BUAD 313 Final Project: Custom Spotify Playlist

Riley Cox, Justin Shih, Matthew Dolins, Brennan Shirer

Professor Vishal Gupta

BUAD 313 - Advanced Operations Management and Analytics

8 May 2025

For our final project, our team decided to design an advanced model that would create a custom Spotify playlist for individual users based on their ratings of various songs, as well as several other key factors. To create our initial base model for the project, we used two data sets that were given to us, which included key information such as artist name, track name, and most importantly, user rating for each song. Our goal for the base model was to build a model that would create a unique playlist with 30 songs for an individual user based on their preference for song ratings.

To begin designing the model, we first had to create decision variables. We created two binary decision variables for the base model, X[i] and Y[i]. These variables represented, respectively, whether or not the song existed in the playlist and whether or not an artist appeared in the playlist. Next, we created constraints to ensure that the playlist was exactly 30 songs and that no artist or song was repeated. With these constraints, we were able to create variety in artists and songs so that they were not too repetitive. Our objective function maximized the user ratings to get the best 30 songs for the individual user. This provided a foundation for our project to curate music selections using specific user preferences.

Once we formulated the optimization problem, we implemented the model in VS Code using Gurobi. Our model worked successfully and was able to generate a 30-song playlist for an individual user based on their preferences. This base model served as a simple yet effective starting point to further build upon, so we could make it more tailored to specific user requests. It is important to note that the base model operated under the assumption that a playlist's predicted rating was solely optimized by aggregate predicted user enjoyment (user rating).

However, there are some major limitations in the base model that severely hinder its ability to create a well-crafted playlist. Its first shortfall is that the playlists it creates are too diverse. The model essentially acts as an extension of Spotify's liked songs playlist, where it only prioritizes a thumbs up/high rating, and nothing else. It doesn't allow for more than one song per artist, which may seem like it's encouraging the user to explore new artists, but artists' songs often go well together. Similarly, the playlist also does not account for genre, mood, or tempo, which makes the model less effective as it only factors in the previous user rating, instead of considering a variety of important factors in music. Because of this, the model lacks a user-centric and context-aware playlist design. For example, a user could be an avid music lover and have liked songs across vast genres. When the base model develops a playlist for this user, it will put all of the top-rated songs on the same playlist, regardless of their genre or vibe, which can create a very chaotic and disordered feel. Our team wanted our playlists to be coherent and have purpose, not just be a jumble of random songs the user might enjoy with no apparent flow. Another key factor when creating a playlist is adding songs that blend well and flow together nicely. Going off of this, another weakness of the base model we found was that the songs that were selected were in no particular order or optimized order. This is fine if the user just wants to press shuffle, but in an ideal playlist, just like any good album, it should tell a story as it plays, with an intro, a middle, a climax, and then a finale. We leveraged these inefficiencies to serve as the foundation to refine our model to better reflect real-world complexities in the next steps of our project.

After developing the base model, our team decided to address the key limitations with a refined model that incorporates song mood, energy, genre, tempo, and other factors to generate more personalized and specific playlists based on the users' requests. This way, we can directly relate emotional and activity-based settings to each other. We found that while the initial data set had useful information, in order to truly create a unique user experience, it would require further considerations such as energy, danceability, and valence etc. With that in mind, we merged additional data from Kaggle with our current dataset to incorporate mood and deeper specificity into the model. To allow for comparison, we used this data for both models so that we'd be able to see the energy, danceability, valence, etc., on the songs in each model's generated playlist. It is important to note that while these additional variables were printed in the base model, they did not change any of the prior constraints. To reflect this additional data on our advanced model, our team created deeper constraints that made these playlists more aligned with real-world consumption behavior.

We defined over twenty specific moods and activities to capture how users feel, each with slightly different constraints, then grouped them into five parent playlist styles: party, study, exercise, relaxation, and emotional. Each parent style has its own set of tailored thresholds that drive our overarching algorithmic constraints. Using our new dataset, we had access to many different variables that could manipulate our songs into subcategories, including genre. When trying to explore genre as a potential constraint, we found that there was greater relevance in the quantitative audio features behind what defines a genre, versus the qualitative genre column itself, as it was often miscategorized. Because of this, we primarily focused on manipulating the energy level, danceability, valence, and other quantitative variables.

To classify the genres, we developed constraints for each of the five parent groups. We decided to test various constraints, such as how "high vs low energy", "high vs low danceability", "high vs low tempo", etc., impact playlist ratings, and we discovered which constraints worked well with each specific playlist. For example, a typical "party" playlist will have songs that require an energy level above a "0.7", a tempo of above "115", and danceability ratings above "0.8". Meanwhile, a study playlist that would want much more calm music has energy levels lower than "0.5", instrumental higher than "0.3", and a max tempo of "120". We continued to create constraints similar to these for each of the mood ratings, in hopes of encapsulating how each genre can be quantitatively classified.

In addition to these constraints for mood categories, we also changed the number of allowed songs per artist. To determine what number to set it to, we ran a sensitivity analysis. Using the original dataset with user ratings, we ran through 5 different values of songs allowed per artist for all users, and compared the amount of average diversity in playlists versus the normalized total average ratings per song in the playlist (Figure 1). It was clear from Figure 1 that increasing songs allowed per artist would lead to a loss in artist diversity; however, as shown in Figure 2, there was also a steep increase in the average ratings of a playlist. Nonetheless, the increase from 1 to 2 allowed songs per artist presented a large increase in user ratings, so we raised it to 2. This served as an optimal point between both considerations, where neither tradeoff (namely the rating and diversity) was too far exploited on either side, since the base model forced there to be thirty unique artists, which greatly sacrificed the user ratings. It is important to note that the testing done was under the assumption that there were 30 songs in the playlist, since that's how our base model was built.

We also ran a sensitivity analysis on the number of songs in a playlist and its effect on average user ratings and diversity. We found that 30 songs in a playlist offered more diversity and artist appearances, however, there was a significant loss in average user ratings. To improve this, we focused on looking for a playlist length that would create as much diversity as possible while still maintaining a higher level of user ratings. From our analysis, we discovered that playlists with 20 songs had the highest increase in artist diversity, while also having the lowest decrease in ratings (Figures 3.1 & 3.2), so we decided to change our playlist length from 30 songs to 20 songs. However, as a future consideration, playlist diversity would benefit from more songs, and provide the user a more balanced listening experience overall, since a big concern with Spotify playlists is that you do not want them to be so short that the user feels like they are always hearing the same songs.

Beyond those changes, we kept the objective function the same, such that we want user ratings to be optimized throughout, so that enjoyment is still our priority. We found this aspect of the base model to be sufficient, especially for low-rated songs, more so than high, as we do not want a user's disliked songs to end up in the playlist. We also kept the constraint that no song should be repeated, as twenty songs should all be counted as unique.

Next, we focused on constraints that would optimize the flow of a playlist. The base model allowed songs with high energy to immediately transition into slow-tempo songs, which did not create a seamless experience. This random order did not allow for an optimal experience, as it was based on random chance instead of curated flow orderings. A good playlist tells a story, such that a user can *feel* smooth transitions throughout the songs and not become emotionally overstimulated by vastly different songs playing back-to-back. We took the playlists that were created from this model and created an additional model to order each of these created playlists in a flow that benefited the user experience.

