OPTIMIZING MUSIC RECOMMENDATIONS: A DEEP DIVE INTO ENHANCING DEEZER'S FLOW RECOMMENDER SYSTEM

Recommender Systems F23

- ***Maha Hazimé- Zayour***

- **Mariela Lizbeth Longines Salgado**

University of Lucerne of Applied Sciences and Arts
Master in Applied Information and Data Science

Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Table of Contents

- Mariela Lizbeth Longines Salgado 0

Introduction & Objectives 2

Introduction to Recommender Systems 2

Deezer's Flow 2

Significance of Study 2

Objectives 2

Methodology 3

Data Description 3

Preprocessing Steps 4

Model Development and Evaluation 4

Algorithms and Architectures 4

Evaluation of Recommender System's Performance 5

Summary of Results 5

Ensemble Learning as a Competitive Advantage 6

Evaluating the Practicality of the Recommendation Engine for Production 7

Considerations Requiring Ongoing Development: 8

Considerations Requiring Ongoing Development and Additional Challenges for Developing a Successful Commercial Recommender System: 8

Innovations in Music Recommendation and Deezer's Path Forward 9

Concluding Remarks and Future Directions 10

Self-reflection (Maha) 11

Self-reflection (Mariela) 12

Appendix 13

Appendix 1 13

Appendix 2 14

Introduction & Objectives

Introduction to Recommender Systems

In the modern digital era, the extensive volume of available data has required the development of systems capable of providing personalized recommendations. Recommender systems have become widespread, offering personalized suggestions ranging from books and movies to restaurants and music. These systems play a crucial role in helping users discover content that is most relevant to their preferences, thereby significantly enhancing user experience.

Deezer's Flow

One notable application of recommender systems in the music streaming industry is Deezer's Flow feature. Flow is an intelligent shuffle mode that offers a personalized playlist based on a user's listening history, preferences, and behavior. As competition in the music streaming sector intensifies, the effectiveness of such features becomes a key differentiator, making the study of their predictive accuracy immensely valuable.

Significance of Study

Improving the accuracy of Flow's recommendations augments user satisfaction and increases the chances of subscriber retention for Deezer. Inaccurate recommendations, conversely, can lead to user churn, which is detrimental to the platform's long-term success. Therefore, the predictive accuracy of Flow is critical for both user engagement and Deezer's commercial success.

Objectives

The main goal of this research is to create and assess a predictive model capable of accurately determining whether a user will listen to the initial track suggested by Flow. We plan to utilize machine learning algorithms for forecasting user interaction with Flow's recommendations. Additionally, this report seeks to achieve the following objectives:

- Clarify the innovative techniques incorporated into our recommendation system.
- Evaluate the practicality of the recommendation engine we've developed for production.
- Delve into potential obstacles and future considerations when implementing a successful, commercially viable recommendation system.

In summary, this study lays the groundwork for exploring ways to optimize and potentially redefine personalized music recommendation systems like Deezer's Flow.

Methodology

Data Description

The dataset used in this project is sourced from Deezer, a leading music streaming service. This dataset is designed to mimic real-world scenarios for music track recommendations based on user interactions.

Key Data Fields:

- **media_id**: Unique song identifier.
- **album_id**: Identifier for the song's album.
- **media_duration**: Duration of the song in seconds.
- **user_gender**: Gender of the user.
- **user_id**: Anonymized user identifier.
- **context_type**: Describes how the song was encountered.
- **release_date**: Song release date in YYYYMMDD format.
- **ts_listen**: UNIX timestamp of song listening.
- **platform_name**: Operating system used by the user.
- **platform_family**: Type of user device.
- **user_age**: Age of the user.
- **listen_type**: Indicates if the song was listened to in "flow."
- **artist_id**: Unique artist identifier.
- **genre_id**: Unique genre identifier.
- **is_listened**: Binary indicator (1 if the song was listened to for >30 seconds, 0 otherwise).

