TASTES: A Taxonomy-Free Music Recommendation Strategy for Interest Expansion on Spotify

José Maria Gouvêa
jose.m.gouvea@tecnico.ulisboa.pt
Instituto Superior Técnico
Lisbon, Portugal

ABSTRACT

The demand for adaptive music recommendation systems has grown with the rise of digital streaming platforms. *Spotify*, a leader in this space, strives to offer new features that connect listeners to the right musical content. To explore new ways of enhancing music discovery, we introduce TASTES, a novel sequential recommendation approach for playlist continuation on *Spotify*, free of genre taxonomy. TASTES dynamically models user preferences within hand-curated playlists on *Spotify*, using their Web API services, to produce suggestions that expand the user's musical interests. To evaluate our approach, we conducted a 30-day within-subjects study with 20 participants, comparing TASTES' aptitude for playlist continuation and interest expansion against *Spotify*'s recommendation method. Assessment based on playlist audio composition impact, suggestion accuracy and diversity, and user experience feedback revealed that while TASTES does not significantly improve interest expansion over *Spotify*'s recommendation strategy, it proved more capable of generating relevant suggestions within the user's preferences and offering an improved music discovery experience in cases where speech content is less relevant to the listener.

KEYWORDS

Music Recommendation; Music Discovery; Musical Interest Expansion; Playlist Continuation; Spotify; Taxonomy-Free Classification; Interest-Aware Recommendation; Sequential Recommendation

1 INTRODUCTION

With the growing popularity of global music streaming platforms, listeners have unprecedented access to vast music libraries, creating more demand for innovative tools that tailor relevant music to listeners. The fluidity of modern listening habits demands systems that introduce new genres, enhancing discovery experiences and diversifying song collections. Research shows that novelty and diversity are crucial for user satisfaction in recommendation tasks. A common challenge is providing suggestions that expand listeners' musical repertoire based on evolving trends, meeting their desire for exploration.

Spotify has become a leader in music streaming with its vast library, user-friendly interface, and powerful recommendation tools. Features like "Discover Weekly" curate tracks outside the user's usual interests, highlighting the importance of introducing new music to the listener. *Spotify* also offers playlist curation features that suggest new songs for specific playlists, but these recommendations often cater to existing themes rather than exploring new musical avenues.

Current music discovery approaches often rely on predefined genre taxonomies, which can be restrictive and fail to capture music's multidimensional quality. Contemporary music frequently defies traditional genres, incorporating elements from various styles, often fitting between existing genres. This limits taxonomy-dependent systems in providing personalised music discovery experiences that adapt to users' unique preferences and classifications.

1.1 Approach and Research Questions

To address these challenges, we propose TASTES, a novel interest-aware recommendation approach aiming to expand users' musical interests within their hand-curated playlists by generating personalised suggestions for playlist continuation in a sequential fashion. These suggestions follow the user's evolving trends across various music categories, guiding them into new discovery horizons and enhancing their listening experience. TASTES uses unsupervised clustering techniques to identify song groupings without relying on predefined genre taxonomies, allowing for a more adaptive categorisation. Accordingly, our primary research questions are:

(1) How effective is a taxonomy-free interest-aware recommendation approach such as TASTES at expanding the listener's musical interests?
(2) Can a taste expansion approach to recommendation improve listeners' music discovery experience?

1.2 Contributions

Firstly, We conduct a **comprehensive literature review**, exploring existing research on music recommendation systems, diversity, and genre taxonomy challenges. This review highlights the aspects which informed our approach's design.

Additionally, we develop a **novel music recommendation model**, TASTES, which focuses on expanding users' musical interests within their hand-curated playlists. By leveraging an evolving model of preferences, TASTES generates personalised recommendations that extend users' inclinations into new areas, enhancing their music exploration journey.

Finally, we conduct a **user study** to evaluate the effectiveness of TASTES for music discovery on *Spotify*. We assess the impact of our approach on playlist continuation tasks using both quantitative and user-centred metrics, contributing to a better understanding of expansion-driven functionalities in recommendation systems.

2 RELATED WORK

2.1 Fundamental Concepts

Recommender Systems. Recommender systems are algorithms designed to suggest items like books, movies, or songs based on

user feedback, past behaviour, and trends. They personalise content delivery and enhance the user's experience.

Diversity/Interest-Aware Recommendation. Diversity-aware recommendations strategies aim to engage users by introducing variety in suggested items, preventing content homogenisation [Di Noia et al. 2014]. This is crucial in music, where preferences span multiple genres and shift based on context and mood. Interest-aware strategies, a subset of diversity-aware systems, track user tendencies within different interest categories, combining diversity with personalisation. They adapt to evolving trends, providing more relevant suggestions by recognising the complexity of musical tastes.

Playlist Continuation. Playlist continuation (PC) involves suggesting songs to be added to a playlist that fit its musical footprint and the user's current inclinations within the represented themes. This tests a recommendation method's ability to cater to changing preferences within a specific interest area, making PC tasks ideal for evaluating interest-aware approaches.

2.2 Spotify Web API (SWAPI)

SWAPI is a powerful interface bridging third-party applications and the *Spotify* database, providing access to a wide range of functionalities for *Spotify* music and profile data collection, analysis, and publishing. SWAPI has been extensively used in various applications, including music analysis and playlist curation. We now detail the API functionalities used in our proposed solution.

2.2.1 ***Audio Feature Requests.*** Provides quantitative metrics for a track's musical characteristics. Data for each track includes 12 audio descriptors, ranging from traditional music metrics (*tempo*, *key*, *mode*, *time signature*) to high-level quantifiers (*acousticness*, *energy*, *danceability*, *instrumentalness*, *liveness*, *loudness*, *speechiness*, *valence*), determined by *Spotify*'s audio analysis methods. The feature descriptions and ranges are available in SWAPI's online documentation [Spotify [n. d.]]. The tracks' *Spotify* ID[1] are required as input for these requests.

2.2.2 ***Playlist Requests.*** Enables publishing of playlists in users' *Spotify Library*[2]. The publish request returns the playlist's *Spotify* ID as confirmation. The endpoint also allows to upload and replace tracks in user-owned playlists and retrieve data about these playlists, including the tracklist IDs and addition timestamps.

2.2.3 ***Recent Streaming History Requests.*** Retrieves users' recent playback history. Requests return the most recently streamed tracks before a provided Unix timestamp. Each response includes each track's ID and playback timestamps. These requests are costly in computation time and frequency-restricted by *Spotify*.

2.2.4 ***Recommendation Requests.*** The recommendation function obtains track suggestions based on up to five seed items (tracks, artists, or genres). The seeds shape the recommendations, determined by *Spotify*'s algorithms. Requests support audio attribute filtering, although we avoid using these filters as they often reduce the accuracy and efficiency of *Spotify*'s responses.

