Team 1 'The Tune Titans♫' - Final Report

Kashyap Dadhaniya, Karlee Harris, Salil Kamat, Rajinder Kaur, Rishabh Surana

Introduction (motivation):

The motivation behind our project is to recommend music that is tailored to the diverse music tastes of users and integration of new music. Our team is developing a music recommendation system which generates 5 ideal song recommendations for a user supplied playlist. We used the Spotify Million Playlist Dataset (https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge) as our foundational data source paired with data extrapolation from the Spotify API (https://developer.spotify.com/documentation/web-api), aiming to enhance user experience by personalizing music recommendations based on their existing playlist preferences. By leveraging our unique approach which requires zero cost, there's an opportunity to revolutionize how music is recommended, tapping into an industry worth $31.2 billion. With nearly 500 million users on Spotify, there's a growing demand for seeking new ways to engage with music.

Problem Definition:

Currently, the landscape of music recommendation systems (MRS) is dominated by two methods: Collaborative Filtering and Content-Based Filtering. Despite their widespread application across leading platforms like Spotify, they exhibit limitations in capturing the multifaceted music preferences of users and often fail to integrate lesser known/new music, we aim to address these concerns.

Literature Survey:

To address the general question of how to recommend music we researched music preference impacts. **Schäfer et al. (2010)** found that amongst many music internal and external factors that the cognitive functions of music and the physiological arousal excited by music had the largest impact on musical preference. **Schäfer et al. (2013)** similarly concurred, finding that the psychological effects of music impacted preference, stating that people often turned to music for companionship and mood enhancement. Furthermore, **Liu et al. (2023)** found that many students used music to regulate mood. These papers all had similar shortcomings of lacking data diversity and size because they used surveys/observational data but with their shared agreement their findings are strengthened. We have combatted this flaw by using the very diverse and large Spotify Million Playlist Dataset. With these findings, **Song et al. (2012)** created a Music Recommendation System (MRS) aiming for heightened personalization and emotional resonance to enhance user satisfaction. Accurately detecting user emotion is significantly difficult, but we hypothesize that many users already group playlists roughly by how music makes them feel, thus we will recommend by playlist rather than by user to overcome the confounding variable of varying situations dictating user music preference. **Donghong Han et al (2022)** offered a comprehensive review of music emotion recognition (MER) which may be useful in classifying playlists with elicited emotion as a consideration.

Next, we researched existing algorithms applied to the problem set of making recommendations. **Gulzar et al. (2023)** introduced an Ordered Clustering-Based Algorithm (OCA) aimed at enhancing e-commerce recommendation systems through clustering methods. This algorithm is an option for our MRS, but we may face challenges with respect to scalability and differences between music preferencing and e-commerce behaviors. Specific to MRS, **Verma et al. (2021)** and **Schedl (2019)** discussed collaborative and/or content-based filtering. Collaborative filtering in the scope of MRS would be finding playlists that have shared attributes (i.e. common artists) and assuming the playlists would 'like' the songs found in the others' playlist. Whereas content-based filtering would look at all the songs in a playlist, find their commonality, and recommend songs similar to that determined commonality. **Luke Barrington et al. (2009)** examined the effectiveness of the content-based filtering Genius MRS using features like artist and acoustic content. This approach will prove in the context of our project as we are considering utilizing the Spotify API to create a database of song metrics such as tone and danceability to utilize in content-based filtering. **Bobadilla et al. (2023)** approached content-based filtering algorithms by using matrix factorization (MF) techniques, highlighting the strengths of approaches like Biased (BiasedMF) and Bayesian Non-negative (BNMF). Additionally, **Wu et al. (2023)** noted that Large Language Models

(LLMs) can be applied in the context of content-filtering. However, applying LLM to analyze lyrics would dramatically increase the computational complexity of our algorithm which may not be practical to implement in our project. There are also AI-based, deep learning algorithms like CNN, RNN, and LSTM which **Anand et al. (2021)** and **Van den Oord et al.(2013)** applied to the music recommendation problem set. Although effective, these models were very black-box and lacked considerations like user context. Lastly, **McFee et al. (2012)** developed a MRS which used a similarity analysis method called metric learning to rank (MLR) to return an ordered list of recommendations. The end goal of our MRS is to provide a ranked list of recommendations, thus MLR will be a very useful approach. One common flaw found in most of the previously discussed MRS is that non-influential or new artists perpetually are not recommended due to popularity bias and the cold start problem. **Christina Bauer et al (2017)** explains these issues at the artist level highlighting the importance of exposure and audience development. These existing flaws are an important consideration for our project as we do not want to unintentionally limit visibility of certain music. We seek to address the cold start issue by using the rank of songs opposed to weights (by frequency) within our model to lessen the overshadowing effect. The user provided playlists will be added to our database to continue to diversify our recommendations and integrate new music.

