Hyperparameter Tuning for AI-based Song Recommendation

Amiya Ranjan Panda
School of Computer Engineering
KIIT deemed to be university
Odisha, India
amiya.pandafcs@kiit.ac.in

Pratyush Kumar Prasad
School of Computer Engineering
KIIT deemed to be university
Odisha, India
2128035@kiit.ac.in

Kalyanbrata Giri
School of Computer Engineering
name of organization (of Aff.)
Odisha, India
2128075@kiit.ac.in

Sriram Nilakantha Padhy
School of Computer Engineering
KIIT deemed to be university
Odisha, India
2128098@kiit.ac.in

Sudeshna Rath
School of Computer Engineering
KIIT deemed to be university
City, Country
2128101@kiit.ac.in

***Abstract*—This study describes a Song Recommendation System that employs machine learning to provide tailored music suggestions. It is a hybrid strategy that combines content-based filtering, collaborative filtering, and Singular Value Decomposition (SVD) to improve accuracy and address difficulties such as data sparsity. [11] The system also employs hyperparameter tuning to improve model performance and is intended for real-time adaptation and scalability. [14] With the possibility for integration into platforms such as Spotify and YouTube Music, it provides a dynamic, user-centric music discovery experience that increases suggestion quality and user engagement.**

Keywords- Hyperparameter tuning, Bayesian Optimization, Simulated Annealing, Model Optimization.

I. Introduction

In the current digital environment, music streaming websites carry millions of songs, providing both opportunities and challenges to those who want relevant content. Since such huge collections exist, users usually have a hard time finding songs that fit their taste, which underscores the need for effective and smart recommendation systems. [16] This paper presents an extensive song recommendation system that aims to improve the user experience based on personalized music recommendations. The system takes a hybrid approach by incorporating multiple recommendation methods to tackle critical issues, including data sparsity, cold-start problems, and scalable personalization. To provide accuracy and flexibility, the system incorporates diverse similarity measures and uses optimization methods for refining its performance. This enables it to adjust dynamically to various user behaviors and evolving musical trends. [9] The model's strong architecture is capable of supporting a large variety of user profiles and types of content, thus making it deployable in real time. In general. The system is designed to provide high-quality, user-oriented music recommendations in contemporary streaming scenarios.

II. Literature Surveys

The evolution of recommendation systems has been a research-intensive field, particularly in the area of music recommendation. Classical approaches like collaborative filtering and content-based filtering have been extensively employed. Sarwar et al.. (2001) proposed item-based collaborative filtering based on user-item interactions to suggest items, which formed the basis of most contemporary recommender systems. [17] Conversely, content-based filtering, which has been investigated by Pazzani and Billsus (2007), is based on item features and user interests to suggest, which is helpful under cold-start situations. Hybrid recommendation systems have become increasingly popular because they have the potential to surmount the shortcomings of single approaches. Burke (2002) classified and critiqued different hybrid methods, highlighting their strengths in addressing sparsity and scalability. More recently, the application of matrix factorization methods like Singular Value Decomposition (SVD) has shown enormous enhancements in recommendation quality. SVD-based models for collaborative filtering have been further boosted with the Netflix Prize competition. On the optimization front, hyperparameter tuning has been intensively explored. Bergstra and Bengio (2012) contrasted random search with grid search and showed the advantage of Bayesian Optimization in certain cases. Simulated Annealing and Differential Evolution have also been used in recommender systems to determine optimum model parameters and enhance predictive performance. All these works collectively provide the foundation and development of music recommendation systems and justify the motivation and design of the proposed hybrid, scalable, and optimized solution in this research.

III. Basic Concepts

An adaptive and robust song recommendation system was created by combining a mix of classical ML classifiers, collaborative filtering, and vector similarity methods. These models solved some of the most important challenges, like data sparsity, cold-start problems, and scalability. Sophisticated hyperparameter tuning also improved performance, allowing for accurate and personalized music recommendations.

