through GAN. By comparing multiple evaluation metrics including Recall (Recall@K) and Normalized Discount Cumulative Gain (NDCG@K), the study showed that AVAE outperforms several top existing recommendation models on all datasets, demonstrating its strength in handling sparse data and complex user behavior patterns.

Ensemble Variational Autoencoders (EnsVAE) was proposed aiming to provide a new solution for recommender systems. It combines the predictive utility matrices of multiple sub-recommenders into a probability distribution of interest that can be represented by a VAE, and in this way improves the performance of recommender systems [17]. The framework of EnsVAE mainly consists of two innovative sub-recommender systems: a Gate Recurrent Unit-based Matrix Factorization recommender (GRU-MF) and a GloVe content-based filtering recommender (GloVe-CBF).GRU-MF utilizes deep learning techniques to perform matrix decomposition of user-item interactions, which is capable of capturing complex relationships between users and items. On the other hand, GloVe-CBF extracts item features from content information for personalized recommendation. The article demonstrated the significant advantages of EnsVAE over other methods in terms of recommendation performance, especially when dealing with new users and new items. Additionally, EnsVAE is able to select appropriate sub-recommenders according to different application scenarios, exhibiting its scalability and adaptability.

Finally, protecting sensitive user information in recommender systems is as important as maintaining high recommendation accuracy. Fang et al. proposed a recommender system that combines DP and VAE named DP-VAE [18]. The article emphasized that although previous studies have explored the combination of DP and recommender systems, most of them have ignored the model's impact on the accuracy of an unbalanced user population. Especially when a large user population controls model training, applying DP may further exacerbate this unbalanced impact. They computed a user-level prior based on user metadata to optimize the VAE model, and added noise to the computation process to protect the privacy of user metadata. Ultimately, they proposed a novel recommender system using DPSGD for gradient updating. Experimental results showed that the system achieves highly accurate recommendations on multiple datasets while maintaining a reasonable level of privacy protection. This study provides a promising direction for future research in privacy-preserving recommender systems.

The above researches not only demonstrated the potential of VAE in recommender systems, but provided valuable insights on how to improve the performance of recommender systems through deep learning techniques.

Chapter 3 Methodology

This section describes the models we use in recommender systems, methods of data preprocessing, how to implement the models and some improvement strategies. Additionally, the technology and tools used to manage and implement the project are presented.

3.1 Approach

3.1.1 Variational Auto-Encoder

VAE is a powerful generative model that is widely used in many fields such as image generation and denoising, anomaly detection, natural language processing, and recommender systems. There are multiple advantages of applying VAE for collaborative filtering in recommender systems. VAE can map users and items into a continuous latent space, which helps to discover potential user interests and item characteristics. Additionally, using VAE can fill in the gaps in the data by learning the representations in the latent space, which improves the system's ability to handle the missing data. VAE can alleviate the cold-start problem in recommender systems by learning the latent space and combining it with information about other relevant features to provide them with meaningful recommendations.

(High dimensional) variable: X – generated from conditional distribution $P_{\theta *}(x|z)$

Unobserved continuous random variable: Z – generated from prior distribution $P_{\theta *}(z)$

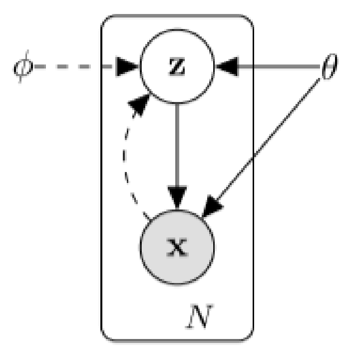

Figure 1 Directed Graphical Model [4]

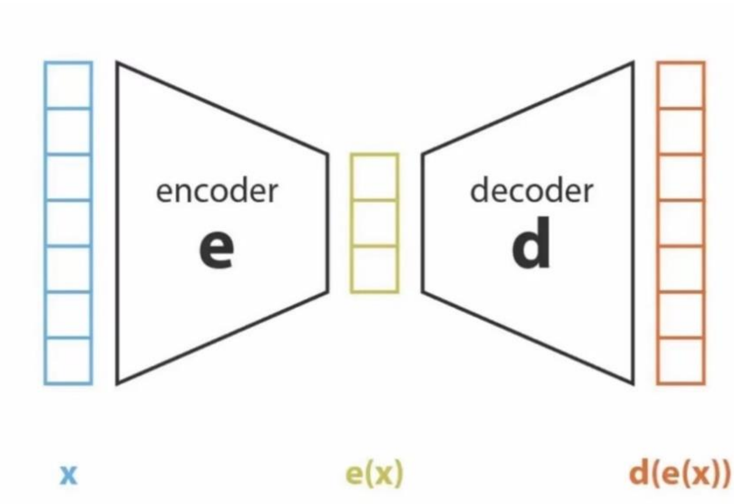

Figure 2 Abstract Figure of VAE

VAE is a network architecture belonging to the probabilistic graphical and variational Bayesian models.

Two neural networks of VAE:

i. Encoder: Encoding and compression of data.
ii. Decoder: Data generation.

However, there is still a prerequisite before decoding: e(x) is a regularized latent space. In order to make latent space regularized, the way to do this is to add some randomness. That means input the same x, output different z every time. The specific mathematical representation of increased randomness is the application of Bayesian Rules.

$$P(z|x) = \frac{P(x|z)P(z)}{P(x)} = \frac{P(x|z)P(z)}{\int P(x|u)P(u)du} \tag{1}$$

Next, calculate loss function to ensure P(z) and P(z|x) are regularized.

$$q_\theta(z)(encoder) \approx p(z|x) \tag{2}$$

$$D(q_\theta(z)||p(z|x))(KL\ Divergence) = E_{z\sim q}\left[\log\frac{q_\theta(z)}{p(z|x)}\right] = E_{z\sim q}[\log q_\theta(z) - \log p(z|x)]$$

$$= E_{z\sim q}[\log q_\theta(z) - \log p(z,x)] + \log p(x)\ \ (evidence\ lower\ bound) \tag{3}$$

$$\log p(x) = E_{z\sim q}[\log p(z,x) - \log q_\theta(z)] + D(q_\theta(z)|(p(z|x))$$

$$\geq E_{z\sim q}[\log p(z,x) - \log q_\theta(z)] \equiv \varsigma_q \tag{4}$$

The objective: minimizing KL divergence leads maximizing ELBO.

Recommender systems based on the VAE model have had numerous improvements and variations to date. the VAE model is not necessarily used in isolation. For example, the Side Information Aided Variational Autoencoder (SI-VAE) [19] uses a user's browsing history and their auxiliary information (e.g., frequency and duration of visits) to predict the websites that may be of interest to a particular user in the future. VAE models have also been used in conjunction with the GAN.

3.1.2 Collaborative Filtering (CF)

Recommender systems are broadly categorized into three different types, as shown in Figure 3.

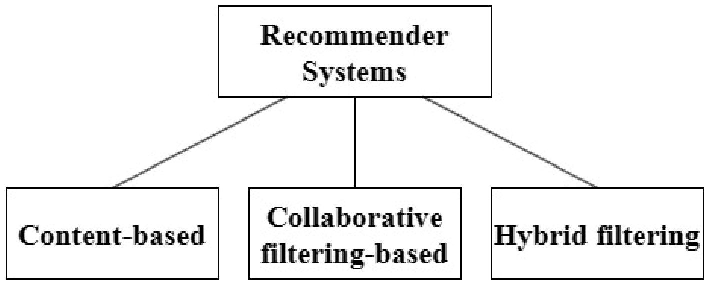

Figure 3 Classification of Recommender Systems

CF is an approach to recommender systems that relies on the similarity between users, and its core mechanism lies in accurately identifying and exploiting the neighboring user groups of the target user, i.e., the "neighborhood". This approach is mainly divided into two ways: memory-based and model-based. CF does not rely on item characteristics, and thus is able to make effective recommendations in the absence of item-specific information. Additionally, it can expand the user's interest range by recommending new items [20].

Model-based approaches utilize data mining and machine learning algorithms to build predictive models. This approach has the characteristic of not relying on a complete dataset when generating recommendations. Instead, it builds and computes models by extracting key features from the dataset, thereby optimizing processing speed and increasing the efficiency of the recommendation system.

Memory-based methods can be divided into two categories: user-based collaborative filtering and item-based collaborative filtering. In the user-based approach, the user rating of a new item is calculated by finding other users from the user neighborhood who have previously rated the same item. If a new item receives a favorable rating from the user neighborhood, the new item is recommended to the user. In the project-based approach, a project neighborhood is created that consists of all similar projects that have been previously rated by users. Then, user ratings for different new items are predicted.

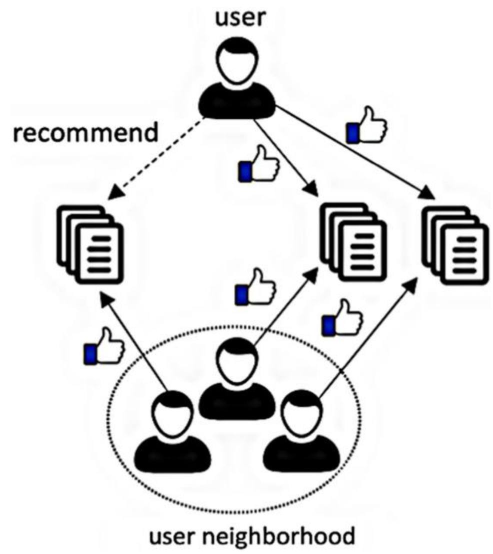

Figure 4 User-based Method [20]

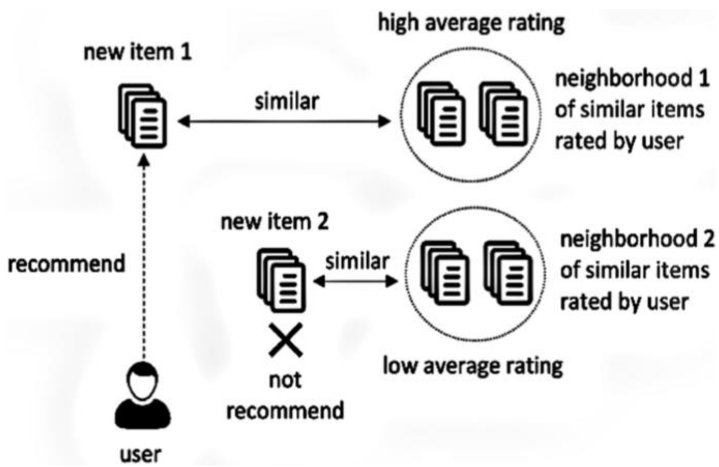

Figure 5 Item-based Method [20]

However, CF systems face some challenges, including the cold-start problem, where new users or new items lack sufficient data to make effective recommendations. Additionally, this raises user privacy concerns due to the need to access and analyze users' historical data [20]. These issues need to be addressed through algorithmic improvements or the introduction of additional privacy protection measures.