While we conducted similar analyses for all moods and parent groups, we will use the party playlist that we generated specifically for "user_b4beed9bf653604a876fdfd9df59e19c" as our example. A good party playlist will begin with a hook, which is 1–3 songs that are known as “familiar bangers.” We enforce this by constraining those songs to be in the top 10% of popularity among the 20 selected songs (Figure 10). By starting a party playlist with hit songs, it draws in the listeners as they will feel encouraged and likely to exclaim things like “I love this song!” The playlist should ramp up with more energy in the next 4–7 songs, raising the energy of the group. To implement this, we added a constraint to have the valence levels above 0.7 to maintain excitement (Figure 11). Songs 8–12 should be the zone of discovery, where songs are not as popular. We have this fragment coming after the energy levels are already high, as the group will already be hooked in from the popularity and high energy. In doing this, we added the constraints to have popularity in the 40th–70th percentile within the playlist and had these songs descending in energy and danceability. After the zone of discovery, there should be a gentle dip for songs 13–15 in which energy is eased and listeners have a breather before the finale. The base model consistently had staggering variability in tempo, so we want this flow to come naturally, which is why we ensured that tempo throughout each category was ordered in an ascending order then descending in the next section. Lastly, there should be a comfort finale for songs 16–18 in which we return to safe songs that are above the 85th percentile in popularity. This not only ensures that the crowd continues to be excited with familiar favorites, but it will also easily flow into the beginning of the playlist again, if left on a loop.

In Figure 4 from the appendix, we observe how the new ordering system helped to create a more stable and predictable energy level throughout the party playlist when compared to the base model. This graph highlights how the base model doesn’t incorporate a consistent flow within the playlist, while the advanced ordering model we created does. The base model would leave a crowd on a high, then move into a slow start with no warning, leading to disappointment in “maintaining the crowd high.” This was shown in the large spikes and dips in the graph, as each song was not dependent on the song before it. Meanwhile, the advanced ordering model stabilizes the energy level throughout the party playlist to create more flow for the user. This is proven in the graph as each song follows a stable pattern where it has a gradual rise and fall as each song plays. It still moves from faster to slower-paced songs, but does so in a way that the user will anticipate and crave it, as they are already mentally moving towards that pace in the song prior.

To test our additions to the model, we decided to run a robustness analysis to measure the average change in tempo between adjacent songs, and “vibe” level between different songs in our playlists. To do this, we added a noise factor to the different user ratings and tested our model to see if it still performed under more uncertainty regarding the user ratings. Using Gaussian noise values from 0.05 to 0.30, we simulated different user ratings than the ones in the dataset, to see how consistent our model would be in creating the same vibe and flow. In Figure 12, the changes in tempo remained remarkably stable, with a standard deviation of only 1.90 BPM. Figure 13 also shows extremely minimal variance with all the different noise, with a standard deviation of 0.01 for average energy, 0.02 for valence, and 0.01 for danceability. These robustness tests help confirm that the model is not too sensitive to user rating data, and can perform well even with noisy or variant data.

However, we recognize that there are many improvements still to make to our new model, starting with the many assumptions made when formulating our mood categories. Without any data to back up our constraint numbers for the different audio features in the specific playlists, since our base model didn't use that information, we used lots of personal judgment calls to assign constraint coefficients. A big improvement for our model would be to gather data to run analyses to test to see what the most optimal constraints would be to fit specific moods (such as energy levels in each section). Another weakness of our model is that we had to sacrifice our user rating objective function in order to try to create flow and theme within a playlist. Our objective function was still focused on maximizing the user ratings, but overall, our curated playlists do have lower ratings than our base model, given that we operated under the assumption that a user would like our overall playlist more for its flow and specificity, versus previous rating alone. We also recognize that when developing the ordering for a playlist, we made assumptions regarding the ideal lengths and styles of each section, such that every user would want the same energy/valence level and build up of ascent/descent per section. In order to improve this, we would ideally gather more data regarding the best hooks, build-ups, and climaxes of different styles of playlists to tweak them further. Additionally, we could also improve our transitions between songs. Currently, our model only takes into account the tempo/BPM, when in reality, there are many things, including a song's EQs, key, and structure, that can help make transitions between songs more seamless. Finally, another weakness we must acknowledge is that while our dataset did contain a lot more individual song information, it did come at the expense of losing a lot of songs. The dataset provided to us at the start had almost 20,000 songs, ours had just under 2,000. Because of this, some of our analyses and tests came out with skewed results reflecting the much smaller pool of songs to choose from for any given playlist.

While the end goal always remained the same- ensuring ratings were high, we discovered that artist variability, tempo, energy, popularity, valence, and other factors were key as well. In order to create an optimal playlist, our team decided there should be a strong underlying flow, both in how the factors (such as energy) are structured, as well as how songs' tempos transition into one another. As we look towards future considerations, we acknowledge the shortcomings of our current model and that a larger database of songs will likely lead to more fruitful findings. Overall, we found that creating a playlist that caters to a user is not just about evaluating their personal song preferences, but also about how best to cater to their current environment and mood, and create a smoother listening experience.

Appendix

Figure 1: Sensitivity Analysis - This figure shows the diversity in artists against the expected ratings from users. As diversity increases, we note that the user rating decreases, which makes sense as the songs have not yet been rated.