<u>Proposed Method:</u>

Playlist Clustering: We began with a dataset of 2.2 million unique song records and conducted data pre-processing and an initial exploratory data analysis (EDA). Subsequently, we implemented a content-based filtering approach by clustering songs and playlists in two sequential steps. In the first step, we clustered songs based on their content features, such as valence, acousticness, danceability, and energy, using the K-means algorithm. These song clusters provided insights into the underlying structure of the song data. Leveraging the song clusters obtained from the first step, we employed a two-tiered clustering technique. Specifically, we used the K-means algorithm again to cluster playlists, this time using the song clusters as input features. This sequential approach ensured that the clustering of playlists was influenced by the characteristics and similarities identified within the song clusters. By adopting novel two-tiered clustering, we created a hierarchical clustering process that optimized the clustering of playlists based on the content features of songs.[1]

Candidate Song Generation: After training our playlist classification model, we used the classified playlists to narrow our song recommendation candidates. Using our trained playlist classification model we classify the user provided playlist using the playlist's songs' features. If we do not already have the songs' features in our database we will conduct a Spotify API request. We will also add user provided playlists to our database (not tied to the user) to increase our pool of candidate songs to be used for recommendations in the future. By continuing to build our database we will allow for new music to be introduced.[1] We then compile a list of songs from playlists of the same classification to form the candidate song pool for recommendations and therefore which songs we conduct similarity analysis on (opposed to conducting similarity analysis on every song in existence).

Metric Learning to Rank (MLR): Applying metric learning to rank (MLR) to the scope of ranked music recommendations, the input is a playlist, and the output is a ranked order of song recommendations from a list of candidate songs. Playlists with ranked songs by preference are required to train the model. We generated a ranked order of songs for each training playlist by selecting diverse songs from within the playlist that promote song variation while still putting an emphasis on most favored song types. To determine our ranked songs for each playlist, we opted to perform k-means clustering with 10 clusters on each playlist. Within each cluster, we chose the song which had the smallest euclidean distance from the median of the features to be the 'cluster representing song'. We sorted the 'cluster representing songs' by the number of songs in their cluster ascending and used their index as their rank. For instance, the 'cluster representing song' belonging to the cluster with the most songs has a rank of 10. The larger the rank, the more weight is given to a song when conducting similarity analysis and ultimately making recommendations. This novel approach allows for the playlists' most common group of songs (largest

[1] Innovation/Novelty of our MRS

cluster size) to have the most influence without drowning out the influence of less common groups of songs (small clusters).[2] With 10 song rankings for each playlist, we trained a metric learning to rank model which will be applied to a user provided playlist to make recommendations. Our MLR ranks all candidate songs (from the candidate song generation) and offers the top 5 to the user, but allows recommendations to be replaced with the next ranked song if the user does not like a recommendation or would like to view more recommendations.

Graphical User Interface: The visualization is a Python Django based application that allows a user to sign in with their Spotify account. The dashboard (first page the user sees once logged in) contains data about the user's top 10 songs that can be filtered by a variety of time frames, including short term, medium term, and long term. Users are able to navigate to the 'Playlists' tab, where all of the user's playlists are shown in a table format, which can be clicked to open a new page that shows all of the songs present in that playlist. The tracks page includes a button that a user can click, which will start the recommendation process based on the songs currently in the playlist. The innovation with this approach is that Spotify itself and other websites do not give the users the information that is presented on our web application. This includes the ability to see the top 10 songs across different time frames, view their playlists and the songs within them, get recommendations using novel and innovative models, preview recommendation audio, directly add songs to their Spotify playlists, replace songs immediately if disliked or if more recommendations are desired, visualize feature data that influenced their unique recommendations, and leave survey feedback for developers.[2]

Experiments/Evaluation:

We first completed Exploratory Data Analysis (EDA) on the dataset which primarily included removing duplicate records of playlists and songs. Next, we created various plots that best summarized the dataset visually. Figure 1 is a histogram which provides a visual representation of the distribution of the various song features, such as danceability, energy, loudness, etc. Figure 2 is a correlation heatmap which shows the associations/correlations between the various song features; it appears that Energy and Loudness of a playlist have the highest correlation (0.8) amongst all the features.