3.1.3 VAE with Multinomial Likelihood (MultVAE)

VAE is considered too regularized. MultVAE extends VAE to CF with implicit feedback and introduces a different regularization parameter for learning objectives. [9]

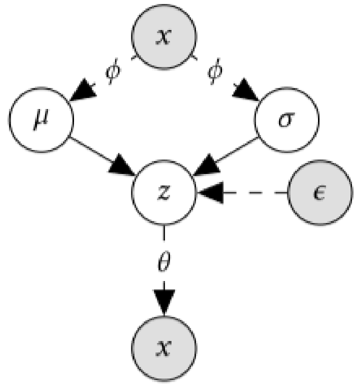

Figure 6 Directed Graphical of MultVAE. Indexing users and items [9]

For each user u, sampling a potential user from the standard Gaussian distribution indicates that z_u. Then, this latent representation is converted to a probability distribution on the items by a non-linear function. The user's item interaction vector x_u is modelled as drawn from a multinomial distribution with this probability distribution.

$$z_u \sim N(0, I_k) \tag{5}$$

$$\pi(z_u) \propto \exp\{f_\theta(z_u)\} \tag{6}$$

$$x_u \sim Mult\big(N_u, \pi(z_u)\big) \tag{7}$$

(where N_u is the total number of interactions of user u and I_K is the unit matrix of the potential space of dimension K)

$$\log p_\theta(x_u|z_u) = \sum_i x_{ui} \log \pi_i(z_u) \tag{8}$$

(Log-likelihood expression for a user given a potential representation)

In the same way as the VAE model, parameters of MultVAE are learned by maximizing ELBO.

However, CF methods often face the problem of data sparsity, which leads to degradation of recommendation performance. So, Noor et al. introduced probabilistic keywords to CF methods, considered book borrowing records and book keyword data, as well as used probabilistic techniques to predict the book borrowing probability under the target user condition to solve the sparsity problem in CF methods [21]. This may provide direction to improve MultVAE.

3.1.4 Disentangled Variational Auto-Encoder (MacridVAE)

MacridVAE, focusing on the underlying decision-making factors behind user behavior, aims to learn disentangled representations of entangled factors. MacridVAE reveals the user's macroscopic intentions through high-level concepts while capturing preferences for specific features at the micro level [10]. For example, the same car is available in different configurations for consumers to choose from. The approach significantly improves the performance, interpretability, and controllability of the recommender system and provides users with a more personalized recommendation experience.

Based on the VAE paradigm, MacridVAE optimizes θ by maximizing $\sum_u \ln p_\theta(x_u)$.

$$\ln p_\theta(x_u) \ \geq \ E_{p_\theta}(C)\left[E_{p_\theta(z_u|x_{u,}C)}\left[\ln p_\theta\left(z_u, C\right)\right] - D_{KL}(p_\theta\left(z_u\middle|x_{u,}C\right)||p_\theta(z_u))\right] \qquad (9)$$

Based on MacridVAE, the authors further considered the semantic information of the items and used visual and categorical signals to initialize the item factors and prototype representations, and created the new model SEM-MacridVAE [22].

Visual semantic information was encoded from the raw images of each item using pre-trained AlexNet, which was matched with item embeddings via principal component analysis (PCA) to initialize item representations and conceptual prototypes.

The classification signal is utilized to achieve better macro-decoupling in a supervised manner by comparing the cross-entropy loss between the true categories of the items and the learned categories.

3.1.5 Recommender VAE (RecVAE)

The RecVAE model is an improved model based on the Mult-VAE model, which uses multinomial distributions as the likelihood function instead of Gaussian and Bernoulli distributions, which are commonly used in VAE models.

RecVAE has a number of innovations. Firstly, the author designed a new encoder network architecture that employs densely connected layers, Swish activation functions and layer normalization techniques to enhance the representation of the model. Secondly, RecVAE introduces a novel composite prior distribution to prevent the "forgetting" problem during training, and trains the model by alternately updating the encoder and decoder, which exploits the idea of denoising the self-encoder.

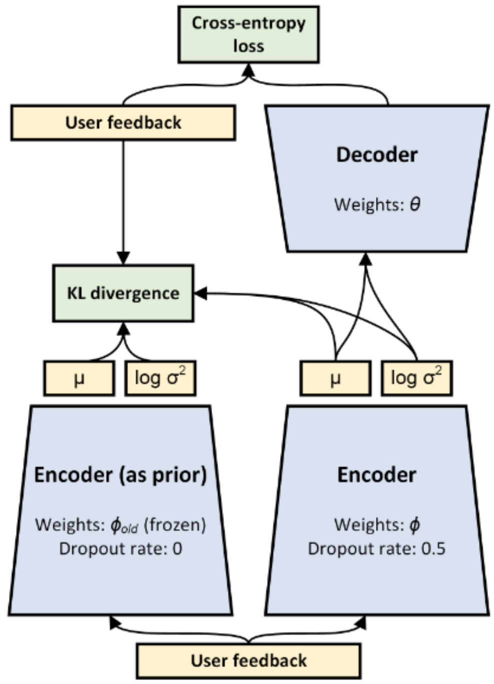

Figure 7 RecVAE Structure [11]

RecVAE is denoising variational autoencoder, it changes ELBO to:

$$E_{q_\phi}(z_u|x'_u)E_q(x'|x)\left[\log p_\theta(x|z) - \beta'(x)KL(q_\varphi(z|x')||p(z|\varphi_{old},x))\right] \tag{10}$$

Next, a method like ALS (a common matrix factorization technique) is utilized to train the model, alternating before user and item embeddings. User embeddings are apportioned by the inference network, while each item embedding is trained separately.

ALGORITHM 1: Proposed training procedure

Data: $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|U|}\}$
Result: $\boldsymbol{\phi}, \boldsymbol{\theta}$
for $n := 1, \ldots, N$ **do**
  for $m := 1, \ldots, M_{\text{enc}}$ **do**
    Sample batch $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_b\} \sim \mathcal{D}$;
    Update $\boldsymbol{\phi}$ based on $\widetilde{\mathcal{L}}$;
  end
  $\boldsymbol{\phi}_{old} := \boldsymbol{\phi}$;
  for $m := 1, \ldots, M_{\text{dec}}$ **do**
    Sample batch $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_b\} \sim \mathcal{D}$;
    Update $\boldsymbol{\theta}$ based on $\widetilde{\mathcal{L}}_{\text{dec}}$;
  end
end

Figure 8 Training Method [11]

3.2 Dataset Collection

In this project, four available book datasets (Book-Crossing, Amazon-Books, Douban, GoodReads) will be used. The above datasets all contain data such as user id, book id (ISBN), ratings, etc., which meets our requirements for the dataset.

3.3 Data Preprocessing

First, we put the user ID, book ID, and rating data for each dataset into a separate csv file. Then, we loaded the data using the pandas library and calculated and printed out the missing values using the isnull() and sum() method. Understanding the distribution of missing values in the data is crucial for effective data cleaning and preprocessing, as the way missing values are handled can directly affect the quality of subsequent data analysis and model training. The describe() function provides a statistical summary of the numeric columns in the dataset, including mean, median, standard deviation, etc. These statistics allow us to quickly grasp the distribution of the data and information on outliers, helping us make more rational data processing decisions and providing preliminary insights for data modeling. We also used the head() method to display the first few rows of the Data Frame. We can quickly review the format and data types of the data table, giving us an intuitive understanding of the structure of the dataset. In summary, these steps provide a solid foundation for data cleaning, exploratory analysis, and further data processing.

The second stage of data processing is a key prerequisite for the implementation of collaborative filtering recommender systems. The core idea is to use similarity to make recommendations. We will use 4 main parameters.

Setting a **threshold** ensures that only positive user preferences are considered. Filtering out data below the rating threshold helps to improve data quality, allowing the model to focus on positive user interactions and enhancing the quality of recommendations. Some datasets have ratings ranging from 0 to 10, so we generally set the threshold to 5. Scores ranging from 0 to 5 we set the threshold to 3 or 3.5.

Using **min items per user** (different users have rated an item at least) and **min users per item** (different items does a user rate at least) perform a second filter. The purpose of such filtering is to effectively filter out extreme and unrepresentative users and items in the dataset, thus improving the overall performance of the recommender system.

After that we will perform data segmentation. Randomly split the users and divide them into training set, validation set and test set users. To evaluate the model on users not available during training, we set the **heldout_users** parameter to ensure that a specific number of users are assigned to the validation set and test set. And used 80% of the ratings in the test set to calculate the user embedding and evaluated the model on the remaining 20% of the ratings.

Considering the different data types in different datasets, we also converted user id and book id to internal numeric identifiers to facilitate model processing.

Finally, the digitized training, validation and test set data are saved as CSV files for subsequent model training and evaluation.