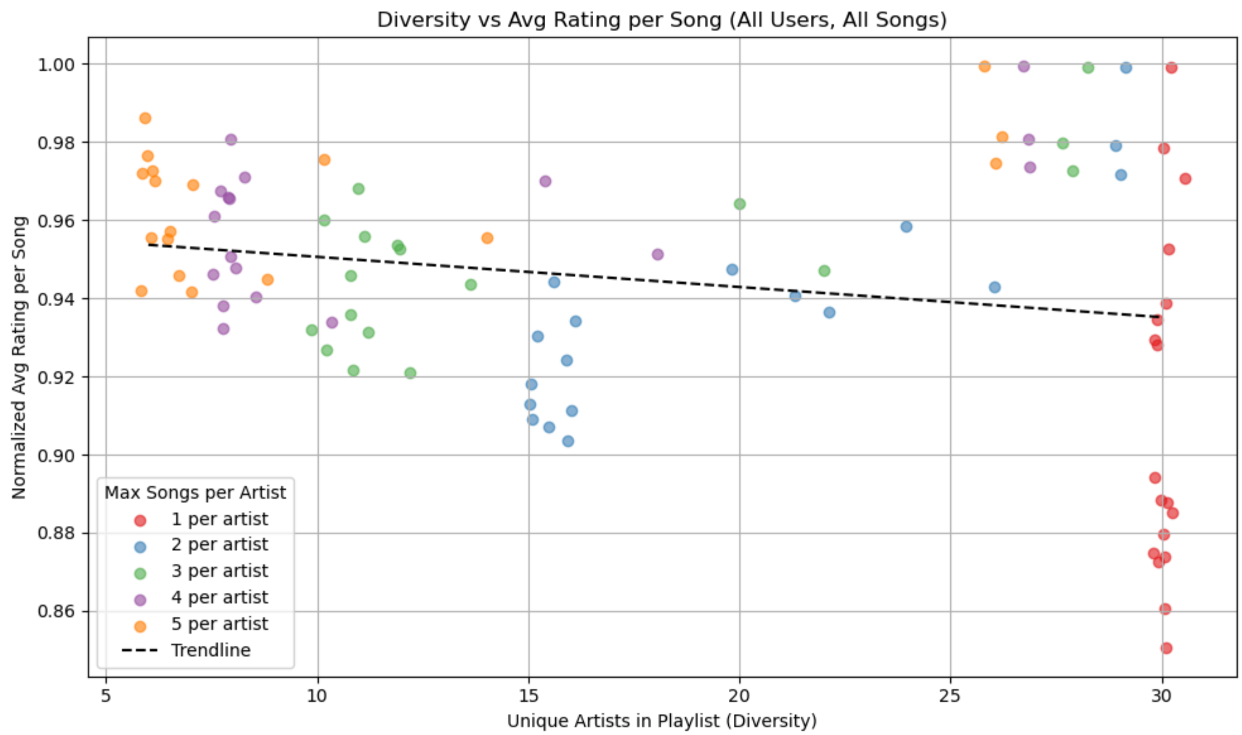

Figure 2: Sensitivity Analysis - This figure compares the average rating per song by the number of songs allowed per artist . As more songs are added from an artist, the increase in average ratings decreases.

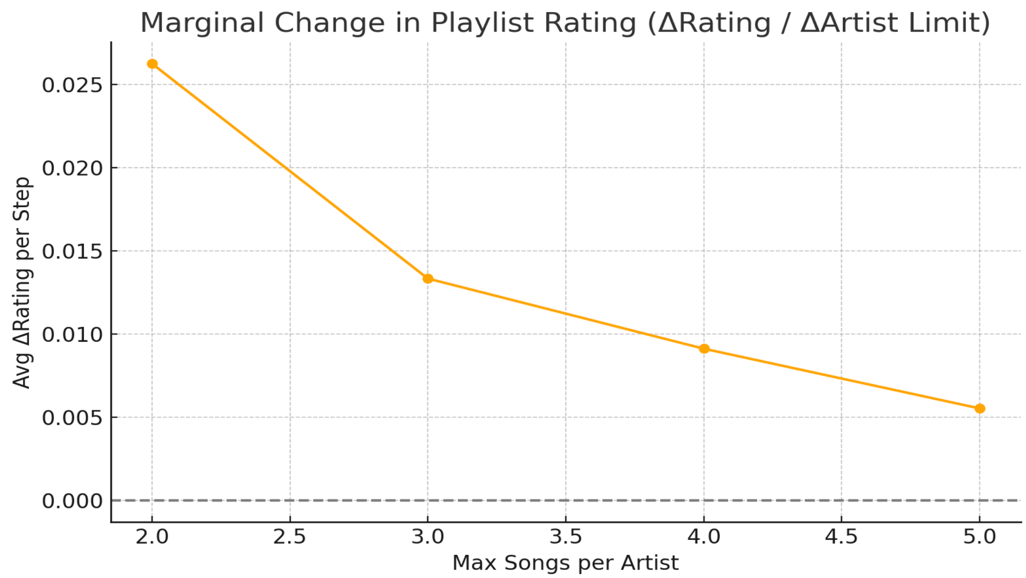
Marginal Change in Playlist Rating (ΔRating / ΔArtist Limit)
Avg ΔRating per Step
0.025
0.020
0.015
0.010
0.005
0.000
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Max Songs per Artist

Figure 3.1

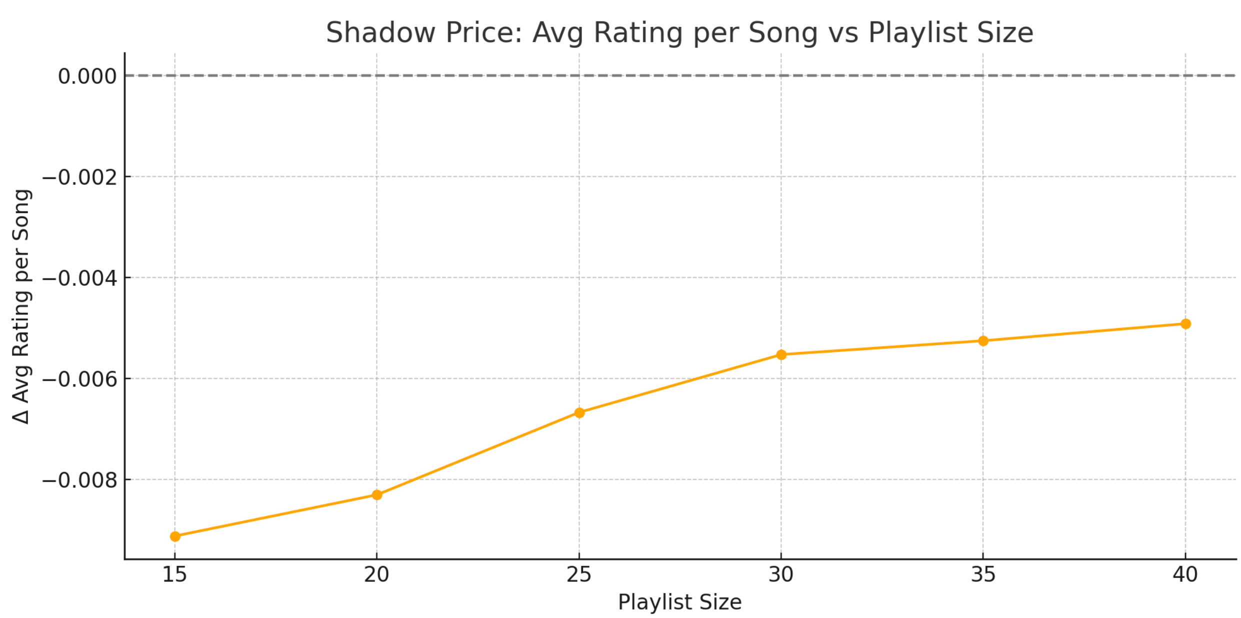
Shadow Price: Avg Rating per Song vs Playlist Size
Δ Avg Rating per Song
0.000
−0.002
−0.004
−0.006
−0.008
15
20
25
30
35
40
Playlist Size

Figure 3.2: Sensitivity Analysis - These graphs (Figure 3.1 & 3.2) compare the relationship between playlist size to the change in average expected rating as well as the change in diversity (unique artists) in a playlist. We found that 20 was a strong trade off point, especially compared to the initial 30.

