Work Project, presented as part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

Improving Personalized Recommendations for Cold-Start Users on the NetEase Cloud Music Platform: A DeCS-Inspired Recommendation System

João António Cordeiro Da Cunha Ribeiro

58876

Work project carried out under the supervision of:

Yufei Shen

17-12-2024

Abstract

The rise of big data and rapid digitization in the digital media industry have made recommendation systems essential for delivering relevant, personalized content to users. In the music streaming sector, platforms like NetEase face the cold-start problem when recommending content to new users with minimal interaction data. To address this, we developed and compared various techniques, including a User Similarity-Based Recommendation Algorithm, a User Preference Elicitation Recommendation Algorithm, a DeCS-Inspired Recommendation Algorithm, and a Discriminative Frequent Itemsets model. Our findings show that the DeCS-Inspired model performs best in data-rich scenarios, while Demographic-Based methods excel in cold-start situations. To optimize performance, we propose a hybrid approach that combines Demographic-Based techniques for cold-starts and transitions to the DeCS-Inspired model as user data grows.

Key Words: Recommendation System, Information System, Cold-Users, Music Streaming Service

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

INDEX

1. Introduction 4

2. Literature Review 5

2.1 Demographic Approaches 6

2.2 Meta-Learning and Reinforcement Learning in Cold-Start Scenarios 6

2.3 Recommendation system techniques and related issues: a survey 7

2.4 Emerging Trends 7

2.6 Limitations of Current Approaches 8

2.7 Summary 8

3. Context and Data 9

3.1 NetEase Research Context 9

3.2 Data Description 9

3.3 Metadata 9

3.4 Cold-Start Problem: Definition and Challenges 10

4. A DeCS-Inspired Recommendation System 11

4.1 Data Preparation and Embedding Initialization 11

4.2 Rating System 12

4.3 User/Item Embeddings 12

4.4 User and Item SI Embeddings 13

4.5 Model structure 15

4.6 Beyond the Model Structure – Intuition, Evaluation and Implementation 17

5. Conclusion and Discussion 22

5.1 Key Findings 22

5.2 Limitations 24

5.3 Research Contributions 24

5.4 Implications for Practice 25

5.5 Future Research Directions 27

6. References 29

7. Appendix 35

1. Introduction

“We engage with automation and machine learning every day in ways we may not realize. Consider the most common online habits: browsing social media, shopping on Amazon, or binge-watching on Netflix. These platforms rely on “recommendation systems”' to analyze tastes and preferences and try to serve up the products, content, or people they believe you want to see.” (What Netflix's Recommendation Systems Can Teach Us About The Computing Challenges Of The Near Future s.d.)

The swift transition to digital systems and services has transformed the way users interact with information across the internet. Information systems now play a critical role in managing vast catalogs of media content, catering to diverse user profiles with varying preferences and behaviors. At the core of these information systems lie recommendation systems, which aim to present users with the most personalized and relevant items. *Figure 1* provides an abstract overview of the key components that constitute a recommendation system.

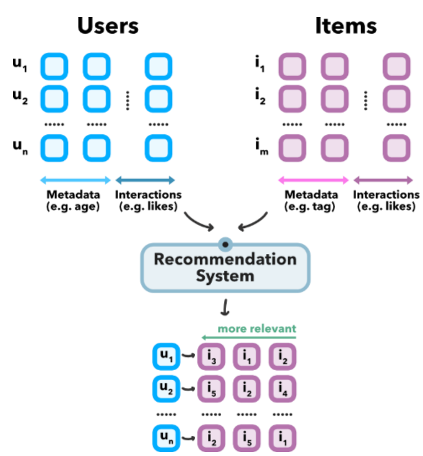


Figure 1 - Recommendation System Key Components Overview

A major challenge recommender systems face is the **Cold Start Problem** (Chadha e Jain 2022), which can be divided into three categories: the **new community problem (**no interactions between any user and items), the **new-item problem** (recently added items), and the **cold-user problem** (brand new users on the platform). Throughout existing solutions, Recommender systems exhibit the "Harry Potter effect", where widely liked items, such as Harry Potter, are frequently recommended, leading to a lack of novelty. Knowledge bases are also leveraged to supplement missing system information, enriching the context and the relevance of content provided to users (See *Figure 2*).

The "cold-start problem" remains a critical bottleneck in the effectiveness of recommendation systems. The consequences are significant: users may exit platforms after unsatisfactory early experiences, and there is little visibility for new talent in an increasingly competitive market. In this work project, we have investigated the challenge of providing accurate/meaningful recommendations to cold users on the NetEase Cloud Music platform, where traditional approaches struggle because of minimal interaction records.

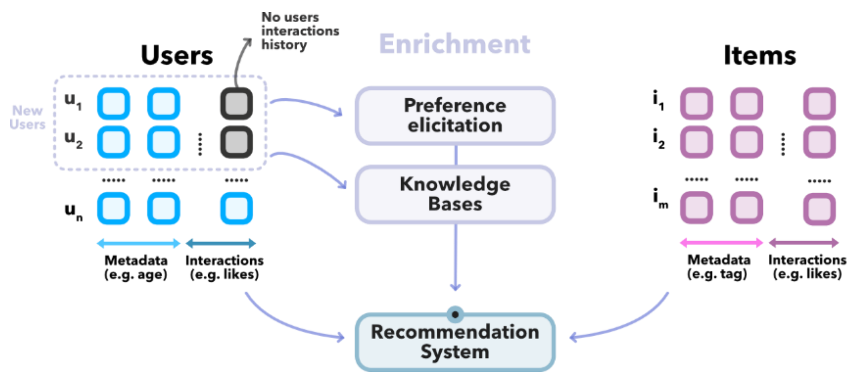


Figure 2 - Cold Users Problem *Abstract Components*

2. Literature Review

The cold-start problem can be a challenge in recommendation systems, particularly when limited user data is available, making it difficult to generate accurate and personalized suggestions. This section discusses key approaches that have been suggested by researchers, for instance, recommendations based on demographic attributes to enhance initial recommendations or combining several approaches to reduce the impact of data sparsity (Lam, et al. 2008), (Lika, Kolomvatsos e Hadjiefthymiades 2014), (Tahmasebi, et al. 2021).

2.1 Demographic Approaches

Demographic data allows systems to group users and items with similar characteristics, improving recommendation accuracy when historical data is lacking (Zhu, et al. 2020). Studies also highlight incorporating Deep Learning using auxiliary information, including - but not limited - to text, user metadata, and other additional information that could generate more accurate predictions (Roy e Dutta 2022), (Steck, et al. 2021).