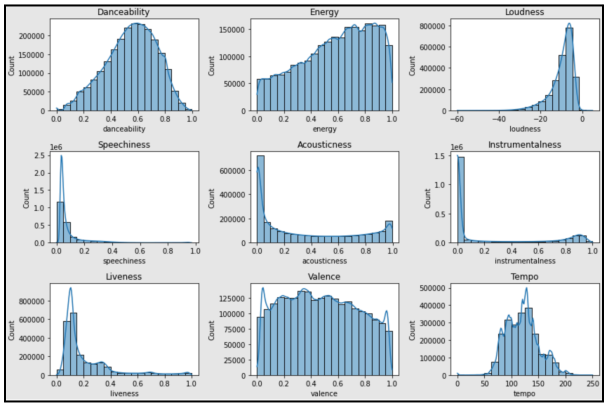


Figure 1: Song Feature Histograms

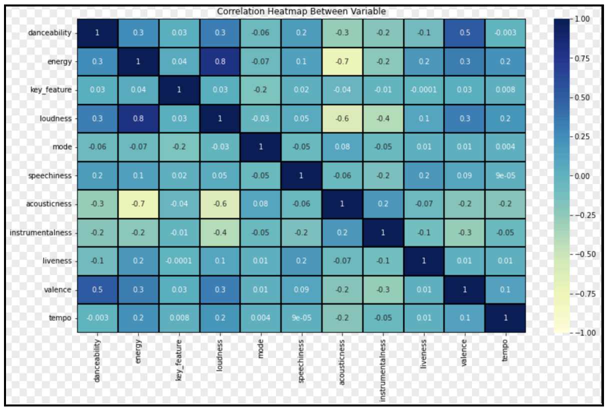


Figure 2: Song Feature Correlation Heatmap

After data pre-processing and EDA, we conducted the 2-way clustering of the playlists. After conducting iterative experiments we determined the optimal number of song clusters and playlist clusters were 100 and 35 respectively. Figure 3 displays a two-dimensional visualization of the playlist clusters.

[2] Innovation/Novelty of our MRS

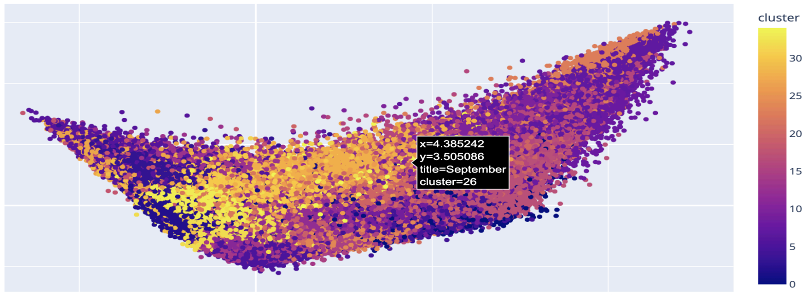


Figure 3: Playlist Cluster 2-Dimensional Visualization

Our first try at forming the training data for the Metric Learning to Rank model (which required performing k-means clustering roughly 1,000,000 times) took nearly seven days to process. We learned a lot about parallelization and were able to use the library pandarallel to distribute the workload across cores. We then faced a RAM constraint and switched to a desktop computer with greater RAM and decreased our training time to just under 10 hours, a 17 times improvement. We left out 20% of our training data for testing and found that our model had an average Normalized Discounted Cumulative Gain (NDCG), a ranking quality metric, of 0.673, with the lowest value being 0.546 and the highest value being 0.768 (1 being perfectly scored/ranked recommendations). NDCG values ranging from 0.5-0.75 are considered very good, and anything above 0.75 is considered excellent (Recommendations, 2024). Figures 5 and 6 below depict the importance of each feature within the final model. Notice ‘speechiness’ had the largest mean Shapley Additive Explanations (SHAP) value which is a metric used to determine feature importance of machine learning models. Spotify defines the ‘speechiness’ of a song as the presence of spoken words in a song. It makes sense that ‘speechiness’ would have a large impact on the model output because intuitively users may separate instrumental-only music and music with lyrics into distinct playlists. ‘Liveness’ was also expected to be an important feature in the model, given our hypothesis of playlists being organized by use-case (environmental situation), for example a user generally would want more lively music at the gym while working out rather than while trying to focus/study. Interestingly, the least important feature was found to be ‘valence’ which refers to how positive/happy a song is, which means playlists most likely contain a diverse range of positivity within their music. This Spotify API link provides a description of each of the features: https://developer.spotify.com/documentation/web-api/reference/get-audio-features.