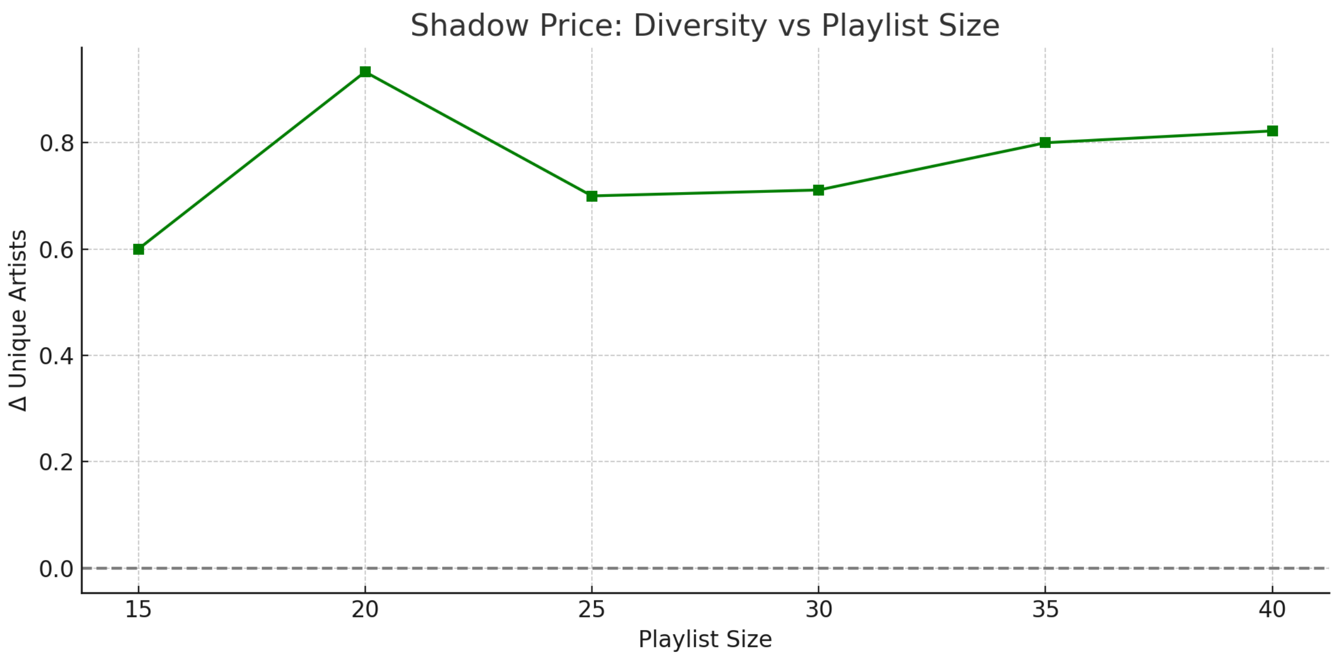

Figure 4: This graph compares the energy flow of the party playlist between the base and advanced models. The advanced model creates a smoother progression of energy levels, avoiding the abrupt spikes and drops seen in the base model, resulting in a more cohesive listening experience.

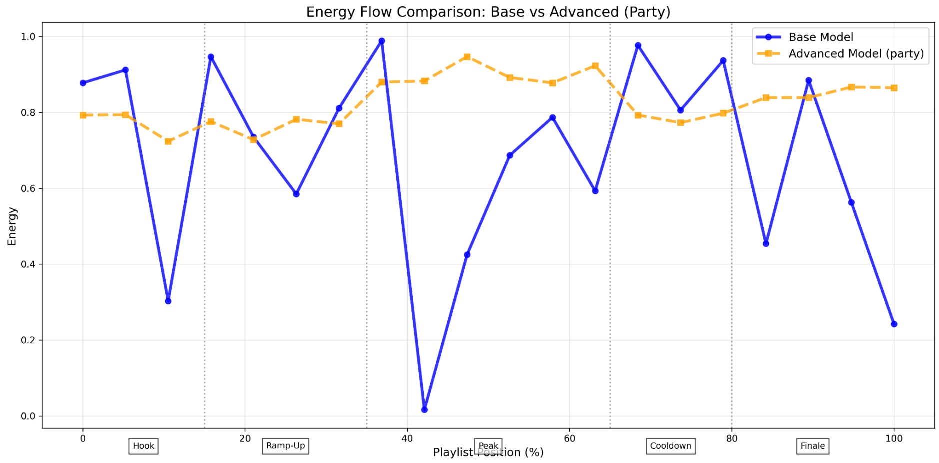

Figure 5: The advanced exercise playlist exhibits a steady tempo curve, unlike the erratic fluctuations of the base model. This structured flow enhances the user's workout by matching tempo with different phases of physical activity.

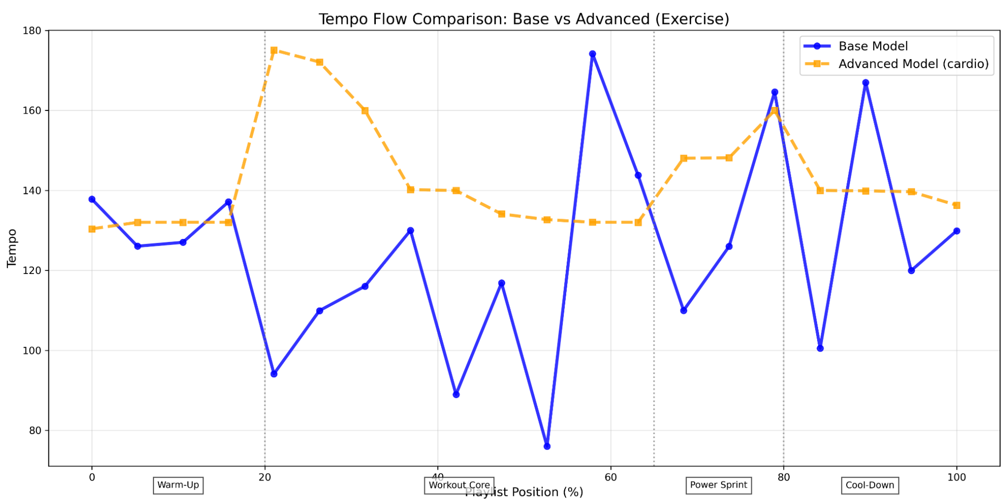

Figure 6: This figure compares the tempo trajectories in the focus playlist. The base model’s selections produce a scattered tempo profile, lacking consistency. In contrast, the advanced playlist demonstrates a thoughtfully designed tempo arc that supports sustained concentration, aligning better with tasks that require focus.

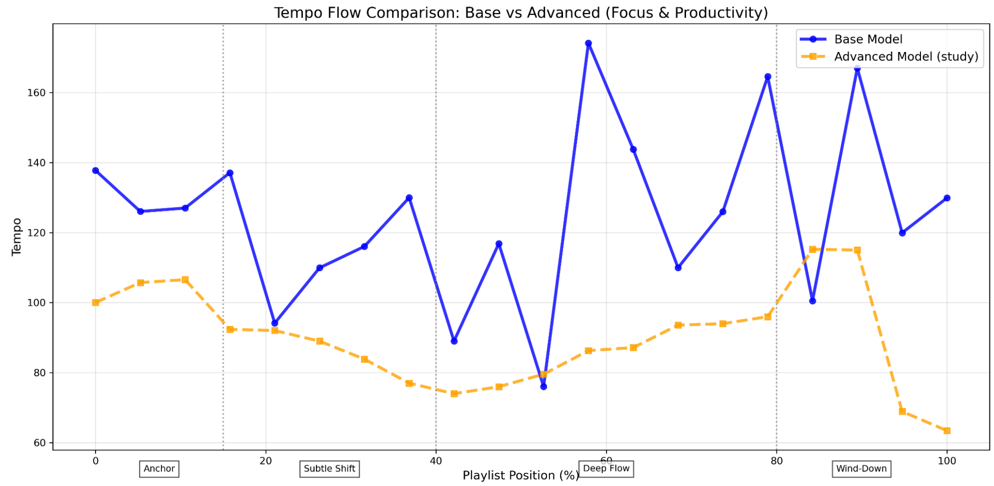

Figure 7: Here we examine the energy level distribution across the focus playlist. The base model shows unpredictable shifts in energy, potentially distracting to users. Meanwhile, the advanced model maintains a balanced energy pattern, steady enough to promote mental clarity while avoiding overstimulation during extended periods of work or study.