Additionally, we are able to calculate the **sparsity** of the filtered dataset. Sparsity directly reflects the information density of the dataset and is a key concept in recommender systems, especially when dealing with user-item interaction data (e.g., ratings, purchases, clicks, etc.). It is used to describe the ratio between unfilled (i.e., users not interacting with items) and the total number of possible interactions in a dataset.

$$sparsity = \frac{Number\ of\ actual\ ratings}{Number\ of\ users \times\ Number\ of\ items} \tag{11}$$

Meanwhile, sparsity is strongly linked to the cold-start problem encountered by recommender systems, particularly when dealing with new users or new items. High sparsity due to lack of sufficient interaction data. This can lead to trained recommender systems that may struggle to accurately predict user preferences.

```
After filtering, there are 32143133 reading events from 134008 users and 72227 books (sparsity: 0.332%)
```

Figure 9 Example of Filtered Statistic

3.4 Improvement Strategies

3.4.1 Implement the Model

Instead of using script files in the project, we implemented the VAE model code and trained the model in Jupyter Notebook, which brings the following main advantages:

i. Jupyter Notebook provides an interactive development environment that allows you to run code line by line or block by block. This means that you can immediately see the execution results of each piece of code, which is particularly helpful for debugging and understanding the execution flow of code. Additionally, during model development, it is often necessary to experiment with different parameter settings or model structures. notebook makes this process very convenient because you can quickly modify and re-run blocks of code without having to execute the entire script each time.

ii. Jupyter Notebook supports Markdown, allowing you to add rich documentation next to the code, which makes the whole project much more readable and maintainable. Moreover, during the research and development of models, it is often necessary to visualize data and results (e.g. loss curves, generated images, etc.) Notebook supports embedded charts and images, making the presentation of results more intuitive and convenient.

iii. Jupyter Notebook supports many data science and machine learning libraries and tools, making it easy to process data, train models and analyze results in a single environment. Jupyter Notebook can also be used in conjunction with virtual environments (such as conda), helping to create isolated development environments.

3.4.2 Data Processing

The original code used the groupby() function with as index equals False, which caused the Series that was expected to be Boolean-indexed to actually be a Data Frame. This triggered an error that attempted to use two indexes on a single data dimension. By removing as index equals False, the groupby() operation defaults to using the grouping key as the index of the returned result, so that the size() method returns a correct Series, which allows subsequent Boolean indexing operations to be performed correctly. In short, the optimization process fixes the mismatch in the data structure by adjusting the way the groupby() method is used, thus fixing the error in the original code.

3.5 Technology

<table>
<tr><td rowspan="4">Software</td><td>Operating System</td><td>Windows 11</td></tr>
<tr><td>Framework</td><td>Pytorch 1.12.1
Cuda 11.6
Cudatoolkit 11.3.1</td></tr>
<tr><td>Language</td><td>Python 3.9.18</td></tr>
<tr><td>Libraries</td><td>Matplotlib 3.8.1
NumPy 1.26.0
Pandas 2.1.3</td></tr>
<tr><td rowspan="2">Hardware</td><td>CPU</td><td>Intel(R) Core™ i7-10750H CPU @ 2.60GHz</td></tr>
<tr><td>GPU</td><td>NVIDIA GeForce GTX 1660 Ti</td></tr>
</table>

Table 1 Technologies and Tools Used

3.6 Project Version Management

Using GitHub to manage the several versions of project code and reports. See Appendix for Git repository link.

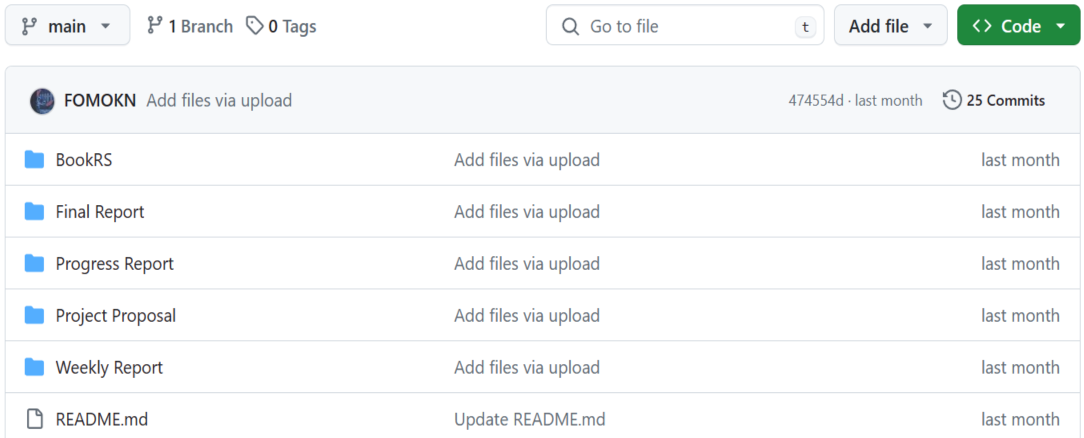

Figure 10 Screenshot of the Git repository

Chapter 4 Implementation and Results

This section will first present the processed dataset and the evaluation metrics used to test the model. Then, it will also show our efforts in seeking models suitable for book recommendation systems, which contains experimental results and visualizations of the results.

4.1 Data Statistic after Processing

In chapter 3, we explained the data processing in detail. In table 2 and table 3, we list the basic statistics of the datasets, which allows us to visualize the differences between the datasets. Note that, although we have different versions of the Douban dataset, they come from the same original dataset. The reason is that we want to test the tolerance of the VAE model to different sparsity.

Dataset	Users	Items	Interactions	Sparsity
GoodReads [23]	134008	72227	32143133	0.332%
Amazon-Books [24]	46276	148785	2453521	0.036%
Douban 1 [25]	22097	29172	855581	0.133%
Douban 2 [25]	152842	78238	2261703	0.019%
Book-Crossing [1]	6053	122740	240664	0.032%

Table 2 Statistics of the Datasets

We attempt to demonstrate that the VAE model may have some mitigating capability for the cold-start problem. VAE is able to generate new data points through continuity and smoothing in the latent space, which can somehow help recommender systems to make recommendations in the absence of user-item interaction data by generating latent user-item preferences. This is why **held out users** need to be set up. In table 3, we used an approximate ratio obviously. The ratio of training set to validation set and test set is 10:1:1.

Dataset	Users	Held out users	Train	Valid	Test
GoodReads	134008	10000	114008	10000	10000
Amazon-Books	46276	3500	39276	3500	3500
Douban 1	22097	1800	18497	1800	1800
Douban 2	152842	10000	132842	10000	10000
Book-Crossing	6053	500	5053	500	500

Table 3 User Statistics of the Datasets

In the field of machine learning and deep learning, dataset splitting is a crucial step. It is useful for evaluating the generalization ability of the model to new data. Meanwhile, it is a key measure against overfitting, ensuring that the model has not simply memorized specific features of the training data, but has learned the broad distributional patterns behind the data. We can optimize model performance without touching the test set used for the final evaluation by hyperparameter tuning on an independent validation set, . In this way, we are able to obtain a fair and accurate model performance metric when using completely unseen test set data for the final evaluation. For VAE models, this approach ensures that the model effectively captures and reproduces the underlying distribution of the data, thereby generating samples that are both novel and representative.

4.2 Evaluation Metrics

NDCG, Recall and Precision are commonly used evaluation metrics in recommender systems and other ranking-related tasks. They each measure different aspects of the model's performance, especially when ranking is taken into account. The use of labels like **@5, @10, @20, @50, @100** is to evaluate the performance of the model at different cut-off levels, reflecting the performance of the model with different Top-N recommendations. For an application that requires extremely high accuracy, performance of @5 may be morimportant than @100.

NDCG is a metric that measures the quality of rankings, especially when the top results are more important than the bottom results. NDCG first calculates the **DCG** and then normalizes it so that NDCG scores for different queries can be compared between 0 and 1.

$$DCG@k = \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2(i+1)} \tag{12}$$

(i is the position of the item in the recommendation list and r_i is the relevance score of the recommended item)

$$IDCG@k = \sum_{i=1}^{|Ideal|} \frac{2^{r_i} - 1}{\log_2(i+1)} \tag{13}$$

(Ideal is the maximum of the top k available items that the user is actually interested in)

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{14}$$

(We added logic in the code to set NDCG@k to 0 if IDCG@k is 0. This handles the case where there aren't any relevant items)

Recall is the proportion of items retrieved from all relevant items. In recommender systems, it reflects the ability of the system to retrieve items that may be of interest to the user. High recall means that more relevant items are captured.

$$Recall@K = \frac{|Recommendation\ items\ \cap Items\ of\ practical\ interest|}{|Items\ of\ practical\ interest|} \tag{15}$$

(Recommended items are the top-N items that the model predicts the user is most likely to be interested in; Items of practical interest are the items that the user actually interacts with; When the denominator is 0, the value of recall will be set as 0)

Precision is the percentage of truly relevant items from the recommended or retrieved items. A high precision means that every item a user views is more likely to be of interest to them.

$$Precision@K = \frac{|Recommendation\ item\ \cap Items\ of\ practical\ interest|}{k} \tag{16}$$

Overall, NDCG, Recall and Precision are used to comprehensively evaluate and compare the performance of recommender systems or ranking models from different perspectives, helping developers to understand the effect of the models in real application scenarios.

4.3 Model Performance in different sparsity

Comparing the performance of the models under different dataset sparsity is an important step in testing the generalization ability and robustness of the models. The key is to ensure that the change in sparsity is the only variable, and the model structure and training process remain consistent. We will use the same model configuration and training procedure to train datasets with different sparsity (Douban 1 and Douban 2 as mentioned earlier).

First, we explicitly defined and quantified sparsity. Before conducting the experiments, it was ensured that the data preprocessing steps were the same so as not to introduce additional variables.

During training, a uniform training strategy is used, including a fixed learning rate, the same batch size and the same number of training cycles. This ensures that the difference in model performance is only caused by the sparsity of the dataset. In order to assess the model performance, the evaluation metrics mentioned above are chosen.