[1]The *Spotify* ID is a unique base-62 identifier for every artist, track, album and playlist included in the platform's database.
[2]Personalised page within the *Spotify* application where users can save their favourite music and podcasts for easy access.

2.3 Literature Review

2.3.1 ***Taxonomy-Free Music Classification.*** Effective music recommendation requires a comprehensive model of musical categories. Scaringella et al.'s 2006 survey [Scaringella et al. 2006] reviews music genre classification, highlighting challenges such as unclear genre definitions, the subjectivity of classification, and the wide range of musical styles. The survey stresses the importance of considering various audio features in music similarity and suggests unsupervised methods for more flexibility. It advocates for multi-label classification and systems supporting genre evolution.

Krebbers' 2020 thesis [Krebbers 2020] compares supervised and unsupervised classification of electronic music using *Spotify* audio features. Supervised methods (SVM and KNN) achieved around 70% accuracy but struggled when classifying between similar genres, aligning with Scaringella et al.'s conclusions. K-Means, an unsupervised method, revealed new ways of categorising music, achieving a Purity Score of 49.7% and a Silhouette Score[3] of .283.

2.3.2 ***Interest-Aware Recommendation.*** Balancing accuracy and diversity in recommendations is crucial for user satisfaction [Di Noia et al. 2014]. Kaya and Bridge's 2019 paper [Kaya and Bridge 2019] introduces SPAD, an interest-aware approach that identifies sub-profiles in the user's items of interest using unsupervised clustering and diversifies recommendations to align with the user's sub-profile trends. SPAD showed improved accuracy and diversity when pitted against similar interest-aware methods, making it more precise while preserving suggestion diversity.

Zhao et al.'s 2021 paper [Zhao et al. 2021] introduces *TecRec*, which improves on Kaya and Bridge's approach to address the rabbit hole effect[4] and taste distortion problem[5]. *TecRec* is an interest-aware approach which adapts to dynamic changes in user preferences, using feedback data to adjust its selection strategy. It measures the diversity of suggestions through the KLD[6] between the recommendation set's sub-profile distribution and the user's sub-profile preference distribution.

2.3.3 ***Playlist Continuation.*** Kaya and Bridge's 2018 paper [Kaya and Bridge 2018] proposes a practical application of SPAD for automatic playlist continuation (APC) on *Spotify* for the ACM RecSys Challenge 2018. Participants had to generate up to 500 candidate songs to be added to a playlist while maintaining its original musical footprint. Kaya and Bridge's adaptation of SPAD performed well in accuracy metrics while maintaining suggestion diversity, coming in 7th out of 32 participants. Furthermore, their analysis of 10000 *Spotify* playlists revealed that most contain distinguishable sub-genres, with 4 being the most frequent number of sub-profiles found, often composed of a small number of tracks each. This shows that diversity-aware approaches can effectively balance recommendation accuracy and diversity in a practical setting.

[3]The Silhouette Score quantifies how well a data point fits into its assigned cluster and how distinct it is from other clusters.
[4]Occurs when a recommender system over-focuses on a user's prominent interests, neglecting others leading to echo chambers and unfair content representation.
[5]The mismatch between the system's suggested content distribution and the user's true preferences.
[6]The Kullback-Leibler (KL) divergence is a measure of how one probability distribution diverges from a second, expected distribution.

3 IMPLEMENTATION

Based on the chosen experimental task and the success found by prior solutions, the following core aspects define our architectural design:

Playlist Expansion: TASTES is designed for playlist continuation (PC) tasks. It uses an initial hand-curated playlist to define the user's area of interest in focus, producing new song suggestions aiming to extend beyond the included themes. TASTES operates independently for each playlist and *Spotify* user.

Spotify Integration: Our framework integrates directly with *Spotify* profiles, leveraging SWAPI for recommendation services to enhance music discovery. Some design choices were taken to accommodate SWAPI's limitations and maximise their functionality.

Taxonomy-Free Classification: Using an unsupervised classification method, TASTES organises music by finding emergent song groupings, or "facets," based on audio similarities. This approach allows flexibility in music classification and independence from predefined genre taxonomies.

Distributed Preference: TASTES mitigates the accuracy-diversity tradeoff of recommendations by catering to the user's multiple interests. It models musical taste as a distribution of weighted preferences towards playlist facets, each weight reflecting the user's relative affinity for that musical grouping.

Sequential Recommendation: The strategy uses an iterative online framework for real-time recommendation generation, equivalent to Zhao et al.'s *TecRec* approach. Each step of execution, referred to as TES[7], is manually triggered. After updating its user preference model based on the user's recent trend data, the strategy finishes each step by outputting a new set of suggestions.

Evolving Interest Adjustment: Inspired by Zhao et al.'s approach to mitigating the rabbit hole effect and taste distortion, TASTES dynamically adjusts the preference distribution model using recent *Spotify* activity and direct feedback data, maintaining an up-to-date representation of the listener's most recent tendencies within the playlist facets.

3.1 User Feedback

TASTES relies on user-generated feedback data between processing iterations to refine its recommendations. This feedback pertains to both the playlist in focus and previous suggestions, allowing the strategy to adjust to the listener's current needs. The user's preferences are updated using three types of feedback data, collected through SWAPI endpoints:

- *Recommendation feedback*: User appraisals of TASTES recommendations via the *Spotify* interface, such as "likes," playlist additions, removals from the recommendation pool, or no reaction, indicating their interest in each track.
- *Playlist feedback*: Changes to the playlist tracklist reflect the user's current inclinations. Added songs are highly representative of the user's current preferences, while removed tracks indicate disinterest.
- *Streaming feedback*: Recent listening history provides insights into the user's favoured songs and playlist facets. Tracks that are played more recently and frequently are considered more representative of the user's present trends.

[7]TASTES Processing Step

3.2 Radio Playlist

TASTES publishes recommendations directly in the user's *Spotify Library* within a designated **radio playlist** through SWAPI. At the end of each TES, the previous recommendations are replaced with new tracks, allowing the user to listen to them and provide more explicit feedback reactions through *Spotify*'s interface. TASTES interprets three types of recommendation interactions:

- Moving a track from the radio playlist to their own indicates interest in similar recommendations.
- Removing a track from the radio playlist indicates disinterest, equivalent to a "dislike."
- Liking a track in the radio playlist shows interest without necessarily wanting to add it to their playlist.

If no reaction is given, TASTES assumes no specific interest. Each track can only be assigned one feedback type, prioritised in the listed order: adding to the playlist, removing, or liking.