2.2 Meta-Learning and Reinforcement Learning in Cold-Start Scenarios

Meta-learning has become an essential tool in addressing cold start challenges by allowing recommendation models to dynamically adapt to sparse data environments. (Wang, et al. 2022) showcase how reliable metadata and meta-learning approaches are in an overview that includes the expansion of existing collaborative filtering recommender systems with an enhanced correlation, inspired by (Vozalis e Margaritis 2003). Meanwhile, (Son 2016) conducted a comparative review on the new user cold-start problem in recommender systems. In this review, the model in question was found to perform less efficiently than its peers—NHSM, FARAMS, and HU-FCF—based on error metrics and computational time. On the opposite end, NHSM (Liu, et al. 2014) emerged as a robust solution to the cold-start problem, emphasizing improved accuracy in collaborative filtering. This new approach evaluated the distance between rating pairs, deviation from the median, and uniqueness of rating pairs, while using the mean and variance of user behavior globally—rather than locally—leading to its high accuracy and reduced reliance on metadata.

Reinforcement learning (RL) is also increasingly applied in recommender systems, enabling them to adapt dynamically to continuously evolving information and user preferences (Singh e Singh 2024).

2.3 Recommendation system techniques and related issues: a survey

There are numerous methods and models that, while attempting to solve certain issues in item recommendation, inadvertently exacerbate others. Common issues (Kumar e Thakur 2018) include data sparsity and high dimensionality, which are pervasive across different recommendation problems. Another issue is limited content analysis, where content-based filtering is restricted by relying solely on explicit characteristics of the recommended item.

2.4 Emerging Trends

(Karabila, et al. 2024) proposes an approach that takes on deep advanced learning and self-supervised learning techniques, that promotes the implementation of a recommendation system that is an ensemble of three models: BERT-enhanced deep neural networks, self-supervised learning and collaborative filtering approaches, capturing relationships in textual context, enhancing the user-item relationship understanding. Here, the combination of item and user similarity metrics with other content-based methods is key into providing accurate and well-founded recommendations.

On the other hand, (Hafnar e Demšar 2024) draws a rather different perspective at the possibility of mitigating this issue, through the usage of the GPT 4 LLM (Large Language

Models) for personalized content generation in a context where there is a need of creating tailored game levels within minimal player interaction data. The relevance this paper brings into this problem is of a complete bypass on the user-item data need through the zero shot capabilities of LLMs, that allow for personalized and relevant outputs with minimal inputs.

2.6 Limitations of Current Approaches

Though current approaches have improved greatly concerning the cold-start problems, there are still limitations. Approaches that are mostly based on demographic information and auxiliary data-based approaches may face issues of data availability or insufficient data granularity. Also, the incorporation of deep learning, although appropriate, can increase the intricacy and resource implications rendering them impractical for smaller resource-constrained (Steck, et al. 2021). Thus, in future work, the emphasis may lay on low-cost, effective models that mitigate biases in recommendations without compromising their efficiency (Pandey e Rajpoot 2016), (Silva, et al. 2019).

2.7 Summary

The more recent strategies have shown us that the cold-start problem has several means to be solved. It portrays something that can be summed up as the contemporary existence of highly contextual, state of the art, data broad approaches to reach a highly reasoned recommendation and a zero shot, more aggressive approach that requires very limited amounts of data, while having a personalization aspect to it.

3. Context and Data

3.1 NetEase Research Context

NetEase Cloud Music (NCM), launched in 2013 by NetEase, Inc., has grown into one of China's leading music streaming platforms. The platform stands out for its innovative features, fostering a community-oriented ecosystem on short-form music content and interactive functionalities. This dataset contains an impression-level interface of over 57 million recordings, reflecting users' behavioral data (clicks, likes,shares…) in addition to demographic/content metadata, all collected in November 2019.

3.2 Data Description

The **interaction_data** dataset from NetEase Cloud Music encompasses over 57 million impressions within the "cloud village" tab of NetEase's app, on a sample with over 2 million users where each impression consists of music content card appearance to any user. Additionally, other datasets capture impression-related data, offering insights into users, content creators, and the impressions themselves. These datasets provide an added layer of detail, enriching the understanding of interactions and its stakeholders.

3.3 Metadata

The dataset captures all aforementioned user interactions through a binary field for each action in particular (**isClick**, **isLike**, **isShare**, and **isComment**). Additionally, the view time a field corresponds to the number of seconds a user has spent on a card after clicking it. On the card level, variables such as **mlogId**, **songId**, and **artistId** uniquely identify each card. Other attributes, including **publishTime**, **contentId**, **talkId**, and **cardType** respectively describe the card's publication timeline, content, topics and the format of each card. The dataset also includes demographic, use metrics(such as **age**, **gender** or **province**).

(*Appendix* 1) provides a comprehensive overview of these metadata variables, summarizing the features that were used in our recommender systems.

3.4 Cold-Start Problem: Definition and Challenges

The cold-start problem is a restriction on the effectiveness of recommender systems, especially when new users or new items are involved. For this thesis, we focus on the **New User Problem**, where little to no data exists for a user's engagement history.

Hence, to define cold-start users in our context, we analyzed the available data and established criteria to identify a cold-start user, to then implement our recommender systems:

- **Registration Date:** Users who registered within the last six months.
- **Interaction Count:** Users with four or fewer actual interactions (e.g., clicks, likes…).
- **Activity Level:** Users with an activity level of three or lower on the platform.

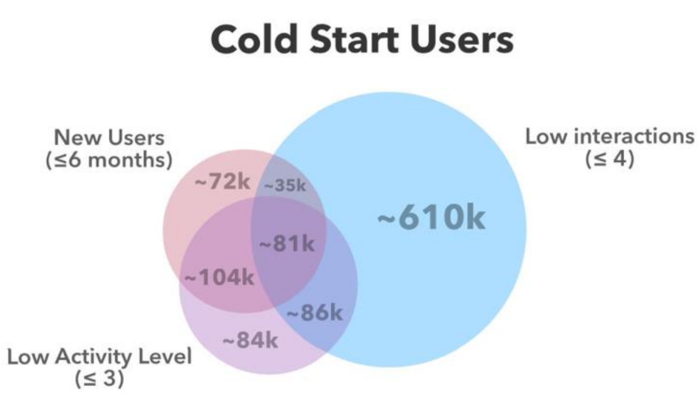


Figure 3 - Cold start users definition

This group accounts for only **3.88%** (*See Figure 3*) of the total user base and includes users who have few records available for making recommendations. The interaction limit was established by analyzing user engagement (*See Appendix* 2). By limiting our focus to users with ≤4 interactions and a platform activity level ≤3, we identified a distinct subset of low-engagement users for whom personalization is most challenging.

