A 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

João Rafael Simão Rocha

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

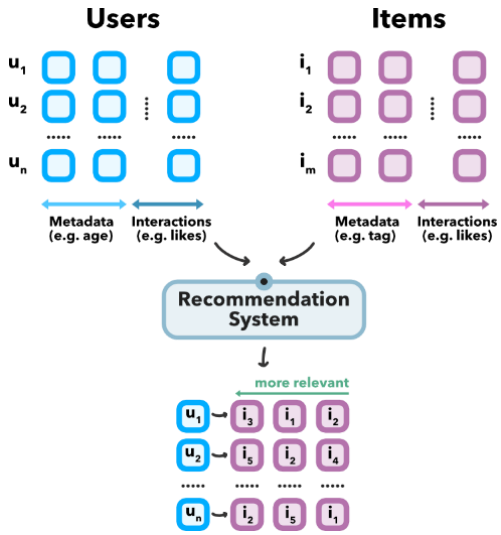
This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab,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

INDEX

1. Introduction

“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. The diverse nature of both content and user needs makes delivering relevant items in response to a user query a complex challenge. 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

The growing reliance on recommendation systems in the music streaming industry highlights the importance of personalization in attracting users. The global market for music streaming is expected to generate $29.60 billion in 2024 and $33.97 billion in 2027, growing through increased affordability, advancement in mobile technology, and greater access in emerging markets (Chadha e Jain 2022; Music Streaming - Worldwide | Statista Market Forecast s.d.). Such a phenomenon leads to a huge amount of data generation, which companies then use to improve their algorithms for recommendations, thereby enhancing the satisfaction and retention rates of users (Sun e Peng 2022).

This growth has also been reflected in the music streaming market in China, which is projected to achieve a CAGR of 4.84% from 2024 to 2027, mainly due to the popularity of local artists as well as the data-driven strategies employed (Music Streaming- China | Statista Market Forecast s.d.). Innovations have been in place at leading platforms such as Tencent Music and NetEase to maximize user engagements through AI-powered recommendations to address competition-related pressure (Financial Times 2023). Globally, AI in music is expected to grow from $2.8 billion in 2023 to $34.4 billion by 2033, with recommendation systems playing a critical role (AI-based Recommendation System Market Report s.d.). These systems help increase acquisition of customers, reduce churn, as well as improve users' satisfaction, subsequently influencing the competitive advantage and long-term loyalty of

music streaming providers (AI in Music Streaming: Personalization at its Best s.d.; How Retailers Can Keep Up with Consumers s.d.).

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**, the **new-item problem**, and the **cold-user problem**. The new community problem arises when a given information system has just booted up and there has been no interaction between any user and items. In the new-item problem, the goal is to recommend a recently added item in the catalog to users, despite lacking information about how much they may like it. Conversely, the cold-user problem arises when a new user joins the system, and the lack of prior knowledge about their preferences makes it difficult to select relevant content from a vast catalog of possibilities. Recommendation systems can also be challenged by returning users whose behavior and preferences may vary between sessions, not just by first-time visitors. A widely used recommendation method for addressing the new-item problem is content-based filtering (CBF). This approach mitigates the issue by leveraging item metadata, such as tags or attributes, to identify item similarities. It then recommends new items to a target user based on their resemblance to those the user has liked in the past. To address the cold-user problem, Preference Elicitation is a popular technique (Pu e Chen 2008). This method gathers user information during the registration process, either by directly asking users to provide their preferences or by leveraging existing data, such as from their social media accounts. This initial data helps generate an embedding for the user, serving as a recommendation starting point. Another approach is categorizing users into groups, such as by demographics, and using the available information from cold users to assign them to a relevant group (Pandey e Rajpoot 2016). This allows for more tailored

recommendations based on the interaction history of other users within the same group. When little to no information is available, a fallback solution is to recommend globally popular items. However, Recommender systems biased toward globally popular items exhibit the “Harry Potter effect”, where widely liked items, such as Harry Potter, are frequently recommended. While users may enjoy such content, these recommendations often lack novelty. Globally popular items can be filtered or weighted down in the recommendation process to mitigate this effect. Knowledge bases are also leveraged to supplement missing system information, enriching the context and the relevance of content provided to users (See *Figure 2* for an abstract overview of the cold-users problem). Users’ new interaction data is typically collected continuously, enabling cold users to gradually become “warmer” as their preferences and behaviors are better understood over time. Consequently, by leveraging Active Learning methods, the cold-user problem can further be mitigated. Nevertheless, the challenge of providing initial content to new users is crucial for the user to trust a given platform

The "cold-start problem" remains a critical bottleneck in the effectiveness of recommendation systems. This issue arises when insufficient data is available for new users or content, leading to suboptimal recommendations. 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. Similar problems occur in other industries, going from e-commerce to online advertising. A new online store may find it difficult to suggest items to a customer with no purchase history. An online advertiser may also find it a challenge to reach users whose browsing history is very limited. Essentially, in this work project, we have investigated the challenge of providing accurate

and meaningful recommendations to new users on the NetEase Cloud Music platform, where traditional collaborative filtering approaches struggle because of minimal interaction records. We developed and compared various techniques, ranging from demographic similarity and preference elicitation to embeddings and discriminating frequent item sets, to improve recommendations for cold-start users. Our results revealed that the **DeCS-Inspired model** performed the best under **data-rich conditions** with higher k values (e.g., k=10), effectively capturing long-term user preferences and achieving superior ranking metrics. However, for **cold-start scenarios** where interaction data is limited, the **Demographic Similarity-Based** and **User Preference Elicitation** models demonstrated their effectiveness in delivering initial, personalized recommendations. By integrating these techniques into a **hybrid approach**, we identified robust recommendation models tailored to sparse first-user data while establishing metrics to evaluate their performance. This comprehensive strategy ensures accurate recommendations, regardless of the amount of user interaction data available.

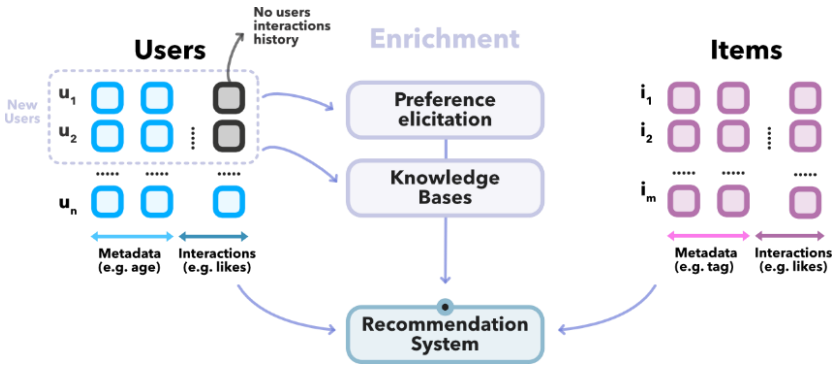


Figure 2 - Cold Users Problem *Abstract Components*

2. Literature Review

Recommendation systems can be understood through three key operations: information retrieval, scoring, and re-ranking. Information retrieval involves filtering and selecting a subset of relevant items from the vast catalog based on initial user input or user implicit signals (e.g. past history). Scoring assigns a relevance score to each retrieved item, often using algorithms that factor in user preferences, item features, or historical data. Finally, re-ranking adjusts the order of the scored items to optimize the presentation, ensuring the most relevant or engaging content appears at the top.

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 is a common issue when new users or items lack interaction and therefore restricts recommendation accuracy. Researchers have proposed several innovative approaches to address this problem, these methods aim to extract more meaningful insights from limited data, allowing recommendation systems to provide relevant suggestions even in the absence of extensive user history. In traditional collaborative filtering (CF) systems, the absence of historical data constrains the system in detecting user preferences, while content-based filtering (CBF) approaches have limitations when item features don't represent user preferences properly (Pandey e Rajpoot 2016), (Li, et al. 2017). This section discusses key approaches that have been suggested by researchers, for instance, recommendations based on demographic attributes, such as age and gender, 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 Challenges in cold-start Information Systems

When information systems are first launched, they face uncertainty regarding the relevance of items to users and users' potential interests—a challenge known as the new community problem. Additionally, as the product catalog evolves and new items are introduced, there is uncertainty around how existing users will respond to these additions, commonly referred to as the new items problem. Finally, as the user base grows over time and individual interests shift, recommending relevant items to users with limited interaction data becomes a significant challenge, known as the cold-users problem. Matrix Factorization (MF) is commonly used to learn vector embeddings for users and items, helping to predict user preferences. However, it faces challenges with the cold-start problem, where there is little to no information about new users or items. To address this, some approaches extend MF by incorporating metadata about users and items or by leveraging pre-trained embeddings for these entities. While effective in some scenarios, these methods often struggle in dynamic environments where new users or items are frequently introduced. To overcome this limitation, recent approaches have reframed the cold-start problem as a meta-learning task. Instead of training a separate model for each user or item, meta-learning focuses on teaching the model how to "learn" user preferences efficiently with minimal data. This approach aims to make recommendations quickly and effectively, even with sparse information, as highlighted in "A Meta-Learning Perspective on Cold-Start Recommendations for Items." In the context of music streaming platforms and the cold-start problem for new users, (A Semi-Personalized System for User Cold Start Recommendation on Music Streaming Apps) proposes using user clustering and preference elicitation to address the lack of user-specific information.