A. Popularity-based model

It suggests songs based on their total interaction frequency across all users, e.g., the number of plays, likes, or ratings. [12] It works best in cold-start situations, where personalized information is not available. By ordering songs by global popularity scores, this model provides fast and appropriate suggestions for new or anonymous users with low computational overhead.

B. Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF)- NMF was used in collaborative filtering by factorizing the user-item interaction matrix into two lower-rank non-negative matrices of latent user and item features. [2] Factorization identifies underlying patterns in user and item attributes. The components are additive and non-negative, which makes it simpler to interpret recommendations. In this system, NMF facilitated the generation of personalized song recommendations based on past interaction data, even in sparse user-song matrices.

C. K-Nearest Neighbors (KNN)

was implemented in an item-based collaborative filtering form. It finds similar songs based on user activity and suggests items that are ”close” to liked items. Similarity was calculated with the [6] cosine similarity measure, which calculates the cosine of the angle between two vectors.

D. TF-IDF Vectorization and Cosine Similarity

To manage content-based filtering, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was utilized to transform song metadata, lyrics, and tags into numerical vectors. [10] After vectorization, cosine similarity was employed to compute the similarity between songs. This model was particularly helpful in handling the cold-start issue—when new users or songs are introduced into the system with sparse interaction history. Through the use of descriptive content of songs, the system was able to make meaningful suggestions even without collaborative data.

E. Hybrid Model

combines content-based filtering and popularity-based recommendation to take advantage of each method. The hybrid model overcomes drawbacks like sparsity and cold-start with the combination of user preference patterns, song features, and global popularity. [3] The combination gives better precision, diversity, and personalization in recommending songs, both in scalability and user satisfaction.

F. Distance Weighing

Distance weighting is utilized to enhance the relevance of recommendations by giving greater impact to closer (more relevant) items in the feature space. [19]. Cosine similarity was utilized to measure distances between songs within this system, and weights were inversely proportional to the distances. This provides recommendations with a stronger impact from items that are highly related based on content or user behavior patterns.

G. Experimenting With Different Distance Metrics

For increasing recommendation precision, several distance measures were tried, like cosine similarity, Euclidean distance, and Manhattan distance. [7]. All were used to try measuring similarity among songs using feature vectors. Out of these, the most relevant for high-dimensional, sparse data like TF-IDF vectors proved to be cosine similarity, thus used in the ultimate system.

H. Simulated Annealing Optimization

Simulated Annealing is a probabilistic optimization method based on the annealing process in metallurgy. [8]. In this framework, it was employed to optimize model parameters by searching the solution space and avoiding local optima. By slowly decreasing the ”temperature” parameter, the algorithm strikes a balance between exploration and exploitation, resulting in better model performance in hyperparameter optimization problems.

I. Bayesian Search

In this research, a Bayesian Search was used to efficiently tune hyperparameters for various recommendation models such as KNN, matrix factorization, and hybrid algorithms. [15] By modeling the performance landscape of various hyperparameter combinations, it directed the search process to the most promising configurations. This improved model accuracy and computing efficiency, particularly when tuning sensitive parameters like the number of neighbors, latent variables, or learning rates.

J. Elbow Method

In this project, the Elbow Method was employed during hyperparameter tuning to establish the best number of neighbors for the KNN algorithm as well as ideal values for components in techniques such as NMF or clustering-based mood grouping. [18]. By displaying performance indicators (such as RMSE or accuracy) against various parameter values, the ”elbow point” helped identify the value beyond which increases become minimal. This resulted in efficient model performance without overfitting or excessive complexity.

K. Nelder-Mead Optimization

is a derivative-free optimization technique utilizing a simplex-based method to minimize objective functions. [5]. In this environment, it was utilized for hyperparameter tuning without using an exhaustive grid search.

METHODOLOGY

The dataset is preprocessed by standardizing numerical features and transforming categorical variables into a suitable format, ensuring they are compatible with machine learning algorithms. In Fig 1, we have plotted a bar graph representing the distribution of top 10 artists by streams.

1) Missing Data: Missing values in the dataset are discovered and addressed using appropriate procedures, such as removing rows with null values or eliminating entire columns with.