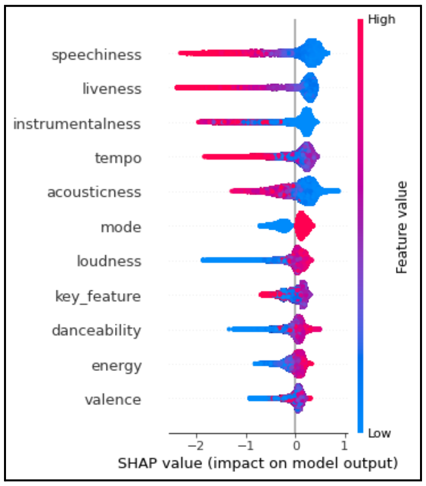


Figure 5

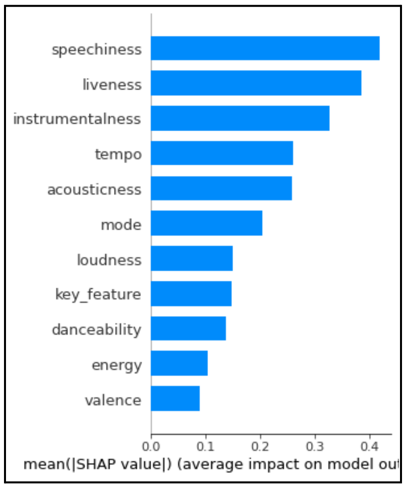
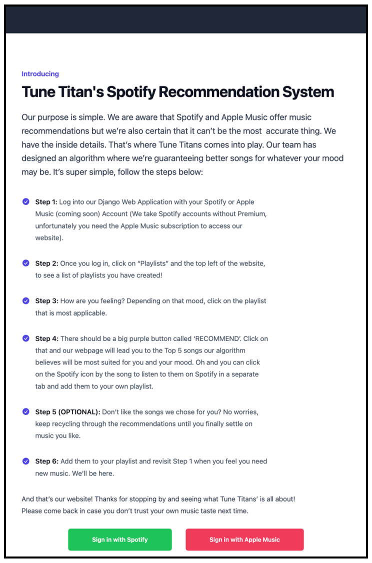


Figure 6

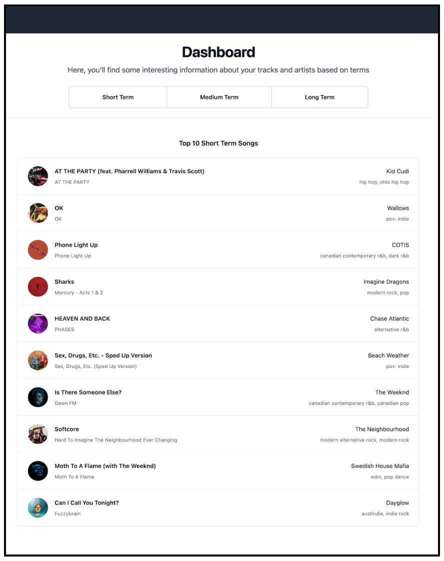

Introducing

Tune Titan's Spotify Recommendation System

Our purpose is simple. We are aware that Spotify and Apple Music offer music recommendations but we're also certain that it can't be the most accurate thing. We have the inside details. That's where Tune Titans comes into play. Our team has designed an algorithm where we're guaranteeing better songs for whatever your mood may be. It's super simple, follow the steps below:

- **Step 1:** Log into our Django Web Application with your Spotify or Apple Music (coming soon) Account (We take Spotify accounts without Premium, unfortunately you need the Apple Music subscription to access our website).
- **Step 2:** Once you log in, click on "Playlists" and the top left of the website, to see a list of playlists you have created!
- **Step 3:** How are you feeling? Depending on that mood, click on the playlist that is most applicable.
- **Step 4:** There should be a big purple button called 'RECOMMEND'. Click on that and our webpage will lead you to the Top 5 songs our algorithm believes will be most suited for you and your mood. Oh and you can click on the Spotify icon by the song to listen to them on Spotify in a separate tab and add them to your own playlist.
- **Step 5 (OPTIONAL):** Don't like the songs we chose for you? No worries, keep recycling through the recommendations until you finally settle on music you like.
- **Step 6:** Add them to your playlist and revisit Step 1 when you feel you need new music. We'll be here.

And that's our website! Thanks for stopping by and seeing what Tune Titans' is all about! Please come back in case you don't trust your own music taste next time.

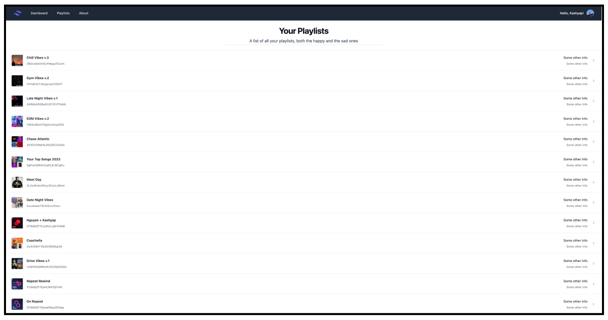

Dashboard

Here, you'll find some interesting information about your tracks and artists based on terms

Top 10 Short Term Songs

AT THE PARTY (feat. Pharrell Williams & Travis Scott) AT THE PARTY	Kid Cudi hip hop, ohio hip hop
OK OK	Wallows pov: indie
Phone Light Up Phone Light Up	COTIS canadian contemporary r&b, dark r&b
Sharks Mercury - Acts 1 & 2	Imagine Dragons modern rock, pop
HEAVEN AND BACK PHASES	Chase Atlantic alternative r&b
Sex, Drugs, Etc. - Sped Up Version Sex, Drugs, Etc. (Sped Up Version)	Beach Weather pov: indie
Is There Someone Else? Dawn FM	The Weeknd canadian contemporary r&b, canadian pop
Softcore Hard To Imagine The Neighbourhood Ever Changing	The Neighbourhood modern alternative rock, modern rock
Moth To A Flame (with The Weeknd) Moth To A Flame	Swedish House Mafia edm, pop dance
Can I Call You Tonight? Fuzzybrain	Dayglow austindie, indie rock

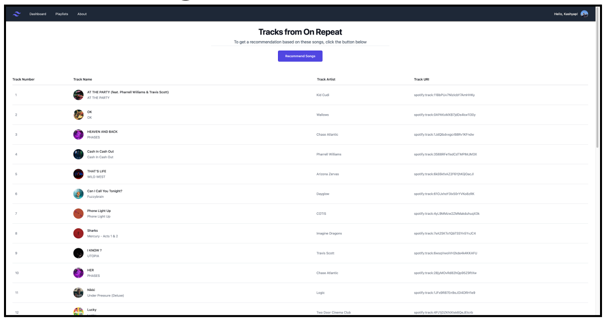
Figure 7 Figure 8



Your Playlists

A list of all your playlists, both the happy and the sad ones

Chill Vibes v.3 2BGnAlbAAVEyPMqqxTGulm	Some other info Some other info
Gym Vibes v.2 7oYwRJCYvAcgxvao1CDtrP	Some other info Some other info
Late Night Vibes v.1 4XRbhAO5BwDUS11PzTPobN	Some other info Some other info
EDM Vibes v.2 1XK5uNbUtYSg0zm4zpOfSL	Some other info Some other info
Chase Atlantic 3X1DYj1WeHbJN32RCG3I3w	Some other info Some other info
Your Top Songs 2023 0gPrehQWXvCpXL5LRCqjEu	Some other info Some other info
Meet Day 3L2wWvbU8Vyy3CsuLcBtmt	Some other info Some other info
Date Night Vibes 0uuoksalcTEr2tiSJu3mvu	Some other info Some other info
Nguyen + Kashyap 37i9dQZF1EJsWwLq9HV94B	Some other info Some other info
Coachella 2w3rD9lIYYBJDr6RjGbp29	Some other info Some other info
Drive Vibes v.1 1zGKWhQ8RohKZSZ5Ql2GOA	Some other info Some other info
Repeat Rewind 37i9dQZF1EpHUR41lQ7rMl	Some other info Some other info
On Repeat 37i9dQZF1EpwaGKyp2D4pp	Some other info Some other info