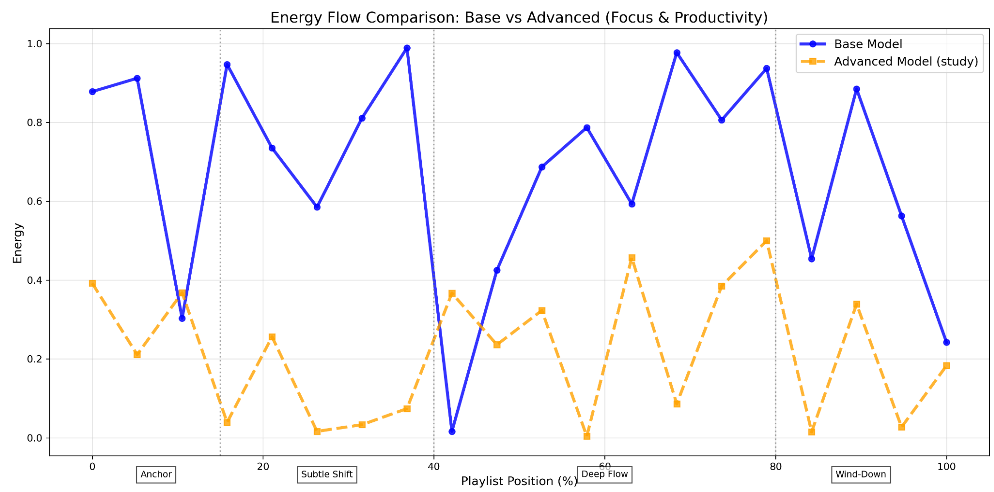

Figure 8: The emotional playlist from the advanced model maintains a steady valence (emotional tone), while the base model's playlist shows sharp variations. This stability creates a more emotionally coherent listening experience.

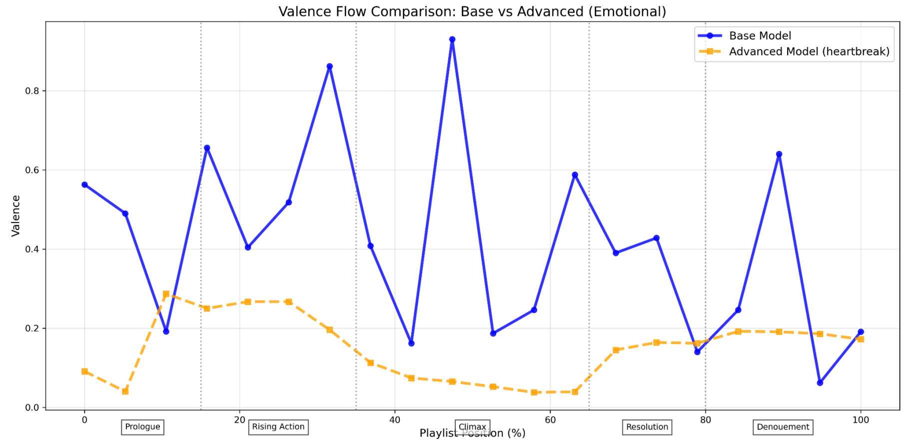

Figure 9: This figure compares tempo consistency in the relaxation playlist. The advanced model maintains a calm, even tempo throughout, whereas the base model fluctuates unpredictably, reducing its suitability for a relaxing setting.

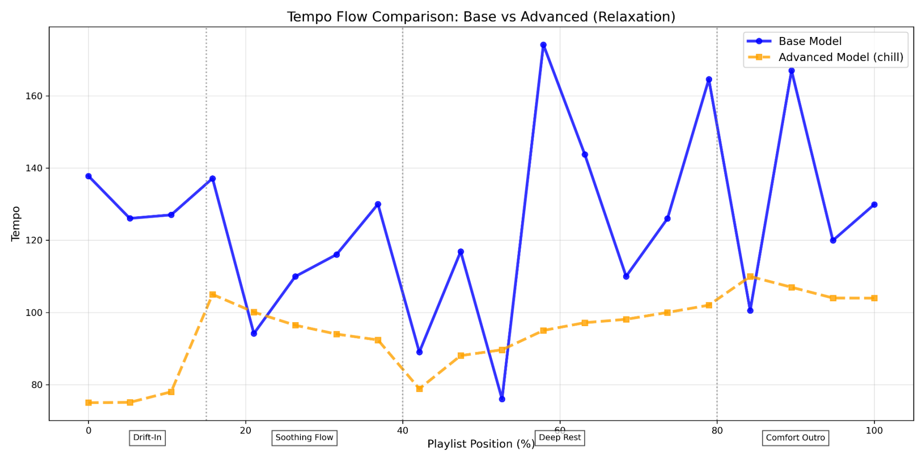

Figure 10: This figure shows the distribution of popularity per song for both the base model, advanced model, and ideally where the popularity aims to be. The base model has consistently higher popularity as it is highly correlated with user rating, however it decreased the amount of possible artist diversity. In ensuring this artist diversity, our playlist still allowed for the 90th percentile of popular songs to be in the hook, however the popularity score was relative to the playlist.