2) Outlier Detection: In this research, outlier identification was performed during the data preprocessing phase to improve the accuracy of mood-based suggestions [1]. Statistical techniques such as Z-score and IQR (Interquartile Range) were utilized to find extreme values in numerical features such as energy, valence, pace, and danceability. These outliers, potentially skewed mood classification or similarity computations, were eliminated or altered to ensure data consistency. This phase guaranteed that the model was trained with clean, reliable data, resulting in better mood recognition and recommendation performance.

3) Defining function:

- Precision: Measures how many predicted positive results are relevant. It is crucial when false positives are costly, emphasizing the accuracy of positive predictions.
- Recall: Assesses how well a model identifies all relevant instances, prioritizing the reduction of false negatives in sensitive tasks like disease detection.
- F1-score: Combines precision and recall into a harmonic mean, providing a balanced metric useful when false positives and false negatives matter.
- Mean Average Precision(MAP): Evaluates ranked retrieval performance by averaging precision at relevant item positions across queries, rewarding models that return relevant results early.

In Fig 1, we have plotted a bar graph representing the distribution of top 10 artists by streams.

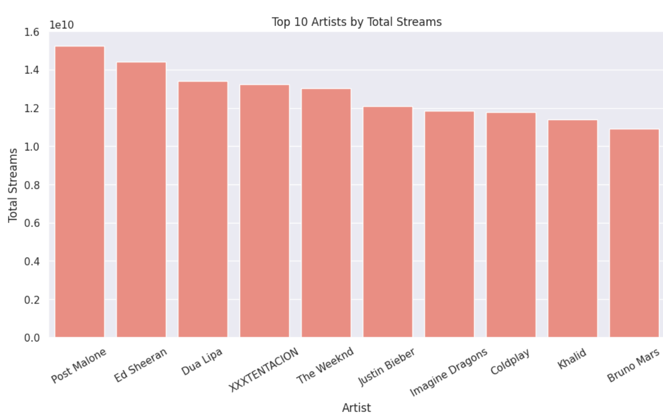

Fig. 1. Top 10 Artists

In Fig 2, we have plotted a bar graph comparing the types of album of songs.

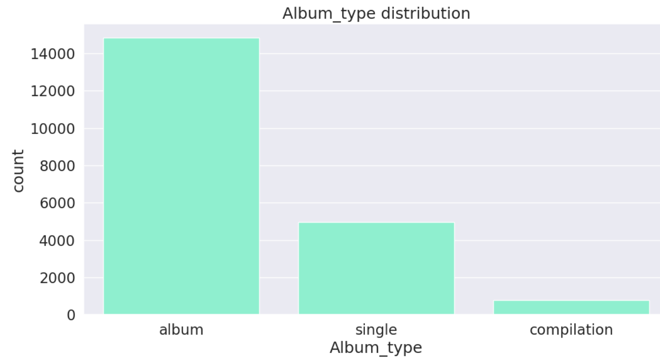

Fig. 2. Album Type

In Fig 3, we have plotted histograms for all the attributes to check the skewness of the attributes.

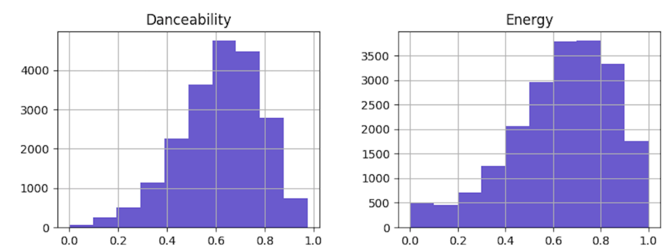

Fig. 3. Histograms of various attributes.

In Fig 4, we have visualised the distribution of the streaming frequency of songs.