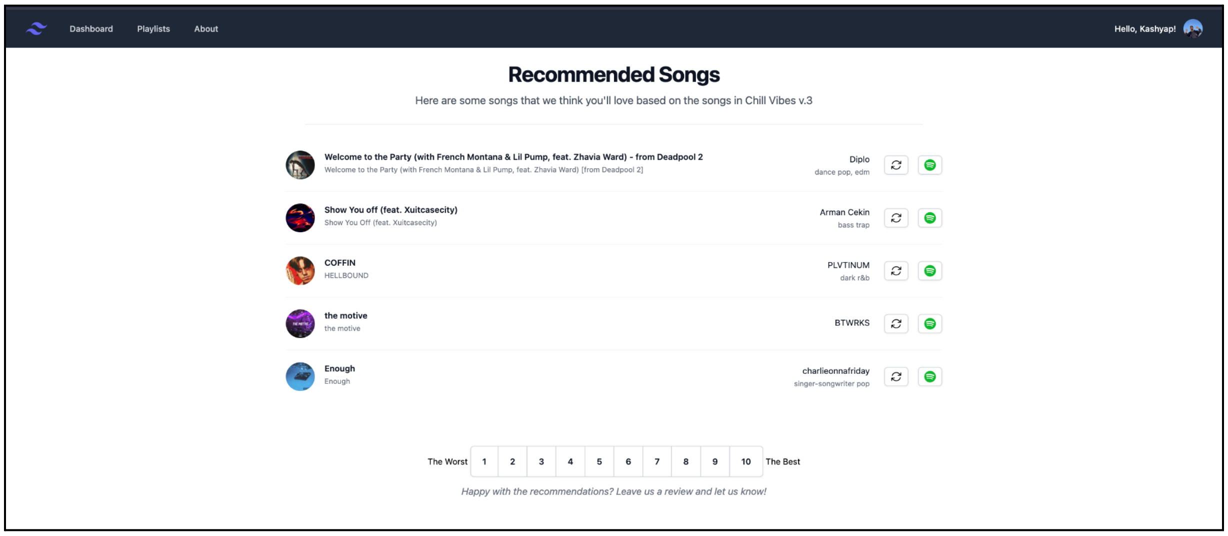


Tracks from On Repeat

To get a recommendation based on these songs, click the button below

Track Number	Track Name	Track Artist	Track URI
1	AT THE PARTY (feat. Pharrell Williams & Travis Scott) AT THE PARTY	Kid Cudi	spotify:track:11BbPUv7NlzIcbY7AmHHKy
2	OK OK	Wallows	spotify:track:0APAKxMXB7jdDs4kw1l30y
3	HEAVEN AND BACK PHASES	Chase Atlantic	spotify:track:1JdQIbdvxgcrB8Rv1KFndw
4	Cash In Cash Out Cash In Cash Out	Pharrell Williams	spotify:track:3S68RFe1lsdCdTMPIMJM3X
5	THAT'S LIFE WILD WEST	Arizona Zervas	spotify:track:6kE6kfxAZ2F6YjhKQOacJI
6	Can I Call You Tonight? Fuzzybrain	Dayglow	spotify:track:61OJxhoY3lx50rYVKo8zRK
7	Phone Light Up Phone Light Up	COTIS	spotify:track:4yL9MMzw2ZMMakduhuqX3k
8	Sharks Mercury - Acts 1 & 2	Imagine Dragons	spotify:track:7sA2SKTo1QbTSSYn5YvJC4
9	I KNOW ? UTOPIA	Travis Scott	spotify:track:6wsqVwoiVH2kde4k4KKAFU
10	HER PHASES	Chase Atlantic	spotify:track:2BjyMOvRd82hQp95Z9ftXw
11	Nikki Under Pressure (Deluxe)	Logic	spotify:track:1JFx9R87En9oJOI4DRH1e9
12	Lucky	Two Door Cinema Club	spotify:track:4PJ1jDZKNXtsk6QeJEtcrb