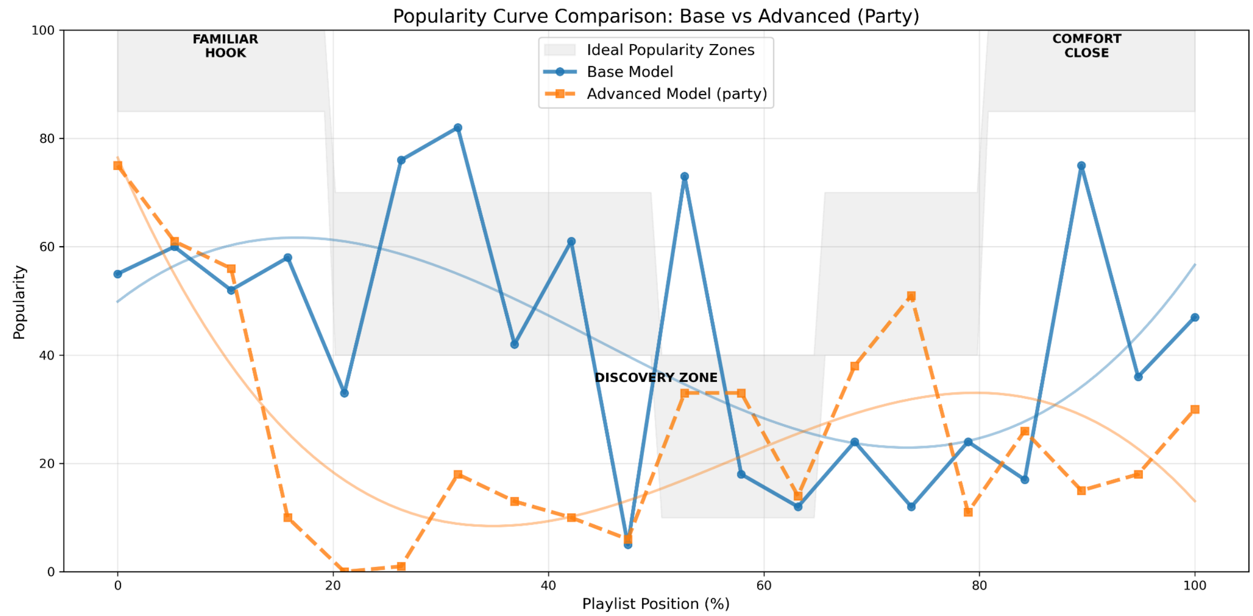

Figure 11: This graph shows the difference in the valence level between the base model and the advanced model. This backs our constraint and recommendation to keep songs above a valence of 0.7 in the ramp-up as it boosts energy and builds up listener morale.

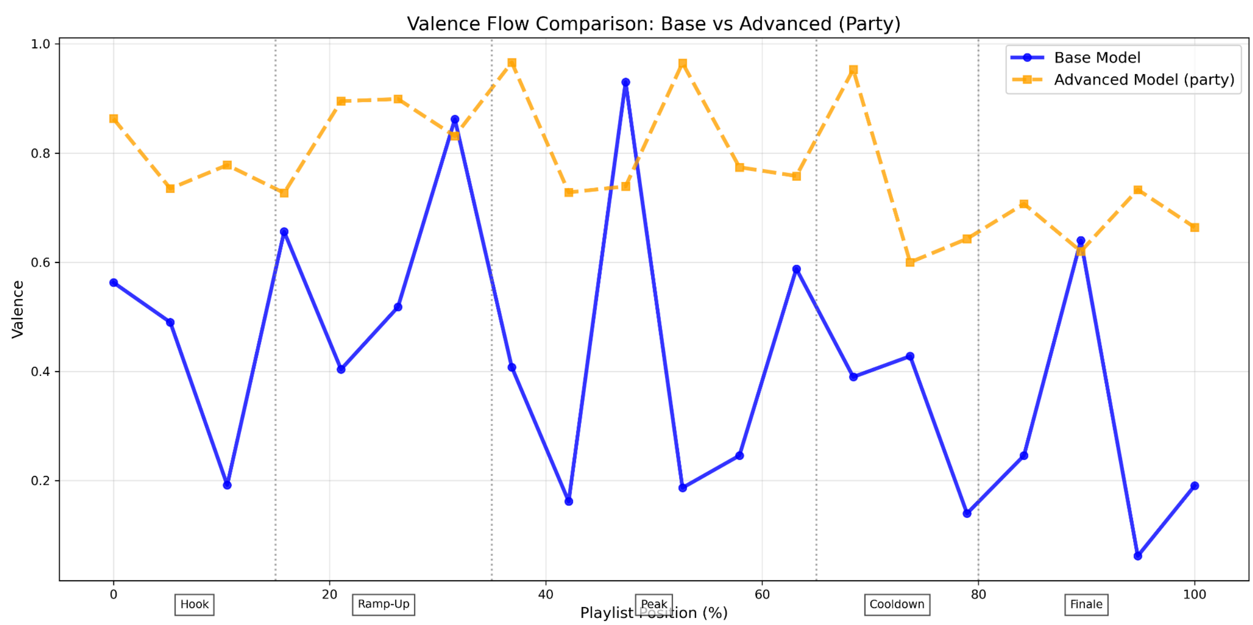

Figure 12: Shows the robustness of playlist flow (average tempo change in between songs) across different noise levels. Made with Party playlist constraints, tested with "user_b4beed9bf653604a876fdfd9df59e19c".

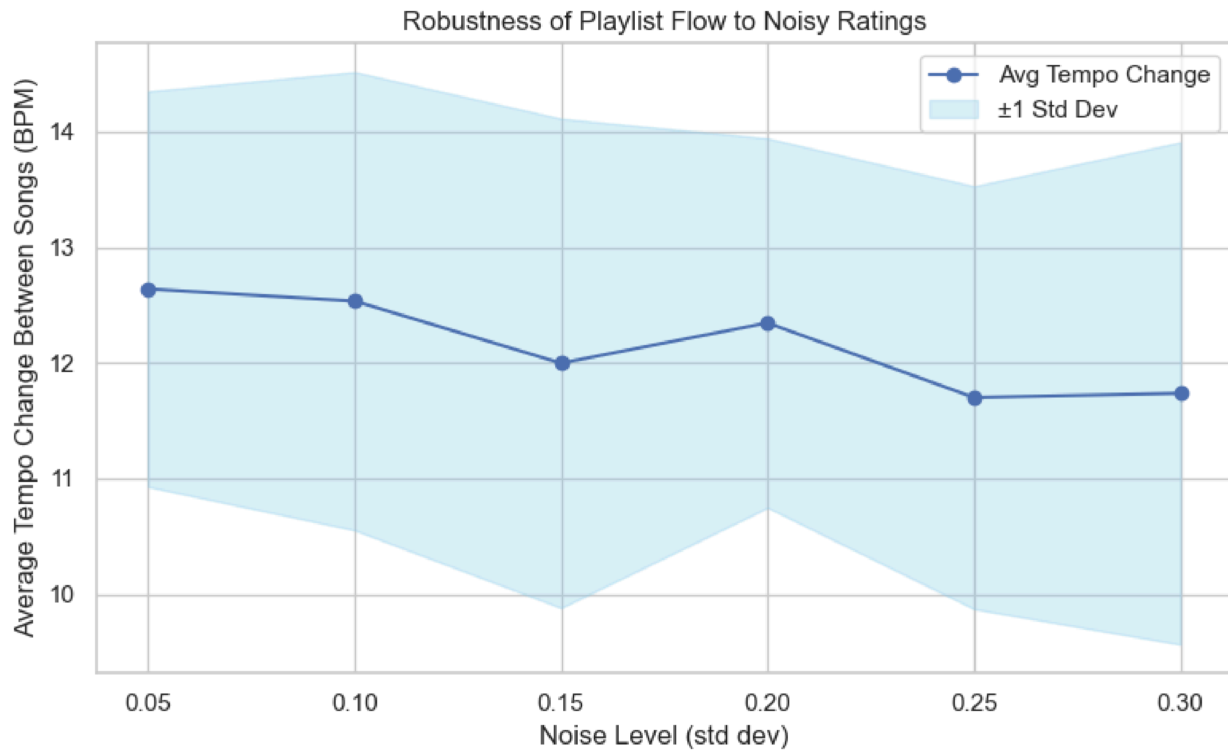

Figure 13: Robustness of average audio features on a playlist to noisy user ratings. Made with Party playlist constraints, tested with "user_b4beed9bf653604a876fdfd9df59e19c".