3.3 User Preference Model

TASTES' user preference model is comprised of several data structures that collectively sustain the most recent state of the user's inclinations within their playlist's area of interest, all maintained and updated using a third-party SQL database called TUDB[8]. At every TES_i the following models are updated:

- **User Facet Preference Distribution Vector (p_i)**: An array of weights (0-1) representing the user's current affinity towards each playlist facet. Each entry $p_{i,f}$ expresses the user's relative inclination towards facet f within the set F of facet labels. A value of $p_{i,f} = 0$ indicates no interest, while 1 indicates a strong preference for facet f.
- **Model Track Set Vector (P_i)**: An array of structures designating each relevant track in the playlist's area of interest. Each structure contains the track's normalised audio feature values and facet classification labels. The audio feature values for a track $s \in P_i$ are represented as s_a for all $a \in AF$, where AF is the set of *Spotify* audio features listed in 2.2.2. The Primary Track Set Vector ($P1_i \subset P_i$) includes only the tracks in the user's own playlist.
- **Relevance Ratings Vector (U_i)**: An array of values (0-1) where each entry $U_{i,s}$ indicates the user's interest in model track $s \in P_i$. A value of 0 indicates no interest and 1 indicates the highest interest within the model.
- **Similarity Threshold Vector (D_i)**: An array of positive values denoting the similarity area of each track $s \in P_i$. Each entry $D_{i,s}$ retains the track's distance to its furthest k-neighbour in P_i, computed in the audio feature space. If the similarity/distance between two tracks s and r is less than $D_{i,s}$, they are considered similar.
- **Audio Feature Weight Distribution Vector (w_i)**: An array of values (0-1) representing the user's perception of audio similarity. Each weight $w_{i,a}$ for all $a \in AF$ indicates the importance given by the user to audio feature a when including tracks in the playlist's area of interest.
- **Tabu Set (TB_i)**: An array of strings comprising the *Spotify* IDs of all songs that have been in the user's playlist

[8]TASTES User Database

or recommended by TASTES since TES_0. This set helps avoid suggesting tracks the user has already encountered, as SWAPI requests do not offer control for this aspect.

3.4 First Execution Step

Before TASTES can produce recommendations for a particular playlist, it must first initialise all data structures comprising its preference model. Consequently, TES_0 follows a distinct processing structure to set up the subsequent iteration steps. Due to the lack of prior usage data, TASTES relies on two initial assumptions: the tracklist represents the user's musical taste in the playlist's context and more recently added tracks better represent the user's current trends. These assumptions help initialise the user's preference model as an approximation of their actual interests, minimising cold-start[9] problems.

Model Set Extraction. The initial track set vector P_0 is assembled by collecting all available audio feature data for each track in the user's playlist. The values are normalised to a 0-1 range. An added vector $T_s, \forall s \in P_0$ is also attained using the timestamps of when each track was added to the playlist. The primary model and tabu set vectors are initialised as $TB_0 = P1_0 = P_0$.

Relevance Rating Estimation. TASTES computes the starting relevance rating vector U_0 based on the recency of tracks in the playlist. Each track s's initial relevance is determined by the following logistic function:

$$U_{0,s} = \frac{1}{1 + \exp\left(-\alpha_{recent}\left(\frac{T_s - T_{\text{oldest}}}{T_{\text{newest}} - T_{\text{oldest}}} - \beta_{recent}\right)\right)}, \quad \forall s \in P_0 \tag{1}$$

where T_s is the s's addition timestamp, T_{newest} and T_{oldest} are the timestamps for the most recent and oldest songs in the playlist, and α_{recent} and β_{recent}[10] represent the function's slope and mid-point, whose values were determined in our preliminary tests.

Audio Feature Weight Distribution Calculation. TASTES uses a weighted audio feature distance metric to determine track similarities within the playlist model set:

$$sim_i(s, r) = \sqrt{\sum_{a \in AF} w_{i,a} \cdot (s_a - r_a)^2}, \quad \text{for any } s, r \in P_i \tag{2}$$

where $w_{i,a}$ expresses the importance of feature a to the user. The initial weight distribution vector w_0 is given by the inverse variance[11] of each audio attribute $var_0(a), \forall a \in AF$ across the model set tracks, given it reflects how much the user cares for it:

$$w_{0,a} = \frac{1}{var_0(a)}, \quad \forall a \in AF \tag{3}$$

Facet Model Classification. TASTES employs a K-Means unsupervised clustering method to identify groupings of tracks based on audio similarity, measured through Equation (2). The algorithm is iteratively applied to P_0 for different values of k, constrained by a maximum (MAX_F) and minimum (MIN_F) number of clusters. The optimal facet partition is chosen based on the highest average Silhouette Score obtained by the distributions, ensuring meaningful and distinct groupings of tracks based on the user's perception of similarity.

Similarity Threshold Calculation. Each track's similarity threshold $D_{i,s}$ defines the minimum distance another track must be to be considered similar to the first, calculated as the distance to its furthest k-neighbour in P_0, where k is given by the average number of tracks per model facet $\frac{|P_0|}{|F|}$.

Facet Preference Distribution Initialisation. The initial user facet affinity distribution p_0 is calculated as the relative frequency of tracks in the model belonging to each playlist facet:

$$p_{0,f} = \frac{\sum_{s \in P_0} \delta(f, s)}{|P_0|}, \quad \forall f \in F \tag{4}$$

where $\delta(f, s)$ assumed the value of 1 if track s belongs to facet f, and 0 otherwise.

3.5 TES Pipeline

In the iterations following TES_0, TASTES updates the user's preference model with newly available *Spotify* feedback data before producing a new set of relevant recommendations for continuation. Each iteration aims to select suggestions that reflect the user's recent trends while encouraging the exploration of new frontiers of musical interest. All the hiper-parameter values used in our final implementation are presented in Table 1.

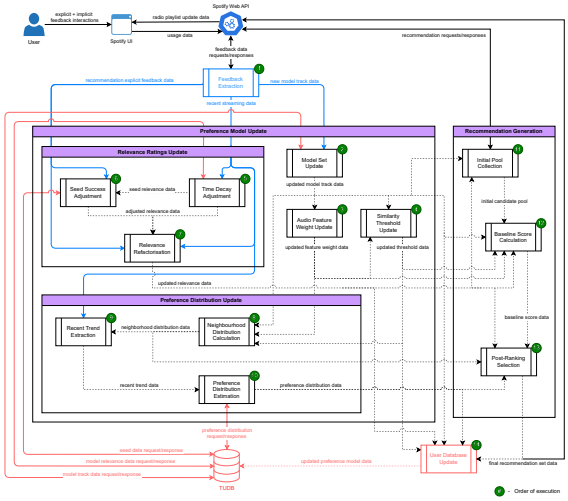

Figure 1: TES Processing Pipeline Diagram

3.5.1 ***Feedback Extraction.*** The first step in every TES_i is to extract the relevant feedback data produced by the user since TES_{i-1}, through *Spotify* usage data, using SWAPI's data retrieval functions:

- *User Playlist Data*: Collect the sets of newly added (A_i) and removed tracks (X_i) by comparing the current state of the user's playlist with the retained primary model track set $P1_{i-1}$.