Training Dataset: Captures user listening history over one month, with each row representing a listened track. It includes all the mentioned data fields and serves as the training data for predicting whether users will listen to Flow's initial track recommendation.

It will also be divided into a training set and a test set. This division is crucial for testing the effectiveness of our recommendation system.

Preprocessing Steps

To prepare the data for modeling, various preprocessing steps were undertaken. we initially focus on data cleaning to ensure the quality and reliability of the dataset, removing any noise or irrelevant features that could skew the model's performance. Following this, we engage in feature engineering to create new variables that better capture the underlying patterns in the data, enhancing the model's predictive power. This involves tasks like converting timestamps to a more manageable date-time format and segmenting user age into distinct groups. Each step serves to prepare the dataset for efficient and accurate modeling, thereby aiding in the development of a robust recommender system.

Model Development and Evaluation

Algorithms and Architectures

How would you develop a recommendation algorithm to win this competition?

In the pursuit of developing a recommendation algorithm to secure victory in a competitive setting. We deploy a suite of algorithms and architectures designed to maximize recommendation accuracy and effectiveness. These algorithms are thoroughly evaluated to identify their individual strengths and potential synergies.

SVD (Singular Value Decomposition)

Our first approach utilized the SVD algorithm, widely used in collaborative filtering techniques. GridSearchCV was used for hyperparameter tuning with 3-fold cross-validation. The parameters tuned included the number of factors, epochs, and learning rates. We then evaluated the model based on RMSE, Precision, Recall, and F1 Score.

KNN (K-Nearest Neighbors)

The second approach was to apply the KNNBasic algorithm, with hyperparameter tuning based on 3-fold cross-validation. The number of neighbors ('k') was tuned, along with similarity options. Model evaluation metrics remained consistent with the SVD approach.

Random Forest Classifier

For our third approach, we employed the Random Forest Classifier, well-known for its high accuracy, ability to run in parallel, and capability to handle a large number of features. Hyperparameters like the number of estimators and maximum depth were fine-tuned using GridSearchCV.

Hybrid Model

In the final stage, we experimented with a hybrid model combining predictions from SVD, KNN, and Random Forest. A simple ensemble technique was used to average out the predictions from the three models. This new approach aims to capture the strengths of each model and mitigate their weaknesses.

In addition to our algorithmic strategies, we establish a baseline model by deriving predictions based on the mean of the 'is_listened' values from the training dataset. These baseline predictions are subjected to rigorous evaluation, utilizing metrics such as RMSE, Precision, Recall, and the F1 Score. The outcome of this evaluation yields an RMSE of approximately 0.497, a Precision of 0.6, a Recall of 1.0, and an F1 Score of approximately 0.75, underscoring the performance of the model.

Throughout our model development and evaluation process, we consistently employ a comprehensive suite of evaluation metrics. These metrics provide an exhaustive assessment of our algorithms and the hybrid model. Our overarching objective is not merely to construct effective models but also to gain a deep understanding of their individual strengths and weaknesses. This knowledge forms the foundation for ongoing improvements and the potential deployment of our recommendation algorithm in a production environment.

Evaluation of Recommender System's Performance

To ensure the system meets both functional and quality criteria, we've employed a diverse range of metrics and machine learning algorithms to quantify the system's efficacy in recommending tracks that users will listen to for more than 30 seconds

Summary of Results

The recommender system was evaluated using various performance metrics—Root Mean Square Error (RMSE), Precision, Recall, and F1 Score. We deployed several machine learning algorithms to gauge these metrics, including Hybrid models, Baseline models, Singular Value Decomposition (SVD), K-nearest neighbors (KNN), and Random Forest.

Below are the summarized results:

	RMSE	Precision	Recall	F1 Score
Hybrid Model	0.447	0.693	0.987	0.814
SVD	0.415	0.772	0.887	0.826
KNN	0.474	0.712	0.979	0.798
Random Forest	0.559	0.687	0.998	0.814
Baseline Model	0.497	0.600	1.000	0.750

Performance metrics indicate the results of the metrics

- RMSE: The SVD model achieved the lowest RMSE, indicating its strong predictive accuracy. This result supports our hypothesis about SVD's efficacy in handling latent factors.