Model	RecVAE
Hidden dim	600
Latent dim	200
Batch size	300
Beta	0.5
Gamma	0.03
Epochs	50
Learning Rate	1e-3

Table 4 Model Parameter Configuration

We visualize the training results as well as the validation results:

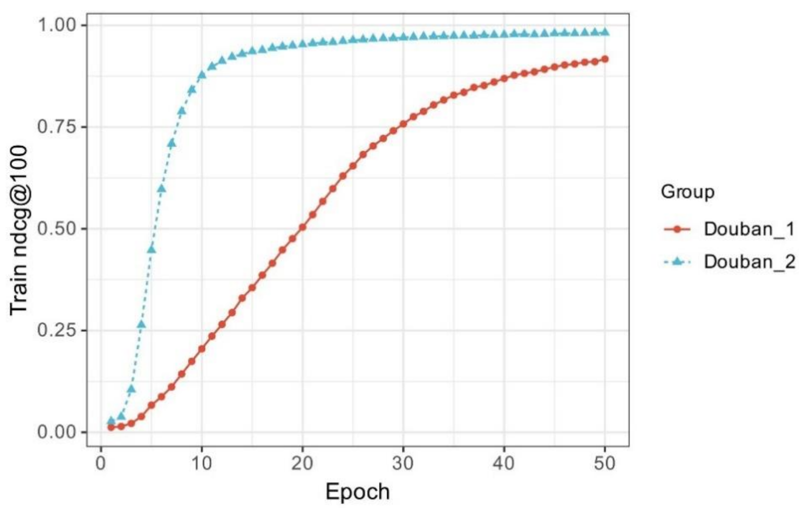

Figure 11 Train Results Comparison

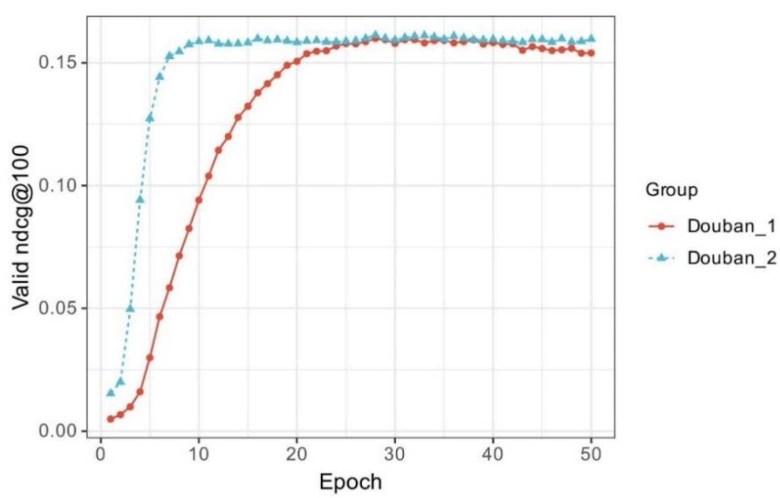

Figure 12 Valid Results Comparison

On the training set, both datasets enabled the models to reach very high NDCG@100 values. On the validation set, both models eventually reached similar levels of NDCG@100, but Douban 2 (the sparser dataset) learnt more slowly in the initial phase. This may be due to the fact that in the sparser dataset, it is harder for the model to learn from fewer user-item interactions. However, this demonstrates that the VAE model is tolerant of data sparsity, especially on the validation set, which shows that the model is able to learn enough information from limited data to make effective predictions.

Subsequently, we performed a more complete test process, which tested the model on users not seen by the model. We used more comprehensive evaluation metrics. The results are presented in the following table.

Metric	**@K**	**Dataset**	
		Douban 1	**Douban 2**
NDCG	5	0.098	**0.110**
	10	0.104	**0.119**
	20	0.118	**0.129**
	50	0.143	0.143
	100	**0.164**	0.153
Recall	5	0.067	**0.122**
	10	0.099	**0.154**
	20	0.138	**0.186**
	50	0.205	**0.241**
	100	0.272	**0.289**
Precision	5	**0.086**	0.047
	10	**0.065**	0.031
	20	**0.046**	0.020
	50	**0.029**	0.011
	100	**0.019**	0.007

Table 5 Evaluation Metrics in Test

According to the results in table 5, the higher precision rate on the test set for Douban 1 suggests that it provides more accurate recommendations when the list of recommendations is shorter. Douban 2 has a higher recall on the test set, possibly because the VAE model is better able to capture patterns in sparse data and thus find more relevant items in a large number of recommendations.

Comparisons of NDCG shows that the model of Douban 2 is more accurate when giving a small number of recommendations, while Douban 1 is more effective when providing a wider range of recommendations.

These results also demonstrate the effectiveness of the VAE model in dealing with data of different sparsity, especially in sparse datasets where it also maintains good performance. Note that both Douban 1 and Douban 2 are derived from the same original dataset, and the reason for their different sparsity is that different parameters (min items per user and min users per item) were set in the data processing. Higher min items per user and min users per item, results in fewer user-item interactions and lower sparsity. Conversely, more user-item interactions and higher sparsity. So, before you train your model, for a dataset such as Douban, you need to decide whether the model should focus more on breadth or precision, or find a balance between two of them.

4.4 Model Comparison

In this section, we perform a critical performance evaluation of three different VAE models (MultVAE, MacridVAE and RecVAE). We trained each model separately using four independent datasets (Book-Crossing, Douban, Amazon-Books and GoodReads) to ensure a thorough evaluation. Our goal is to measure the performance of each model on the test set using comprehensive evaluation metrics that allow for meaningful comparisons in terms of prediction accuracy, generalization ability, and ability to handle data sparsity. This nuanced approach will allow us to identify the models that perform best on unseen data. Note that we ensured that the hyperparameters shared by the models such as learning rate, hidden dim and latent dim were consistent.

Next, we will show the performance data of each model on different test sets in table form. There are significant advantages when presenting model performance data in table form, which not only provides clarity and intuition, making the information readily available at a glance, but supports both horizontal and vertical comparisons. Additionally, the table is able to present a large amount of detailed data completely in a limited space, ensuring that the information is comprehensive and accurate. This structured representation of data also greatly facilitates subsequent data analysis, as data can be exported directly from the table for in-depth statistical analysis or visualization, enhancing the practicality and analytical depth of the report.

Dataset	Model	Top-N Rate					
		Metrics	**@5**	**@10**	**@20**	**@50**	**@100**
GoodReads	MultVAE	NDCG	0.431	0.396	0.345	0.319	0.358
		Recall	0.420	0.375	0.312	0.295	0.371
		Precision	0.420	0.375	0.312	0.218	0.153
	MacridVAE	NDCG	**0.535**	**0.485**	**0.418**	**0.377**	**0.416**
		Recall	**0.517**	**0.454**	**0.374**	**0.343**	**0.421**
		Precision	**0.517**	**0.454**	**0.374**	**0.257**	**0.179**
	RecVAE	NDCG	0.505	0.460	0.397	0.359	0.397
		Recall	0.064	0.111	0.178	0.295	0.400
		Precision	0.490	0.433	0.356	0.245	0.170
Amazon-Books	MultVAE	NDCG	0.060	0.064	0.075	0.097	0.116
		Recall	0.058	0.068	0.094	0.153	0.211
		Precision	0.056	0.046	0.037	0.026	0.019
	MacridVAE	NDCG	**0.115**	**0.117**	**0.128**	**0.153**	**0.174**
		Recall	**0.107**	**0.117**	**0.146**	**0.213**	**0.278**
		Precision	**0.102**	**0.081**	**0.059**	**0.037**	**0.026**
	RecVAE	NDCG	0.087	0.091	0.103	0.126	0.148
		Recall	0.049	0.079	0.117	0.185	0.252
		Precision	0.078	0.063	0.049	0.032	0.023
Douban 1	MultVAE	NDCG	0.076	0.085	0.099	0.118	0.135
		Recall	0.073	0.094	0.127	0.185	0.246
		Precision	0.059	0.044	0.032	0.020	0.013
	MacridVAE	NDCG	**0.139**	**0.148**	**0.165**	**0.188**	**0.205**
		Recall	**0.129**	**0.152**	**0.196**	**0.265**	**0.324**
		Precision	**0.106**	**0.074**	**0.051**	0.029	0.018
	RecVAE	NDCG	0.098	0.104	0.118	0.143	0.163
		Recall	0.067	0.099	0.138	0.205	0.272
		Precision	0.086	0.065	0.046	0.029	**0.019**

Table 6 Evaluation Metrics in Test (1)

Dataset	Model	Top-N Rate					
		Metrics	**@5**	**@10**	**@20**	**@50**	**@100**
Book-Crossing	MultVAE	NDCG	0.030	0.033	0.039	0.046	0.054
		Recall	0.032	0.040	0.059	0.083	0.120
		Precision	0.018	0.012	0.009	0.006	0.004
	MacridVAE	NDCG	**0.031**	**0.033**	**0.037**	**0.047**	**0.052**
		Recall	**0.032**	**0.039**	**0.052**	**0.088**	**0.110**
		Precision	**0.020**	**0.015**	**0.010**	**0.007**	**0.005**
	RecVAE	NDCG	0.024	0.027	0.031	0.039	0.047
		Recall	0.022	0.032	0.046	0.075	0.109
		Precision	0.016	0.011	0.008	0.005	0.004

Table 7 Evaluation Metrics in Test (2)

4.4.1 Performance Comparison

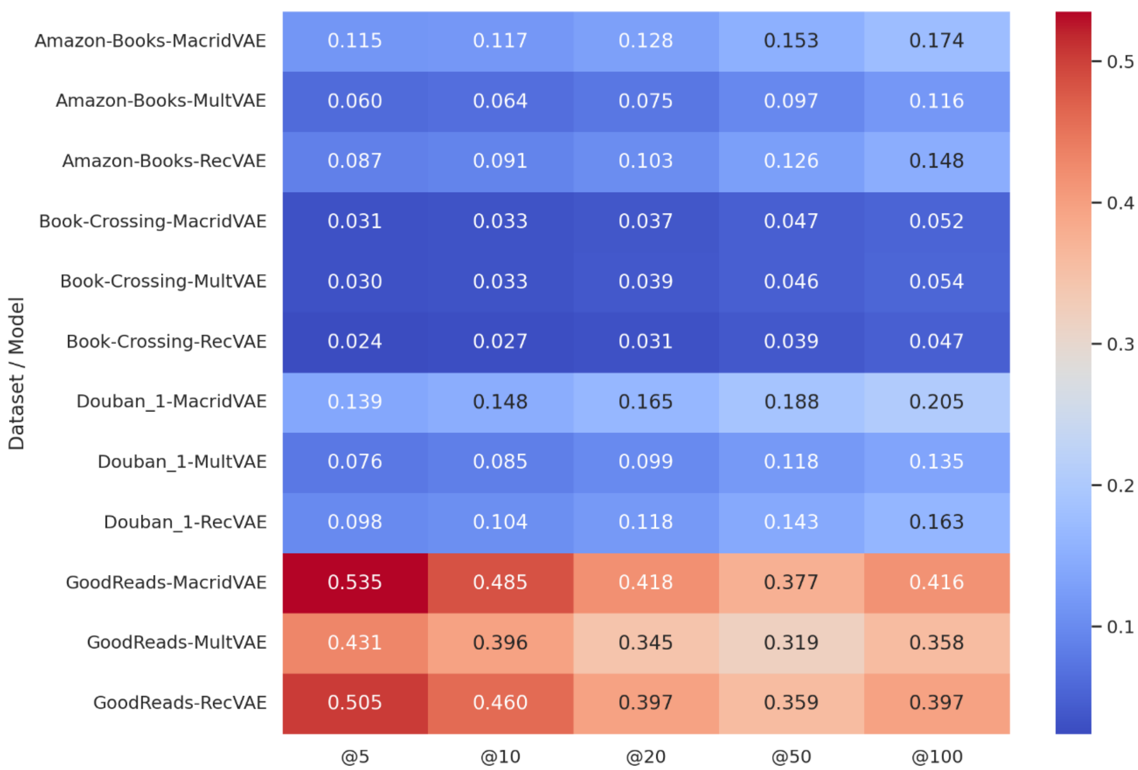

Figure 13 Heatmap of NDCG Performance

The heat map in Figure 13 shows the NDCG performance of different models for each dataset and Top-N value. The colors in the graph represent the high or low NDCG values: the color closer to red, the higher the NDCG value, which indicates better model performance; the color closer to blue, the lower the NDCG value, which indicates worse model performance. This color coding helps to quickly identify the strength of a model performance in a particular setting.