Figure 9 Figure 10



Recommended Songs

Here are some songs that we think you'll love based on the songs in Chill Vibes v.3

Welcome to the Party (with French Montana & Lil Pump, feat. Zhavia Ward) - from Deadpool 2 Welcome to the Party (with French Montana & Lil Pump, feat. Zhavia Ward) [from Deadpool 2]	Diplo dance pop, edm
Show You off (feat. Xuitcasecity) Show You Off (feat. Xuitcasecity)	Arman Cekin bass trap
COFFIN HELLBOUND	PLVTINUM dark r&b
the motive the motive	BTWRKS
Enough Enough	charlieonnafriday singer-songwriter pop

Happy with the recommendations? Leave us a review and let us know!

Figure 11

Figures 7 through 11 displayed above are in chronological order of the display pages a user will interact with as they iterate through the steps of requesting song recommendations for a playlist. After logging in (Figure 7) using spotify account credentials, a dashboard of Top 10 Songs filterable by terms will display (Figure 8). When a user selects the 'Playlists' tab in the navbar, the user's playlists will display (Figure 9). When a user clicks on a playlist name the songs within the playlist will display (Figure

10) with a "Recommend Songs" button which will open a new page where recommended songs (along with the ability to directly listen to and add songs to their spotify playlist) will display (Figure 11). There is also a replace song button next to each recommendation that will replace a recommendation with the next ranked recommendation in the case that someone does not like their recommendation or wants to see more recommendations. The recommendations page also allows the user to leave feedback regarding their recommendations. Lastly, the about page allows a user to view the MLR feature importance charts (Figures 5 and 6) to better understand what contributed to their recommendations. While in developer mode, to use the app you must email kdadhaniya3@gatech.edu your spotify affiliated email and wait for access to be granted. For proof of concept, you may use the following demo Spotify account credentials - username: **tunetitansdemo@gmail.com** password: **TuneTitansDemo1!** To view our app follow this link: https://streamify.azurewebsites.net

The user provided feedback consists of ordinal values ranging from 1 to 10 inclusively which represent how well they liked their recommendations overall. While providing limited access, thus far, we have received 25 ratings, the most common rating was 10, with an average rating of 5.92, and a median Rating of 6. We anticipated our ratings to be low at first considering our base dataset of the Spotify Million Playlist Dataset only consisted of playlists created from January 2010 to October of 2017, thus no songs after 2017 were initially recommended. We store users' playlists (anonymously) to contribute to our database of possible songs to be recommended in the future, thus as more users utilize our MRS, the more 'newer music' will be integrated, thus our recommendations will only get better/ more diverse with time.

Conclusions and Discussion:

Based on the comprehensive analysis presented in our experiments/evaluation section of our project, we have successfully created a music recommendation system. We trained a two-tiered clustering model, which after testing multiple k values, we found that the optimal number of clusters to be 100 and 35 clusters respectively for songs and playlists. We made significant improvements in the training time of our Metric Learning to Rank algorithm model. Our finalized model achieved an impressive average Normalized Discounted Cumulative Gain (NDCG) score of 0.673, indicating a high quality of song ranking which is well above industry standards for this metric. We also discovered new findings within our feature analysis which revealed that 'speechiness' significantly influenced playlist categorization, demonstrating the importance of the presence of lyrics in user preferences. Additionally, we succeeded in creating an interactive Python Django web application which provides users with an intuitive interface to receive song recommendations. Currently, our web application is a minimum viable product. Our project team would require additional feedback from future users to improve both our algorithms as well as provide the team with honest insights on how to improve the user interface (UI) of our Django Web app. In addition to user feedback based on our surveys, we hope to grow our database of daily active users who frequently utilize our app for their song selection requirements. One benefit of having a growing database of users is that our algorithms will constantly pull from new playlists, and will be more robust in its prediction accuracy of song recommendations. Another benefit of growing our user database is that in the future, we can potentially pitch our idea to venture capital investors for us to acquire funding to further develop our application and algorithm based on the growing demand for its capabilities. Future work may include creating a mobile application that could reach millions of additional users in major digital distribution platforms such as the Apple store, Android Store, and Google Play Store. Additionally, as the team has focused on developing the music recommendation system for the Spotify Platform, a promising future project would be to integrate these features for the Apple Music platform as well. Apple Music is arguably the second biggest music streaming platform used in the world. Incorporating the integration of Apple Music to our platform would only raise the user outreach and increase the potential of our application. Overall, we are very pleased to have created a product that provides accurate song recommendations and we are excited to work as a team outside of this course to further develop our application to possibly reach a multitude of new users. Lastly, **all team members have contributed an equal amount of effort in their respective fields towards this project.**