2.2 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). One example of a Deep Learning based approach is DropoutNet which focuses on training neural networks for cold-start situations, using dropout techniques to create the effect of missing data during the training phase (Steck, et al. 2021).

2.3 Hybrid Systems for Enhanced Personalization

A further promising strategy is the implementation of hybrid systems which integrate different recommender techniques. For instance, several approaches combined CF and CBF using neural network architecture like self-organizing maps to capture the user's preference better during the initial interactions. These methods can be particularly effective in RS, where additional features like song metadata and user activity context have been shown to improve during cold-start scenarios (Nguyen, Denos e Berrut 2007).

Further, on prediction accuracy, recent developments in personalized RS go far beyond improving prediction accuracy, which has been the primary focus, into factors that enhance user experiences like novelty, diversity, relevance, serendipity, and user satisfaction (Lam, et al. 2008). For example, while creating music playlists, MRS has started to incorporate additional context-aware features, such as user's activity or mood, which adds a layer of personalization that traditional systems lacked. Additionally, it has been noted that hybrid

models that dynamically adapt the weights of different recommender techniques based on user feedback have been successful in running with changes in user preferences over time (Bobadilla, et al. 2012), (Song, Dixon e Pearce 2012). Spotify's "Discover Weekly" playlist exemplifies the effectiveness of this approach, leveraging a hybrid of content analysis, contextual data, and CF to suggest songs, tackling both cold-start issues and long-tail item challenges (Kowald, Schedl e Lex 2020).

2.4 Novel Architectures for Cold-Start Solutions

Some researchers have explored novel model architectures to address cold-start limitations. (Feng, et al. 2021) addressed the cold-start issue by developing a model-agnostic interest learning framework. Their approach pulls user attributes and behaviors using a two-tower system. The first tower generalizes user behavior across both new and existing users, effectively mitigating the sparsity of data for cold-start users. The second tower then refines this information to generate personalized recommendations. This method is particularly effective because it balances generalization with personalization, ensuring that new users receive relevant recommendations without needing extensive interaction history.

Another solution comes from (Lin, et al. 2021), who introduced a task-adaptive neural process (TaNP). This approach is rooted in meta-learning, where each user is treated as a separate task. It uses limited user interactions to quickly adapt and make personalized recommendations by mapping observed behaviors to a predictive distribution. This allows the system to leverage global knowledge and apply it to individual users, enabling accurate predictions even with sparse data. The flexibility and adaptability of this method make it

highly effective for cold-start users, as it customizes recommendations based on just a few initial interactions.

In a more data-driven approach, (Han, et al. 2022) applied Empirical Bayes to model user engagement in large-scale e-commerce systems. This method combines behavioral and non-behavioral signals to generate more accurate recommendations for new items and users. In the context of a music platform like NetEase Cloud Music, such a model could help overcome the lack of initial user interactions by predicting user preferences based on general patterns of engagement and content popularity. This approach not only improves recommendation accuracy for cold-start items but also ensures a balanced user experience by integrating business metrics.

A novel approach by (Wang, Dai e Li 2024) operates with Large Language Models (LLMs) to address the cold-start problem by generating synthetic user behaviors. LLMs analyze textual descriptions of items and users' previous interactions to simulate potential user preferences, filling in the data gaps for cold-start items. This extended data provision is then used to train the models more effectively, in which the ability to simulate user behavior without needing actual interaction data improves the system's capacity to recommend new items.

2.5 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 measurement derived from the implementation of a demographic-based correlation into the predicted rating, inspired by (Vozalis e Margaritis 2003). This integration enhances correlation accuracy in predictions, essential for improving recommendations for new users. Meanwhile, (Son 2016) conducted a comparative review addressing the performance of various methods for 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 metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and computational time, ultimately making it the least effective among the compared methods. 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. Instead of employing the Proximity-Impact-Popularity (PIP) similarity measure, NHSM introduced the Proximity-Significance-Singularity (PSS) measure. This new approach evaluated the distance between rating pairs, deviation from the median, and uniqueness of rating pairs, thereby improving the accuracy by combining PSS with the Jaccard similarity coefficient. Moreover, NHSM used the mean and variance of user behavior globally—rather than locally—which contributed to its high accuracy and reduced reliance on metadata, without the fuzziness found in other models.

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). However, in an environment where new items are added with high frequency, RL provides low-quality results.

2.6 Recommendation system techniques and related issues: a survey

The conceptualization of a recommendation system is crucial for effectively addressing the challenges associated with cold-start users. There are numerous methods and models that, while attempting to solve certain issues in item recommendation, inadvertently exacerbate others. Therefore, it is essential to outline these challenges comprehensively and be prepared to address them effectively. 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, which limits the system's ability to make distinctions.

Privacy concerns also arise due to the way user data is utilized for improved recommendations, highlighting the need for careful data management practices. Synonymy, which can cause confusion in recommendations, can be mitigated through methods such as constructing a thesaurus, using singular value decomposition, and latent semantic indexing.

Given these challenges, an optimal model would involve a genetic approach combined with K-means clustering. This approach would decompose similarity values using vector cosine and other types of metrics, apply a genetic algorithm for optimization, and provide improved rating predictions for users.

2.7 Emerging Trends

The latest research performed around the topic of is comprised of two papers from this year that aim at tackling this issue with two different approaches that seek to fight a common

threat to the efficiency of any recommendation system and the sustainability of users within the platform it acts upon: cold-start users.

(Karabila, et al. 2024) proposes an approach that takes on deep advanced learning and self-supervised learning techniques. It 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, namely UserKNN and Item KNN. It provides insightful contributions on each one of the models within the ensemble, as it relies on a hybrid approach that integrates BERT with Collaborative filtering where BERT-based embeddings captures relationships in textual context, enhancing the user-item relationship understanding, while it leverages the collaborative filtering side of it as a baseline that BERT is complementary to. The SSL (Self Supervised Learning) comes in as a management factor in regards to unlabeled data, which comes in handy into addressing what is a commonality across recommender systems ready data/embeddings - data sparsity - and can therefore play a key role in the enhancement of the recommendation accuracy. Its role in addressing the cold-start problem derives from the insightfulness it can retrieve from the embeddings through textual item profiles, besides the ability it has through SSL to uncover data behavior and patterns, reducing the historical data dependence. It is also important to note that the role of combining 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 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. This answers the needs of a truly cold-start

ready recommender system, as it leverages the GPT 4 LLM to generate game levels using zero-shot reasoning, in an attempt to have a personalization as effective as possible based on a residual amount of available data, and therefore without the need of a heavy training process. This is enabled by the reliance on a framework that combines the little gameplay data with predefined, structure prompts that guide the LLM to achieve consistent and tailored results, with better player engagement and retention rate results when compared with other level generation methods. The relevance this paper brings into this problem is of a complete bypass on the extensive user-item data need through the zero shot capabilities of LLMs, that allow for the production of personalized and relevant outputs with minimal inputs. It is, after all, a demonstration of the strength that zero-shot learning can have as a cold-start mitigation technique.

2.8 Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns

(Panteli e Boutsinas 2023) proposed a methodology to solve the cold-start problem in recommender systems by integrating user clustering with the extraction of frequent discriminative patterns. Unlike conventional collaborative filtering approaches that struggle to recommend items to new users or products due to a lack of historical interactions, their method employs user features and previous purchase data from so-called "hot" users to generate recommendations. Users are first grouped based on demographic or contextual characteristics. Within each group, frequent item sets are extracted to identify combinations of products that existing users consistently like. Discriminative frequent item sets, which are patterns that are significantly more prevalent in one group of users than in others, are then extracted. These discriminative item sets serve as 'mind-blowing' buying behaviors for new

and cold users who share similar characteristics with the group. By assigning relevant frequent items to these users, the method can effectively improve the user-product matrices, reducing dispersion problems and allowing for precise recommendations, even without initial rankings.

Empirical evaluations on the MovieLens dataset show that this hybrid approach can achieve higher accuracy and handle both out-of-matrix (pure cold-start) and in-matrix (sparsity-related) prediction scenarios. The main contributions of this work include improving the diversity of recommendations, handling zero-rating scenarios without requiring extensive user interaction.

2.9 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 personalization if the additional data lacks relevance. 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 be on developing low-cost yet effective models that include strategies for mitigating biases in recommendations without compromising their efficiency (Pandey e Rajpoot 2016), (Silva, et al. 2019).

2.10 Summary

To conclude this section, recent strategies in recommendation systems address cold-start challenges by integrating methods from demographic clustering and deep learning to meta-learning, hybrid approaches, and reinforcement learning. These approaches aim to improve personalization on the problem of cold-start users, enhancing user satisfaction, and fostering long-term engagement from the very first interaction.