[9] Problem of starting a program or operating system from a state where there is no prior knowledge or data available.
[10] The values used for the relevance function slope and mid-point are shown in Table 1.
[11] Measures the dispersion of a set of data from its mean value.

- *Recommendation Interaction Data*: Determine the sets of liked (RL_i), added ($RA_i \subset A_i$), and removed recommendations (RR_i) from the suggestions generated in the last execution, by analysing the user's radio playlist.
- *Recent Streaming Activity Data*: Create a recent stream count vector SC_i where each entry $SC_{i,s}$ gives the number of times the user has listened to each track $s \in P1_{i-1} \cup A_i \cup RL_i$ since TES_{i-1}.

Audio feature data for tracks in sets A_i, X_i, LR_i, AR_i, and RR_i are extracted and normalised.

3.5.2 ***Model Set Update***. The new relevant songs $s \in \{A_i \cup LR_i\}$ are classified within the model's facets based on the most common facet classification in their k-neighbourhoods in P_{i-1}. The model track set is then updated: $P_i = \{P_{i-1} \cup A_i \cup LR_i\} \setminus X_i$, allowing the playlist facets to evolve with new songs, adapting to the users changing preferences, in line with the research presented in Section 2.3.1. The audio feature weight distribution vector w_i and similarity thresholds D_i are recalculated for the new model set, using equations 3 and 4 respectively, and the taboo set is updated with the newly added tracks: $TB_i = TB_{i-1} \cup A_i$.

3.5.3 ***Relevance Ratings Update***. To adjust the relevance ratings U_{i-1}, TASTES first accounts for the time elapsed since the previous update:

$$U_{i,s} = \left[U_{i-1,s} \times \left(1 + \eta_{stream} \times \frac{SC_{i,s}}{days\_elapsed_i(s)}\right)\right] \times \phi_{decay}^{days\_elapsed_i(s)}, \quad \forall s \in P_{i-1} \tag{5}$$

where $days\_elapsed_i(s)$ gives the fraction of days passed since s was last streamed, or if not it returns the time passed since TES_{i-1}. The decay η_{decay} and stream ϕ_{stream} factors' values were respectively determined under the following assumptions: it should take 20 days for a track which is not streamed to become irrelevant, and 2 streams per day elapsed for it to become twice as relevant. Subsequently, given each recommendation is produced through a SWAPI request which takes a model track as seed, all past seeds' relevances are again readjusted based on the success they produced:

$$U_{i,seed} = U_{i,seed} \times (1 + \zeta_{ss} \times success(seed)) \tag{6}$$

where the seed success factor ζ_{ss} determines the impact on relevance caused by the seed's success score $success(seed)$, calculated as follows:

$$success(s) = \frac{(\theta_{add}, -\theta_{rej}, \theta_{like}, \theta_{str}) \cdot (add(s), rej(s), like(s), str(s))}{total(s)} \tag{7}$$

where $add(s)$, $rej(s)$, $like(s)$ and $str(s)$ is the number of added, rejected, liked and streamed recommendations resulting from the seed, the corresponding weights $\theta_{add}, \theta_{rej}, \theta_{like}$, and θ_{str} define the influence of each interaction on the seed's success score and $total(s)$ gives the total number of previous radio suggestions obtained using the seed. A seed s is said to have been successful when $success(s) > 0$ and unsuccessful when $success(s) < 0$. Finally, all relevances are normalised and refactored to reflect the user's trends according to the most recent feedback, from most to least relevant:

(1) $\theta_{added} \leq U_{i,s}$: Tracks newly added to the playlist (A_i), sorted from highest to lowest stream count $SC_{i,s}$.
(2) $\theta_{success} \leq U_{i,s} \leq \theta_{added}$: Successful seeds, from highest to lowest relevance $U_{i,s}$.
(3) $\theta_{liked} \leq U_{i,s} \leq \theta_{success}$: Liked recommendations (RR_i), sorted from highest to lowest recent count $SC_{i,s}$.
(4) $\theta_{model} \leq U_{i,s} \leq \theta_{liked}$: All other model tracks, from highest to lowest relevance $U_{i,s}$.
(5) $U_{i,s} \leq \theta_{model}$: Unsuccessful seeds, from highest to lowest relevance $U_{i,s}$.

The relevance thresholds defined by the track weight factors θ_{added}, $\theta_{success}$, θ_{liked}, and θ_{model} ensure each track category's relative relevance is maintained while still allowing differentiation within them. The weight factor values were determined in preliminary testing, based on the average normalised relevance of each type of track in the testers' model sets, while ζ_{ss} was determined assuming each seed's relevance should increase by half per two added recommendations produced.

3.5.4 ***Recent Trend Extraction***. The next step in the TES is to determine the user's recent trend distribution vector t_i, representing the user's recent relative preferences towards each playlist facet:

$$t_{i,f} = \sum_{s \in RT_i} \left(\frac{KNN_{i,f}(s)}{k} \times W_{i,s} \times \phi_{decay}^{days\_elapsed_i(s)}\right), \quad \forall f \in F \tag{8}$$

where $KNN_{i,f}(s)$ gives the number of k-neighbors of track s in the model set belonging to facet $f \in F$ and $RT_i = \{\{P_{i-1} \setminus X_i\} \cup A_i \cup LR_i \cup RR_i \cup X_i\}$. The influence weight $W_{i,s}$ for each type of track is determined as follows :

$$W_{i,s} = \begin{cases} \theta_{added} + (1 - \theta_{added}) \times \frac{SC_{i,s}}{\max\left(\sum_{r \in A_i} SC_{i,r}, 1\right)} & \text{if } s \in A_i, \\ \theta_{liked} + (\theta_{added} - \theta_{liked}) \times \frac{SC_{i,s}}{\max\left(\sum_{r \in LR_i} SC_{i,r}, 1\right)} & \text{if } s \in LR_i, \\ \theta_{model} + (\theta_{liked} - \theta_{model}) \times \frac{SC_{i,s}}{\max\left(\sum_{r \in RP_i} SC_{i,r}, 1\right)} & \text{if } s \in MT_i, \\ \theta_{rem} & \text{if } s \in X_i, \\ \theta_{rej} & \text{otherwise.} \end{cases} \tag{9}$$

Similarly to the previous TES process, the weight of each track type towards the recent trend estimation is determined by the weight factors, guaranteeing their relative influence aligns with the user's feedback.

3.5.5 ***Current Trend Estimation***. The facet preference distribution vector p_i is then adjusted the retained preference vector p_{i-1} and recent trend vector t_i, as follows:

$$p_{i,f} = (1 - \gamma \times days\_elapsed_i) \times p_{i-1,f} + \gamma \times days\_elapsed_i \times t_{i,f}, \quad \forall f \in F \tag{10}$$

where $\gamma = 1 - \phi_{decay}$ is the daily update factor. If $days\_elapsed_i \geq 20$ then $p_{i,f} = t_{i,f}$.