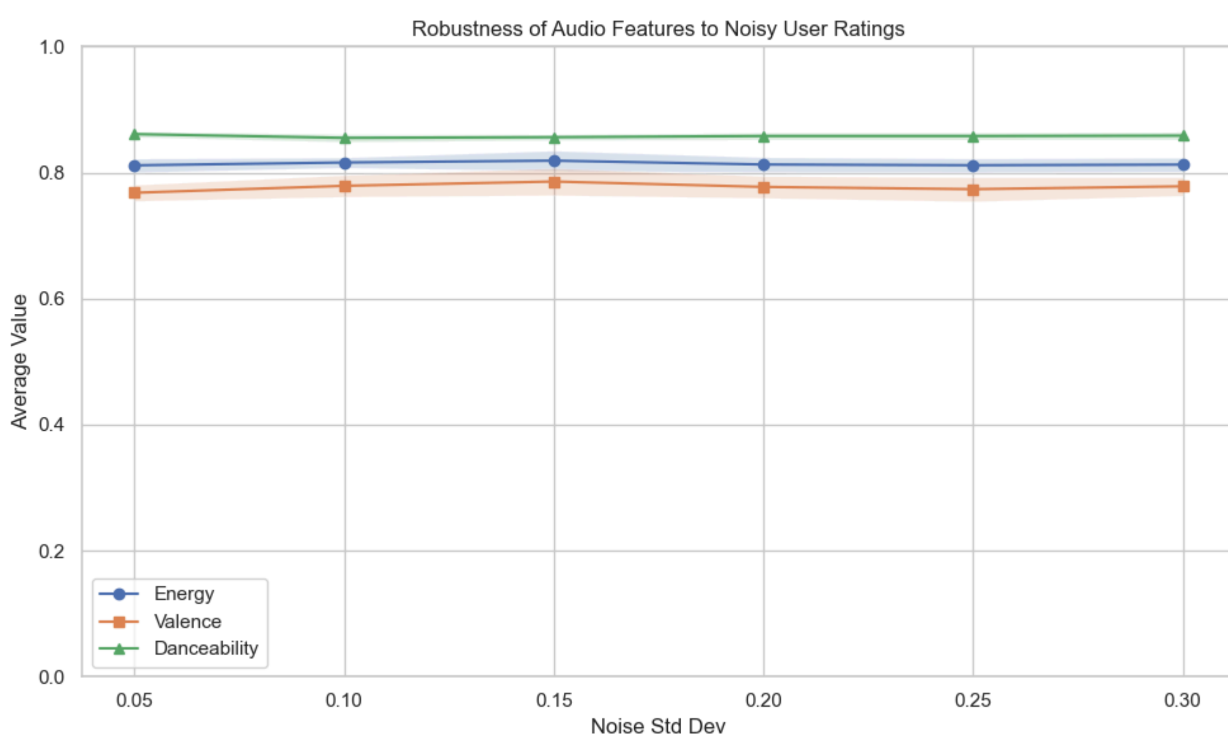

Figure 14: This figure formulates the base model with decision variables, constraints, and an objective function.

Decision Variables

Binary Variables:

- Let X_i be the binary variable such that it is 1 if the song exists in the playlist, and 0 otherwise!
- Let Y_i be the binary variable such that it is 1 if the the artist is in the playlist, and 0 otherwise!

Additional Data:

- Let there be a list of ratings "ratings"
- Let there be a list of possible track names "track_names" (i)
- Let there be a list of possible artist names "artist_to_song" (j)
- Let there be a list of the average ratings "rating"
- Let unique_artists be the set of unique artist names

Constraints

- There can only be 30 songs

$$\sum_{i=1}^{n} X_i = 30$$

- No artist can be repeated twice

$$\forall j \in \{0, 1, \ldots, |\text{unique\_artists}| - 1\} : \sum_{i \in \text{artist\_to\_songs}[j]} X_i \leq Y_j \quad \text{(link constraint)}$$

$$\forall j \in \{0, 1, \ldots, |\text{unique\_artists}| - 1\} : \sum_{i \in \text{artist\_to\_songs}[j]} X_i \leq 1 \quad \text{(uniqueness constraint)}$$

- No song can be repeated twice

$$\forall i = 1, \ldots, n : \quad X_i \leq 1$$

Objective Function

Maximize the user ratings to get the best songs!

$$\max \sum_{i=1}^{n} (\text{rating}_i \times X_i)$$

Figure 15: This figure formulates the advanced model with the new constraints regarding mood, tempo, energy, etc. We also increased the artist song allowance to 2.

Decision Variables

Binary Variables:

- Let X_i be the binary variable such that it is 1 if the song exists in the playlist, and 0 otherwise!

Additional Data:

- Let there be a list of ratings "ratings" (i)
- Let there be a list of possible track names "track_names" (i)
- Let there be a list of possible artist names "artist_names" (j)
- Let there be audio features for each song: "tempo", "energy", "valence", "acousticness", "instrumentalness", "danceability", "loudness", "speechiness", "popularity"

Constraints

- There can only be num_songs songs

$$\sum_{i=1}^{n} X_i = \text{num\_songs}$$

- No artist can be repeated more than twice

$$\forall j = 1, \ldots, m : \sum_{i \in \{i \mid \text{artist\_names}[i] = \text{artist\_names}[j]\}} X_i \leq 2$$

Mood-Specific Constraints:

Pregame

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{danceability}_i \geq 0.7$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.8$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.6$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{popularity}_i \geq 60$$

Party

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{danceability}_i \geq 0.8$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.7$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.6$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \geq 115$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{popularity}_i \geq 70$$

Karaoke

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{popularity}_i \geq 70$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \leq 0.2$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.5$$

Cardio

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \geq 130$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.8$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{danceability}_i \geq 0.7$$

Weight Training

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.8$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.6$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \geq 100$$

Yoga

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.4$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \geq 0.4$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \leq 100$$

Running

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \geq 140$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.8$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.5$$

Deep Work

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \geq 0.5$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.5$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{speechiness}_i \leq 0.1$$

Study

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.5$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \geq 0.3$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \leq 120$$

Creative Work

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \geq 0.3$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.6$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.4$$

Heartbreak

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \leq 0.3$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{acousticness}_i \geq 0.4$$
$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.5$$

Melancholy

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \leq 0.4$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.4$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{acousticness}_i \geq 0.3$$

Euphoric

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.8$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.7$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{danceability}_i \geq 0.6$$

Romantic

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.5$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.6$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{acousticness}_i \geq 0.3$$

Angry

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \geq 0.8$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \leq 0.4$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{loudness}_i \geq -7.0$$

Sleep

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.2$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{loudness}_i \leq -12.0$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \geq 0.4$$

Meditation

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.3$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{instrumentalness}_i \geq 0.5$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \leq 80$$

Chill

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.5$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.4$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{tempo}_i \leq 110$$

Sunday Morning

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.5$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.5$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{acousticness}_i \geq 0.3$$

Beach

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{valence}_i \geq 0.6$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{energy}_i \leq 0.7$$

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \text{acousticness}_i \geq 0.2$$

Flow Pattern Constraints:

Party Flow Constraints:

- Each song must be assigned to exactly one section

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \sum_{j=1}^{5} S_{i,j} = 1$$

- Section size constraints (proportional to playlist length)

$$\sum_{i=1}^{n} S_{i,1} = \text{hook\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,2} = \text{ramp\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,3} = \text{peak\_size} \ \ (\approx 0.30 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,4} = \text{cooldown\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,5} = \text{finale\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

- Section content constraints
  - Hook: High popularity bangers

$$\forall i = 1, \ldots, n : S_{i,1} = 1 \Rightarrow \text{popularity\_percentile}_i \geq 90$$

- Ramp-Up: Energy build with high valence

$$\forall i = 1, \ldots, n : S_{i,2} = 1 \Rightarrow \text{valence}_i \geq 0.7$$

- Peak: Mid-popularity discovery tracks

$$\forall i = 1, \ldots, n : S_{i,3} = 1 \Rightarrow \text{popularity\_percentile}_i \in [40, 70]$$

- Finale: High popularity comfort tracks

$$\forall i = 1, \ldots, n : S_{i,5} = 1 \Rightarrow \text{popularity\_percentile}_i \geq 85$$

Exercise Flow Constraints:

- Each song must be assigned to exactly one section

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \sum_{j=1}^{4} S_{i,j} = 1$$

- Section size constraints

$$\sum_{i=1}^{n} S_{i,1} = \text{warmup\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,2} = \text{core\_size} \ \ (\approx 0.45 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,3} = \text{sprint\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,4} = \text{cooldown\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

- Section content constraints
  - Warm-Up: Progressive tempo (100-110 BPM)

$$\forall i = 1, \ldots, n : S_{i,1} = 1 \Rightarrow \text{tempo}_i \in [100, 110]$$

- Workout Core: High-BPM power songs

$$\forall i = 1, \ldots, n : S_{i,2} = 1 \Rightarrow \text{tempo}_i \geq 130$$

- Power Sprint: Ultra-high BPM tracks for final push

$$\forall i = 1, \ldots, n : S_{i,3} = 1 \Rightarrow \text{tempo}_i \geq 140$$

Focus Flow Constraints:

- Each song must be assigned to exactly one section

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \sum_{j=1}^{4} S_{i,j} = 1$$

- Section size constraints

$$\sum_{i=1}^{n} S_{i,1} = \text{anchor\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,2} = \text{shift\_size} \ \ (\approx 0.25 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,3} = \text{flow\_size} \ \ (\approx 0.40 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,4} = \text{winddown\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

- Section content constraints
    - Anchor: Consistent groove (90-110 BPM)

$$\forall i = 1, \ldots, n : S_{i,1} = 1 \Rightarrow \text{tempo}_i \in [90, 110] \wedge \text{instrumentalness}_i \geq 0.3$$

- Subtle Shift: Textural novelty with mid-high popularity

$$\forall i = 1, \ldots, n : S_{i,2} = 1 \Rightarrow \text{popularity\_percentile}_i \in [50, 80]$$

- Wind-Down: Familiar return with top focus tracks

$$\forall i = 1, \ldots, n : S_{i,4} = 1 \Rightarrow \text{popularity\_percentile}_i \geq 80$$

Emotional Flow Constraints:

- Each song must be assigned to exactly one section

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \sum_{j=1}^{5} S_{i,j} = 1$$

- Section size constraints

$$\sum_{i=1}^{n} S_{i,1} = \text{prologue\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,2} = \text{rising\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,3} = \text{climax\_size} \ \ (\approx 0.30 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,4} = \text{resolution\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,5} = \text{denouement\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

- Section content constraints
    - Prologue: Safe neutral (mid-valence)

$$\forall i = 1, \ldots, n : S_{i,1} = 1 \Rightarrow \text{valence}_i \in [0.4, 0.6]$$

- Climax: Peak emotion (low valence, high energy)

$$\forall i = 1, \ldots, n : S_{i,3} = 1 \Rightarrow \text{valence}_i < 0.4 \wedge \text{energy}_i > 0.6$$

- Resolution: Gentle lift (increasing valence, acoustic tracks)

$$\forall i = 1, \ldots, n : S_{i,4} = 1 \Rightarrow \text{acousticness}_i > 0.3$$

- Denouement: Comfort close (familiar, hopeful)

$$\forall i = 1, \ldots, n : S_{i,5} = 1 \Rightarrow \text{valence}_i > 0.5 \wedge \text{popularity\_percentile}_i \geq 80$$

Relaxation Flow Constraints:

- Each song must be assigned to exactly one section

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \sum_{j=1}^{4} S_{i,j} = 1$$

- Section size constraints

$$\sum_{i=1}^{n} S_{i,1} = \text{drift\_size} \ \ (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,2} = \text{soothing\_size} \ \ (\approx 0.25 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,3} = \text{deep\_size} \ \ (\approx 0.40 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,4} = \text{comfort\_size} \ \ (\approx 0.20 \cdot \text{num\_songs})$$

- Section content constraints
  - Drift-In: Slow fade (60-80 BPM, low dynamics)

$$\forall i = 1, \ldots, n : S_{i,1} = 1 \Rightarrow \text{tempo}_i \in [60, 80] \wedge \text{energy}_i < 0.5$$

- Soothing Flow: Textural novelties, relaxing

$$\forall i = 1, \ldots, n : S_{i,2} = 1 \Rightarrow \text{energy}_i < 0.6$$

- Deep Rest: Uniform warmth (constant energy/tempo)

$$\forall i = 1, \ldots, n : S_{i,3} = 1 \Rightarrow \text{energy}_i < 0.5$$

- Comfort Outro: Safe return to familiar relaxation tracks

$$\forall i = 1, \ldots, n : S_{i,4} = 1 \Rightarrow \text{popularity\_percentile}_i \geq 80$$

- Section content constraints
  - Hook: High popularity bangers

$$\forall i = 1, \ldots, n : S_{i,1} = 1 \Rightarrow \text{popularity\_percentile}_i \geq 90$$

- Ramp-Up: Energy build with high valence

$$\forall i = 1, \ldots, n : S_{i,2} = 1 \Rightarrow \text{valence}_i \geq 0.7$$

- Peak: Mid-popularity discovery tracks

$$\forall i = 1, \ldots, n : S_{i,3} = 1 \Rightarrow \text{popularity\_percentile}_i \in [40, 70]$$

- Finale: High popularity comfort tracks

$$\forall i = 1, \ldots, n : S_{i,5} = 1 \Rightarrow \text{popularity\_percentile}_i \geq 85$$

Exercise Flow Constraints:

- Each song must be assigned to exactly one section

$$\forall i = 1, \ldots, n : X_i = 1 \Rightarrow \sum_{j=1}^{4} S_{i,j} = 1$$

- Section size constraints

$$\sum_{i=1}^{n} S_{i,1} = \text{warmup\_size} \;\; (\approx 0.20 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,2} = \text{core\_size} \;\; (\approx 0.45 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,3} = \text{sprint\_size} \;\; (\approx 0.15 \cdot \text{num\_songs})$$

$$\sum_{i=1}^{n} S_{i,4} = \text{cooldown\_size} \;\; (\approx 0.20 \cdot \text{num\_songs})$$