The more recent strategies have shown us that the cold-start problem has several means to be solved, with the arrival of deep learning, contextual embeddings and adequately leveraged one-shot learning being key into creating what is, as of this day, the state-of-the-art panoply of means to tackling this problem. It portrays something that can be summed up as the contemporary existence of highly contextual, data broad approaches to reach a highly reasoned recommendation and a zero shot, more aggressive approach that requires very limited amounts of data, and accounts for the creation of recommendations by focusing on the abilities to be consistent, while having a tailoring 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, including the "Cloud Village" tab, which fosters a community-oriented ecosystem through short-form music content and interactive functionalities such as liking, sharing, and commenting on music cards. This unique approach not only enhances user engagement but

also incentivizes creators to contribute, making NCM a central player in China's competitive music streaming market. (Zhang, et al. 2022)

This dataset contains an impression-level interface of over 57 million interactions as recorded by NCM, reflecting users' behavioral data such as clicks, likes, and shares in addition to demographic and content metadata, all collected in November 2019. The dataset opens a window into exploring some of the major problems faced by the platform:

- **Cold-Start Users:** this dataset allows research into ways of alleviating the cold-start user challenge through demographically similar recommendations for users who have little or no interaction history.
- **User engagement optimization:** understanding specific user behaviors can help create an effective strategy to ensure meaningful interaction for prolonged loyalty to the platform.
- **Creator incentivization:** looping back and forth between users and creators will be analyzed in the dataset and open up avenues for future considerations on how to motivate and reward creating varied and high-quality content.
- **Personalization and diversity** bring out the contradiction between hyper-personalized recommendations and diversity for content, which is an important equation for keeping the user happy and growing.

The digital services and products industry in Asia is characterized by rapid growth, diverse user preferences, and intense competition, with the music sector being no exception. China alone boasts over 800 million active internet users, with average daily usage surpassing the global average of 5.9 hours. Despite this, only about 60% of China's population is online,

indicating significant potential for further expansion in the digital economy. This substantial online presence has positioned China at the forefront of global e-commerce markets (Chakravorti 2018).

In this dynamic environment, effective personalized recommendations serve as a key differentiator for success. Such strategies are crucial for entering and sustaining a presence in industries where traditional media is steadily losing ground to the Internet due to its content limitations. Urbanization has surged, with the urban population increasing from a quarter to half of the total, leading many individuals to turn to the Internet to reconnect with their cultural roots.

Furthermore, factors such as rising income inequality and the loneliness associated with the long-standing one-child policy gave a huge push to Internet usage and involvement in digital platforms.

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, which contains two feeds of short music content cards that correspond to the gathered data scenario, specifically on the discovery subtab. With a sample made of over 2 million users over November 2019, each impression consists of music content card appearance to any user. Furthermore, the data fields for each impression are set to portray all possible actions within the UI possibilities NetEase provides data on various user interactions with content cards, including clicks, likes, shares, comments, views, and follows. These data fields are designed to capture the precise actions users take when engaging with each card in the dataset. They highlight key behaviors,

such as whether a user opens a card, leaves a comment, likes or shares it, and, in the case of a click, how long the user spends viewing it.

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 by incorporating information about the identities and relationships of users, creators, and third parties within the platform's ecosystem.

Therefore, besides the impression-level data, card-level, creator-level, and user-level datasets capture different aspects of each interaction, providing us with a broader sense of engagement dynamics on the platform. For example, while the card demographic table goes over the attribution of a song, creator, artist, content, and talk IDs, the card stats table is comprised of metrics on impressions, likes, shares, or comments for a given day within the period being analyzed. While the creator tables outline demographic attributes, follow information, and attribute a type and level to every user, the user tables similarly categorize users in light of their demographic information, and activity intensity, besides followage and month registration info. Lastly, there is also a creator stats table, which has a residual number of features and serves the purpose of identifying the number of publications during the sample period, for every identified creator.

3.3 Metadata

The dataset captures all aforementioned user interactions through a binary field for each action in particular: whether a user has (or hasn’t) clicked, liked, shared, or commented (**isClick**, **isLike**, **isShare**, and **isComment**) a card. Additionally, being the view time a field

in which the number corresponds to the number of seconds a user has spent on a card after clicking it ranging up to 41,186 seconds. The position (**ImpressPosition**) of a card in the user's feed and the timestamp of its display (**ImpressTime**) provide more context about when and where interactions occur. On the card level, variables such as **mlogId**, **songId**, and **artistId** uniquely identify each card. Other attributes, including **publishTime**, **contentId**, and **talkId**, respectively describe the card's publication timeline, content, and topics, which are essential for understanding trends and patterns in user preferences. The **cardType** categorizes the format of each card, differentiating between video content and images with background music. The dataset also includes demographic and activity metrics at the user and creator levels (**age**, **gender**, **province**, **registeredMonthCnt**, and **userLevel**).

This data is key when it comes to the development of a recommendation system, as it provides **important insights into user's engagement preferences and behavioral patterns**. The previously mentioned features are the foundation of the recommendation algorithm, as these are the primary source of essential knowledge/information on a user, card, and creator. On that account, these features act as guides within the system by presenting content that aligns with the user's past behavior and preferences. The integration of these data nuances may be key to an adequate balancing of cold user satisfaction. With that in mind, pleasing consumers is not only essential to fulfill the platform's long-term objectives but mainly to ensure that it is possible to get the best out of cold-start users by providing them with the best content possible under a broad and deep set of information.

The table below (*Table 1*) provides a comprehensive overview of these metadata variables, summarizing the features that were used in our recommender systems:

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 millisec	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

Table 1 - NetEase Table Metadata: Variable Descriptions by Category

3.4 Cold-Start Problem: Definition and Challenges

The cold-start problem is a restriction to the effectiveness of recommender systems, especially when new users or new items are involved. It is chiefly caused by the absence of previous records, making it hard to make relevant recommendations. The NetEase Cloud Music Platform's dataset also experiences the cold-start issue, relative to the challenges presented by the nature of the users, the nature of the items, and the evolution of user-content interaction over time.

We can explain the issue of cold-start in three different problems as follows:

- **New User Problem:** In the case of a new user coming to a platform, therefore there is no past data that allows us to predict the user preferences and recommend suitable content.
- **New Item Problem:** As the items in the catalog get updated, new items added to the platform do not have enough data from the users to know how relevant they are to the present users.
- **New Community Problem:** When the system is first implemented, the community is new and there isn't enough information about its members or the content provided, so there is trouble with how to make good recommendations.

Furthermore, as users' choices change over time and both the number of users and items increase, achieving this level of personalization becomes increasingly challenging. 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. The criteria were the following:

- **Registration Date:** Users who registered within the last six months (on or after June 2019).
- **Interaction Count:** Users with four or fewer recorded interactions (e.g., clicks, likes, comments).
- **Activity Level:** Users with an activity level of three or lower on the platform.

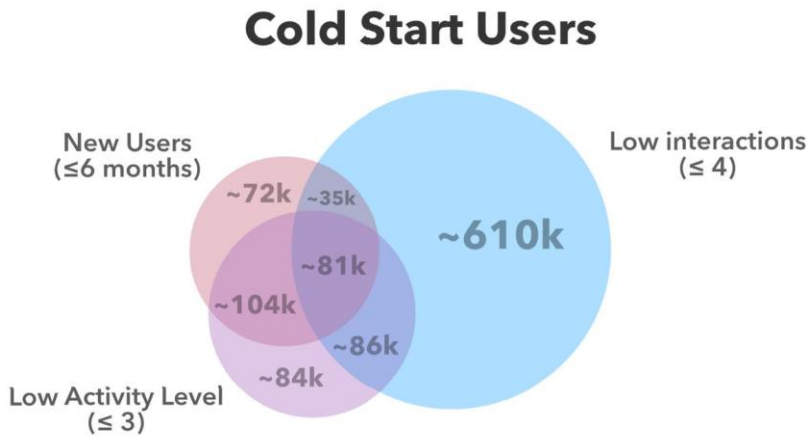


Figure 3 - Cold start users definition

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 focus on users from the past six months was necessary, as older users lacked sufficient interaction records dating from November 2019, and thus their engagement history could not be measured effectively. The interaction limit was established by analyzing user engagement (*See Table 2*):

- **25%** of users had interactions less than or equal to four.
- **50%** had below seven interactions.
- **75%** had more than seventeen, with the average being 26.

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

Table 2 - User impressions statistics

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. These individuals, with minimal engagement and platform familiarity, lack sufficient interaction data to generate reliable recommendations, making this group a prime target for addressing the cold-start problem effectively.

Despite the dataset's considerable size, comprising nearly 60 million observations, the cold-star problem remains significant due to several inherent limitations and complexities. We can name a few limitations:

Implicit Feedback: The dataset provides implicit feedback with the use of features such as "*isClick*", "*isComment*", "*isLike*", and "*isShare*". These features help estimate surface user activity, and more importantly, they do not provide explicit preference information which assists in the creation of explicit user interest profiles.

Exploration vs Engagement: For cold start users, who have very little interaction with the platform, the little available data may be about exploration rather than engagement. Therefore, understanding their initial suggestions may prove challenging. This results in the suggestions being off the mark for the users, leading to dissatisfaction and withdrawal.

Demographic Constraints: Understanding user characteristics involves the use of some basic demographic data such as province of residence, gender, and age. In any case, these are not comprehensive or detailed enough to facilitate the development of advanced user profiles. For example, we believe, an over-representation of age or province can skew the recommendations to default content that is active among the active users of the platform and thereby disregard other cold-start users with different content preferences.

New Content: The cold start does not only concern users, but it also applies to new content added to the system. As the new content does not have enough interaction history, its relevance to the users cannot be determined thereby making it difficult to recommend the content. Furthermore, because the platform is a dynamic catalog that is growing in both the number of content and users, there are difficulties in forecasting how the current users will react to new content.