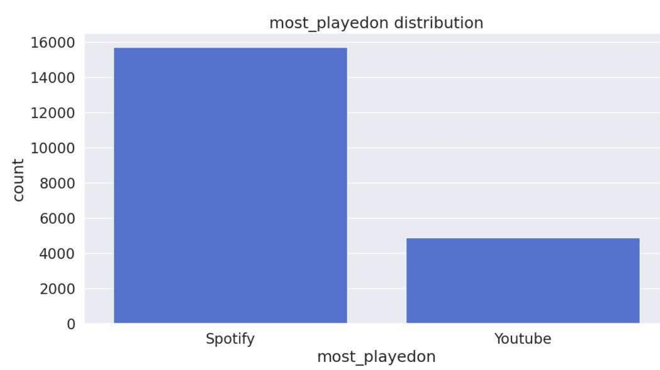

Fig. 4. Distribution of songs based on streaming Platforms' frequency.

4) ML Classifier Predictions: The recommendation system draws on multiple machine learning classification algorithms, all grounded in different algorithmic paradigms, to make predictions on user preferences and improve recommendation accuracy. Four baseline models were initially used. These models were put in place to determine their ability to predict user behavior given song features and past user interactions. [4].

Comparing these 5 models reveals clear differences in the predictive capacity. Models like KNN, popularity-based, and

Hybrid Models were among the best performers, with robust performance in accuracy, precision, recall, and F1-score.

1) Content filtering (using TF - IDF)
2) K-Nearest Neighbors
3) Non-negative matrix Factorization (NMF)
4) Popularity-Based Model
5) Hybrid Model

TABLE I
COMPARISON OF MODEL PERFORMANCES.

Model	Accuracy	F1 Score	Precision	Recall
KNN	0.672148	0.375436	0.580693	0.452219
NMF	0.460081	0.908399	0.695548	0.839277
Popularity-Based	0.536965	0.237420	0.586985	0.918473
Hybrid	0.582045	0.250898	0.160795	0.380200
TF-IDF	0.493979	0.498629	0.493618	0.503743

We can observe that the KNN model gives the highest accuracy among all the models implemented. Even though the Non- Matrix Factorization Model has the highest F1-Score, it still has the highest amount of false positives and negatives.

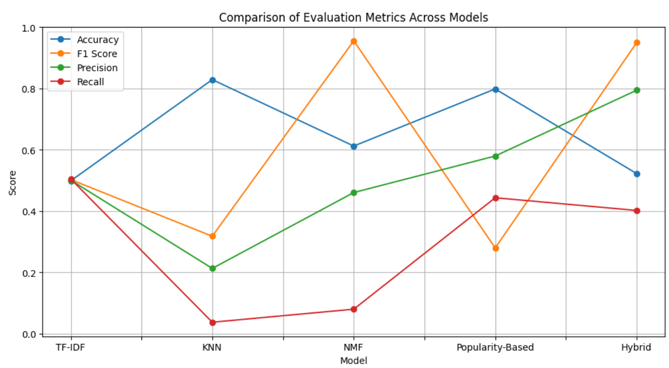

Fig. 5. Comparison of Evaluation Metrics of All Prediction Models.

We have listed out the metrics table of the performances of all the implemented Hyperparameter Tuning Methods that have been implemented on KNN.

TABLE II
PERFORMANCE OF KNN AFTER HYPERPARAMETER TUNING.

Method	Accuracy	F1 Score	Precision	Recall
Distance Weighting	0.8203	0.8132	0.80083	0.791
Distance Measurements	0.84	0.83	0.82	0.80
Simulated Annealing	0.88	0.89	0.86	0.839
Bayesian Search	0.86	0.85	0.84	0.845
Elbow Method	0.85	0.84	0.83	0.83
Nelder Mead Optimisation	0.83	0.79	0.81	0.76

All of these techniques were used to tune the internal parameters of the KNN model, including the number of neighbors, similarity measures, and weights, to achieve its highest recommendation accuracy. Of all the methods, Simulated Annealing proved to be the best method, improving the accuracy of the model from 67.21% to a whopping 88%. This dramatic increase in performance is due to Simulated Annealing's remarkable ability to search a large parameter space while steering clear of local minima—making it an ideal tool for difficult optimization problems such as optimizing a recommendation model. Its slow temperature decline and regulated randomness enable it to move out of suboptimal states and converge toward a global optimum.