3.5.6 ***Recommendation Generation***. With the preference model updated, TASTES moves on to generate the new set of N_FINAL recommendations in the following steps:

Initial Pool Collection. Retrieve the initial set of tracks for recommendation I_i through single-seeded SWAPI recommendation requests of size N_REQ. The likelihood of a model track being selected as a seed is given by its relative relevance in P_i. The requests

Identifier	Value	Identifier	Value	Identifier	Value
α_{recent}	1.0	ζ_{ss}	.25	θ_{rej}	-.20
β_{recent}	.50	θ_{add}	.80	θ_{rem}	-.40
MIN_F	4	$\theta_{success}$	.70	γ	.10
MAX_F	15	θ_{liked}	.50	λ	.50
η_{decay}	.90/.80	θ_{str}	.30	N_REQ	20
ϕ_{stream}	.50	θ_{model}	.20	N_FINAL	15

Table 1: Final Implementation Hiper-Parameter Values

continue to be performed until $|I_i| = N\_REQ \times N\_FINAL$ tracks are obtained. If any suggestion's track ID is already in the tabu set TB_i, it is discarded before adding the rest to I_i.

Baseline Rating Calculation. Calculate a baseline rating B_i for each track, determining its overall alignment with the model set based on its similarity with each track in P_i and their relevance U_i:

$$B_{i,r} = \sum_{s \in P_i} kmd(r,s) \times U_{i,s}, \quad \forall s \in I_i \tag{11}$$

$$kmd(r,s) = \begin{cases} \frac{D_{i,s} - sim_i(r,s)}{D_{i,s}} & \text{if } sim_i(r,s) < D_{i,s}, \\ 0 & otherwise. \end{cases} \tag{12}$$

Post-Ranking Selection. Following Zhao et. al.'s re-ranking process [Zhao et al. 2021], the final set is iteratively built by selecting the track $r \in I_i$ which maximises the inclusion value function:

$$V_{i,r} = \lambda \times \sum_{s \in R_i \cup \{r\}} \frac{D_{i,s}}{|R_i|+1} + (1-\lambda) \times kld(p_i \| \sum_{s \in R_i \cup \{r\}} \frac{N_{i,s}}{|R_i|+1}), \quad \forall r \in I_i \tag{13}$$

where $kld(a\|b)$ is the Kullback-Leibler divergence between distributions a and b and $N_{i,s}$ is the k-neighborhood facet distribution of track s. The continuation/expansion tradeoff factor λ balances the relative weight of the two aspects which determine the inclusion value: the average baseline rating of R_i after adding a candidate r to the set, and the KLD between the user's preference distribution p_i and the average facet distribution $N_{i,s}$ of tracks in set $R_i \cup \{r\}$. Once the final set R_i is attained, it is then published in the user's radio playlist, and all track IDs are added to the taboo set TB_i, concluding the TES.

3.6 SRSE

SRSE is a sequential recommendation approach similar to TASTES but without employing any of its interest-aware or taxonomy-free framework features. Its goal is simply to produce the most relevant suggestions at each iteration based on *Spotify*'s methodology by leveraging the services offered through SWAPI. SRSE and TASTES always operate independently of each other.

3.6.1 Retained Preference Structures. SRSE only requires three of the data structures employed by TASTES: the **model track set** vector P_i, containing all audio feature data for relevant tracks, without facet classification; the **relevance ratings** vector U_i, expressing the user's affinity towards each relevant track and the **tabu set** vector TB_i, to avoid repetition in recommendation sets. These data structures follow the architectural description provided in 3.3 and are maintained through TUDB, similar to TASTES.

3.6.2 Setup Step. The strategy's initial step is nearly identical to TASTES, initialising the retained model structures through the same processes, up to the relevance rating estimation stage. Afterwards, it moves directly to recommendation generation, described ahead.

3.6.3 Standard Processing Pipeline. After the initial execution step, SRSE follows a simplified version of the TASTES pipeline, with only three processing steps:

Feedback Extraction. SRSE extracts all new feedback data since the previous iteration, similar to TASTES, including playlist modifications, interactions with previous recommendations, and recent *Spotify* streaming data. Sets P_i and TB_i are updated with this data following the same process described in 3.5.2.

Relevance Adjustment. After updating the preference model set, SRSE adjusts relevance ratings for all tracks in P_i. The first stage is identical to TASTES' time decay adjustment, described in 3.5.3. Afterwards, SRSE normalises and refactors all relevance $U_{i,s}$ using three value tiers instead of five, from most to least relevant:

(1) $U_{i,s} \geq \theta_{added}$: Newly added recommendations, from highest to lowest $SC_{i,s}$.
(2) $\theta_{added} \geq U_{i,s} \geq \theta_{liked}$: Liked recommendations, from highest to lowest $U_{i,s}$.
(3) $\theta_{liked} \geq U_{i,s}$: All remaining model tracks, from highest to lowest $U_{i,s}$.

The values for track weight factors θ_{added} and θ_{liked} are the same for both TASTES and SRSE.

Recommendation Generation. SRSE generates the final set of recommendations R_i using single distinct seeded SWAPI recommendation requests, selecting the model track seeds similarly to its counterpart. Unlike TASTES, SRSE solicits a single suggestion per request ($N\_REQ = 1$) to ensure the most relevant track is generated by *Spotify* for each seed. SRSE continues to make requests until it has N_FINAL tracks. If a suggestion's track ID is already in the tabu set TB_i, it is discarded. Once R_i is attained fully, the user's radio playlist is updated with the new tracks whose IDs are added to the tabu set TB_i, concluding the SRSE execution step.

4 EVALUATION

To assess TASTES' capabilities, we devised an evaluation framework centred on capturing its performance for PC tasks on *Spotify* compared to the baseline strategy SRSE. In line with our research questions, our comparative analysis was informed by both strategies' performance in two aspects:

(1) The ability to generate relevant suggestions that extend beyond the existing musical themes in a playlist.
(2) User satisfaction with the overall music discovery experience provided.

We looked to inform these aspects both from a quantitative perspective and according to the user's perceptions after experiencing each strategy.

4.1 Experimental Setup

Our testing setup was designed to emulate a PC scenario on *Spotify*, allowing testers to access and react to recommendations through

their existing profiles. The testing process consisted of a series of Recommendation Feedback Cycles (RFC) structured as follows:

(1) TASTES or SRSE produce a set of suggestions for continuation using the user's most recent preference data.
(2) The recommendations are published in the user's radio playlist.
(3) The user provides feedback on the suggestions through *Spotify* interface interactions.
(4) A new processing cycle is manually triggered by the evaluators when the user is ready for more recommendations.
(5) The user's new feedback is processed, updating the strategy's preference model and restarting the RFC.