On the GoodReads dataset, MacridVAE significantly outperforms the other two models. Specifically, MacridVAE reaches 0.535 at NDCG@5, which is much higher than MultVAE 0.431 and RecVAE 0.505. This trend is consistent across all top-N recommendations, where MacridVAE also reaches a maximum of 0.416 at NDCG@100, compared to MultVAE 0.358 and RecVAE 0.398. and 0.398 for RecVAE. Similarly, MacridVAE shows similar superiority on Recall and Precision, especially on smaller recommendation lists (@5 and @10), suggesting that it is able to provide more relevant items at the top of the recommendation list.

On the Amazon-Books dataset, MacridVAE reaches 0.278 at NDCG@100, significantly higher than MultVAE 0.116 and RecVAE 0.148. Meanwhile, On the Douban dataset, MacridVAE reaches at 0.324 in NDCG@100 which is ahead of MultVAE 0.205 and RecVAE 0.163. MacridVAE also maintains the lead in the performance of Recall and Precision. This suggests that MacridVAE is better able to maintain the quality of long list recommendations on this dataset.

Likewise, with the values of Precision@K and Recall@K, we can still conclude that MacridVAE is the better model and GoodReads the better dataset. Meanwhile, we calculated the ratio of Precision and Recall for each Top-N and visualized them according to different datasets as shown in Figure 14. Normally, the ratio of Precision to Recall decreases with increasing the value of Top-N. This suggests that although recall increases, the decrease in precision may be due to the system recommending more less relevant books. The performance of RecVAE is consistent with the above mentioned. However, the ratios of Precision and Recall for MacridVAE trained on the GoodReads dataset tend more towards 1, indicating that it is able to maintain a high recall while maintaining a high precision. Different characteristics of the dataset may also affect the model's ratio of Precision to Recall. For instance, it might be more difficult for a model to increase the Recall while maintaining a high Precision for a dataset with the high sparsity. This is why MacridVAE is unable to keep the ratio tending towards 1 with increasing Top-N after being trained on three other datasets. Depending on the needs of different application scenarios (e.g., whether precision or recall is more important), these results can help in selecting suitable models.

The weak performance of all models on the Book-Crossing dataset may be due to the extreme sparsity of the Book-Crossing dataset.

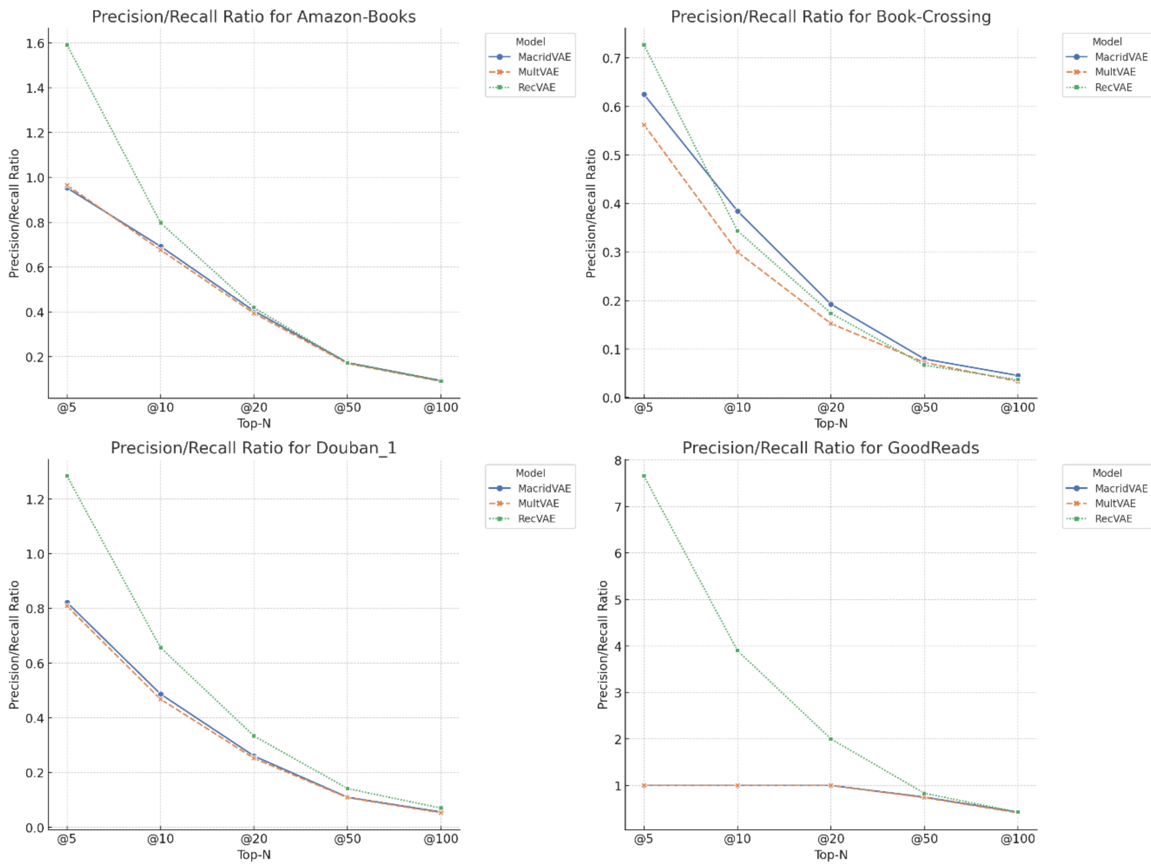

Figure 14 Precision/Recall Ratio for Datasets

4.4.2 Modelling Advantages

MacridVAE leads on most of the metrics, especially on the GoodReads and Douban datasets, suggesting strong overall performance of the model. This is due to the fact that MacridVAE employs learning from decoupled representations, which models the high-level concepts of user intent separately from the low-level concepts that are used in executing the intent. It also introduces micro-decoupled regularizes to independently capture implicit low-level factors such as colour or size, which helps the model to capture the finer nuances of user preferences.

RecVAE typically has the most balanced performance of the three models, especially on the Douban dataset. This may indicate that the RecVAE model structure better balances the various evaluation metrics in some cases. RecVAE introduces a new encoder network architecture and introduces the method of alternately training the encoder and decoder, with this method RecVAE is able to use corrupted inputs when training the encoder and at the same time use clean input data when training the decoder, improving the generalization of the model.

Additionally, both RecVAE and MacridVAE introduce composite prior distributions. The composite prior distribution brings a number of benefits, including better generalization, stability of the training process, richer latent representations, and the ability to model implicit user feedback in more detail. These benefits enable models to capture user preferences more accurately in recommender systems, provide more accurate personalized recommendations, and remain robust in a variable data environment.

In a nutshell, MacridVAE may be more effective in extracting user preferences because of its ability to decouple the user's intention and detail preferences. On the other hand, RecVAE may show better performance on certain datasets or evaluation metrics through its alternating training and encoder architecture optimization.

4.4.3 Problem Analysis

The lower performance of all models on the Book-Crossing dataset, may be due to the fact that these datasets have higher complexity and sparsity, which demand more robustness from the models and require further optimization and tuning.

Based on the previous statistics, it can be obtained that the Book-Crossing dataset has two main characteristics, high sparsity as well as low number of user-item interactions. However, in our previous experiments, the tolerance of the VAE model for high sparsity was demonstrated. Moreover, all 3 VAE models performed well on the GoodReads dataset (a large amount of interactions data). Combined with the above information, this means that a large number of user-object interactions are important for training and validating VAE models. This is due to the fact that the VAE model can generate the user's potential interest items in the latent space with the help of a large amount of interaction data.

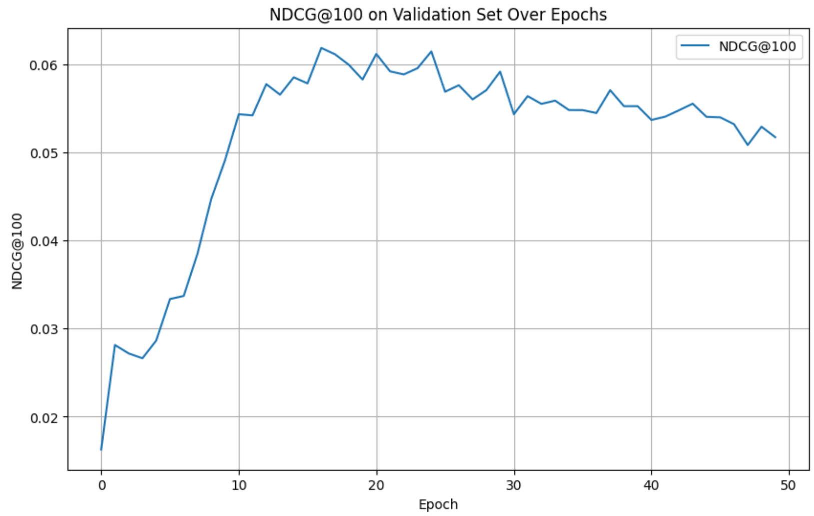

Figure 15 Example of Low Performance (MacridVAE on Book-Crossing)

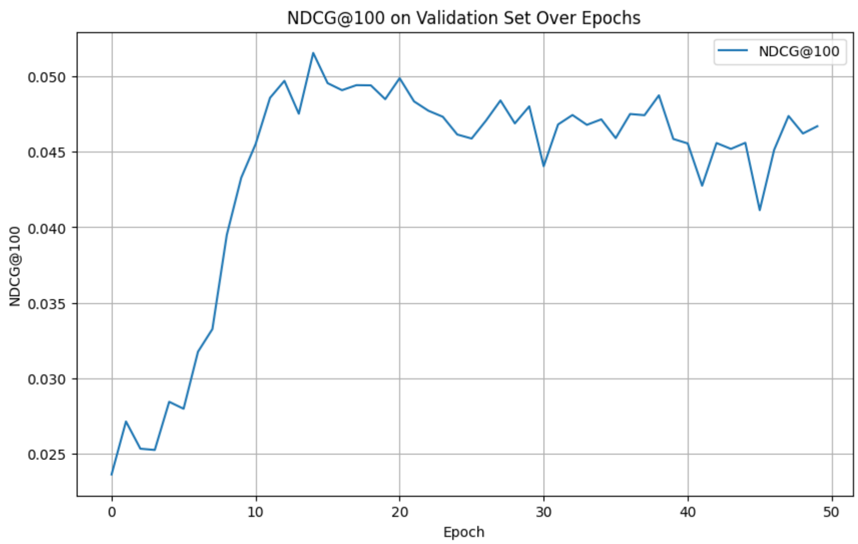

Figure 16 Example of Low Performance (MultVAE on Book-Crossing)

4.4.4 Practical Significance

From a practical application perspective, these test results provide important insights into the selection and application of VAE models.