4. A DeCS-Inspired Recommendation System

This recommendation system draws from the principles of DeCS (Deep Neural Network Framework for Cold Start) to handle one of the major challenges in recommendation systems, the cold-start problem. The DeCS-inspired architecture aims to mitigate this problem by creating **a robust recommendation framework that leverages both user and item data, enhanced by embedding techniques and side information**. To perform this analysis, a share of 5% from the original dataset was used as the sample, being it derived from a sampling probability on interactions and non-interactions.

4.1 Data Preparation and Embedding Initialization

In any recommendation system, data preparation is foundational and fundamental. Here, the input data consists of user and item-specific features based on side information/metadata available on them, besides a record of user/item interactions that is translated into item and user embeddings based on the proposed rating system. These elements are processed to create dense, low-dimensional representations called embeddings, which simplify and capture the essential relationships within the data.

4.2 Rating System

A rating formula was designed to address this unary, binary (isClick, isComment, isIntoPersonalHomepage, isShare, isViewComment, isLike') **and non-negative** (mlogViewTime) **data in** a way that the rating portrays all the nuances of these features in a **fair, understandable and concise manner**. With this in mind, we first sought to understand the proportion of each interaction kind within two datasets. Thus, we could adjust the weight each interaction is going to have by leveraging the proportion of instances in which it hasn't appeared, thus highlighting the rarity of each interaction as the driver of its value.

Nevertheless, there was still a data field not carrying any value or significance towards these same ratings: mlogviewTime – the time a user spends on a card he/she has clicked (in seconds). To incorporate such an important feature, we've set it as a bonus to a rating, so that individuals are not penalized and have a rating bump for how further they have taken their view time with cards, meaning that its scaled mlogview value is added to one and later multiplied by the previous result, which is finally min/max method normalized.

4.3 User/Item Embeddings

Finally, we're ready to set up the user and item embeddings used as inputs to the DeCS method. Firstly, it is necessary to have our first rating matrixes, by pivoting the dataset that attributes a user id, item id, and rating to each interaction so that we can get both **user/item and item/user matrixes**. These sparse matrixes have, respectively, a feature size of 932 and 4087 which **will require some feature dimensionality reduction techniques** to provide adequate, not too computationally heavy and quality-oriented user/item embedding. To fulfill these needs**, we have opted to use the SVD approach** as the dimensionality reduction method, since it has proven itself efficient in handling very sparse matrixes while extracting the essential out of the existing features. Besides, it is not computationally expensive and allows us to perform an analysis where a certain degree (preferably 95% or 90%) of variance can be explained. Thus, the number of dimensions will be of 330 dimensions on the user dataset and 17 as of the item dataset. **These dimensionalities are tied to the sample size** (of 5%) **and reflect this method's ability to achieve a 95% explainability on both datasets**.

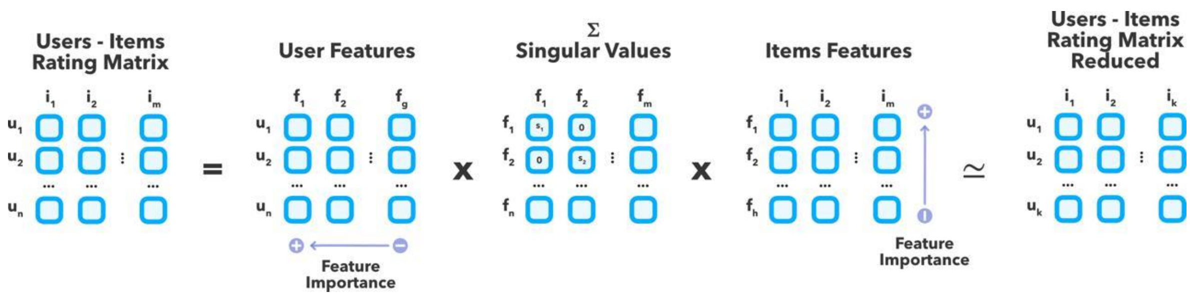


Figure 4 – SVD Dimensionality Reduction Illustration

4.4 User and Item SI Embeddings

Then, it is time to prepare the embeddings that correspond to the user and item side information. In this context, side information refers to the data that isn't directly related to the ratings used in the former embeddings, and speaks to the nature of users and items. This is the true differentiator of an item/user from its peers, playing a key role on the model fit for the mitigation of the cold start user problem. Thus, this role is played by a **universal, singular identity embedding** (all users and items identified will have it).

On that account, we define user-side information by the user demographics dataset, where we can find the existing metadata of each user: province, age, gender, activity level, and registration month (number of months before November 2019). Before one hot encoding of all the data, we first needed to bin both age and registration month, as one hot encoding them in the raw dataset would create an excessive amount of embedding features. We end this embedding preparation with one hot encoding of the entirety of the now well-suited data, leaving us with 63, user-side information features. Similarly, it is also necessary to have such embedding on the item side: as we dive into the mlog (card/item) demographics dataset, it is possible to observe that it's a broader metadata description than the user one. Out of all interactions, 39% are a repetition of artists in a card, 33% are song duplicates and only 6.1% are creator duplicates. While the average artist interaction duplicates per user is 8.55, the average for songs is 8.65 and the creator's one stands at 4.99. With this, 30% of users have repeated interactions with artists, 25.5% have done so with songs, and only 8.18% have done so in with card creators. This shows a clear intent of following the same artists and songs, thus emphasizes the similarity NetEase Music has with Tik-Tok, a platform with a very similar *modus operandi*, set of features, and model as a social network. Pointing out these similarities in the context of item information is important considering that this similar platform (in purpose) has proven how important artists and songs are to the platform's existence: the Music Impact Report has proven that users are nearly twice as likely to discover new music in comparison to other platforms, that the usage of such platforms is related to the likeliness of adhering to music streaming (62% of Tik-Tok users / 43% overall consumers -USA) and that, in the UK, Tik-Tok users spend 49% more on music purchases. This indicates that the app itself promotes awareness, interest, and closer following behavior of both artists and songs. Besides these three identifiers tied to each card, it is also important to take into account the (binned) publish time date in days from the end of the month followed by type, talk and content id, on what is a granular overlook of each card/item, being it possible to have an undisclosed talk id and various content ids, while also having a type that can be either 1 (text with background music) or 2(music video). After the single necessary binning, we end up with a staggering amount of 77824 features. With that in mind, the final number of dimension to which this dataset is logically reduced to depends on the size of the sample relative to the impression data dataset. In accordance to the analysis performed for all datasets in need of reduction (*Appendix* 3), it was decided that the item SI dataset was reduced to 157 dimensions, achieving a 95% explainability value from the initial number of dimensions.