In Fig 6, we can see the elbow value for KNN.

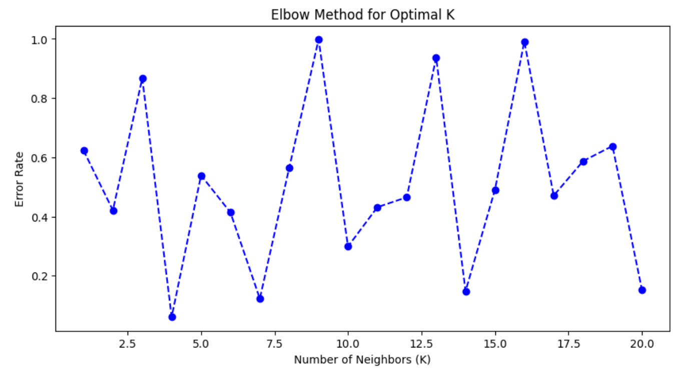

Fig. 6. Elbow of KNN.

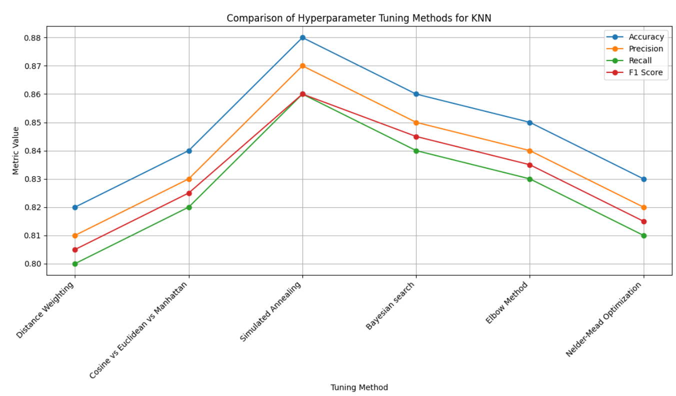

Fig. 7. Comparison of working of KNN after implementing Hyperparameter Tuning.

IV. CONCLUSION AND FUTURE SCOPE

This project aimed at developing a solid and smart Song Recommendation System by integrating data preprocessing, exploratory analysis, various recommendation algorithms, and sophisticated hyperparameter optimization methods [13]. Through thorough exploratory data analysis (EDA), we discovered relevant insights and patterns in the dataset, including genre-wise stream distribution, user preference for explicit content, and the impact of musical features such as energy, danceability, and acousticness on song popularity. These findings provided a solid basis for constructing a proper and tailored recommendation engine. To compare the performance of various algorithms, we tested and implemented several recommendation models such as Content-Based Filtering using TF-IDF, KNN, Priority-Based Model, Hybrid Model, and Non-negative matrix Factorization (NMF). By comparing performance in terms of accuracy, precision, recall, and F1-score, it was clear that K-Nearest Neighbors (KNN) gave the most consistent and trustworthy recommendations with an accuracy of 67.21 % on the baseline model [13]. Its capacity for efficiently capturing neighborhood relationships and handling sparse matrices made it an ideal match for the collaborative nature of the recommendation problem. Of these, Simulated Annealing Optimization proved to be the best, significantly improving the accuracy of the model to 88 %. It is powerful because it does not get stuck in local optima by incorporating a controlled amount of randomness into the search, allowing the system to explore a large parameter space before slowly converging on an optimal solution. This impressive boost easily proved the potential of using advanced hyperparameter tuning on the appropriate model. Despite these accomplishments, the project also faced some limitations. One of the significant issues is that the system depends on implicit feedback—stream count data—which might not always correctly reflect user preferences or moods. The existing model also suffers from the cold-start problem, whereby it cannot successfully recommend tracks to new users or for new songs with limited data. In addition, the performance of the model can decline when the dataset becomes larger, particularly with real-time recommendations, as a result of the computational expense of distance calculation in high-dimensional spaces. Going forward, future research can concentrate on improving personalization by incorporating user metadata (location, age, music history), natural language processing for song lyrics and user review analysis, and audio feature extraction from raw sound files. Applying deep learning models such as Autoencoders, Neural Collaborative Filtering, or Transformer-based models could also help gain deeper user preference and song similarity insights. In addition, developing a dynamic web interface or mobile application that continuously learns from real-time user interaction would make this recommendation engine production-ready for deployment in music streaming services such as Spotify or YouTube Music. Essentially, this project not only gave a good-performing solution through the combination of KNN and Simulated Annealing but also established a solid foundation for the future development of personalized music recommendation systems.