MacridVAE shows adaptability and robustness in dealing with datasets of varying types and sparsity, especially in scenarios where the quality of long-listed recommendations needs to be maintained, which is particularly important for recommender systems whose goal is to enhance user engagement over the long term. On the other hand, although RecVAE does not perform as well as MacridVAE on some datasets, it demonstrates a balanced performance in specific sparse environments, suggesting that the selection of a model needs to take into account the specific attributes of the data and the specific needs of the recommender system, such as the priority of the recall rate or the precision rate, as well as the environment in which the system will be ultimately used. Therefore, it is crucial to balance the performance of the model with the alignment of business goals in the decision-making process.

When training a VAE model, the selection and use of high-quality datasets is crucial, except for a well-designed network structure and an appropriate training strategy. From our experimental results, the **GoodReads** dataset gained good model performance on each VAE model (better than the other datasets in this experiment). Hence, we propose that the GoodReads dataset is valuable for reference. Its specific statistics can be viewed in Table 2 and Table 3.

4.5 Hyperparameter Tuning

In this section, we will show that debugging hyperparameters makes RecVAE have better model performance in specific task and dataset.

4.5.1 Learning Rate

Learning rate is one of the most important hyperparameters in model training as it controls the magnitude of weight updates. Choosing the right learning rate is crucial to ensure that the model converges efficiently. If the learning rate is too high, the training may become unstable; if the learning rate is too low, the training process may be very slow or even fall into local minima.

In RecVAE, the Adam optimizer is used. The Adam optimizer is a very popular algorithm in the field of machine learning that combines the advantages of two other extensions of stochastic gradient descent, the Adaptive Gradient Algorithm (AdaGrad) and the Root Mean Square Propagation (RMSProp). Adam calculates an adaptive learning rate for each parameter in an attempt to achieve faster and more stable convergence. Adam is efficient in computation, with low memory requirements, and is particularly well suited to problems with large amounts of data and many parameters.

We use the control variable approach in which only one experimental variable is altered (Learning Rate) while the other parameters are held constant to ensure that any observed changes in the results can be directly attributed to changes in the experimental variable. This approach helps to accurately assess the impact of learning rate on model performance.

Table 9 shows the model, dataset, and learning rates that we will use. Table 10 shows the other parameter settings.

Model	Dataset	Learning Rate
RecVAE	GoodReads	1e-3
		5e-4

Table 8 Different Learning Rate Set Up

Hyperparameters	Value
Hidden dim	600
Latent dim	200
Batch size	500
beta	None
gamma	0.005

Table 9 Hyperparameters except Learning Rate

In Figure 17, during the 10 training epochs, the model with a learning rate of 0.001 started with better performance than the model with a learning rate of 0.0005, achieving higher NDCG@100 values. The model with a learning rate of 0.001 improved its performance rapidly in the initial few cycles, but the performance gain diminished over time, showing a convergence trend. On the other hand, the lower learning rate (0.0005), although initially slow to improve, shows a steady improvement in the subsequent cycles and approaches the performance level of the higher learning rate in the 10th epoch. This suggests that although a higher learning rate accelerates the initial learning process, both are likely to achieve similar performance over a longer period of time, and that the lower learning rate provides a smoother and more consistent performance boost.

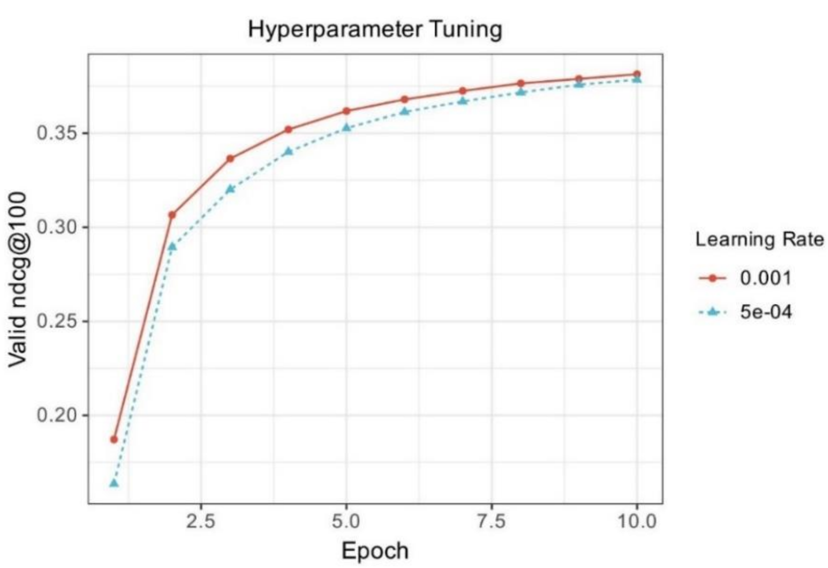

Figure 17 Results of Hyperparameter Tuning

Obviously, the model with a learning rate of 0.001 is better within 10 epochs. However, our training resources (time and computational power) are relatively sufficient and the goal is to achieve more stable convergence, then a lower learning rate (0.0005) may be more appropriate. Although it learns slower in the beginning, it is eventually able to approach the performance of a higher learning rate in a more stable way. Higher learning rates may cause the model to exhibit oscillatory behavior as it approaches the best solution. Consequently, the model might miss its optimal performance. To support our findings, we conducted extensive training and listed the test results in the following table.

Metric	**@K**	**Learning Rate**	
		1e-3	**5e-4**
NDCG	5	0.505	**0.519**
	10	0.460	**0.472**
	20	0.397	**0.404**
	50	0.359	**0.364**
	100	0.397	**0.401**
Recall	5	0.064	**0.065**
	10	0.111	**0.113**
	20	0.178	**0.180**
	50	0.295	**0.297**
	100	0.399	**0.400**
Precision	5	0.490	**0.503**
	10	0.433	**0.444**
	20	0.356	**0.361**
	50	0.245	**0.247**
	100	0.170	**0.171**

Table 10 Test Results in Different Learning Rate

After 50 training epochs, for two models with different learning rates (0.001 and 0.0005), we observed that the model with a learning rate of 0.0005 outperforms the model with a learning rate of 0.001 on the test set for all evaluation metrics (NDCG, Recall and Precision). These results validate that choosing a lower learning rate helps the model to achieve superior long-term performance in a more stable manner when resources allow. Although the model with a lower learning rate learns slower at first, it shows better performance and generalization ability as the number of training epochs increases.

In general, it is important to choose a learning rate that is appropriate for a specific task and dataset. In practice, we need to experiment by adjusting the learning rate several times to find the balance point that achieves the best model performance.

4.5.2 Gamma (γ) in RecVAE

γ is a hyperparameter related to the size of the dataset that controls the KL divergence scaling factor. It determines the scaling factor together with the known number of user interactions with the item. γ can be chosen to regulate the weight of the KL divergence term through cross-validation, enabling the model to adapt to the data volume of different users, and enhancing the stability and accuracy of the model in handling different kinds of user behavior data.

For datasets with dense user-item interactions, a smaller γ may be needed to allow for richer user representations. On the contrary, for sparser datasets, a larger γ may be needed to strengthen the independence of potential features.

To verify the sensitivity of the RecVAE model to γ and that γ plays a role on sparse datasets. We pick the Douban 2 dataset (sparsity 0.019%), which has high sparsity, to try to find the right value of γ on RecVAE. Similarly, we use the control variable method, keeping the other configurations constant and changing the value of γ.

Hyperparameters	Value
Hidden dim	600
Latent dim	200
Batch size	500
Learning rate	1e-3
Beta	None
Epochs	30
Gamma	0.005 0.01 0.02 0.03 0.04 0.05

Table 11 Different gamma in RecVAE

In the end we get the following results as shown in table 12.

Gamma	Valid ndcg@100
0.005	0.1494
0.01	0.1546
0.02	0.1602
0.03	0.1613
0.04	**0.1615**
0.05	0.1603

Table 12 Results of Different gamma in RecVAE

According to the results, when γ is set to 0.005, NDCG@100 is 0.1494, this may indicate that for very sparse datasets, a small value of γ is not enough to provide sufficient regularization, potentially indicating that the potential representation may be too free to capture the sparse structure of the data well. As γ is increased to 0.01 and 0.02, there is a significant improvement in performance, reaching NDCG@100 of 0.1546 and 0.1602, respectively, suggesting that increasing the strength of regularization can help improve the quality of the recommendations. Further increasing γ to 0.03 and 0.04 resulted in a further but reduced increase in performance, NDCG@100 reaches a maximum of 0.1615. However, there was a slight decrease in performance when γ was increased to 0.05, NDCG@100 is 0.1603, which may imply that beyond a certain point the regularization of the model is too strong, resulting in the degrees of freedom of the potential representations being too constrained, thus affecting the performance of the model.