- Precision and Recall: The Hybrid model and the SVD had closely comparable F1 scores, suggesting a balanced trade-off between Precision and Recall. Interestingly, the Random Forest also posted a high F1 score, but its Precision was slightly lower.

- F1 Score: The highest F1 Score was achieved by the SVD model, followed closely by the Hybrid and Random Forest models. This suggests that these models can be considered robust in terms of both Precision and Recall.

Key Takeaways

- The SVD model outperforms all other models in RMSE and F1 Score, making it an excellent choice for predicting latent factors in user-item interactions.

- The Hybrid model performs well across all metrics, indicating that the ensemble approach successfully leverages the strengths of its component algorithms (SVD, KNN, Random Forest).

- The Random Forest model, despite having the highest RMSE, posts strong Recall and F1 Score, suggesting it is particularly useful when the focus is on reducing false negatives.

By systematically evaluating our recommender system through these diverse metrics and algorithms, we've gained a comprehensive understanding of its performance capabilities and limitations. These insights will be invaluable for future optimization and scalability efforts.

Ensemble Learning as a Competitive Advantage

What would you propose to solve Deezer's general recommendation problems?

To address Deezer's general recommendation problems, we propose a comprehensive solution rooted in innovative techniques that significantly enhance the quality of music recommendations. The core of our approach lies in multi-faceted feature engineering and the strategic implementation of ensemble learning.

Multi-faceted Feature Engineering: Our recommendation system leverages a wealth of user and item metadata to construct a feature-rich dataset. This approach involves segmenting users based on age groups, evaluating artist popularity metrics, and assessing listening frequency. By doing so, we gain deep insights into user behavior and preferences, allowing us to offer more dynamic and user-specific recommendations. This multi-faceted feature engineering forms the foundation of our approach, enabling a nuanced understanding of user interactions with music content.

Ensemble Learning: A standout innovation in our recommendation system is the integration of ensemble learning, a departure from traditional single-model approaches. Ensemble learning seamlessly combines three diverse machine learning algorithms: Singular Value Decomposition (SVD), K-Nearest Neighbors (KNN), and Random Forest. This ensemble approach is pivotal in achieving unparalleled recommendation accuracy and robustness.

Strengths of Individual Models:

- SVD excels in matrix factorization, making it adept at capturing latent features in user-item interactions, particularly beneficial for handling large datasets with high dimensionality.
- KNN effectively captures local data patterns, making it suitable for user-based and item-based collaborative filtering, especially in the presence of explicit data clusters.
- Random Forest's ability to handle both linear and non-linear data patterns, coupled with its inherent randomness, introduces an additional layer of diversity to the ensemble, enhancing its predictive power.
-

Ensemble Learning Methodology: The ensemble model operates by fitting each model (SVD, KNN, and Random Forest) to the training data. When a new recommendation request arises, each model generates its prediction. These predictions are skillfully combined, typically through averaging, to produce a final recommendation. This ensemble approach serves as the cornerstone of achieving superior recommendation accuracy.

Ensuring Robustness and Accuracy: Ensemble learning offers inherent improvements in robustness. Even if one model occasionally makes suboptimal predictions, the ensemble's collective intelligence ensures that these inaccuracies are mitigated or averaged out, ultimately resulting in more precise recommendations.

Ready for Production: The ensemble learning approach yields models that are not only highly accurate but also robust, making them ideal for production environments. The ensemble model can be seamlessly retrained to accommodate new data or algorithms, ensuring adaptability and longevity in a real-world recommendation system.

In conclusion, our proposal addresses Deezer's general recommendation problems by incorporating innovative techniques such as multi-faceted feature engineering and ensemble learning. This approach promises to deliver superior music recommendations that align with user preferences, solidifying our commitment to enhancing the quality of recommendations for Deezer's Flow users.