4.5 Model structure

The DeCS model is a **deep neural network designed specifically to address the cold-start problem in recommender systems**. Its structure is built to effectively process sparse user-item interaction data by combining embeddings, side information, and a multiview neural network architecture. Its illustration corresponds to the *Figure* 5.

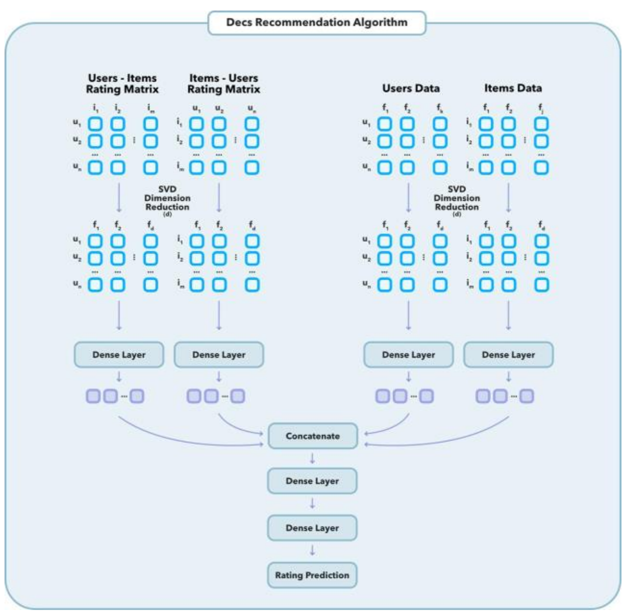


Figure 5 - A DeCS-Inspired Recommendation System Algorithm Structure

The model begins by taking as input a user-item and the item-user rating matrixes, which contain sparse interaction information. These matrixes aren't suitable for direct use in machine learning models due to their sparsity and to the high dimensionality. To address this, the DeCS framework uses embedding layers to transform the matrix's lower-dimensional datasets, made out of vectors attributed to each item/user. Each embedding maps a user or an item into a shared lower-dimensional space of size, which is chosen to balance the trade-off between computational efficiency, information retention, and variance explainability. In addition to embeddings, the model takes advantage of what is called side information, which corresponds to almost all metadata fields that are solely focused on items (mlogs) and users. All these fields are finally transformed into binary vectors using one-hot encoding and then processed through a dense layer to produce a compact vector representation, leveraging the ReLU activation function to introduce non-linearity.

The user embeddings and side information vectors are also passed through a couple of dense layers. Similarly, item embeddings and side information are processed through their couple of dense layers which are structurally identical to the previously mentioned layer, allowing the model to identify patterns or significant behaviors.

This same concatenation implies the combination of the four embeddings present in this neural network architecture in a unified vector. Here, concatenation means merging all existing features to that point into a common feature vector, with a size of 1024. This concatenated, unified representation is then passed by another set of two dense layers, where ReLU is once again employed into, respectively, a 256 and 128-neuron set of layers. Ultimately, the model outputs a predicted rating for each user/item pair, in a layer made out of a dense layer with a single neuron, activated by a linear function, so that it can produce continuous output values, correspondent to the final, predicted user rating.

4.6 Beyond the Model Structure – Intuition, Evaluation and Implementation

This model aims for better scenario in a recommendation process directed towards cold users in a recommendation context, but the model itself is far from all the thought process and intuition inherent to the proposed approach.

Since we wish to build a model focused at predicting items to recommend cold users, one must prepare it to perform in the worst case scenario: the total absence of information about cold users. With this in mind, it was decided that the model would be trained with all non-cold users data on 100 epochs, while the testing phase would only include cold users. Since all the cold user (actual) information is being tested, we are able to ensure that the model is being tested on a "pure" cold user basis, while having an actual data based testing. This will allow us to understand how the model truly behaves in this scenario, where it is most important to achieve the best performance in virtue of user retention with minimal information.

Then, it was necessary to perform initial model tests to understand its behavior and how it could be optimized. From the beginning, it became clear that the aforementioned intuition was resulting in very weak results on cold start users, against all available items in the dataset, no matter the sample size. This brought the need of a different approach that could ease the complexity of the task while accomplishing the goal the model was set to achieve. In light of these drawbacks and analysis, two item related solution rose: either the items with whom all users registered within the last six months have interacted, or the items for the users with a level of three or less interaction level were considered. This way, it would be possible to consider a smaller set of items from one of two universes that encompassed cold users, in a smaller, more objective and minded set of items, essential to ease the process complexity and provide better recommendations, in a "pure" cold user context.

Taking this into consideration, and for a different sample size, both item sets were tested in the prediction making process. In this analysis, there will be no concern in regards to the k (number of predicted items) number, being this something that will be approached and evaluated afterwards. As a result, the outcomes obtained are those presented in *Table 1*.

SAMPLE SIZE (DATASET %)	5		10	
ITEM SETS (RECENT USERS - R, LOW INTERACTION LEVEL USERS - L)	R	L	R	L
METRICS (K = 5)				
PRECISION	0.03	0.05	0.02	0.02
RECALL	0.11	0.22	0.08	0.09
HIT RATE	0.13	0.24	0.09	0.11
MRR	0.05	0.12	0.03	0.04
NDCG	0.07	0.16	0.04	0.05

Table 1 – Recent and low interaction users' itemset analysis (different samples)

Here, the recent user item related dataset performed better on all metrics, across both sample sizes, making it the itemset with the best ability to provide predictions, leading better model efficiency/precision. Despite this, it is still possible to observe that even though the model is working at its best, the results still seem to be poor, leaving one last parameter (k) to be shifted in order to mitigate so. The distribution of impression value works on a general basis is skewed to the right, with a very frequent set of values (up to four impression positions - *Appendix* 4). Here, the statistics tell us that the average number of impression is 24.

It is important to note that this number is the combination of all impression positions ,and does not accurately portray what we need to know: how far do users go, on average, when it comes to the number of cards seen. By knowing this, we would be able to adapt k into a new value that better portrays the reality within user behavior in the network, giving us a value to work with and reach a realistic set of results/ model performance. In that regard, the aforementioned analysis was conducted, and the maximum number of consecutive impressions per user in the platform presents a somewhat similar profile compared to the last impression position analysis - but a set of more modest statistics (*Figure 6*).