Hence, for the sparsity (0.019%) of the dataset used, a range of γ (e.g., 0.02 to 0.04) seems to provide the best model performance. Furthermore, considering the trend of performance with γ values, the optimal γ may lie around 0.04. From the above experiment, it can be seen that the performance of RecVAE is very sensitive to the value of γ. In a practical application, once the initial range has been set and preliminary experiments have been carried out, the range can be further narrowed and refined based on the results obtained.

Chapter 5 Professional Issues

5.1 Project Management

5.1.1 Activities

To complete each objective, some details will be given in the table.

Objectives	Activities
Learning the VAE Model	Research articles and codes about the model like VAE, Mult-VAE, MacridVAE, and RecVAE. **(Completed)**
Dataset	Download suitable datasets about books and data processing. **(Completed)**
Book Recommendation	Using the book dataset, implement it in VAE models. Train, valid, and test models to get the initial results. **(Completed)**
Comparative Study	Evaluate the performance of different models and analyze their strengths and weaknesses. **(Completed)**
Better Model Performance	Based on RecVAE, it will be tuned and tested with a dataset to demonstrate the performance improvement. **(Completed)**
Write Reports	1) Weekly report **(Completed)** 2) Record test results and improvement measures **(Completed)** 3) Final report **(Completed)**

Table 13 Relevant Activities

5.1.2 Schedule

Using Gantt charts to show the activities and their deadlines.

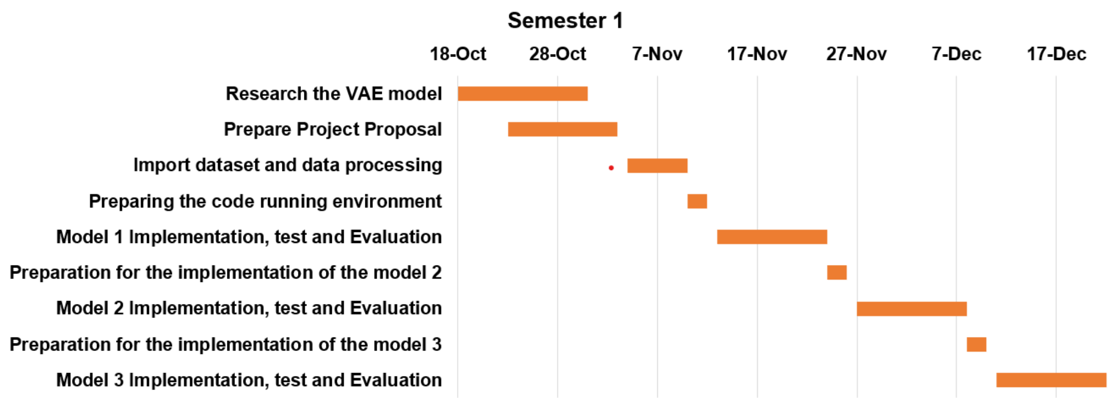

Figure 18: Schedule (Semester 1)

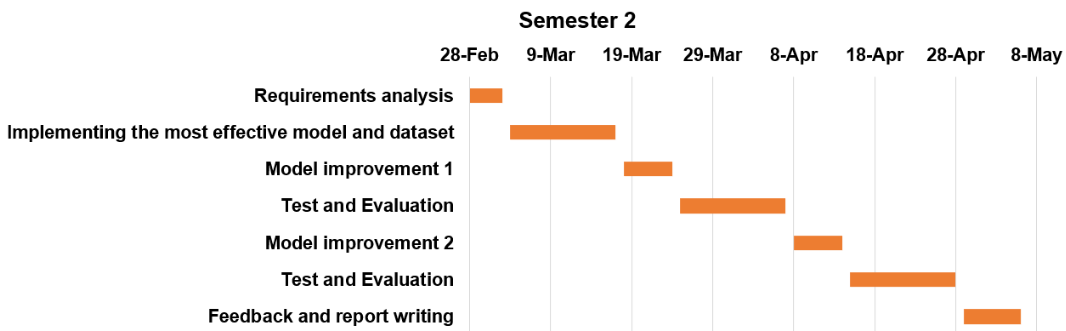

Figure 19: Schedule (Semester 2)

5.1.3 Project Data Management

All references in the project will be saved and managed using Zotero.

Git repositories are used to store files for the entire project and to upload the latest code for the project, which prevents files from being lost. See Appendix for Git repository link.

5.1.4 Project Deliverables

To ensure that all project resources to be submitted for evaluation are clear, they are listed below.

i. Project Proposal
ii. Ethic Forms
iii. Weekly Report
iv. Progress Report
v. Slides
vi. Finial Report
vii. Project Code

5.2 Risk Analysis

Risk identification helps in advance planning and management, as well as in developing appropriate measures to address risks. The method for quantifying risk is likelihood multiplied by severity, where both likelihood and severity are categorized into four levels, from 1 to 4.

5.2.1 Risk Assessment

The minimum and maximum values of risk are 1 and 16 respectively. Thus, we can categorize risk into three levels. 1 to 6 are considered low risk. 7 to 12 are moderate risk. 13 to 16 are high risk. Through ratings, it is possible to determine which risks are more serious or more likely to occur, which facilitates the rational allocation of resources and ensures that the most important issues are prioritized for solution.

In table 14, we will list the possible risks and assess their risk.

Risk ID	Potential Risk	Cause ID	Potential Causes	Severity	Likelihood	Risk
R1.1	Missed deadline	C1.1.1	Illness	1	3	3
		C1.1.3	Poor time management	4	3	12
R2.1	Loss of data	C2.1.1	Poor version control	4	4	16
R2.2	Insufficient data	C2.2.1	Difficulties in data collection	3	2	6
R2.3	Data leak	C2.3.1	Security hole	4	3	12
R3.1	Software bugs	C3.3.1	Non-modular design	1	3	3
		C3.3.2	Poor test plan	4	2	8
R4.1	Unsuitable model selection	C4.1.1	Model is not suitable for the task	4	1	4
R5.1	Model overfitting	C5.1.1	Poor choice of model parameters	4	3	12

Table 14 Risk Analyze

5.2.2 Potential Mitigation Strategy

Rating all risks and proposing mitigation methods facilitates comprehensive and systematic risk management, supports decision-making and optimizes resource utilization. In the following table, we will list the mitigation for every risk in table 14.

Mitigation ID	Potential Causes	Mitigation Strategy
M1.1.1	Illness	Register exceptional circumstances if ill.
M1.1.2	Poor time management	Make a Gantt plan early
M2.1.1	Poor version control	Implement version control strategy at start.
M2.2.1	Difficulties in data collection	Broaden data sources and increase data collection channels
M2.3.1	Security hole	Strengthen security measures and conduct regular system security checks
M3.3.1	Non-modular design	Create highly modular design before implementation
M3.3.2	Poor test plan	Create test plan at start
M4.1.1	Model is not suitable for the task	Review exist literatures to learn the techniques
M5.1.1	Poor choice of model parameters	Perform cross-validation and select appropriate model complexity

Table 15 Mitigation Strategies for Every Risk

5.3 Professional Issues.

5.3.1 Legal Issue

When developing a book recommendation system using VAE, it is important to strictly comply with data protection regulations such as the GDPR to ensure that user privacy is not violated and that data processing is kept legal, fair and transparent during the data collection phase. As our project involves the use of four different datasets, the provenance of the datasets should be clearly stated, ensuring that the datasets do not contain any personally identifiable information such as ID numbers and gender. When using the model for commercial purposes, it is important to explicitly state the specific purpose for which the API is being used, and to ensure that the data processing process is transparent by way of writing the relevant documentation, giving users a clear understanding of how their data will be used and seeking their consent in advance.

5.3.2 Social Issue

In building a book recommendation system, social concerns central on ensuring algorithmic fairness and maintaining user trust. According to the ACM's Code of Professional Conduct, recommendation algorithms need to avoid potential bias and ensure fairness to all user groups, which requires that measures be taken to identify and eliminate possible discrimination during the design and training process. At the same time, according to the BCS code, the operation of recommender systems and the basis for recommendations need to be fully transparent to users, providing interpretability and allowing users to understand and question the system's decisions. At the same time, according to the BCS code, the operation of recommender systems and the basis for recommendations need to be fully transparent to users, providing interpretability and allowing users to understand and question the system's decisions. Besides, the overall impact on society should be considered during the development and application of the model, including the impact on reading habits, cultural diversity, and the possible long-term effects of the recommended content on users' psychology and behaviour. We need to pay sustained attention to these social issues to ensure that advances in technology contribute to the overall well-being of society, rather than becoming a tool for division and inequality.

5.3.3 Ethical Issue

Ethical issues faced when implementing a book recommendation system project include ensuring the privacy of user data, ethical responsibility of algorithms and transparency of conflicts of interest. According to the ACM and BCS codes of ethics, it is important to obtain explicit consent from users for the collection and use of data, to protect their privacy and to prevent data leakage. Developers should be held accountable for decisions made by recommendation algorithms, especially when these recommendations may have a significant impact on users. Additionally, conflicts of interest should be disclosed to ensure objectivity and impartiality of research and development efforts and to prevent improper influence on the results of recommendations driven by personal interests. The entire project team should be guided by the principle of safeguarding the well-being of users and the public interest of society, and promoting ethical, fair and sustainable technological development.

5.3.4 Environmental Issue

When implementing the book recommendation system, it is important to consider environmental issues, which relate to algorithmic efficiency and data central energy consumption. According to ACM's Green Computing Code and BCS's Sustainable IT Strategy, we should minimize the environmental impact of the algorithms, which means that when designing models and selecting computational resources, preference should be given to solutions with high energy efficiency ratios. At the same time, algorithms should be selected or developed to ensure that energy consumption is minimized without sacrificing performance, for example by optimizing the algorithm's data processing efficiency and reducing unnecessary data transfer and storage. Also, we should promote sustainable practices, including the use of renewable energy sources and optimized hardware utilization, in an effort to achieve the coexistence of environmental protection and technological innovation.