The evaluation involved putting a group of *Spotify* users through a series of RFCs fueled by both TASTES and SRSE, collecting the same performance data for each series to conduct a posterior comparative analysis to inform our research questions.

Personal Playlist. Before starting the evaluation, each tester selected a playlist from their *Spotify Library*. We refer to it as the participant's **personal playlist**, defining the area of musical interests the strategies will attempt to cater to. The only requirement for the personal playlist, besides being curated by the participant, was a minimum length of 20 distinct tracks, to ensure enough data is available for TASTES to build its initial facet model. The playlist remained the same for each participant throughout the entire evaluation period to ensure consistent testing conditions for both approaches.

4.2 Preliminary Testing

In preparation for the primary evaluation, a preliminary assessment was conducted with three main objectives: to calibrate the tested strategies' hyper-parameters to maximise performance within the limited timeframe of the primary evaluation, to test all components of TASTES and SRSE to confirm the processing pipeline works as intended and to gauge the impact of each approach on recommendations from the userperspectiveive. We selected 3 testers (1 female and 2 male, ages 23-25) based on availability, usage frequency, and familiarity with *Spotify*. The tester selected a personal playlist from their *Spotify Library* with the following genres: Hip-Hop (313 tracks), Electronic (156 tracks), and Indie Pop (72 tracks).

4.2.1 Testing Process. The preliminary tests consisted of 6 RFCs, divided into 2 series of 3, where testers were suggested 3 sets of 20 tracks produced by each tested strategy for the continuation of their playlists. Testers were informed about the explicit recommendation feedback responses interpreted by the strategies. Figure 2 shows the user's radio playlist view in *Spotify*'s UI, where the suggestion sets were published at the end of each RFC, awaiting new reactions. The preliminary testing lasted 8 days. Testers' feedback confirmed noticeable differences between suggestions from TASTES and SRSE, helping refine the testing methodology.

4.2.2 Resulting Adjustments. Based on preliminary tests, we adjusted TASTES' and SRSE's parameters for the primary evaluation (see Table 1 for final values). Key changes included reducing recommendation sets to 15 tracks per RFC, setting a fixed 3-day interval for user feedback, and developing a comprehensive guideline document for participants.

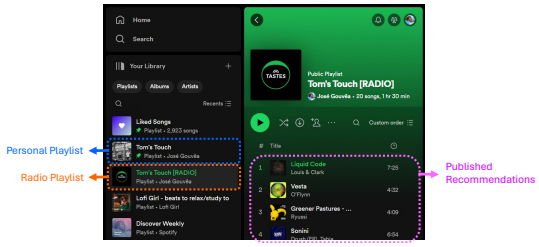

Figure 2: *Spotify Library* User View Example

4.3 User Study

As the primary method for evaluating our approach, we conducted a 30-day within-subjects study with a total of 20 participants. The aim of the study was to expose each tester to two identical series of RFCs while employing each recommendation strategy, similarly to the preliminary testing approach, and collect the results produced in each set for our final analysis.

Sampling. Over 30 *Spotify* Premium users with manually curated playlists were interviewed for the study. The final 20 participants met the following criteria:

- At least one year of active *Spotify* usage.
- Minimum of 3 hours per week listening to their personal playlist.
- A personal playlist with at least 20 tracks.
- Availability for the entire study period.

Participants were asked to select personal playlists they listened to regularly, preferably during the same activity, to maximise relevant data and control contextual variables. The playlist remained the same throughout the entire study, to ensure consistency in the focal area of the user's interests, contributing to a fair assessment of both strategies.

4.4 Procedure

The primary evaluation followed a structure similar to the preliminary testing. Conducted over 30 days, it was divided into two 15-day segments. Each participant received 5 sets of 15 recommendations, produced by each tested strategy every 3 days, resulting in five 3-day-long successive RFCs per segment. The participant group was split in half to control for sequential variables: one half tested SRSE in the first segment and TASTES in the second, while the other half did the reverse. All were provided with an overview of the study's structure and timeline as well as an explanation of the explicit recommendation feedback reactions and how TASTES interprets them, as detailed in Section 3.2. Participants were also told they were testing two configurations of the same strategy, without any comparative descriptions, to avoid bias and ensure impartial evaluations.

4.5 Data Collection

During the study, three types of data were collected for our comparative evaluation of the strategies' performance:

4.5.1 ***Playlist Audio Composition Data.*** The variance of each audio feature and the number of facets found by TASTES, before and

after each segment. This data was collected through TUDB and used to analyse the strategies' impact on the playlist's audio composition, informing our quantitative assessment of the strategies' interest expansion ability.

4.5.2 ***Explicit Recommendation Feedback Data.*** For each segment, we collected the total number of radio playlist tracks which were added to the user's playlist, liked by the user and removed from the radio playlist. This data was logged into TUDB, at the end of each TES, and used to evaluate the accuracy of both strategies.

4.5.3 ***User Feedback Data.*** We designed two surveys aimed at capturing the user's overall inclinations as music listeners and *Spotify* users, and their perception and satisfaction with different aspects of the recommendation experience in each segment.

Initial Profiling Survey. Before the start of the study, participants completed a preliminary questionnaire about their demographics, music preferences, *Spotify* usage habits, and personal playlists. This data helped us understand the study sample's characteristics and their potential correlation with the results obtained from the remaining data collected.

User Feedback Survey. Participants filled out a feedback survey at the end of each segment to capture their impressions of the recommendation experience, including perceived diversity, accuracy and overall satisfaction. The surveys were identical for both segments, allowing a direct comparison between TASTES and SRSE, as participants provided feedback knowing they were evaluating distinct methodologies. We also provided the participants with their answers to the first survey before filling out the second survey, to facilitate a more accurate comparative assessment. The surveys were performed using Google Forms. We discuss the relevant findings in the following section.

5 RESULTS

5.1 Participant Sample

A total of 20 candidates participated in the evaluation study. All participants were *Spotify* Premium users with at least one year of active usage. The following data was collected through the initial profiling questionnaire. The group consisted of users aged 21-26, with 40% females and 60% males, mirroring *Spotify*'s majority user base demographics [Shepherd [n. d.]].

5.1.1 ***Musical Preference.*** The survey data (Figure 3) regarding the participants' music listening habits reflects the following relevant findings:

- IM1: 85% regularly integrate music into their daily lives.
- IM2: Mixed responses on mainstream preference alignment.
- IM3: 50% favour listening to music they already know.
- IM4: 75% claim not to get tired of listening to new music.
- IM5: Mixed regarding certainty of music they wish to hear.