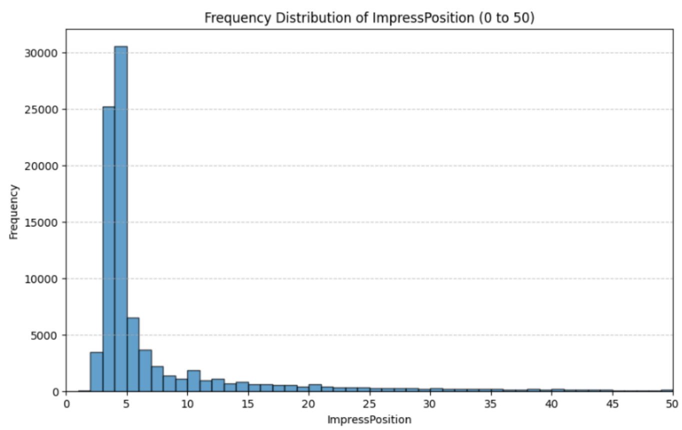


Figure 6 - Maximum Impress Position (per user) Frequency Distribution

	impressPosition
count	91149.000000
mean	10.220079
std	28.267617
min	1.000000
25%	3.000000
50%	4.000000
75%	6.000000
max	1337.000000

Table 2 - Maximum Impress Position (per user) Statistics

In *Table 2* we can see that the majority of users have only gotten to a maximum of 3 to 4 cards in a row at most. But just like the last distribution, this is also a highly skewed distribution to the right, being the statistics also a factor that demonstrates such thing, mainly by allowing us to know that, for example, that the first quartile stand at the value of three, while the second one only goes up by one, being the third equal to 6. The fact that the maximum value is of 1337 shows how far the tail of this distribution goes and how much does it influence the value that is perhaps the most important in the purpose we wish to achieve: the mean value, standing at a value a little over 10 impression seen at most per user, on average.

Both these analyses allow us to know the reality of the average user in the platform and therefore enable us to tune k to a different value than the 5 used to perform the tests between the two item sets we have evaluated. After getting to know the model, the data on impressions and the limitations it has in light of the analytic and predictive nature inherent to the purposes it wishes to serve it is possible to conclude that the wisest way to evaluate the model is by using the low interaction level user set of interaction items, along with the least computationally expensive sample (5% of the total impression data) and finally by setting k to 10, since it is the average number of interactions a user will go through at once, at most. Based on these insights, the final model was developed, and its predictions were computed, yielding the following results:

Metric	k=10
Precision	0.23*
Recall	0.29
HitRate	0.33
MRR	0.14
nDCG	0.16

Table 3 - A DeCS-Inspired Recommendation System Results

Throughout the given results (*Table 3*), it is possible to see that the precision level seems pretty low. Nevertheless, the actual prediction level turns out to be slightly higher than the existing one, due to the fact that the best case scenario in this precision evaluation is of 40% (if the maximum 4 interactions -or less- of a cold user are within the 10 recommended ones) which brings our actual prediction value to a staggering *23% (cold users have an average of 1.3 items they have interacted with – 0.03 / (1.3/10) = 0.23). This shows that the models performs somewhat well under a total lack of information on user preferences and achieved almost one in every four correct predictions in accordance with the precision adaptation that accounts for the average cold user actual interactions.

Then, recall also presents a somewhat impressive value of a 29% rate of all relevant items (actual interactions) that ended up showing in the predictions made for each user, which is a positive metric considering there was no item information in regards to any of the users within the test set.

Thirdly, the hit rate metrics enables us to observe what is perhaps the most important metric: one in every three cold users got recommended an item they actually interacted with, and with whom therefore showed signs of engagement and satisfaction. This comes as particularly important as it shows that, generally speaking, this model would have the ability to retain at least 33% of users, let alone with an online scenario where the model would progressively have more information to work with, as a user would interact (or not) with other items and explore the platform, in a hypothetical online recommendation system.

Moving on to MRR, a value of 0.14 indicates that, on average, the relevant items arrive between rank 7 and 8 of the predictions made by the model. In opposition with the positive idea some metrics may leave us with, this one arrives as something that needs to be improved in the model. It is crucial for a given model that it has the ability to come up with the best items as early as possible, for the platform to have a better chance of retaining and further entertaining/satisfying the user with relevant, recommended and carefully selected content based on all the information it has on a given user. This is not being achieved as much as we'd like for DECS and should therefore be a matter to be concerned in further versions/improvements these kinds of models might go through.

Lastly, and with a somewhat similar indication to MRR, NDCG has a low value of 0.16, indicating that the rankings provided by this DECS inspired model are of low quality, further indicating that relevant items are not showing up as early (and perhaps as frequently) as they should. This corroborates the previous metrics and highlights, once again, the importance of improving the model in virtue of better, stronger, and earlier relevant items within the recommendations.

5. Conclusion and Discussion

5.1 Key Findings

The findings from the **DeCS-inspired Recommendation System** revealed several key insights. First, the importance of an adequately sized **side information embedding layer** for both users and items became evident. The use of **SVD** for dimensionality reduction proved beneficial—not only for its computational efficiency but also for its ability to ensure a post-reduction explainability rate of **95%**. Second, the inclusion of **viewing time** for card interactions significantly improved the differentiation of item ratings. Third, the selection of a specific item set for predictions played a critical role in achieving reasonable model performance. Contrary to initial assumptions, basing item selection on **user activity levels** proved slightly more effective than focusing on items for recent users. Following this, analyzing **impression distributions** allowed for setting an appropriate standard *k* value, striking a balance between realism and model complexity. Lastly, the results revealed that the model struggled to position the most relevant recommendations at the top of the result sets. This indicates a need for structural or data-driven improvements to better identify and differentiate the most relevant items, ensuring they are appropriately prioritized.

SSZTo evaluate and compare these systems, we conducted a comprehensive analysis using **five recommendation evaluation metrics** which are now consolidated into a single representation for clarity and comparison (*Table 4*).

Metric	k	DeCS-Inspired
Precision	1	2.15
	3	2.49
	5	5.23
	10	23.44
Recall	1	1.97
	3	2.87
	5	5.5
	10	28.75
HitRate	1	2.15
	3	3.23
	5	6.45
	10	32.56
MRR	1	2.15
	3	2.51
	5	3.21
	10	13.66
nDCG	1	2.15
	3	2.48
	5	4.02
	10	15.81

Table 4 – DeCS Recommendation Model Across Evaluation Metrics

Overall, the **DeCS-Inspired** model excels at higher *k* values. By leveraging the user's interaction history, it achieves superior recall and hit rates, effectively ranking relevant items higher and capturing complex user behaviors. This model performs especially well for users with substantial interaction data, as it can better identify items aligned with long-term engagement and preferences. This model presents itself as suitable for Long-Term, Data rich scenarios as it offers the best balance between relevance and precision, with a singular ease in adaping to large scale personalization and evolving user engagement.