Chapter 6 Conclusion

6.1 Reflection and Conclusion

In this study, we have successfully explored the application of three VAE models (MultVAE, MacridVAE, RecVAE) in personalized book recommendation systems using the variational autoencoder (VAE) technique. By carefully preprocessing and analyzing four book rating datasets, we addressed data sparsity, improved recommendations for new users and books, and tackled the cold-start problem. The experimental results show that the MacridVAE model, in particular, achieves the best NDCG@100 of 0.416 on all datasets, which is significantly better than the other models, proving its advantages in long-list recommendation and handling sparse data. GoodReads dataset is the most suitable for training the VAE models. Additionally, in specific cases, we use the control variable method to obtain progress in several evaluation metrics after adjusting the learning rate and γ parameter by the RecVAE model, demonstrating the powerful adjustment and adaptability of the model. These findings validate the usefulness and effectiveness of VAE techniques in book recommendation systems. Through the experiments and analyses in this project, we have gained a deeper understanding of the VAE model's ability to handle complex user behaviours and item attributes in recommender systems, laying a solid foundation for further improving the performance of recommender systems and user satisfaction. However, we also encountered some challenges in our experiments, especially in understanding and interpreting the high-dimensional representation learned by the VAE model. Although the model achieved significant results in some evaluation metrics, we also found fluctuations in the model's performance under different datasets and different parameter settings, suggesting that we need to be more careful and prudent in the process of model design and parameter tuning.

6.2 Future Work

6.2.1 Hyperparameters Tuning

Although tuning model hyperparameters using the control variable method is intuitive and simple, it suffers from limitations such as high computational cost, tendency to fall into local optimal solutions, neglect of parameter interdependencies, and inefficient search in high-dimensional parameter spaces. To effectively overcome these challenges, more advanced hyperparameter optimization strategies such as grid search, stochastic search, Bayesian optimization, etc., or automating the search process using AutoML tools, should perhaps be employed. For example, the RecVAE model suggests specific settings for the β hyperparameter, but actually finding the optimal γ value (for β′ tuning) may still require more widely applicable hyperparameter optimization tools. Additionally, the use of incremental tuning and multi-stage tuning methods, which quickly narrow down the parameter range before fine tuning, can effectively balance search efficiency and model performance. Together, these methods provide an efficient and economical path for deep learning model hyperparameter optimization.

6.2.2 Algorithm Exploration

Methods of combining the VAE with other machine learning techniques should be explored in the future. For example, existing hybrid VAE Models and combining VAE with GAN demonstrate the potential of utilizing complex model structures and generative adversarial techniques to enhance recommendation quality and efficiency. In particular, extending the application of VAE in CF by introducing probabilistic keywords demonstrates a new way to solve the data sparsity problem. The introduction of probabilistic keywords is expected to improve the model's understanding of user behaviours, enabling recommender systems to discover more fine-grained and specific patterns in potential user-item interactions. The successful implementation of these approaches will be an important direction for future research in the field of recommender systems, and also brings new challenges, including the design of models, the optimization of training strategies, and the effective integration of probabilistic keywords.

6.2.3 Data Extension

Expanding and improving the quality of the dataset is key in the further development of recommender systems. In order to better understand user preferences and book characteristics, the system needs to incorporate more diverse user behavioral data including, but not limited to, browsing history, time-series analysis (e.g., analyzing user's reading preferences over time), and contextual information (e.g., location of the user, type of device used, etc.). Moreover, it is crucial to increase the number of interactions and reduce the sparsity of the data, as this can greatly improve the representativeness of the dataset and the predictive accuracy of the system. In terms of book features, the extraction of richer content features, such as topic models and sentiment tendencies, through text analytics allows recommender systems to analyze book content more deeply and thus match user interests more precisely. By integrating these multi-dimensional data, the recommender system can not only enhance the diversity and novelty of the recommendation results, but also ensure the accuracy of the recommendations, and ultimately provide users with a more personalized and satisfying reading experience.

6.2.4 Cross-domain Recommendations

Cross-domain recommender systems aim to utilize the knowledge gained in one domain to enhance the recommendation effect in another domain. For VAE-based models, knowledge migration between different domains can be achieved by sharing latent factor models and mapping functions. For example, users' behaviors and preferences in the movie domain may help predict their interests in the book domain. When implementing cross-domain recommendation, it is important to identify and exploit user characteristics and behavioral patterns common to both domains, while dealing with inter-domain variability. In this way, cross-domain recommender systems can not only improve the quality of recommendations, but also expand the scope of applications to provide richer and more personalized services to users.

References

[1] Möbius, 'Book Recommendation Dataset'. Accessed: Dec. 23, 2023. [Online]. Available: https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

[2] P. Kerudi, T. Srujan, and R. C K, 'BOOK RECOMMENDATION SYSTEM: A SYSTEMATIC REVIEW AND RESEARCH ISSUES', *Huazhong Keji Daxue Xuebao Ziran Kexue BanJournal Huazhong Univ. Sci. Technol. Nat. Sci. Ed.*, vol. 50, p. HST-0621, Jun. 2021.

[3] H. Alharthi, D. Inkpen, and S. Szpakowicz, 'A survey of book recommender systems', *J. Intell. Inf. Syst.*, vol. 51, no. 1, pp. 139–160, Aug. 2018, doi: 10.1007/s10844-017-0489-9.

[4] D. P. Kingma and M. Welling, 'Auto-Encoding Variational Bayes'. arXiv, Dec. 10, 2022. doi: 10.48550/arXiv.1312.6114.

[5] M. Á. D. Miguel, J. M. Armingol, and F. García, 'Vehicles Trajectory Prediction Using Recurrent VAE Network', *IEEE Access*, vol. 10, pp. 32742–32749, 2022, doi: 10.1109/ACCESS.2022.3161661.

[6] O. Pooladzandi, X. Li, Y. Gao, and L. Theverapperuma, 'Exploring the Potential of VAE Decoders for Enhanced Speech Re-Synthesis', in *2023 IEEE Statistical Signal Processing Workshop (SSP)*, Jul. 2023, pp. 359–363. doi: 10.1109/SSP53291.2023.10207969.

[7] F. Barreto, S. Yadav, D. S. Patnaik, and D. J. Sarvaiya, 'MMCD VAE Model for Deep Facial Scale-Invariant-Feature-Translation', in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2021, pp. 526–530. doi: 10.1109/ICICCS51141.2021.9432318.

[8] M. Krishnathasan, 'Movie Recommendation System Using Concurrent Hybrid Variational Autoencoders', in *2021 21st International Conference on Advances in ICT for Emerging Regions (ICter)*, Dec. 2021, pp. 1–5. doi: 10.1109/ICter53630.2021.9774813.

[9] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, 'Variational Autoencoders for Collaborative Filtering', arXiv.org. Accessed: Oct. 24, 2023. [Online]. Available: https://arxiv.org/abs/1802.05814v1

[10] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, 'Learning disentangled representations for recommendation', in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 5711–5722.

[11] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, ‘RecVAE: a New Variational Autoencoder for Top-N Recommendations with Implicit Feedback’, in *Proceedings of the 13th International Conference on Web Search and Data Mining*, Jan. 2020, pp. 528–536. doi: 10.1145/3336191.3371831.

[12] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, ‘AutoRec: Autoencoders Meet Collaborative Filtering’, in *Proceedings of the 24th International Conference on World Wide Web*, in WWW ’15 Companion. New York, NY, USA: Association for Computing Machinery, May 2015, pp. 111–112. doi: 10.1145/2740908.2742726.

[13] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola, ‘COFIRANK Maximum Margin Matrix Factorization for collaborative ranking’, in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, in NIPS’07. Red Hook, NY, USA: Curran Associates Inc., Dec. 2007, pp. 1593–1600.

[14] J. Weston, S. Bengio, and N. Usunier, ‘WSABIE: scaling up to large vocabulary image annotation’, in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, in IJCAI’11. Barcelona, Catalonia, Spain: AAAI Press, Jul. 2011, pp. 2764–2770.

[15] J. Liu, Y. Xiao, K. Zhu, W. Zheng, and C.-H. Hsu, ‘Hybrid Variational Autoencoder for Collaborative Filtering’, in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, May 2022, pp. 251–256. doi: 10.1109/CSCWD54268.2022.9776247.

[16] G. Zhang, Y. Liu, and X. Jin, ‘Adversarial Variational Autoencoder for Top-N Recommender Systems’, in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Nov. 2018, pp. 853–856. doi: 10.1109/ICSESS.2018.8663730.

[17] A. Drif, H. E. Zerrad, and H. Cherifi, ‘EnsVAE: Ensemble Variational Autoencoders for Recommendations’, *IEEE Access*, vol. 8, pp. 188335–188351, 2020, doi: 10.1109/ACCESS.2020.3030693.

[18] L. Fang, B. Du, and C. Wu, ‘Differentially private recommender system with variational autoencoders’, *Knowl.-Based Syst.*, vol. 250, p. 109044, Aug. 2022, doi: 10.1016/j.knosys.2022.109044.

[19] P. Wang, W. Li, Z. Yu, B. Lu, and S. Lu, ‘Website Recommendation with Side Information Aided Variational Autoencoder’, in *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*, Nov. 2020, pp. 1–6. doi: 10.1109/IPCCC50635.2020.9391524.

[20] D. Roy and M. Dutta, ‘A systematic review and research perspective on recommender systems’, *J. Big Data*, vol. 9, no. 1, p. 59, May 2022, doi: 10.1186/s40537-022-00592-5.

[21] N. Ifada, I. Syachrudin, M. K. Sophan, and S. Wahyuni, ‘Enhancing the Performance of Library Book Recommendation System by Employing the Probabilistic-Keyword Model on a Collaborative Filtering Approach’, *Procedia Comput. Sci.*, vol. 157, pp. 345–352, Jan. 2019, doi: 10.1016/j.procs.2019.08.176.

[22] X. Wang, H. Chen, Y. Zhou, J. Ma, and W. Zhu, ‘Disentangled Representation Learning for Recommendation’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 408–424, Jan. 2023, doi: 10.1109/TPAMI.2022.3153112.

[23] M. Wan and J. McAuley, ‘Goodreads Datasets’. Accessed: Mar. 27, 2024. [Online]. Available: https://mengtingwan.github.io/data/goodreads

[24] Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and M. Julian, ‘Amazon Reviews’23’. Accessed: Mar. 27, 2024. [Online]. Available: https://amazon-reviews-2023.github.io/

[25] Cui B., ‘DOUBAN BOOK RATING DATA’. Peking University Open Research Data Platform, Nov. 19, 2015. doi: 10.18170/DVN/LA9GRH.

Appendices

i. Data and Software Repository:

The project's data and software repository can be reached on GitHub in the URL: https://github.com/FOMOKN/Project-RS