5.1.2 ***Spotify Usage.*** In regards to streaming habits, most participants have used *Spotify* for at least 5 years, with an average of 7.8 years, and spend 5-20 hours per week on the platform, aligning with the most recent *Spotify* usage statistics [Shepherd [n. d.]] which show an average of about 16 hours of usage per week. Most participants rated their familiarity with *Spotify* features at least a

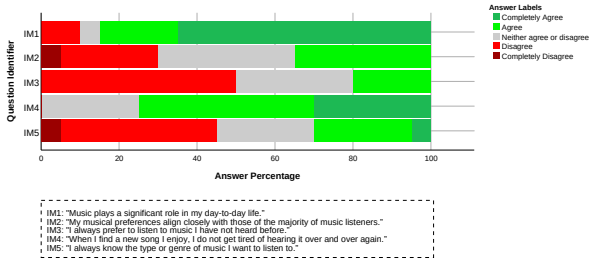

Figure 3: Participant Music Preferences

4/5. Figures 4 and 5 show that 75% of participants spend at least 75% of their *Spotify* time listening to personal playlists, but only half add new tracks weekly, with another 30% stating it is a monthly occurrence. In regards to the sample's general satisfaction with *Spotify*'s recommendation capabilities, the results express a mean score of 3.6 out of 5.

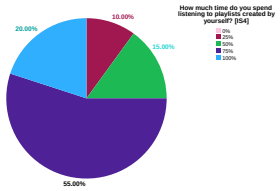

Figure 4: Time Spent Listening to Self-Curated Playlists

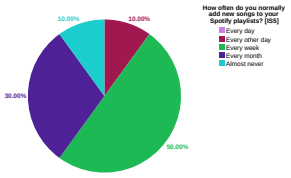

Figure 5: Frequency of Additions to Personal Playlists

5.1.3 ***Personal Playlist.*** The initial user playlist pool analysis is shown in Figures 6, 7 and 8. Playlists varied widely in size, with a mean facet count of 6.4 and an average of 46.27 songs per facet, in line with the results of Kaya and Bridge's sub-profile analysis [Kaya and Bridge 2018]. Figure 9 shows the playlists' genre distribution, provided by the applicants. The participants were also asked to rate their playlists' diversity and repetitiveness on a 1-9 Likert scale. Both distributions show diverse responses with a mean close to 5. The perceived diversity of the personal playlists leans slightly towards homogeneity, with a median score of 4, while the perceived repetitiveness indicates a slight tendency towards monotony, with a median score of 6. Despite this, most participants rated their playlists between 6 and 8 on the listening experience provided, with a mean rating of 7.25.

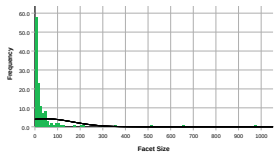

Figure 6: Personal Playlist Facet Size Distribution

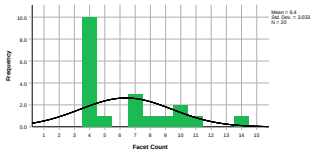

Figure 7: Personal Playlist Facet Count Distribution

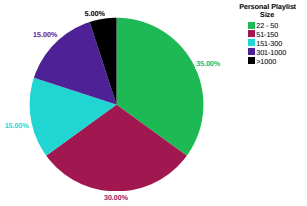

Figure 8: Personal Playlist Size Distribution

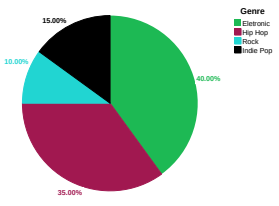

Figure 9: Personal Playlist Genre Distribution

5.2 Study Results

We now present and detail our analysis of the study results. All study participants scored at least a 4 out of 5 when asked if they had usage difficulties with the experimental setup during the study, indicating no significant influence of user error in our analysis.

5.2.1 ***Musical Interest Expansion.*** To evaluate the first research question, we conducted quantitative data analysis by comparing the relative changes in the area of musical preferences covered in the personal playlists as a result of each strategy. Additionally, we performed a qualitative data assessment focused on capturing the participants' perceived interest shifts after testing each recommendation approach.

Quantitative Data Analysis. The quantitative assessment of the strategies' capacities for taste expansion centred around two key aspects: the impact of each strategy on the playlists' audio feature distribution, informed by changes to the features' variances, and the difference in the number of facet clusters detected by TASTES' unsupervised classification framework before and after each testing segment. We employed Wilcoxon's non-parametric signed-rank test for the comparative analysis of playlist audio feature impact., the results revealed the following observations:

- TASTES' suggestions led to an average decrease in the variance of all audio features. SRSE produced similar effects with some exceptions which increased slightly in variance.
- Both strategies caused an average decrease in the number of facets identified by the unsupervised classification method, with a mean change of -.75 facets for TASTES and -1.05 facets for SRSE, showing no significant difference in impact.
- Wilcoxon's signed-ranks results show that TASTES impacts five audio features significantly more than SRSE: *liveness* ($Z = -2.389$, $p = .017$), *loudness* ($Z = -2.912$,$p = .004$), *speechiness* ($Z = -2.016$, $p = .044$), *tempo* ($Z = -2.464$, $p = .014$), *valence* ($Z = -2.688$, $p = .007$)

Qualitative Data Analysis. Participants were inquired about 4 core aspects after experiencing each tested approach, using the same 9-point Likert scaled questions:

- Overall diversity of the participant's personal playlist.
- Strategy's ability to produce recommendations outside of the music interests represented in the personal playlist.
- Accuracy of recommendations outside of the music interests represented in the personal playlist.
- Overall diversity of the produced recommendations.

After performing a Wilcoxon's signed-rank test over the answers for each aspect, the results indicate no significant correlation between the responses given regarding SRSE and TASTES. Ultimately, both approaches seem to produce identical results concerning the perceived diversity of the track selections and interest expansion ability.

5.2.2 ***Music Discovery Experience.*** Regarding the second research question, our quantitative data analysis focused on feedback responses (i.e., "likes", additions, and rejections) to track suggestions produced by each methodology, while the qualitative evaluation was informed by participants' impressions of the music discovery journey during each testing segment collected through the feedback surveys introduced in section 4.5.3.

Quantitative Data Analysis. We evaluated SRSE and TASTES' overall accuracy through the number of relevant tracks produced as suggestions. A recommendation is considered relevant if the user attributes a positive reaction, i.e. a "like" or adding it to their playlist. A Wilcoxon's signed-rank test was conducted on the total count of added, liked, and rejected recommendations, and the percentage of songs added to the user's playlist during each segment which originated from the radio playlist.

The results showed no significant differences between the methodologies. Although TASTES had a higher mean of 14.8 total added recommendations produced per participant compared to SRSE's 12.7, the p value is not small enough to be considered a relevant improvement. Thus, both approaches show similar accuracy in the recommendations produced.