5.2 Limitations

The **DeCS-inspired Recommendation System** presents a more complex set of limitations. Its exclusion of cold-start users restricts its applicability to a broader and more diverse audience. The model's heavy reliance on a small, interconnected set of evaluation criteria narrows the scope of its analysis. The use of **one month's data** from the NetEase platform introduces **temporal and coverage constraints**, causing the model to miss unseen patterns—a challenge not unique to this model but shared across others as well. To achieve computational feasibility, dimensionality reduction and partial sampling are required, which, in turn, constrain the granularity of insights.

5.3 Research Contributions

Despite the limitations discussed earlier, this study makes significant contributions to the field of personalized recommendation systems, specifically in tackling the **cold-start user problem** on the NetEase Cloud Music platform. A major contribution of this study is the demonstration that integrating **diverse data sources** offers a holistic approach to understanding user preferences. The tested model showcased how combining multiple data layers—such as user behavior, metadata, and side information—can produce nuanced user profiles. These profiles serve as the foundation for generating personalized recommendations, particularly in cold-start environments.

Complementing this, the study highlights the effective use of **metadata fields** to tailor recommendations and evaluate initial recommendation sets.
This research also introduces a **rigorous set of evaluation metrics** to systematically assess model performance. These carefully selected metrics not only highlight each algorithm's strengths and weaknesses but also provide deeper insights into their real-world applicability. By establishing an **evaluation framework**, this study paves the way for scalable solutions, offering a standardized methodology that remains insightful regardless of the computational complexity or sophistication of the model.
Beyond academic contributions, this work delivers a **practical framework** for digital platforms like NetEase Cloud Music to improve **user retention** and **engagement**. By specifically addressing the needs of cold-start users, the methods developed in this study aim to foster **long-term user satisfaction**, which is critical in maintaining a competitive advantage in the fast-paced digital music industry.

5.4 Implications for Practice

The cold start problem is a problem faced by every music streaming service. But it is also an opportunity to **build trust, encourage user participation,** and create the foundation for a long-term relationship. Considering this particular situation as an increasingly common scenario in any digital, online platform, this research seeks to contribute to both academic knowledge and practical industry solutions by advancing the understanding of recommender systems. With a particular focus on addressing the cold-start problem, the conducted study approaches a **critical challenge** that hinders effective personalization in both regional and global markets.

It is not a merely technical problem, but it is very crucial when it comes to starting the user's journey on a platform like NetEase Cloud Music. With some demographics involved, such as age and geographic and cultural context, the platform can perhaps make recommendations that seem to reside within easy reach in a very intuitive way. The implications of solving the cold-start problem extend far beyond that first impression. It should incite exploratory activity and discovery. Meanwhile, based on demographic trends, new content—emerging artists/new releases—could find the audience compellingly through publicity in an already congested crowd. This serves both users and creators, thereby strengthening the platform's ecosystem.

Cold-start solutions are also important for returning users because their preferences might have changed or evolved. The same strategy can be employed on returning users as on new users, to spark that interest at the beginning and ensure that submissions on the platform remain relevant and consistently in touch with time.

This cold-start problem is not solely about filling the normatively structured void in the data; **it is also about shaping the user's experience from that initial interaction**. For NetEase Cloud Music and similar platforms, solving this challenge transforms the act of streaming into a **journey of connection and discovery**, one that builds loyalty and fosters a deeper love for music.

5.5 Future Research Directions

To address the need for an interactive and personalized user journey—especially for newer users—no amount of research alone will suffice due to ongoing technological advancements and recommendation methodologies. It is essential to explore further possibilities within this research context to understand what can still be improved or innovated. This exploration focuses on three key fronts: enhancing data strategies, improving model development, and addressing concerns related to data privacy and security.

It has become evident that the existing item and user data lacks critical informational and analytical components necessary for both interpretability and efficiency in model development. While privacy concerns are acknowledged, incorporating more identity-related and meaningful data about users, creators, and cards would allow for better insights and more robust models.

This is namely related to the song-related data. On a platform centered around music streaming services, specific attributes such as genre, beats per minute, key and mode, danceability, energy, or liveness would be crucial. The inclusion of such metrics would enable advanced techniques like data clustering, embedding creation, and other manipulations, offering higher-quality solutions to the cold-start user problem.

Furthermore, a broader temporal dataset would allow for analyzing temporal patterns in user preferences, creator behaviors, and card characteristics. Such insights could significantly improve model performance by understanding and even leveraging the impact of time on recommendations. However, working with larger datasets brings the need of higher computational capacity to process increased data complexity effectively.

The existing models, while functional, are not perfect and can benefit from structural enhancements. One major improvement is transitioning to an online recommendation system. This approach would enable real-time evaluations and updates, ensuring recommendations reflect the most recent and accurate information, thereby improving user retention and satisfaction.

Computational complexity has also posed significant limitations. The **DeCS-inspired neural network approach**, while promising, remains computationally expensive due to embedding processes, training, and implementation. Further experimentation with its network structure was constrained by time and resources.

Data privacy is a cornerstone of recommendation systems (RS), balancing the need for effective models with legal and ethical requirements for user privacy. Future work must address threats like adversarial attacks and recommendation poisoning to develop robust, secure, and fair systems.

This research lays the groundwork for solving the cold-start problem and advancing beyond it. Future directions will involve smarter algorithms, secure and ethical data management, and a vision for inclusive, user-centric platforms. For NetEase Cloud Music and the broader music streaming industry, this represents not just a technical challenge but an opportunity to redefine how music connects users in the digital age. **In an industry where the sheer abundance of content can overwhelm users, recommendation systems serve as a critical and indispensable differentiator.**

6. References

s.d. “AI in Music Streaming: Personalization at its Best.” *AI in Music Streaming: Personalization at its Best.*

s.d. “AI-based Recommendation System Market Report.” *AI-based Recommendation System Market Report.*

Bobadilla, JesúS, Fernando Ortega, Antonio Hernando, e Jesús Bernal. 2012. “A collaborative filtering approach to mitigate the new user cold start problem.” *Knowledge-based systems* (Elsevier) 26: 225–238.

Chadha, Aman, e Vinija Jain. 2022. “Recommender Systems Primer - Cold Start.” *Recommender Systems Primer - Cold Start.* www.vinija.ai.

Chakravorti, Bhaskar. 2018. “Competing in the Huge Digital Economies of China and India.” *Harvard Business Review* 6.

Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, e Hervé Jégou. 2024. “The Faiss library.”

Elahi, Mehdi, Farshad Bakhshandegan Moghaddam, Reza Hosseini, Mohammad Hossein Rimaz, Nabil El Ioini, Marko Tkalcic, Christoph Trattner, e Tammam Tillo. 2021. “Recommending videos in cold start with automatic visual tags.” *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization.* 54–60.