New Community Problem: In the same way, at the initial stages of launching the system to the market (the new community problem), the system faces item visibility and users' interests without historical data. This absence of data makes it difficult to come up with effective solutions to different problems for different users.

4. Exploratory Data Analysis

In the direction of understanding user behavior and data quality within the NetEase Cloud Music platform, we carefully analyzed key variables and interaction patterns present in the dataset. The analysis also addressed data inconsistencies, outliers, and the distribution of user engagement across the platform. Below is a detailed breakdown of the key findings and steps taken during the EDA.

4.1 Feature Explanation

Some part of the analysis was about understanding temporal variables, like the **impressTime** feature, which originally was recorded as a Unix timestamp, however, we converted it into a more readable datetime format using pandas' **to_datetime()** method. To facilitate further

analysis, additional columns were created, namely **day**, **hour_of_day**, and **day_of_week** from the converted datetime values. However, during this process, some discrepancies were found between the **dt** column (representing day numbers) and the corresponding **impressTime** values. These discrepancies might indicate some flaws during data gathering or processing. Moreover, results along with certain extremes such as exceptionally large **mlogViewTime** logs were dealt with properly.

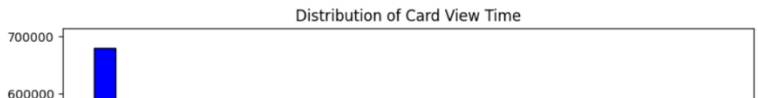
An examination of the **impressTime** variable indicates that there are differences in the way the users engage with the platform on a day of the week as well as an hour of the day. Considering the pattern of engagement over the week (Appendix 1), it can be observed that Friday and Saturday are the busiest days implying that users are more active around the weekends. Other weeks have good records in terms of engagement but not as much as Friday and Saturday. This suggests that the use of the platform could be affected by leisure time or weekend patterns of activities for more people. Moreover, a closer analysis of how this activity varies throughout the day reveals another interesting pattern. The uptake of user activity goes up as the day progresses with sustained peak activity noted in the afternoon hours, especially between 2 PM and 3 PM (Appendix 2) which could be a result of resuming from lunch or coming back from work. Activities start slowing during the early morning hours then pick up as the day wears on. There is a low engagement curve in the evenings, but interactions begin to rise from 10 PM showing that users are very active at night.

These temporal dependencies highlight the significance of user action patterns in making content delivery decisions. The weekly peaks and late-night activity spur curiosity on the relationship between users' activities in relation to content and content categories as well as the varying activity levels among different users. This could extend to include the types of

content and or the age brackets that are engaged during this night's activity. In this way, consisting of these latent temporal dimensions, user content abstinence periods between peak hours, and content release strategies can be enhanced for particular group audiences' engagement at any time.

The **impressPosition** variable, during data validation, the number of records were found to have been registered in error. We noted that 3.21% of observations (3,205,640 records) had an **impressionPosition** of 0, which is invalid as positions should always start at 1. To keep the dataset clean and maintain the dataset's integrity, such observations were excluded from the dataset. Additionally, the distribution of **impressPosition** was analyzed to compute descriptive statistics and identify potential outliers.

In exploring user interaction features such as **isClick** and **mlogViewTime**, some logical inconsistencies arised. Within the dataset, there were 135,677 observations in which there was a click (isClick = 1), but the time spent viewing was either zero or not available (mlogViewTime = 0 or NaN). Such cases were at odds with the presumption that there would be some time spent viewing any card that was clicked upon. To correct this, these cases were discarded from the dataset. We also found 7,938 records that had positive mlogViewTime without any clicks (isClick = 0), which also represented a logical inconsistency. These records were flagged and excluded to maintain dataset accuracy. Notably, the mlogViewTime column, representing the time users spent on viewing cards, showed that 95.2% of mlogViewTime were 0 or null, which suggests that, in most cases, the impression generated did not result in any meaningful view time. This finding highlights the platform's overall low engagement levels and the challenges of capturing user attention. Furthermore,



the analysis revealed a classic long-tail distribution in user and content engagement, where a small subset of users and items accounted for the majority of interactions (*See Fugure 4*).

Other key user interaction features such as **isViewComment**, **isComment**, **isLike**, **isShare**, and **isIntoPersonalHomepage** were analyzed in detail. Logical errors were identified where interactions like likes, comments, or shares were recorded without an initial click (e.g., “isClick” = 0). Since such actions require opening the card, these discrepancies were anticipated as data collection artifacts and were removed.

User behavior was further analyzed using the **userId** column, where we calculated the ratio of clicks to impressions for each user. This metric allowed us to better understand the engagement levels across the user base. These were combined into a new data frame and subsequently analyzed on the distribution of user interactions and engagement rates. The same type of analysis was performed on **mlogId** to identify cards with high impressions and clicks, along with their corresponding engagement rates.

When examining the **songId** column, we found that the most used song and the most popular artist were both identified by the same ID: "AFBHAHLH". Besides, we have analyzed the

distribution of **songId** to identify popular songs, showing that the average popular song was used around 5 times per month, with some notable outliers indicating highly popular tracks. Similarly, we analyzed the distribution of **artistId** to identify popular artists. This revealed that some artists have more songs used by users, indicating potential artist popularity or a larger catalog of appealing music.

To understand the types of content shared, we analyzed the distribution of **type** column, which revealed that image-based content is more prevalent (53.9%) than video content (46.1%) in music logs. Additionally, the **publishTime** column revealed that the average creation date for content is 2 months before the data collection period.

Demographic features, such as **age** and **gender**, provided more insights about the users. The **age** feature was categorized into age groups, creating a new variable that enabled a deeper exploration of user preferences and behavior across demographics. Regarding gender distribution, we found out that male users were 35.31% of the platform's audience, while a significant portion (37%) of users had an unknown gender. Moreover, registration data was also refined by transforming the **registeredMonthCnt** column into a **registrationDate** feature. Rows with registration dates before April 2013 were excluded, assuming these were erroneous as the application launched in that month.

4.2 Summary

The EDA revealed significant insights into user behavior and data quality on the NetEase Cloud Music platform. This was aimed at understanding the key variables structure and their relationship with one another to help explain user activity and identify some data quality concerns. The data showed a very low rate of user engagement, with only 3.82% of

impressions resulting in actions, which brings out the difficulties encountered in eliciting and predicting actions when designing recommendation systems. There is a need to understand the drivers of engagement as well as develop advanced techniques towards the enhancement of engagement levels. When analyzing types of interactions, a huge disparity was observed, with comments being very few as compared to clicks, likes, and shares. This result calls for an examination as to how spouses engage in commenting, as it may entail changing the way the system operates to elicit the desired behavior from the users. Additionally, we observed some discrepancies in the data where users appeared to engage (like, share, comment) before clicking on the card, which was flagged for deletion in order to preserve the data for future research.

We further looked at the data in its temporal aspects by analyzing the interactions by considering the day of the week and the hour of the day. The analysis on the other hand needs to be more careful as these engagement variations span more than two time periods. Interaction distribution concerning cards (mlogID) and users was characterized by a famous "long-tail" distribution where a few cards and users made the majority of the interactions. This particular aspect, which is usually found in social media sites, underlines the importance of effective recommendation systems for both the appropriateness of content and the extent to which it's extreme and includes both highly demanded and less utilized resources.

The outcomes of this first probe provide a basis for the disentangling of user behavior and data quality within the realm of the online platform. By spotting inconsistencies in the data, dealing with sparse and imbalanced interaction patterns, and the long tail of user-item mappings, we will be able to formulate better and more precise strategies for modeling and making predictions.

5. Methods

While developing our recommender systems, we applied a diverse range of approaches to address the unique challenges of the NetEase Cloud Music platform. We designed some methodologies integrating similarity-based techniques, user preference elicitation, content-based recommendation systems, and deep neural network models inspired by the DeCS framework.

1. **Discriminant Frequent Item-sets Recommendation Algorithm:** A K-means clustering algorithm was applied to segment users based on their demographic profiles. The most frequent music sets were then extracted from each resulting cluster.

5.1 Evaluating Recommender Systems

Our dataset includes information such as dense user demographic data, creator statistics, impression data, and detailed activity logs. The entire computational effort is highly demanding due to its large number of users and interactions, especially when doing one-hot encodes, creating new variables, and other types of machine learning methods. Given the high number of processing threads, the dataset poses significant challenges and joint limitations in terms of computational load, particularly concerning RAM requirements for processing and analyzing the data. We wanted to refer to the capacities used, which in this case were a maximum of 51 GB of RAM, 225 GB of disc storage and an additional premium of 100 compute units. To manage these computational limitations, we chose to work with a reduced version of the data set - selecting a 5 % sample of all the data. This sampling approach allowed us to carry out an initial exploratory analysis and test our models without overloading our available system memory. While working with a larger subset or the full

dataset could potentially produce more far-reaching perceptions, the computational resources available limited our ability to do so effectively.
By first analyzing this sample, we ensured that our methodology could be tested and repeated before attempting to scale up to a more demanding computational workload. Not only did this approach allow us to validate our pipeline in more manageable segments, but it also provided a practical basis for potential scaling as the appropriate resources become available.