Evaluating the Practicality of the Recommendation Engine for Production

To address Deezer's overarching recommendation challenges, we propose an innovative and scientifically grounded approach rooted in the enhancement of recommendation techniques. Our recommender system embodies a multifaceted strategy that leverages several pioneering techniques, distinguishing it as both unique and highly effective.

Considerations Requiring Ongoing Development:

However, while our evaluation demonstrates the practicality of the recommendation engine, the journey toward a truly production-ready system demands continued attention to several critical aspects. Scalability, for instance, necessitates the implementation of advanced techniques to accommodate a burgeoning user base. Achieving real-time response necessitates the development of a streamlined real-time recommendation pipeline to meet user demands promptly. To bolster robustness and adaptability, an ongoing process of model retraining with fresh data must be seamlessly integrated into the system, ensuring its resilience in the face of evolving user preferences and behavior. Resource efficiency can be elevated through code optimization for parallel processing, coupled with diligent resource monitoring and management tools during deployment. User experience monitoring, a vital element, can be actualized through sophisticated techniques such as A/B testing, user feedback analysis, and continuous user behavior tracking. Lastly, the critical domain of maintenance requires meticulous documentation, version control, and the establishment of a robust deployment pipeline, encompassing regular updates, model retraining, and bug fixes.

Considerations Requiring Ongoing Development and Additional Challenges for Developing a Successful Commercial Recommender System:

Our evaluation of the recommendation engine's practicality for production underscores the need for ongoing attention to critical aspects such as scalability, real-time responsiveness, robustness, adaptability, resource efficiency, user experience monitoring, and meticulous maintenance. In addition to these considerations, several significant challenges and considerations emerge in the development of a successful commercial recommender system. These encompass diverse domains, including technical, ethical, user-centric, and strategic aspects:

1. Content Diversity: The vast array of content available on platforms like Deezer, encompassing various languages, genres, and cultural nuances, presents the challenge of recommending the right content to the right user amidst this extensive inventory.

2. Temporal Dynamics: User preferences are not static and can shift based on mood, time of day, or seasonal influences. Building a system that adapts to these temporal dynamics is essential for maintaining relevance.

3. Cross-Platform Experience: Users engage with music streaming services across multiple devices and platforms, necessitating the creation of a seamless, cross-platform experience while preserving user context and preferences.

4. Monetization vs. User Satisfaction: Balancing user satisfaction with revenue generation strategies is an ongoing challenge. A commercial system must align with business goals while ensuring a positive user experience.

5. Competitive Landscape: The music streaming industry's competitive landscape demands continuous innovation to stay ahead in refining recommendation algorithms and enhancing user engagement.

6. Equity and Fairness: Ensuring equity and fairness in recommendations is essential. The algorithm should not favor specific artists, genres, or user demographics, ensuring equitable representation.

7. Explainability: Implementing effective explainability in recommendations remains a challenging task to foster user trust and understanding.

8. Global vs. Local Content: Balancing global chart-toppers with local favorites is pivotal for platforms catering to an international audience.

9. Legal Considerations: Navigating legal obstacles related to song copyrights, artist agreements, and international licensing impacts the range of music that can be recommended.

10. Infrastructure Reliability: The recommendation system must be robust and fault-tolerant, delivering consistent performance as it scales to accommodate a growing user base and expanding music catalog.

11. User Feedback Loop: Establishing an efficient user feedback loop is indispensable, allowing users to express their satisfaction or dissatisfaction with recommendations and fostering continuous improvement.

Addressing these multifaceted challenges requires a comprehensive strategy that harmonizes technical innovation, user-centric design, ethical obligations, and strategic business objectives. The development of a successful commercial recommender system is contingent on navigating these complexities effectively in the dynamic and competitive music streaming landscape.

Innovations in Music Recommendation and Deezer's Path Forward

Do the two solutions above overlap, in what way, and why or why not?