Feng, Philip J., Pingjun Pan, Tingting Zhou, Hongxiang Chen, e Chuanjiang Luo. 2021. “Zero shot on the cold-start problem: Model-agnostic interest learning for recommender systems.” *Proceedings of the 30th ACM international conference on information & knowledge management.* 474–483.

Hafnar, Davor, e Jure Demšar. 2024. “Zero-Shot Reasoning: Personalized Content Generation Without the Cold Start Problem.” *arXiv preprint arXiv:2402.10133.*

Han, Cuize, Pablo Castells, Parth Gupta, Xu Xu, e Vamsi Salaka. 2022. “Addressing cold start in product search via empirical bayes.” *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 3141–3151.

s.d. “How Retailers Can Keep Up with Consumers.” *How Retailers Can Keep Up with Consumers.*

Karabila, Ikram, Nossayba Darraz, Anas El-Ansari, Nabil Alami, e Mostafa El Mallahi. 2024. “Addressing data sparsity and cold-start challenges in recommender systems using advanced deep learning and self-supervised learning techniques.” *Journal of Experimental & Theoretical Artificial Intelligence* (Taylor & Francis) 1–31.

Kowald, Dominik, Markus Schedl, e Elisabeth Lex. 2020. “The unfairness of popularity bias in music recommendation: A reproducibility study.” *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42.* 35–42.

Kumar, Pushpendra, e Ramjeevan Singh Thakur. 2018. “Recommendation system techniques and related issues: a survey.” *International Journal of Information Technology* (Springer) 10: 495–501.

Lam, Xuan Nhat, Thuc Vu, Trong Duc Le, e Anh Duc Duong. 2008. “Addressing cold-start problem in recommendation systems.” *Proceedings of the 2nd international conference on Ubiquitous information management and communication.* 208–211.

Li, Jingjing, Ke Lu, Zi Huang, e Heng Tao Shen. 2017. “Two birds one stone: on both cold-start and long-tail recommendation.” *Proceedings of the 25th ACM international conference on Multimedia.* 898–906.

Lika, Blerina, Kostas Kolomvatsos, e Stathes Hadjiefthymiades. 2014. “Facing the cold start problem in recommender systems.” *Expert systems with applications* (Elsevier) 41: 2065–2073.

Lin, Xixun, Jia Wu, Chuan Zhou, Shirui Pan, Yanan Cao, e Bin Wang. 2021. “Task-adaptive neural process for user cold-start recommendation.” *Proceedings of the Web Conference 2021.* 1306–1316.

Liu, Haifeng, Zheng Hu, Ahmad Mian, Hui Tian, e Xuzhen Zhu. 2014. “A new user similarity model to improve the accuracy of collaborative filtering.” *Knowledge-based systems* (Elsevier) 56: 156–166.

Melville, Prem, e Vikas Sindhwani. 2010. “Recommender systems.” *Encyclopedia of machine learning* 1: 829–838.

s.d. “Music Streaming - China | Statista Market Forecast.” *Music Streaming - China | Statista Market Forecast.*

s.d. “Music Streaming - Worldwide | Statista Market Forecast.” *Music Streaming - Worldwide | Statista Market Forecast.*

Nguyen, An-Te, Nathalie Denos, e Catherine Berrut. 2007. "Improving new user recommendations with rule-based induction on cold user data." *Proceedings of the 2007 ACM conference on Recommender systems.* 121–128.

Pandey, Anand Kishor, e Dharmveer Singh Rajpoot. 2016. "Resolving Cold Start problem in recommendation system using demographic approach." *2016 International Conference on Signal Processing and Communication (ICSC).* 213-218. doi:10.1109/ICSPCom.2016.7980578.

Panteli, Antiopi, e Basilis Boutsinas. 2023. "Addressing the cold-start problem in recommender systems based on frequent patterns." *Algorithms* (MDPI) 16: 182.

Pu, Pearl, e Li Chen. 2008. "User-involved preference elicitation for product search and recommender systems." *AI magazine* 29: 93–93.

Roy, Deepjyoti, e Mala Dutta. 2022. "A systematic review and research perspective on recommender systems." *Journal of Big Data* (Springer) 9: 59.

Silva, Nícollas, Diego Carvalho, Adriano C. M. Pereira, Fernando Mourão, e Leonardo Rocha. 2019. "The pure cold-start problem: A deep study about how to conquer first-time users in recommendations domains." *Information Systems* (Elsevier) 80: 1–12.

Singh, Neetu, e Sandeep Kumar Singh. 2024. "A systematic literature review of solutions for cold start problem." *International Journal of System Assurance Engineering and Management* (Springer) 1–35.

Son, Le Hoang. 2016. "Dealing with the new user cold-start problem in recommender systems: A comparative review." *Information Systems* (PERGAMON-ELSEVIER

SCIENCE LTD THE BOULEVARD, LANGFORD LANE, KIDLINGTON …) 58: 87–104.

Song, Yading, Simon Dixon, e Marcus Pearce. 2012. "A survey of music recommendation systems and future perspectives." *9th international symposium on computer music modeling and retrieval.* 395–410.

Steck, Harald, Linas Baltrunas, Ehtsham Elahi, Dawen Liang, Yves Raimond, e Justin Basilico. 2021. "Deep learning for recommender systems: A Netflix case study." *AI Magazine* 42: 7–18.

Sun, Anchen, e Yuanzhe Peng. 2022. "A survey on modern recommendation system based on big data." *arXiv e-prints* arXiv–2206.

Tahmasebi, Faryad, Majid Meghdadi, Sajad Ahmadian, e Khashayar Valiallahi. 2021. "A hybrid recommendation system based on profile expansion technique to alleviate cold start problem." *Multimedia Tools and Applications* (Springer) 80: 2339–2354.

Vozalis, Emmanouil, e Konstantinos G. Margaritis. 2003. "Analysis of recommender systems algorithms." *The 6th Hellenic European Conference on Computer Mathematics & its Applications.* 732–745.

Wang, Chunyang, Yanmin Zhu, Haobing Liu, Tianzi Zang, Jiadi Yu, e Feilong Tang. 2022. "Deep meta-learning in recommendation systems: A survey." *arXiv preprint arXiv:2206.04415.*

Wang, Shixiong, Wei Dai, e Geoffrey Ye Li. 2024. "Distributionally Robust Outlier-Aware Receive Beamforming." *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP).* 1–6.

s.d. “What Netflix's Recommendation Systems Can Teach Us About The Computing Challenges Of The Near Future.” *What Netflix's Recommendation Systems Can Teach Us About The Computing Challenges Of The Near Future.* https://www.forbes.com/councils/forbestechcouncil/2021/02/19/what-netflixs-recommendation-systems-can-teach-us-about-the-computing-challenges-of-the-near-future/.