6. Discriminant Frequent Item-sets Recommendation Algorithm

The code implementation is based on a simple recommendation algorithm using Discriminant Frequent Itemsets (*See Figure 13*), as illustrated in the provided flowchart and pseudocode from the paper (Panteli e Boutsinas 2023).This algorithm aims to generate personalized product recommendations for users by leveraging clustering techniques and discriminant frequent item sets. All the steps are listed in the below structured points:

9.1 Feature Selection for Users

The code first performs one-hot encoding on the categorical variables "gender" and "province." This transformation converts categorical data into binary columns, allowing machine learning algorithms to process them effectively. This is essential as it ensures the data can be interpreted numerically, since most machine learning models require numerical input.

9.2 Handling Missing Age and Registration Data

First, the registration date, originally recorded in a Year-Month format, was converted into the number of days since the user's registration. This was achieved by computing the

difference between the current date and the registration date. This transformation provides a **numeric representation** of the time elapsed since user registration. We also applied the same process to the age_group column, which contained categorical values representing age ranges (e.g., '18-24', '25-34'). Missing values were replaced with an appropriate proxy. Each age group was assigned an average age (e.g., '18-24' was set to 21). The average age for the entire dataset was then calculated, and the closest age group to this average was determined by minimizing the absolute difference between the average age and the midpoints of the predefined age ranges. This closest age group was used to fill in the missing entries in the age_group column.

9.3 Variable Selection for the User Matrix

To ensure that the clustering process focuses on the **most relevant** and **non-redundant features**, we implemented a systematic approach to variable selection. The objective was to eliminate multicollinearity, reduce dimensionality, and retain only the most informative features, thereby enhancing the quality of the clustering process and subsequent analyses. The process began with calculating the **Variance Inflation Factor** (VIF) for all numeric variables in the dataset. VIF measures how much the variance of a regression coefficient is inflated due to multicollinearity among independent variables. Features with VIF values exceeding 10 were considered highly collinear and unsuitable for clustering (*See Table 9*). These variables were subsequently excluded to prevent redundancy and improve model interpretability. In addition to VIF analysis, domain knowledge was applied to refine feature selection further.

Highly correlated variables, such as dt and day, were evaluated, and only one was retained based on relevance. Similarly, redundant variables like mlogViewTime and mlogViewTime_minutes, as well as registeredMonthCnt and days_since_registration, were compared, and only the more meaningful feature from each pair was retained. Additionally, isClick, being the target variable for recommendation, was excluded from clustering to avoid data leakage. The final set of selected features comprised a mix of behavioral, demographic, and temporal variables. The following information represents the variables used for the user matrix and their VIF values to justify their usage:

Variable	VIF Value
impressPosition	1.114139
isIntoPersonalHomepage	1.03266
isShare	1.044347
isViewComment	1.215754
isLike	1.098962
hour_of_day	1.052796
day_of_week	1.055234
publishTime	1.18548
type	1.307284
talkId	1.115851
province	1.277776
age	4.458588
gender	1.141855
followCnt	1.419745
level	2.313002

Table 3 - VIF Values

For age-related variables, only one between age and age_avg was retained to prevent redundancy. To prepare the data for clustering, the selected features were standardized using the StandardScaler. However, only continuous variables (publishTime and followCnt) were scaled, as standardizing categorical or binary variables may distort their interpretability. This preprocessing ensured that all features contributed equally to the clustering process,

preventing dominance by variables with larger numerical ranges. By combining statistical measures such as the Variance Inflation Factor (VIF), the cold user's domain knowledge, and appropriate data curation for clustering, we ensured that the user matrix was optimized for k-means initialization. This process retained only the most relevant and interpretable features, removing redundant and highly collinear variables. Following the VIF-based selection, we further evaluated the retained features by analyzing their correlations. Since the correlation matrix did not reveal any particularly strong relationships (i.e., correlations exceeding 40%), we decided to include all the variables that passed the VIF selection threshold. This approach ensured that we did not inadvertently exclude potentially useful features, given the relatively low levels of multicollinearity and weak correlations observed among the variables. These previously mentioned features were extracted from the filtered dataset, and duplicate entries were removed to ensure that each user was represented uniquely. This matrix provides a structured representation of users, capturing their distinct profiles and behaviors, which are essential for clustering users into groups with similar characteristics.

9.4 User-Product Interaction Matrix

To analyze user engagement with the different music pieces, we created the User-Product Interaction Matrix. This matrix captures interactions between users (userId) and music items (mlogId), using weighted interaction scores to represent the intensity of engagement. Interaction types: clicks, likes, and shares, were assigned different weights (isClick: 1, isLike: 2, isShare: 3) to reflect their relative significance in terms of user interest. Those weights were given based on the priority of importance of each meaning, for instance a share is much more important than a like, and the same we can say regarding one like and ine click. The interaction score for each user-item pair was calculated as a **weighted sum of these**

interaction types. To account for multiple interactions with the same item by a user, the maximum interaction score was retained for each user-item pair. The resulting data was then pivoted into a matrix format, with users as rows, content items as columns, and interaction scores as cell values. Missing values were filled with zeros to indicate no interaction, and all values were converted to integers for consistency.

9.5 User and Item Coverage and next steps

The metrics for **user coverage** (19.26%) and **item coverage** (12.86%) highlight the significant sparsity of the user-product interaction matrix in this specific sample dataset. Specifically, the low user coverage indicates that most users (over 80%) have no recorded interactions, classifying them as cold users. Similarly, the low item coverage shows that most items lack sufficient user interactions, further emphasizing the sparsity challenge.

This sparsity can be seen as a limitation of the model, as it inherently reflects the extent of the cold-start problem. However, this limitation also serves to accentuate the cold-start challenge, which is a fundamental issue in recommendation systems. To address this, we adopted two types of strategies in this model:

1. **Leveraging Similar User Characteristics:** By clustering users based on similar demographics or characteristics, combined with interaction scores, recommendations can still be generated for users with minimal or no prior interaction history. This approach helps to mitigate the lack of interaction data for many users.
2. **Inclusion of External Features and Advanced Techniques:** External features, such as user demographics and item metadata, were previously incorporated into the model

to provide additional context. Moreover, advanced techniques like discriminant frequent itemsets were applied to improve recommendation quality.

9.6 Cluster Extraction and Selection of Optimal Number of Clusters

Initially, the user matrix was constructed using selected variables that passed the variance inflation factor (VIF) and correlation filtering criteria. However, the variable impressPosition was excluded from the analysis due to recurring computational issues. Including this variable caused the clustering code to run out of memory, leading to crashes. Despite attempts to optimize the code, errors persisted, such as the one arising when the cluster_users indexer contained multiple dimensions instead of a simple one-dimensional index. These limitations mandated its removal from the user matrix. The clustering process utilized the KMeans algorithm with a predefined range of cluster numbers (2 to 5). The algorithm was applied to the standardized user matrix, excluding userId and talkId, to ensure unbiased clustering results. Three primary evaluation metrics were employed to identify the optimal number of clusters:

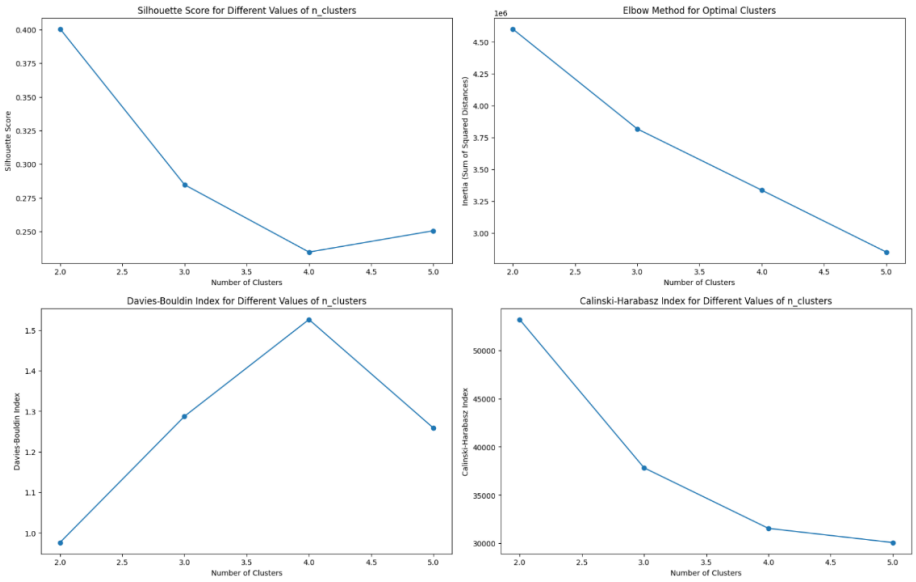


Figure 5 - Number of cluster analysis

- **Silhouette Score:** This metric measures the cohesion and separation of clusters. Higher silhouette scores indicate well-defined clusters. As shown in the silhouette score graph, the score declined steadily with an increasing number of clusters, peaking at 2 clusters and showing a reasonable value at 3 clusters.
- **Elbow Method:** This method evaluates the within-cluster sum of squares (inertia) across different cluster numbers. The elbow point, where the inertia curve begins to flatten, suggested 3 clusters as a balanced choice.
- **Davies-Bouldin Index and Calinski-Harabasz Index:** These metrics provided additional insights into cluster compactness and separability. The Davies-Bouldin index favored 4 clusters, while the Calinski-Harabasz index indicated 2 or 3 clusters as optimal options.