Qualitative Data Analysis. Participant feedback was collected through surveys filled out after each 15-day segment. All testers rated their perception and/or satisfaction regarding the same aspects of their discovery experiences in each segment, through 1-9 Likert-scaled questions pertaining to the following factors:

- Level of repetitiveness/monotony of the personal playlist.
- Overall rating of the personal playlist.
- Genre coverage of the presented suggestions belonging to the musical themes included in the personal playlist.
- Accuracy of recommendations belonging to themes included in the personal playlist.
- Accuracy of recommendations belonging to themes not included in the personal playlist.
- Overall satisfaction with the PC experience provided.

Once again, a Wilcoxon's signed-rank test was performed with the responses to each segment, revealing no significant differences between the strategies perceived experience for the evaluated aspects, when considering the totality of the sample. Further analysis using Spearman rank-order correlation coefficients between the differences in survey responses and initial audio feature variances of their playlists revealed the following significant correlations:

- A positive correlation between the accuracy of recommendations within playlist genres and the strategy's capacity to cover all playlist genres ($\rho(18) = .553$, $p = .011$).
- A negative correlation between the accuracy of recommendations outside the playlist genres and perceived monotony of the user's playlist ($\rho(18) = -.532$, $p = .016$).

- A positive correlation between participants' rating of their playlists and both the strategy's coverage of themes in the playlist($\rho(18) = .499$, $p = .025$) and the accuracy of the suggestions within those themes ($\rho(18) = .470$, $p = .036$).
- A positive correlation between satisfaction with the music discovery experience and the accuracy of recommendations belonging to themes in the playlist ($\rho(18) = .666$, $p = .001$).
- A positive correlation between personal playlist initial *instrumentalness* variance and greater overall experience satisfaction with TASTES over SRSE ($\rho(18) = .601$, $p = .005$).

The only significant correlation found involving playlist audio features pertained to *instrumentalness* variance. When analysing the playlist sample according to this audio characteristic, we found that 8 of the 20 playlists used in the study show a considerably lower value than the rest, as illustrated in Figure 10. Accordingly, a Mann-Whitney U test was conducted between two participant sub-groups for further investigation:

- **Group A** (8 participants): Initial *instrumentalness* variance between .00 to .03.
- **Group B** (12 participants): Initial *instrumentalness* variance between .06 to .20.

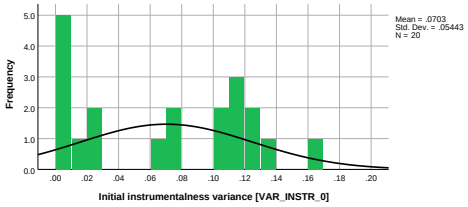

Figure 10: Initial Instrumentalness Variance Distribution

The Mann-Whitney results revealed several key findings. Group A's playlists exhibited significantly higher speechiness variance ($U = 16$, $p = .014$) than Group B's. Group A also scored higher on survey questions IM2 ($U = 17$, $p = .012$) and IM4 ($U = 22$, $p = .031$), as detailed in Figure 3, reflecting more alignment with mainstream trends and less susceptibility to being fatigued with new songs. Additionally, Group A spends more time on *Spotify* per week ($U = 14$, $p = .007$) compared to Group B. On the other hand, Group B scored higher for TASTES' ability to suggest relevant songs within playlist themes ($U = 18.5$, $p = .017$) and for achieving a more satisfying music discovery experience ($U = 7.5$, $p = .002$) over SRSE. A Wilcoxon's signed-rank test using only Group B's satisfaction survey responses also confirmed that TASTES had significant improvements in both these aspects ($Z = 2.124$, $p = .034$ and $Z = 2.546$, $p = .011$). Furthermore, Group A included 6 Hip-Hop, 1 Electronic, and 1 Indie Pop playlist, while Group B comprised 7 Electronic, 2 Indie Pop, 2 Rock, and 1 Hip-Hop playlist.

6 CONCLUSION

Our findings addressed the two primary research aspects:

Effectiveness at expanding musical interests. TASTES showed no measurable improvement in expanding users' interests over SRSE, indicating its interest-aware framework does not offer added capability for musical taste expansion over conventional methods.

Impact on music discovery experience. TASTES performed better than SRSE for users who value instrumental content over spoken word. It generates more relevant suggestions within the user's existing interests and provides a superior music discovery experience overall. When considering the entire study sample, TASTES matched *Spotify*'s recommendation approach results for this aspect.

The results indicate that TASTES is more suited for exploration rather than expansion, especially where instrumental similarity is significant. The successful integration with *Spotify* demonstrates the feasibility of integrating external recommendation systems within established platforms.

6.1 Limitations

Several limitations to our proposed solution and evaluation methodology were identified: The **sample size** was small and may not capture the diversity of *Spotify* users' preferences; the **evaluation time frame** was limited and may not adequately assess the system's long-term effectiveness and adaptability; frequent **user interaction** requirements may have influenced user behaviour, deviating from typical usage patterns; **self-reported data** may contain potential biases and inaccuracies, impacting the reliability of evaluated metrics; reliance on **audio features** ignored intangible factors like lyricism and cultural impact, potentially limiting TASTES' ability to fully capture user preferences; and **data collection limitations** due to dependence on SWAPI's data services imposed constraints that ultimately affected the strategy's performance.

6.2 Future Research

Our proposed future research directions include: **parameter tuning** to conduct extensive experiments with different hyper-parameter values to optimise the strategy's performance; a **long-term study** to evaluate the system's adaptability over a longer time frame with a larger, more diverse sample of *Spotify* users; **enhanced data collection** to incorporate objective measures of user engagement and satisfaction to complement self-reported data; **real-time adaptation** to improve preference adaptability with dynamic adjustments based on immediate user feedback; and **hybrid integration** to explore incorporating TASTES with strategies that account for intangible factors beyond audio features.

REFERENCES

Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. 2014. An analysis of users' propensity toward diversity in recommendations. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 285–288.

Mesut Kaya and Derek Bridge. 2018. Automatic Playlist Continuation Using Subprofile-Aware Diversification. In *Proceedings of the ACM Recommender Systems Challenge 2018*. 1–6.

Mesut Kaya and Derek Bridge. 2019. Subprofile-aware diversification of recommendations. *User Modeling and User-Adapted Interaction* 29 (2019), 661–700.

Noah Dani Krebbers. 2020. *Automatic Categorization of Electronic Music Genres*. Master's thesis. Utrecht University.

Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. 2006. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine* 23 (2006), 133–141.

Jack Shepherd. [n. d.]. *23 Essential Spotify Statistics You Need to Know in 2024*. https://thesocialshepherd.com/blog/spotify-statistics

Spotify. [n. d.]. *Web API*. https://developer.spotify.com/documentation/web-api/reference/get-audio-features

Xing Zhao, Ziwei Zhu, and James Caverlee. 2021. Rabbit Holes and Taste Distortion: Distribution-Aware Recommendation with Evolving Interests. In *Proceedings of the Web Conference 2021*.