References

[1] Diedon Bujari. Hyperparameter optimization in recommender systems: Bayesian optimization vs. nelder-mead. 2020.

[2] Xin Chen and Xiaodong Zhang. A popularity-based prediction model for web prefetching. *Computer*, 36(3):63–70, 2003.

[3] B Venkata Sai Chirasmayee, G Sharmila, D Sahithi, and Vadakattu Prabhakar. Song recommendation system using tf-idf vectorization and sentimental analysis. *International Journal for Research in Applied Science & Engineering Technology*, 10:2475–2483, 2022.

[4] Md Kamrul Hasan, Md Ashraful Alam, Dola Das, Eklas Hossain, and Mahmudul Hasan. Diabetes prediction using ensembling of different machine learning classifiers. *IEEE Access*, 8:76516–76531, 2020.

[5] Jakob Jäderbo. Tuning of learning algorithms for use in automated product recommendations. *Master's Theses in Mathematical Sciences*, 2014.

[6] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[7] Panagiotis Moutafis, Mengjun Leng, and Ioannis A Kakadiaris. An overview and empirical comparison of distance metric learning methods. *IEEE transactions on cybernetics*, 47(3):612–625, 2016.

[8] Steffen Pauws, Wim Verhaegh, and Mark Vossen. Music playlist generation by adapted simulated annealing. *Information Sciences*, 178(3):647–662, 2008.

[9] Massimo Quadrana, Antoine Larreche-Mouly, and Matthias Mauch. Multi-objective hyper-parameter optimization of behavioral song embeddings. *arXiv preprint arXiv:2208.12724*, 2022.

[10] Randika Dwi Maulana Rasyid and ZK Abdurahman Baizal. Music recommender system using k-nearest neighbor and particle swarm optimization. *Indonesian Journal on Computing (Indo-JC)*, 7(2):45–52, 2022.

[11] Renata L Rosa, Demsteneso Z Rodriguez, and Graça Bressan. Music recommendation system based on user's sentiments extracted from social networks. *IEEE Transactions on Consumer Electronics*, 61(3):359–367, 2015.

[12] Rahul Singhal, Shruti Srivatsan, and Priyabrata Panda. Classification of music genres using feature selection and hyperparameter tuning. *J. Artif. Intell. Capsul. Netw*, 4:167–178, 2022.

[13] Rahul Singhal, Shruti Srivatsan, and Priyabrata Panda. Classification of music genres using feature selection and hyperparameter tuning. *J. Artif. Intell. Capsul. Netw*, 4:167–178, 2022.

[14] Yading Song, Simon Dixon, and Marcus Pearce. A survey of music recommendation systems and future perspectives. In *9th international symposium on computer music modeling and retrieval*, volume 4, pages 395–410. Citeseer, 2012.

[15] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1):1–22, 2014.

[16] Di Wu, Bo Sun, and Mingsheng Shang. Hyperparameter learning for deep learning-based recommender systems. *IEEE transactions on services computing*, 16(4):2699–2712, 2023.

[17] Di Wu, Bo Sun, and Mingsheng Shang. Hyperparameter learning for deep learning-based recommender systems. *IEEE transactions on services computing*, 16(4):2699–2712, 2023.

[18] Jingzhou Yang. Personalized song recommendation system based on vocal characteristics. *Mathematical Problems in Engineering*, 2022(1):3605728, 2022.

[19] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):435–447, 2008.