The two solutions presented, one for developing a recommendation algorithm to win the competition and the other for solving Deezer's general recommendation problems, exhibit a degree of overlap in the sense that both emphasize the strategic utilization of ensemble learning as a core component of the recommendation system. However, the specific context and objectives differ.

In the first solution, the focus is on developing a recommendation algorithm to excel in a competitive setting, such as a competition. The strategy involves deploying a suite of algorithms and architectures, including Singular Value Decomposition (SVD), K-Nearest Neighbors (KNN), Random Forest, and a hybrid model, to maximize recommendation accuracy and effectiveness. The goal is to achieve victory by outperforming other algorithms in the competition.

In the second solution, the emphasis is on addressing Deezer's general recommendation problems in a real-world production environment. The proposed solution also incorporates ensemble learning but places a stronger emphasis on multi-faceted feature engineering, including user segmentation based on age groups, artist popularity metrics, and listening frequency. The objective is to enhance the quality of music recommendations for Deezer's Flow users.

While both solutions leverage ensemble learning and share a common commitment to improving recommendation accuracy, they differ in their contexts and objectives. The first solution is tailored for a competitive scenario where victory is the primary goal, while the second solution is aimed at solving practical recommendation challenges faced by Deezer, with a broader focus on user satisfaction and real-world deployment.

Concluding Remarks and Future Directions

In summary, this research project focused on enhancing the predictive accuracy of Deezer's Flow, a music recommendation system, through machine learning algorithms and innovative techniques. We clarified the innovative methods used, evaluated the practicality of the recommendation engine, and discussed ongoing development considerations. Additionally, we explored challenges in developing a successful commercial recommender system. The findings highlighted the effectiveness of ensemble learning and emphasized the need for ongoing attention to scalability, real-time response, user experience, and more. This research lays the groundwork for improving personalized music recommendations and navigating the complexities of the music streaming industry.

Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Self-reflection (Maha)

During my journey in the realm of recommendation systems, I've gained a deep understanding of the essential scientific principles that underlie this field. These fundamental ideas, like collaborative filtering, content-based filtering, and hybrid models, serve as the foundation for building effective recommendation systems.

One crucial area where I've improved my skills is feature engineering. This involves working with data to create features that enhance the accuracy of recommendations. For example, I've become adept at tasks like converting timestamps, grouping users by age, and incorporating popularity metrics for artists, albums, and genres.

Understanding the importance of evaluation metrics is another key takeaway. Metrics like RMSE, Precision, Recall, and F1 Score help assess how well a recommender system performs and guide adjustments for better results.

Ensemble learning, which combines different algorithms like SVD, KNN, and Random Forest, has broadened my knowledge. I've learned how blending these techniques can improve recommendation accuracy.

Data preprocessing, essential for ensuring data quality and preparing it for modeling, has become a strong skill. My practical experience with machine learning algorithms, including their implementation and parameter tuning, has grown significantly.

In addition to these learnings, I've gained insights into debugging, handling errors, and problem-solving during the development process. Recognizing the importance of aliasing when working with different libraries and frameworks has ensured smooth data flow and operations.

In summary, my exploration of recommendation systems, especially through the Deezer Flow project, has not only expanded my scientific and technical knowledge but also nurtured a problem-solving mindset. These skills are versatile and applicable beyond music recommendation, extending to industries like e-commerce and content streaming. This journey highlights the value of continuous learning and adaptability in the ever-evolving fields of data science and machine learning, setting a solid foundation for my future endeavors.

Self-reflection (Mariela)

Embarking on the journey of developing a recommendation algorithm was both enriching and challenging. It demanded a thorough exploration of various aspects of machine learning, including data preprocessing, exploratory data analysis (EDA), feature engineering, model selection, and evaluation.

