HSLU Lucerne University of Applied Sciences and Arts

Recommender Systems Report

Recommender Systems
Block Seminar
Spring Semester 2025

Team Members:
Rafaella Miranda-Sousa Wasser
Luca Renz

Lecturer:
Guang Lu

February 2025

Table of Contents

1. Introduction

1.1. Background

Deezer, established in France in 2007, is a dynamic music streaming service boasting over 90 million tracks, spanning various genres along with podcasts and radio stations. Available globally in over 180 countries, Deezer supports a myriad of devices, enhancing accessibility for users worldwide. As of 2025, the service has attracted about 9.6 million subscribers, establishing itself as a prominent independent music streaming platform. A key feature of Deezer is its "Flow" technology, which leverages advanced recommender systems to craft personalized playlists, significantly enhancing user experience (Benitez, 2025).

This paper delves into several strategies to refine these recommender systems, which is part of an academic project within the "Master of Science in Applied Information and Data Science" program at the Lucerne University of Applied Sciences and Arts (Lu, 2025).

1.2. Motivation

As the first recommender system to be developed by the project team, the goal was to find a topic of interest. Soon it has been decided to not follow the crowd exactly but rather by adding a small point of innovation to it. Music recommendations will be given in descending order, best recommendation first. However, the recommendations were split into three different classes. Liked, Enjoyed and Loved, marked by 1 or 3 stars depending on the assigned class. This should appeal the user to listen to the songs that are considered to be a match that will be listened for a long time by the user and added to the all-time favourite songs of that specific user.

1.3. Research Questions

The given assignment forces the team to address the following questions exclusively in separate chapters followed:

1. How would the project team develop a recommendation algorithm to win this competition?
2. What would the project team propose to solve Deezer's general recommendation problems?
3. Do the two solutions above overlap? If so, in what way and why or why not?

1.4. Applied Strategy

How would the project team develop a recommendation algorithm to win this competition?

To develop such an algorithm, the team followed a simple, yet effective plan as followed:

1. Exploratory Data Analysis (EDA)

To understand, with what kind of data the team was working on, the first phase was the EDA. By doing this, insights were gathered into the distributions, correlations between features, missing or invalid data as well as getting a good understanding of tracked users, its songs, when they listened to their favourite songs and artists and much more.
Details for EDA and Data Cleansing and Preparation can be found in this Jupyter Notebook.

2. Data Cleansing and Preparation

Once the data was understood and basic data cleansing was done such as setting up data types, etc., to ensure better generalization and remove what was identified as outliers or records out of scope., the following steps were taken:

- Rare songs (media_id): Songs with very few interactions were removed.
- Users with limited listening history: Users with very few streams were filtered out.
- Underrepresented genres (genre_id): Genres with low representation were excluded.

Thresholds were dynamically determined using the 25th percentile rather than hardcoded values.
The next step was feature engineering for which additional features were extracted to enhance the dataset's contextual understanding, such as User Activity-, Time-Based, Popularity-Based and Temporal Features to capture general user preferences and potentially find upcoming trends.

3. Define Evaluation Metrics and Models to apply

With the given response variable being of type Binary, it has been decided to not go for accuracy but rather for the F1-Score. More on this to come below.

When defining the evaluation metrics and models to be applied, the team wanted to answer the simple question: "Can simple and traditional model-based collaborative filtering methods outperform (hybrid) Neural Collaborative Filtering when it comes to music recommendations?"

4. Developing Models

After learning about multiple models to apply, the team decided to first implement a baseline model, followed by more advanced models for evaluation and comparison. The goal is to assess performance improvements across different approaches. Each model will be explained in detail in the following subchapters.

Baseline Model: Model-Based Collaborative Filtering

- **Alternating Least Squares (ALS)** was chosen as the baseline model, representing a traditional matrix factorization approach commonly used in collaborative filtering.

Advanced Models

- **Singular Value Decomposition (SVD),** another matrix factorization method, is evaluated as an alternative to ALS.
- **Neural Collaborative Filtering (NCF)** extends traditional methods by incorporating deep learning to model complex user-item interactions.
- **Hybrid Neural Collaborative Filtering (Hybrid-NCF)** further enhances NCF by integrating additional features or combining multiple recommendation approaches to improve predictive accuracy.

5. Evaluate Models

Each model was trained and evaluated using F1-score, precision, and recall, allowing for a clear comparison with more advanced techniques. Nevertheless, the F1-score was primarily chosen as it provides a harmonic mean of precision and recall, making it robust for imbalanced binary classification.

6. Compare Models

After evaluating the models, the evaluation-scores were compared and the best model based on the chosen metric was chosen.

The entire codebase can be found in the gitHub repository.