Zhang, Dennis J., Ming Hu, Xiaofei Liu, Yuxiang Wu, e Yong Li. 2022. “NetEase cloud music data.” *Manufacturing & Service Operations Management* (INFORMS) 24: 275–284.

Zhu, Ziwei, Shahin Sefati, Parsa Saadatpanah, e James Caverlee. 2020. “Recommendation for new users and new items via randomized training and mixture-of-experts transformation.” *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1121–1130.

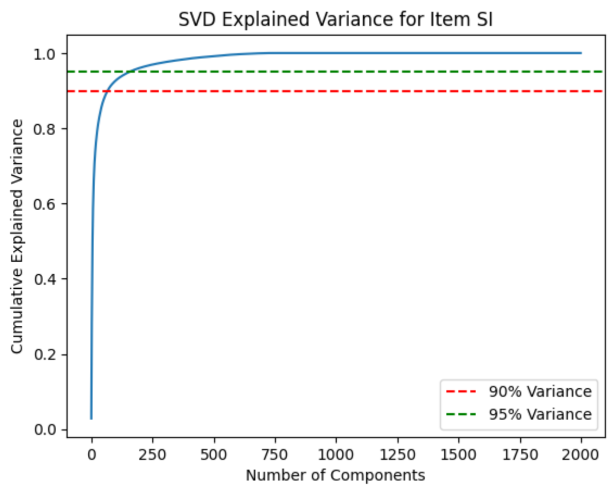
7. Appendix

Element	Variable	Description	Data Type
Position Element	**ImpressPosition**	Position of the card in the user's feed, starting from 1.	Integer
Engagement Element	**isClick**	Binary value indicating if the user clicked on the card (1 for yes, 0 for no).	Binary
	isLike	Binary value indicating if the user liked the card.	Binary
	isComment	Binary value indicating if the user commented on the card.	Binary
	isViewComment	Binary value indicating if the user viewed card comments.	Binary
	isShare	Binary value indicating if the user shared the card on social platforms.	Binary
	isIntoPersonalHomepage	Binary value indicating if the user accessed the creator personal homepage.	Binary
	ImpressTime	Epoch timestamp representing when a card was displayed to a user in millisecc	Numeric
	mlogViewTime	Duration (in seconds) the user spent viewing the card.	Numeric
Card Element	**mlogId**	Unique identifier for each card displayed to users.	String
	songId	Unique identifier for each song in a card.	String
	artistId	Unique identifier for each artist in a card.	String
	creatorId	Unique identifier for the card creator.	String
	contentId	A set of content identifiers attributed to a card.	List
	publishTime	The number of days when the card is published till December 1st, 2019.	Integer
	talkId	Unique level that represents the anonymized topic of the card.	Integer
	cardType	Type of card (e.g., video or image with background music).	Categorical
User Element	**likesCount**	Total number of likes received by the card until the timestamp.	Integer
	userId	Unique identifier for each user.	String
	registeredMonthCnt	Number of months between a user's registration time and December 1st, 2019.	Integer
	userLevel	Activity intensity of a user, ranging from 0 to 10.	Integer
Demographics Element	**province**	Province of residence of the user (in Pinyin format).	String
	age	The age of a user.	Integer
	gender	The gender of a user.	String

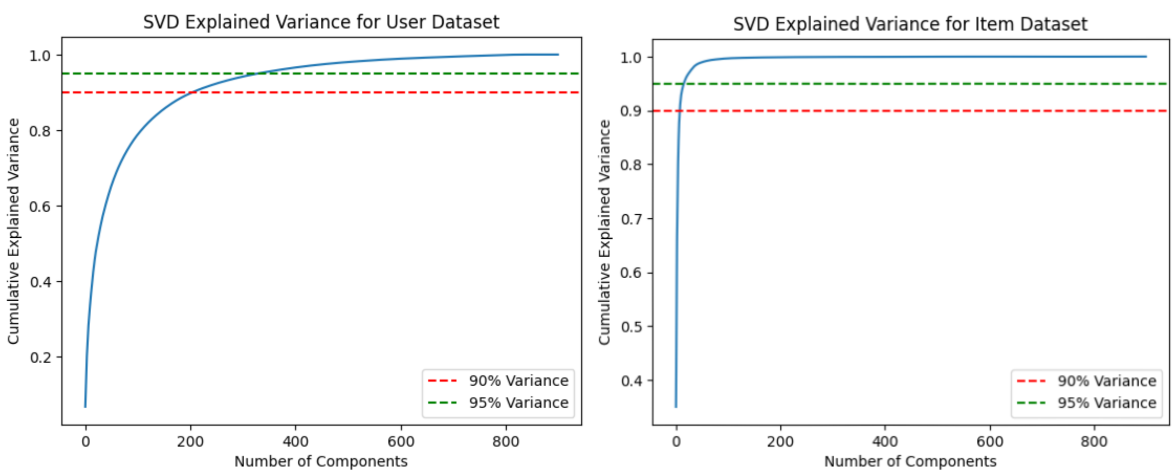
Appendix 1 - NetEase Table Metadata: Variable Descriptions by Category

	Interactions
count	2085480
mean	26.08
std	112.52
min	1
25%	4
50%	7
75%	17
max	22637

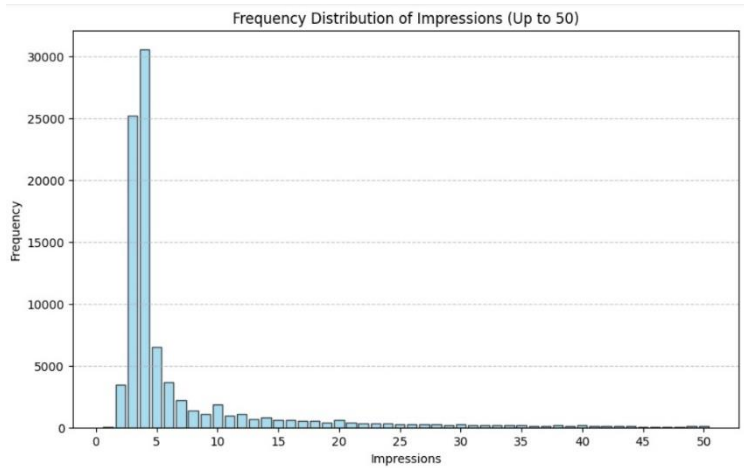
Appendix 2 - User impressions statistics



Appendix 3 - Cumulative Explained Variance of SVD components fot the Item SI Embeddings



Appendix 3 - Cumulative Explained Variance of SVD Components for User and Item Embeddings



Appendix 4 – Frequency Distribution of all Impression Position Values