During the initial phase, I delved into the dataset, finding it well-maintained and requiring minimal cleaning. This streamlined the process, but it emphasized the pivotal role of meticulous data preparation in ensuring accurate recommendations. Additionally, though omitted from the final document, the visualizations I created were invaluable for examining variable correlations, identifying outliers, and revealing that a minority of users engaged with only one or two songs at most. Providing detailed variable descriptions early on in the documentation underscored the importance of intimately understanding the dataset for effective model creation.

The visualizations played a crucial role in gaining a comprehensive understanding of the distribution of certain variables. They shed light on the most popular day among users, as well as the distribution of time, gender, and age demographics.

In my pursuit of creating a baseline model, my aim was to compare various models within a single code. Unfortunately, my limited RAM memory became a significant hindrance, impeding smooth code execution.

A lesson that I will always carry with me is the importance of upgrading RAM memory in a notebook. I encountered multiple instances where insufficient RAM hindered code execution. Mastering techniques to optimize code for efficiency became paramount in my learning journey.

Appendix

Appendix 1

List of the feature engineering steps in our Deezer recommender system along with how each one contributes to improving our recommender system:

1. Timestamp to Datetime Conversion
   - Enhances the system's ability to analyze user interactions over time, allowing it to consider historical listening patterns when making recommendations.

2. Age Grouping
   - Helps personalize recommendations by categorizing users into age groups, ensuring that the system tailors suggestions to the preferences of different age demographics.

3. Artist Popularity
   - Enables the system to identify and recommend tracks by popular artists, increasing the likelihood of suggesting songs that align with users' preferences.

4. Album Popularity
   - Contributes by recommending tracks from popular albums, which are more likely to resonate with a broader audience.

5. Genre Popularity
   - Assists in recommending tracks from trending genres, ensuring that the system stays current and suggests music that aligns with users' evolving tastes.

6. User Listening Frequency
   - Identifies highly engaged users by measuring their listening frequency, allowing the system to provide tailored recommendations to active users.

7. Release Year
   - Helps recommend newer or older songs based on user preferences, accommodating those who prefer contemporary music as well as those who enjoy classics.

8. Favorite Artist
   - Prioritizes tracks by users' most-listened-to artists, ensuring that the system highlights music that aligns with their established preferences.

9. Recent Trend
   - Recommends tracks based on users' most recent genre preference, reflecting their current music interests.

10. Season of Release
   - Enhances user experience by offering seasonally appropriate music recommendations, considering the time of year and associated listening preferences.

11. Album Loyalty
   - Increases user satisfaction by recommending complete albums, appealing to those who prefer to listen to full albums rather than individual tracks.

12. Artist Loyalty
   - Enhances user engagement by recommending more tracks from favorite artists, catering to users' specific artist preferences.

13. Artist-Album Interaction
   - Considers the combined popularity of artists and albums when making recommendations, potentially leading to more appealing and well-rounded suggestions.

14. Temporal Features (Day of the Week)
   - Incorporates weekly patterns in user behavior for recommendations, recognizing that users may have different music preferences on specific days of the week.

Appendix 2

```
Hybrid_model - RMSE: 0.44561765184188334, Precision: 0.6949172601519044, Recall: 0.9878571357882965, F1 Score: 0.8158894928276595
Baseline - RMSE: 0.49691834367486337, Precision: 0.6, Recall: 1.0, F1 Score: 0.7499999999999999
SVD - RMSE: 0.4145986665645214, Precision: 0.7744776377136937, Recall: 0.8877057211000822, F1 Score: 0.8272351256761058
KNN - RMSE: 0.47418246892133264, Precision: 0.7133863047386321, Recall: 0.9071818064860906, F1 Score: 0.7986965461959363
Random Forest - RMSE: 0.5610229163094291, Precision: 0.6849159116782897, Recall: 0.9983306670376295, F1 Score: 0.8124449116564912
Hybrid_model - RMSE: 0.44561765184188334, Precision: 0.6949172601519044, Recall: 0.9878571357882965, F1 Score: 0.8158894928276595
```
final_maha.py