Based on the mentioned metrics (*Table 10*), the decision was made to proceed with 3 clusters. While the silhouette score slightly favored fewer clusters, 3 clusters provided a balance

between interpretability and granularity, aligning with the objectives of the study (*Figure 12*). We have the following Silhouette Score values:

# Clusters	Silhouette Score
2	0.400434255
3	0.284716956
4	0.234710158
5	0.250539831

Table 4 - Silhouette Score Values

9.7 Frequent Itemsets Extraction for Clusters

To gain deeper insights into the behaviors and preferences of users within each cluster, the Apriori algorithm was applied to identify frequent itemsets for each cluster. This process was executed by first grouping users based on their assigned clusters and forming a user-product binary matrix for each cluster. This matrix was derived from the user-product interaction matrix, ensuring that values were transformed into binary (1 or 0) to indicate interaction or no interaction, respectively. The Apriori algorithm was used to extract frequent itemsets for each cluster, using a minimum support threshold of 0.05. This threshold ensured that only individual or groups of music with interactions from at least 5% of users in the cluster were included in the analysis, despite the cold start problem affecting approximately 80% of the initial users from the beginning. The results revealed distinct patterns of interaction within clusters, capturing specific sets of items that were frequently interacted with by users in the same group. All clusters produced frequent itemsets, with Cluster 1 having 11, Cluster 0 having 54, and Cluster 2 having 64 frequent itemsets. These itemsets represent groups of

items frequently interacted with by users within each cluster. The presence of frequent itemsets across all clusters confirms that the clustering process successfully grouped users with similar interaction behaviors. The variation in the number of itemsets per cluster reflects the diversity of user behaviors and cluster sizes. The inclusion of frequent itemsets as discriminant factors within clusters enhances the recommendation system by enabling tailored suggestions based on shared user behaviors. This approach aligns with the goal of addressing cold-start and sparse data challenges by using patterns unique to each cluster. Limitations in this process include potential performance bottlenecks due to large matrix dimensions and sparse data. These issues were mitigated by adjusting the minimum support threshold and ensuring computational efficiency throughout the analysis.

9.8 Extracting Discriminant Frequent Itemset Per Cluster

To better put on practice the interpretability and specificity of the frequent itemsets derived from each cluster, we extracted discriminant frequent itemsets. Discriminant itemsets are that music or set of music that are unique to a particular group of individuals, meaning they are not present in any other cluster. This step ensures that the itemsets identified are distinct to the behavioral patterns of users within a specific cluster, making them valuable recommendations.

- **Frequent Itemsets Extraction for Each Cluster:** The frequent sets previously generated for each cluster were gathered. These sets represent groups of different musics commonly interacted with by users within that cluster.

- **Comparison Across Clusters:** For each cluster, the frequent itemsets from all other clusters were combined into a single set. This formed a reference point for identifying uniqueness.
- **Identifying Discriminant Itemsets:** The frequent itemsets of the current cluster were compared against the combined set of itemsets from all other clusters. Discriminant itemsets were defined as those appearing exclusively in the current cluster, ensuring their uniqueness, in other words, their likes.

This method ensures that the extracted itemsets truly represent the distinct interaction patterns and preferences of users within each cluster. By doing this method, the recommendation model avoids generic and popular suggestions and instead divides different recommendations to the unique characteristics of each cluster. This specificity is important to improve recommendation quality and addressing the cold-start problem, as these itemsets are derived from user groups with shared behavioral traits, highlighting differences in user-product interactions across different people.

9.9 Extracting Discriminant Frequent Itemsets Per Cluster

Now we try to recommend music by managing the discriminant frequent itemsets extracted for each cluster to each user with no prior interactions (cold-start users). We begin by evaluating each user in the user-product matrix to determine if they are a cold-start user. This is done by checking whether the sum of interaction scores (e.g., clicks, likes, shares) across all items is zero or not. Then, for each cold-start user, their corresponding cluster is identified using the user matrix. This association ensures that the recommendations are tailored to the user's group. Discriminant frequent itemsets specific to the user's cluster are retrieved, and

thus, for each itemset in the user's cluster, the music is marked as interacted with (value of 1) in the user-product matrix. This simulates recommendations adapted to the user's cluster.

9.10 Offline Evaluation of the Recommendation System

The aim of this evaluation is to analyze the methodology's performance in forecasting user preferences and to confirm its efficacy in relation to conventional methods. The metrics for evaluation used were determined through a K-fold cross-validation procedure.

To evaluate the model, we implemented a 45-fold cross-validation technique, as stated in (Panteli e Boutsinas 2023). The user matrix was split into ten equal subsets, with 44 subsets used for training and one for testing in each iteration. This process ensures that all users are evaluated and prevents overfitting by testing on unseen data. The number of folds was determined by ensuring a balance between variance and bias. When we increase k quantity of folds, each fold is getting smaller in size for testing, and the training set becomes relatively larger. That means that we have more of the data used for training in each fold, reducing bias in the evaluation. Otherwise, it increases variance, because smaller subsets may not represent the full distribution of the data as it is. So we found a balanced point where For the training set, the user-product interaction matrix was constructed to capture the relationship between users and items. The discriminant frequent itemsets, previously derived for each user cluster, were used to generate recommendations for users in the test set. For the training set, the user-product interaction matrix was constructed to capture the relationship between users and items. The discriminant frequent itemsets, previously derived for each user cluster, were used to generate recommendations for users in the test set.

The evaluation focused on three key metrics:

- **Precision:** The proportion of recommended items that were relevant.
- **Recall:** The proportion of relevant items that were successfully recommended.
- **F1-Score:** The harmonic means of precision and recall, providing a balanced measure of the system's performance.
- **Hit Rate:** The proportion of users for whom at least one true item is present in the recommendations.
- **MRR (Mean Reciprocal Rank):** Measures the rank of the first relevant item in the recommendation list. A higher score means relevant items are ranked higher on average.
- **NDCG (Normalized Discounted Cumulative Gain):** Measures the ranking quality of recommendations. Rewards place true items at higher ranks while normalizing for the ideal ranking.

The results shown are extremely low in all the recommendation metrics, and this is true even when they are converted into percentages. For example, when recommending just one song (Top 1), the precision is around 0.66%, while the recovery is less than 0.4%. When we increase the number of recommendations, from Top 1 to 3, 5 and 10, we see a slight improvement in recall because suggesting more items naturally increases the chances of including something that the user might consider relevant. However, adding more recommendations also weakens precision, as additional, less relevant items are more likely to be included. Since this model works on the basis of the most popular recommendation that a consumer is most likely to like, depending on the particular preferences of each cluster, increasing the number of k's increases the variance contained and is therefore a limitation. Other metrics such as the MRR and NDCG, which consider both the correctness and quality

of the classification, suggest that even when the model guesses correctly, these relevant recommendations are not prominently displayed. This is an underlying recommendation strategy. If the model is based only on the popularity of songs, similar to the use of the **Apriori algorithm**, which identifies frequent item sets, it can recommend items that are generally popular but not necessarily aligned with the user's individual preferences. Which means that in this method, without the personalization factor, there is little chance of getting it right

To initialize the model and conduct the experiments (*See Figure 13*), we initially used 5% of the impression data as a sample, aligning with the approach adopted by the previous models. This initial percentage allowed us to explore the model's behavior with a significant data volume, ensuring a robust preliminary evaluation. However, during the process, we encountered limitations related to computational costs and memory usage, particularly given the maximum **RAM capacity** of 51 GB available for execution. Processing the initial sample proved computationally expensive, regarding this type of analytical method, leading to execution errors and excessive resource consumption. To address this, we opted to reduce the sample size to 2.5% of the impression data, maintaining a sufficient volume to test the model while reducing the computational load to manageable levels. This reduction was crucial to ensure the stability of the execution environment and the feasibility of completing the experiments.

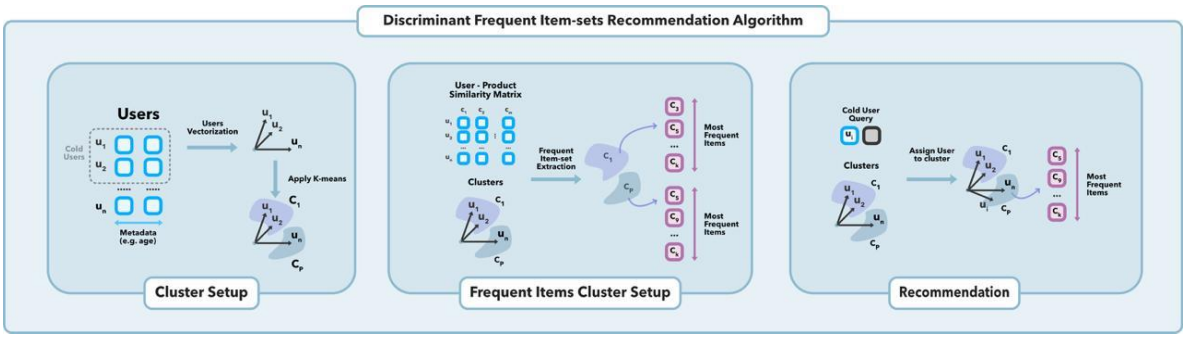


Figure 6 - Discriminant Frequent Item-sets Recommendation Algorithm structure

10. Conclusion and Discussion

10.1 Key Findings

This document explores how data manipulation, exploration, and machine learning can be leveraged to develop recommendation systems for the **NetEase Cloud Music Platform**, specifically targeting *cold-start users*. Cold-start users are those with low activity levels and recent interactions, making it difficult to provide accurate and meaningful recommendations due to the lack of sufficient user information.