2. Applied Models

2.1. Model-Based Collaborative Filtering

The following two approaches with all details such as the training process can be found in this notebook.

Alternating Least Squares (ALS) – Baseline Model

The first approach implemented was Alternating Least Squares (ALS), which served as the baseline model.

ALS is particularly effective for implicit feedback scenarios, such as user-song interactions, where explicit ratings are not available (Sidana et al., 2021). Unlike other factorization methods, ALS handles sparse data efficiently by iteratively updating user and item matrices in a computationally scalable manner.

Despite its advantages, ALS has limitations, including cold-start issues, where it struggles to recommend items for new users with little interaction history. Additionally, ALS assumes that all missing interactions are negative (i.e., unobserved interactions mean no preference), which can sometimes introduce biases in recommendations.

The model was trained and evaluated using F1-score, precision, and recall, allowing for a clear comparison with more advanced techniques.

Singular Value Decomposition (SVD)

As an alternative to ALS, Singular Value Decomposition (SVD) was implemented as another matrix factorization approach for collaborative filtering.

SVD transforms the user-item interaction matrix into a lower-dimensional space, capturing hidden patterns and relationships between users and items. Unlike ALS, which is optimized for implicit feedback, SVD is particularly effective for explicit rating-based systems, where users provide direct feedback, such as star ratings or numerical scores.

By identifying trends in these interactions, SVD can generate highly personalized recommendations, especially in datasets with well-structured rating distributions. A key advantage of SVD is its ability to extract latent factors, representing user preferences and item characteristics, which helps improve recommendation accuracy even when data is sparse (Sarwar et al.,2002).

To evaluate its effectiveness, F1-score, precision, and recall were used as performance metrics.

2.2. Neural Collaborative Filtering

The second method that was applied was the Neural Collaborative Filtering, in short NCF.

NCF is a recommendation system framework that uses deep neural networks to model the complex and non-linear interactions between users and items. Research has shown that NCF consistently outperforms traditional matrix factorization methods on several metrics (He et al., 2017).

Therefore, team hoped that this model would significantly outperform the baseline model and improve the quality of recommendations.

Important to mention is the challenge of cold-start issues with the application of NCF without adding Content-Based features into the NCF framework. Nevertheless, in order to be able to compare how much difference such features can make, it has been decided to develop a somewhat basic NCF and apply a second, more complex model that includes such features.

After loading the dataset, the data was appropriately transformed and encoded with the two features *user_id* and *media_id*. Subsequently, it was split into training and test datasets before being processed using various libraries such as Torch and TensorFlow.

Details how the model was defined and trained can be found in the Notebook NCF.

2.3. Hybrid Neural Collaborative Filtering

As stated in the previous subchapter, the NCF got improved by incorporating Content-Based features, making it a Hybrid Filtering Method.
In addition to the media- and user-id, the following other relevant features were added.

- Gender of user
- Platform (used operating system by user)
- Genre
- Artist
- Age of user
- Song popularity based on the last 7 days
- When the user last listened to a song (last_listen)

The application of the model is almost identical to the previous model. However, with the inclusion of above user- and item-based features, the cold-start issues have been reduced.

For details, please refer to the Notebook Hybrid-NCF.

3. Evaluation and Comparison

3.1. Evaluation Metrics

The list below shows the metrics which were chosen by the project team to evaluate and compare the performances of each model.

Precision: In the context of Deezer, precision indicates the accuracy of song recommendations — it measures the fraction of recommended tracks that users actually find appealing. A high precision score means that most of the songs suggested to users are likely to align with their tastes and be enjoyed. This is enhancing user satisfaction.

Recall: Recall assesses the recommender system's ability to identify all the relevant tracks a user might like. It gauges the system's effectiveness in capturing a broad array of potentially enjoyable songs, minimizing the chance that desirable tracks are overlooked. For Deezer, a high recall ensures that the platform offers a comprehensive and satisfying selection of music recommendations with reducing the risk of missing out on tracks that the user might enjoy.

F1 Score: The F1 score is crucial for balancing precision and recall, especially when it is important to optimize both the accuracy of hitting the mark with song recommendations (precision) and the completeness of the recommendation pool (recall). At Deezer, this metric helps in fine-tuning the recommender system to enhance the overall quality of the user experience by ensuring a balanced approach to song suggestions.
Furthermore, it ensures a balanced evaluation by penalizing both false positives and false negatives, leading to a more reliable assessment of model performance in recommendation

tasks. This helps ensure that recommended songs contribute to an optimal user experience, maximizing both relevance and user satisfaction. This reasoning in addition to the binary response variable, has led to the decision to be the primary evaluation metric.
One may realise that the accuracy is not used in this project as an evaluation metric as it seems inappropriate due to the binary nature of the response variable and the sparsity of the data in Deezer's recommender system.