References

Anand, R., Sabeenian, R. S., Gurang, D., Kirthika, R., & Rubeena, S. (2021). AI based Music Recommendation system using Deep Learning Algorithms. IOP Conference Series: Earth and Environmental Science, 785(1), 012013. https://doi.org/10.1088/1755-1315/785/1/012013

Barrington, L., Oda, R., & Lanckriet, G. (2009). Smarter than genius? Human evaluation of music recommender systems. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=65aa5dd1d217c9eda40905142ec903b09e2e37e6

Bauer, C., Kholodylo, M., & Strauss, C. (2017). Music Recommender Systems Challenges and Opportunities for Non-Superstar Artists. *BLED 2017 Proceedings*, (48). http://aisel.aisnet.org/bled2017/48

Bobadilla, J., Dueñas-Lerín, J., Ortega, F., & Gutierrez, A. (2023). Comprehensive evaluation of matrix factorization models for collaborative filtering recommender systems. *International Journal of Interactive Multimedia and Artificial Intelligence, 8*(4), 65-72. https://doi.org/10.9781/ijimai.2023.04.008

Gulzar, Y., Alwan, A. A., Abdullah, R. M., Abualkishik, A. Z., & Oumrani, M. (2023). OCA: Ordered Clustering-Based Algorithm for E-Commerce Recommendation System. *Sustainability,* Volume 15, 1-22. https://www.mdpi.com/2071-1050/15/4/2947

Han, D., Kong, Y., Han, J., et al. (2022). A survey of music emotion recognition. *Frontiers of Computer Science, 16*(166335). https://doi.org/10.1007/s11704-021-0569-4

Liu, Z., Xu, W., Zhang, W., & Jiang, Q. (2023). An emotion-based personalized music recommendation framework for emotion improvement. Information Processing & Management, 60(3), 103256. https://doi.org/10.1016/j.ipm.2022.103256

McFee, B., Barrington, L., & Lanckriet, G. (2012). Learning content similarity for music recommendation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8), 2207–2218. https://doi.org/10.1109/tasl.2012.2199109

Recommendations performance. mParticle Cortex. (2024, March 20). https://www.vidora.com/docs/recommendations-performance/

Schäfer, T., & Sedlmeier, P. (2010). What makes us like music? Determinants of music preference. *Psychology of Aesthetics, Creativity, and the Arts*, 4(4), 223–234. https://doi.org/10.1037/a0018374

Schäfer, T., Sedlmeier, P., Städtler, C., & Huron, D. (2013). The psychological functions of music listening. Frontiers in Psychology, 4(511). https://doi.org/10.3389/fpsyg.2013.00511

Schedl, M. (2019). Deep Learning in Music Recommendation Systems. *Frontiers in Applied Mathematics and Statistics*, 5. https://doi.org/10.3389/fams.2019.00044

Song, Y., Dixon, S., & Pearce, M. (2012). A survey of music recommendation systems and future perspectives. *Conference: The 9th International Symposium on Computer Music Modeling and Retrieval (CMMR),* 395-410. https://www.researchgate.net/pu

Van Den Oord, A., Dieleman, S., & Schrauwen, B. (Year). Deep content-based music recommendation. *Electronics and Information Systems department (ELIS), Ghent University*. Retrieved from https://papers.nips.cc/paper_files/paper/2013/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pd fblication/277714802_A_Survey_of_Music_Recommendation_Systems_and_Future_Perspective s

Verma, V., Marathe, N., Sanghavi, P., & Nitnaware, P. (2021). Music recommendation system using machine learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *Volume 7*(Issue 6), 80-88. https://ijsrcseit.com/CSEIT217615

Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., Xiong, H., & Chen, E. (2023). A survey on large language models for recommendation. *arXiv:2305.19860v4 [cs.IR]*. Retrieved from https://arxiv.org/abs/2305.19860v4