To address this challenge, four distinct approaches were implemented:

1. **Frequent Item Set Approach** – leveraging item clustering to generate recommendations.

Finally, in the **Discriminant Frequent Item Set Model**, the identification of frequent item sets within clusters enabled the system to capture shared user behaviors unique to each group. These discriminant frequent item sets allowed for **personalized recommendations** for cold-start users, ensuring the suggestions closely aligned with the group's specific interaction

patterns. This approach improved the relevance of recommendations for users with no prior activity. The successful identification of item sets across all clusters further validated the clustering process, demonstrating its effectiveness in grouping users with similar behaviors and preferences. By leveraging the most popular songs as the foundation for recommendations, the system remains **simple to implement** and easy to interpret, as it relies primarily on counting frequencies and selecting the most played items. This **popularity-based approach** is particularly effective for new users without listening histories, as it exposes them to well-known and appealing tracks. However, this simplicity comes with trade-offs. Key metrics such as **precision**, **recall**, **MRR**, and **NDCG** were notably low, indicating that the model struggles to adapt its suggestions compared to more sophisticated systems. While increasing the recommendation list size from **Top 1 to Top 10** improves recall slightly, it simultaneously reduces precision, resulting in a diluted and less meaningful user experience.

To evaluate and compare these systems, we conducted a comprehensive analysis using **five recommendation evaluation metrics per model**, which are now consolidated into a single representation for clarity and comparison (*Table 11*).

Metric	k	User Similarity-Based	Users Preference Elicitation	DeCS-Inspired	Discriminant Frequent Item-sets
Precision	1	2.27	0.49	2.15	0.66
	3	3.03	0.52	2.49	0.79
	5	2.27	0.47	5.23	0.68
	10	2.27	0.49	23.44	0.43
Recall	1	2.27	0.02	1.97	0.38
	3	9.09	0.08	2.87	1.36
	5	11.36	0.12	5.5	1.96
	10	22.73	0.12	28.75	2.45
HitRate	1	2.27	0.49	2.15	0.17
	3	9.09	1.36	3.23	0.55
	5	11.36	2.14	6.45	0.69
	10	22.73	2.14	32.56	0.71
MRR	1	2.27	0.49	2.15	0.17
	3	5.3	0.83	2.51	0.33
	5	5.76	0.98	3.21	0.37
	10	7.35	0.98	13.66	0.37
nDCG	1	2.27	0.49	2.15	0.17
	3	6.28	0.50	2.48	0.24
	5	7.16	0.46	4.02	0.25
	10	10.91	0.30	15.81	0.23

Table 5 - Comparison of Recommendation Models Across Evaluation Metrics

The **Discriminative Frequent Itemsets** model introduces a unique framework by clustering users and items to recommend frequently occurring yet highly discriminative items. While this approach marginally improves recall and hit rates at larger *k* values, it falls short in comparison to the DeCS-Inspired model when evaluated using ranking metrics like MRR and nDCG. This indicates that its optimization for ranking quality and personalization remains less competitive.

- **Cluster-Based Recommendations**: The **Discriminative Frequent Itemsets** model is valuable for specific applications that benefit from group-level clustering and frequent item identification. However, it is less effective for personalization and ranking optimization.

To maximize performance across diverse user scenarios, a **hybrid approach** could combine the strengths of these models. For instance, integrating demographic-based methods for cold-start users with the DeCS-Inspired model for data-rich personalization could create a flexible solution that adapts to varying levels of user engagement and data availability.

10.2 Limitations

Understanding the inherent limitations of our recommendation models is critical for their ongoing development. By identifying weaknesses, we can prioritize improvements and fine-tune parameters to deliver more individualized, effective, and equitable experiences for both new and returning users.

The **Discriminant Frequent Item-Set Recommendation Algorithm** also encounters limitations tied to its dependence on **clustering** and **frequent pattern mining**. To avoid memory and stability issues, the dataset must be significantly reduced (to **2.5%** of its original size), which restricts the depth and generalizability of the analysis. The exclusion of certain variables to maintain execution viability further limits the range of signals the model can leverage, impacting both accuracy and recommendation diversity. While the use of *k-fold cross-validation* provides a straightforward evaluation approach, the reliance on group-level item sets, rather than personalized classification strategies, underrepresents the potential of more robust and individualized methods. Additionally, the evaluation process does not exclusively focus on cold-start users—those with no prior interactions or minimal data—

which means the metrics may not fully reflect the system's performance in handling completely new audiences. While this is not a critical flaw, it does indicate that the current approach may overlook challenges faced when encountering data-poor users or scaling to larger, more complex algorithms.

In summary, each model faces distinct limitations, including a lack of dynamic adaptation, limited feature granularity, computational bottlenecks, and exclusionary strategies that fail to account for specific user segments. A shared challenge across all models is the reliance on **offline evaluation**, which cannot fully capture the platform's dynamic nature or evolving user preferences. This reduces the long-term relevance and adaptability of the recommendations.

By acknowledging these shortcomings, future research can build upon these findings to develop more **integrative**, **scalable**, and **context-aware** solutions, ultimately improving the handling of the cold-start problem and enhancing user experiences across diverse contexts.

10.3 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. By leveraging user interaction data, card metadata, and user metadata, the research successfully addresses the challenge of providing accurate and relevant recommendations for new users with limited interaction histories. A major contribution of this study is the demonstration that integrating **diverse data sources** offers a holistic approach to understanding user preferences. The tested models 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.

Furthermore, the study provided **four distinct approaches** to mitigate the cold-start problem:

- Two **metadata-based methods** to infer similarities and recommend items.
- A **clustering-based approach** that extracts discriminant item patterns to enhance recommendations for specific user segments.
- A **state-of-the-art-inspired method** combining user and item embeddings for advanced personalization.

This variety of approaches challenges the prevailing trend in recent research, which often prioritizes machine learning techniques while neglecting other effective strategies. The success of these alternative methods represents an **innovative and out-of-the-box contribution**, demonstrating their viability as solutions to the cold-start problem. Another key contribution lies in the **rating computation framework**, where interaction weights are attributed based on the rarity or frequency of each interaction type. This approach provides a **simple, interpretable, and intuitive** way to evaluate interactions while aligning with each model's specific data requirements.

Complementing this, the study highlights the effective use of **metadata fields**—such as registration dates, interaction counts, and activity levels—to tailor recommendations and evaluate initial recommendation sets. The combination of metadata with side information serves as a powerful method to improve personalization, as demonstrated in one of the models.

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.

Ultimately, this research highlights the potential for **economic and reputational gains** by leveraging technology to enhance user experience—from initial onboarding to long-term engagement. By utilizing recommendation systems as a **differentiating factor**, platforms can ensure sustainable growth and greater user loyalty in an increasingly competitive landscape.

10.4 Implications for Practice

A new user joins NetEase Cloud Music, hungry to know more yet not knowing how to begin. They open the app, and, instantaneously, the platform must ask itself: How do we make this moment matter? Without a listening history, the system relies on what it knows—demographic data, registration context, and perhaps a hint of regional trends. Referred to as the cold start problem, it 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. From a practical perspective, the findings of this study aim to provide actionable insights for enhancing user experiences on NetEase's digital music platform, fostering greater user engagement, and boosting retention in the competitive and volatile digital media landscape.

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. Possibly, an 18-year-old who lives in a teeming city might be pulled into the trendiest pop hits, while someone who is older and comes from a quieter environment might appreciate instrumental or folk tracks. These would give someone that personal welcome—a very impersonal content library becomes a more curated invitation into exploration. The implications of solving the cold-start problem extend far beyond that first impression. It should incite exploratory activity and discovery. A listener on pop should be interested into a more nearby genre, such as indie or electronica, which would broaden horizons and create a deeper experience. Meanwhile, based on demographic trends, new content—emerging artists or new releases without much interaction data—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. For the music streaming industry, the cold-start solution is a step toward making the whole system understood by users and embracing all creators with equal opportunities to be discovered.

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.

10.5 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 data lacks critical informational and analytical components necessary for both interpretability and efficiency in model development. For example, there is no clear understanding of activity levels, creator types, context, or talk IDs, which could have been leveraged for Exploratory Data Analysis (EDA) or clustering.

Additionally, the meaning of a "card" remains unclear, hindering deeper analysis. 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 also true for 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, data spanning only one month introduces inefficiencies. 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. For instance:

- The **User Preference Elicitation Recommendation method** depends solely on existing data and lacks a learning process. Incorporating machine learning into this

method, such as a dynamic weighting system for metadata, could lead to more personalized and effective recommendations.

Across all these models, additional computational resources and larger datasets are critical for improving results, especially for cold-start users.

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. Protecting system integrity ensures both user trust and platform authenticity.

Recommendation systems hold significant cultural power by influencing what users discover and how creators are perceived. Future systems must prioritize diversity, exposing users to new genres, artists, and perspectives. Beyond fulfilling technical goals, these systems should encourage exploration and cultural growth, enriching the user experience.