3.2. Comparison

The table below presents a comparative analysis of four recommendation models, with ALS serving as the baseline model. The evaluation metrics include F1-score, precision, recall, and the optimal threshold, which was selected based on the highest F1-score to ensure a balanced trade-off between precision and recall. The models were each optimized based on the F1-score by adjusting the threshold accordingly.

Model	F1-Score	Precision	Recall	Optimal Threshold
ALS (Baseline model)	0.8043	0.7158	0.9177	0.2
NCF	0.8091	0.7189	0.9353	0.1
Hybrid NCF	0.8193	0.7308	0.9322	0.3
SVD	0.8693	0.8100	0.9379	0.5

Table 1: Evaluation scores of applied models

Alternating Least Squares (ALS) – Baseline Model achieves an F1-score of 0.8043, with a relatively high recall (0.9177) but struggles with accuracy (0.6974) and precision (0.7158), suggesting it tends to produce more false positives which is something that the team wants to avoid.

Neural Collaborative Filtering (NCF) improves upon ALS with a higher F1-score of 0.8091, as well as increased recall (0.9353), but it does not reach the precision (0.7189) levels of the SVD model below.

Hybrid Neural Collaborative Filtering (Hybrid NCF), which combines features from multiple models, achieves an F1-score of 0.8193, showing an improvement in precision (0.7308), offering a more balanced performance compared to standard NCF.

Singular Value Decomposition (SVD) significantly outperforms all other models, achieving the highest F1-score of 0.8693, with the best precision (0.8100), making it the most effective recommendation approach in this comparison.

The optimal threshold values indicate how each model calibrates its classification boundary. ALS, with an optimal threshold of 0.2, suggests a tendency to generate lower prediction scores, while SVD, with an optimal threshold of 0.5, produces more confident recommendations.

These results confirm that Singular Value Decomposition (SVD) is the best-performing model in terms of F1-score (0.8693), while Hybrid Neural Collaborative Filtering (Hybrid NCF) with an F1-score of 0.8193 provides a strong deep learning alternative.

This was a surprise to the project team but has shown that in specific cases, complex algorithms such as NCF cannot replace simple yet effective algorithms such as the SVD. While SVD has issues with sparse data, the decision on removing users, songs and genres, which were rarely used or active during the preprocessing steps, may have reduced this disadvantage and could therefore perform better than initially expected.

4. Recommendation

Above findings indicate that some approaches perform better than others. Nevertheless, limitations were on the computational power of the team setup. Therefore, it is suggested to run the models with a more complete dataset.

In summary, the results indicate that the SVD model achieved the highest predictive accuracy compared to all other models, including more sophisticated approaches. Consequently, the research question:
"Can simple and traditional model-based collaborative filtering methods outperform (hybrid) Neural Collaborative Filtering in the context of music recommendations?"
can, with the necessary considerations, be answered affirmatively based on the presented findings. Further discussions and insights regarding this question are provided in this chapter.

4.1. Strategic Recommendations

While one can simply argue that SVD is the best method to be used, the team has decided to go into each chosen model and analyse its performance while also informing on when to use this specific model.

ALS – Baseline Model

- **Performance**: ALS demonstrates strong recall (0.9177) with a relatively good precision (0.7158), which suggests it effectively captures a broad set of relevant items for each user.
- **Use Case**: The weakest model according to the table above can be chosen when resources are limited and the goal is to ensure comprehensive coverage of potentially relevant songs, maximizing user discovery and engagement**.**

NCF

- **Performance**: NCF slightly improves on the precision of ALS while maintaining a high recall (0.9353), making it a more balanced choice with the cost of additional complexity.

- **Use Case**: NCF can be used when improving user satisfaction by balancing the trade-off between finding relevant items (high recall) and ensuring these items are genuinely of interest (moderate precision). It's particularly useful in dynamic environments where user preferences may rapidly evolve.

Hybrid NCF

- **Performance**: Hybrid NCF offers a slight improvement in F1 score over NCF, suggesting a better balance between precision and recall.
- **Use Case**: This model is effective in situations where both aspects of recommendation quality are equally important. It integrates content-based features, making it most suitable amongst all other models for overcoming the cold start problem by leveraging item metadata in addition to user-item interactions.

SVD

- **Performance**: SVD outperforms all other models in terms of precision (0.8100) and F1 score (0.8693), while also achieving a very high recall (0.9379).
- **Use Case**: SVD is the best choice for scenarios where high precision is crucial. Its ability to achieve high recall and F1 score makes it versatile for both general use and scenarios requiring high-quality recommendations. This model is especially recommended for top-tier user experiences where the goal is to provide highly accurate recommendations.