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.**

11. 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.*

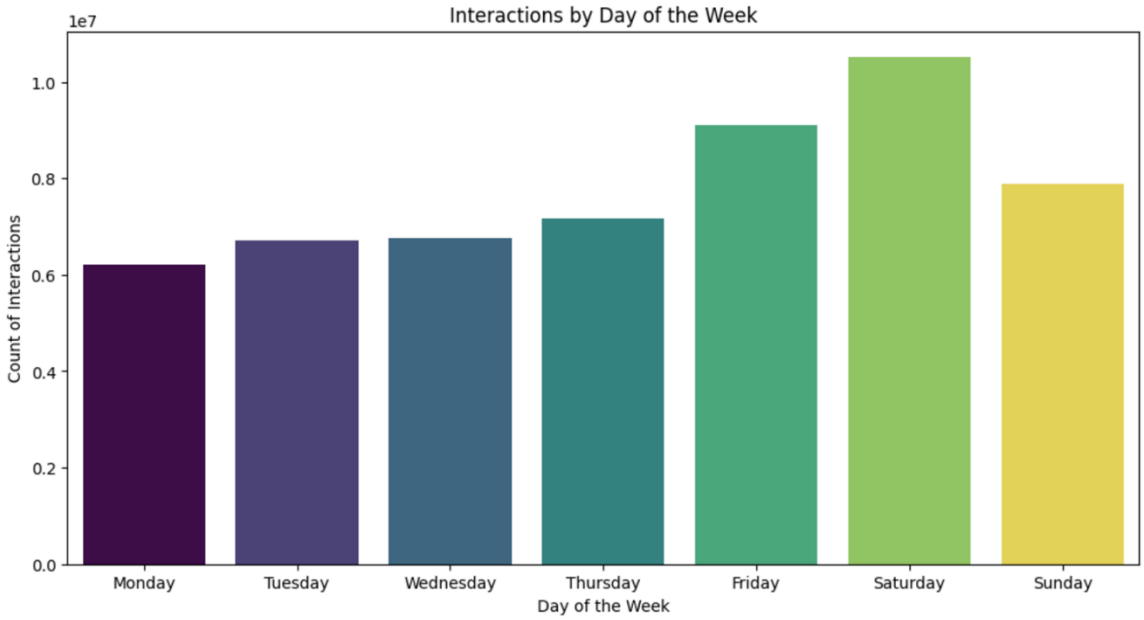
Wang, Shixiong, Wei Dai, e Geoffrey Ye Li. 2024. “Distributionally Robust Outlier-Aware Receive Beamforming.” *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP).* 1–6.

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/.

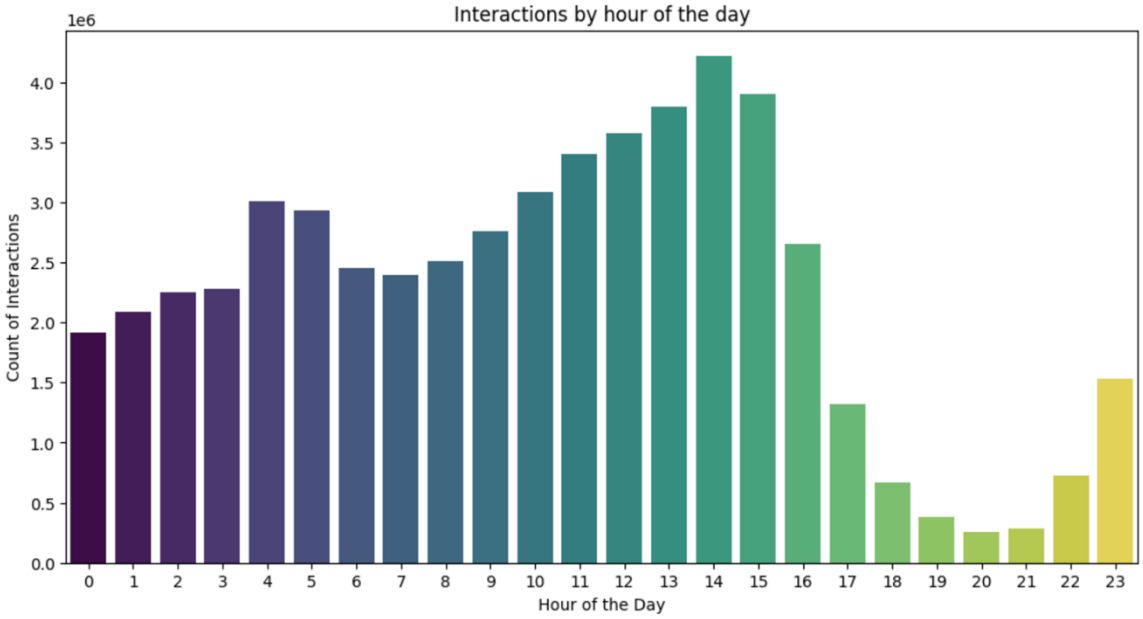
Zhang, Dennis J., Ming Hu, Xiaofei Liu, Yuxiang Wu, e Yong Li. 2022. “NetEase cloud music data.” *Manufacturing & Service Operations Management* (INFORMS) 24: 275–284.

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.

12. Appendix



Appendix 1 - Interactions by day of week



Appendix 2 - Interactions by hour of the day

```
all_int = len(sample_interactions)

rates = {}
columns_to_calculate = ['isClick', 'isComment', 'isIntoPersonalHomepage', 'isShare', 'isViewComment', 'isLike']

for column in columns_to_calculate:
    event_count = sample_interactions[column].sum()
    rate = (event_count / all_int) * 100  # Calculate rate as a percentage
    rates[column + '_rate'] = rate

# Display the results
for column, rate in rates.items():
    print(f"{column}: {rate:.2f}%")
```

```
isClick_rate: 99.99%
isComment_rate: 0.12%
isIntoPersonalHomepage_rate: 1.55%
isShare_rate: 0.43%
isViewComment_rate: 17.80%
isLike_rate: 4.19%
```

```
total_impressions = len(all_impression)

rates = {}
columns_to_calculate = ['isClick', 'isComment', 'isIntoPersonalHomepage', 'isShare', 'isViewComment', 'isLike']

for column in columns_to_calculate:
    event_count = all_impression[column].sum()
    rate = (event_count / total_impressions) * 100  # Calculate rate as a percentage
    rates[column + '_rate'] = rate

# Display the results
for column, rate in rates.items():
    print(f"{column}: {rate:.2f}%")
```

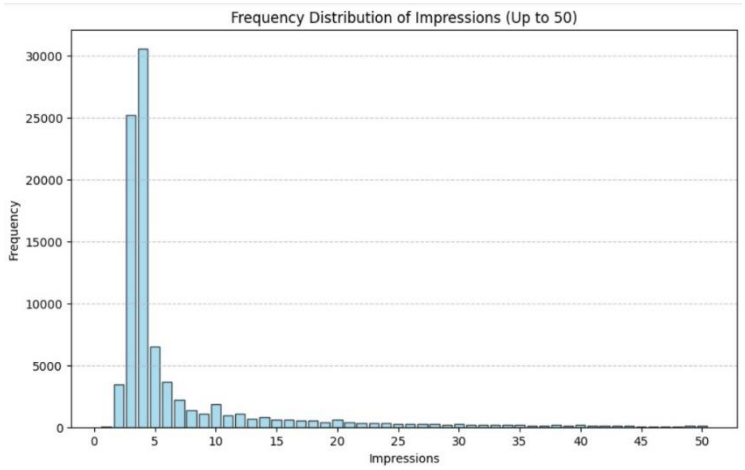
```
isClick_rate: 4.97%
isComment_rate: 0.01%
isIntoPersonalHomepage_rate: 0.06%
isShare_rate: 0.03%
isViewComment_rate: 0.74%
isLike_rate: 0.24%
```

Appendix 3 – Interaction Rarity across interactions and impressions (DeCS inpired approach)

```
sample_rating = sample_rating.copy()

adjustment_factors_imp = {
    'isClick': 0.9503,
    'isComment': 0.9999,
    'isIntoPersonalHomepage': 0.9994,
    'isShare': 0.9997,
    'isViewComment': 0.9926,
    'isLike': 0.9976
}

adjustment_factors_int = {
    'isClick': 1,
    'isComment': 0.9988,
    'isIntoPersonalHomepage': 0.9845,
    'isShare': 0.9957,
    'isViewComment': 0.822,
    'isLike': 0.9581
}

weights = {
    'isClick': 1,
    'isComment': 1,
    'isIntoPersonalHomepage': 1,
    'isShare': 1,
    'isViewComment': 1,
    'isLike': 1
}

sample_rating['impression_rating'] = 0

for column in adjustment_factors_imp.keys():
    factor = adjustment_factors_imp[column]
    factor2 = adjustment_factors_int[column]
    weight = weights[column]

    sample_rating[column] = sample_rating[column] * factor * weight * factor2

    sample_rating['impression_rating'] += sample_rating[column]

sample_rating['weighted_time_rating'] = sample_rating['impression_rating'] * (1 + sample_rating['scaled_mlogviewtime'])
```

```
min_rating = sample_rating['weighted_time_rating'].min()
max_rating = sample_rating['weighted_time_rating'].max()
sample_rating['weighted_time_rating'] = (sample_rating['weighted_time_rating'] - min_rating) / (max_rating - min_rating)
```

Appendix 4 – Rating Making Code: DeCS Inspired approach



Appendix 5 – Frequency Distribution of all Impression Position Values