Context Recommendations

- **For General Use**: SVD offers the best overall performance with high scores across all metrics, making it the primary recommendation for Deezer's core recommendation engine.
- **For Niche User Groups**: ALS and NCF are preferable when the focus is on uncovering a wide array of user preferences, useful for users with eclectic or less mainstream music tastes.
- **For New Item Integration**: Hybrid NCF, with its content-based component, provides a robust solution to integrate new songs into the recommendation pool effectively, addressing new items' cold start issues.

To address some challenges that were experienced in the project and are well known in the environment for data scientists working on recommender systems, the team proposed the following recommendations to solve Deezer's general recommendation problems:

- **Add further content-based metadata**: Enriching the records with further information such as beats per minute (bpm), lyrics, vocals, artist age, nationality of artist/user, etc. may allow for further similarity calculations and might recommend unknown artists that share such characteristics with a famous artist, which leads to the next bullet point.
- **Exploring unknown artists**: The project team initially desired to not focus on the popular songs and artists but rather give new artists a chance to become more visible.
- **Improved User-Experience (UX)**: The team members aimed at improving the UX by innovating the way recommendations are shown, in this project visualized with stars. Visually showing the user that specific songs are not only recommended, but actually highly recommended, may have a significant effect on it.
- **Cold-Start Problem:** To mitigate this issue, it is recommended to use hybrid models, such as the presented Hybrid-NCF. While it performs slightly worse than SVD overall, it may yield better results in cold-start situations.

4.2. Overlap

This subchapter aims to answer the question
"Do the two solutions above overlap, in what way and why or why not?".

In short, they do overlap in many but not all ways. With the application of the collaborative filtering methods, a starting point was set to mitigate the cold-start problem and allow for further exploration and creation of content-based metadata that could allow the model to make recommendations better than shown in this project.

While the competition solution might focus on maximizing the accuracy metrics like F1-Score through sophisticated model tuning and threshold optimization, the team suggested integrating user and item metadata to improve recommendation diversity and address new users or items effectively. This holistic approach not only serves the immediate competitive objectives but also builds a robust framework that supports long-term user engagement and retention.

Unfortunately, not everything was able to be covered in this project due to technical-, scope- or time-reasons. The team emphasizes the importance of also recommending new and unknown artists to the population, rather than keep recommending songs that are already known to active listeners.

5. Reflection

Bringing it down to code: As mentioned various times in this paper, the team wanted to develop an unconventional recommender system that focuses on unpopular artists that may become famous in the short- or long-run. However, soon enough the team realized that bringing the idea down to the code given a dataset that does not specifically offer specific metadata for this use-case could become a big challenge, especially with the limited timeframe, which is why the focus was shifted to a conventional approach.

Computational power: Initially, the team started on local environments using PyCharm but even during data cleansing the team switched to Google Colab due to a better computational power. This showed the team, how important an appropriate infrastructure can be and make the difference between a high-quality and a failed project.

Machine-readable data: While it is not important for a machine to know if genre 123 is pop or reggaeton, it may be especially useful for humans and especially the engineers developing the algorithms to interpret and verify the given recommendations at least up to some level.

Implementation difficulty: Even though the lessons covered most methods and were explained clearly, implementing and developing a model on new data was a major challenge. Working with new libraries, new environments, new methods and developing a recommender system required a good team spirit and support of newest technologies, including generative AI.

Teamwork: The collaborative approach of the team significantly contributed to the success of this project. Working together allowed us to leverage each other's strengths, share knowledge, and find innovative solutions to challenges. Through regular discussions and collaborative problem-solving, the team not only improved the technical skills but also enhanced the ability to work efficiently as a team, making the overall experience highly valuable and enriching.

6. References

6.1. Literature

- Benitez, C. (2025). 37 Deezer Music Statistics For 2025 - Users, Growth, Revenue. Tone Island. Retrieved from https://toneisland.com/deezer-statistics/
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web, 173-182.
- Lu, G. (2025). "Recommender Systems" Blockweek at HSLU. Retrieved from https://elearning.hslu.ch/ilias/ilias.php?baseClass=ilrepositorygui&ref_id=6324650
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2002, December). Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science* (Vol. 1, No. 012002, pp. 27-8).
- Anwar, T., Uma, V., Hussain, M. I., & Pantula, M. (2022). Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: A comparative analysis. *Multimedia tools and applications*, *81*(25), 35693-35711.
- Sidana, S., Trofimov, M., Horodnytskyi, O., Laclau, C., Maximov, Y., & Amini, M. R. (2021). User preference and embedding learning with implicit feedback for recommender systems. *Data Mining and Knowledge Discovery*, *35*, 568-592.

6.2. Tables