Song Recommendation and Automatic Playlist Continuation

Drona Jagad  
Artificial Intelligence  
San Jose State UniversitySan Jose, CA  
dronakaushik.jagad@sjsu.edu

Nemil Panchamia  
Artificial Intelligence  
San Jose State UniversitySan Jose, CA  
nemilnilesh.panchamia@sjsu.edu Hewitt Kothari  
Computer Engineering  
San Jose State UniversitySan Jose, CA  
hewitt.kothari@sjsu.edu

Hardi Trivedi  
Artificial Intelligence  
San Jose State UniversitySan Jose, CA  
hardi.trivedi@sjsu.edu

***Abstract*— With the popularity of music streaming services growing, so has the use of music recommendation systems. The objective of this project is to create a music recommendation system that can provide consumers tailored music suggestions based on their musical tastes. A list of suggested tracks that the user is likely to appreciate will be generated by the algorithm using user behavior data like as listening history, favorite artists, and genres along with the features of music like explicitly, dancability, energy, key, loudness and a dozen more features. The system will employ machine learning algorithms to evaluate user data and produce recommendations, such as user based and item based collaborative filtering and content-based filtering. A knowledge graph-based strategy will also be investigated, which adds semantic connections between musical pieces to improve the precision of suggestions.**

# Introduction

## Motivation and Objective

Music recommendation systems are now essential for giving customers a tailored listening experience due to the growing accessibility of digital music. The quantity of songs customers have access to has skyrocketed with the introduction of digital music services. Users will very certainly never be able to find and listen to every music that is offered to them. Massive amounts of information have been produced by the quick advancement of Internet and information technology, which has both substantially enhanced people's personal requirements and contributed to the issue of information overload. As a result, recommendation systems have been developed in an effort to better match the individualized needs of users and address the information overload issue. The Internet has taken over as the most significant aspect of people's lives in this age of information overload. However, it can be difficult to sift through the vast amount of information on the Internet to find the interesting pieces. Recommender systems have been extensively utilized in e-commerce, information portals, social networks, mobile location services, multimedia entertainment, and other industries. These systems customize and propose items to fulfill users' requirements based on their preferences and other attributes.

The project's goal is to investigate alternative approaches for handling playlist and music recommendation issues using the Spotify 1.2M+ song dataset. The intent of our group, Rockstars, which consists of Drona Jagad, Hewitt Kothari, Hardi Trivedi, and Nemil Panchamia, is to increase the precision and variety of song choices and produce more individualized playlist recommendations for consumers. The project will be carried out as a component of our spring 2023 CMPE 256 - Advance Data Mining course. The project's goal is to make more personalized playlist suggestions for consumers while also increasing the precision and variety of music choices.

The project will use a number of approaches to accomplish the goal, including Knowledge Graph Based Recommendation System, Hybrid Recommendation System, ANNOY combined with Matrix Factorization, and Sequential Recommendation for Automatic Playlist Continuation. A graph database is used by a knowledge graph-based recommendation system to describe relationships between various items and produce user-specific suggestions. To increase the precision and effectiveness of recommendations, hybrid recommendation systems incorporate various recommendation algorithms. High-dimensional data may be efficiently searched for nearest neighbors using ANNOY, while matrix factorization creates suggestions according to the user's tastes. Deep neural networks are used to model a sequential recommender system based on the song's information in sequential recommendation for automated playlist continuation.

We used deep neural networks to learn and reflect the underlying structure of the music data in our music recommendation algorithm. The music data is represented as a series of characteristics, such as tempo, rhythm, melody, and harmony, using a sequence-based method. A deep neural network is then trained to recognize the patterns and traits that are distinctive to the user's musical preferences using this sequence data as input.

A sizable dataset of musical songs and the metadata they are connected with, such as the artist name, track title, album title, and genre, is used to train the deep neural network. In order to provide consumers with individualized music suggestions, the network learns to recognize patterns and attributes that are shared by the music recordings in the dataset. The neural network may propose new tracks and playlists that are likely to be of interest to the user by examining the user's listening history and preferences.

By combining these methodologies, the project aims to develop an efficient and accurate music recommendation system that provides users with personalized song and playlist recommendations. The project will use the Spotify 1.2M+ songs dataset that contains various metadata such as track title, album title, artist name, track number, explicitness, and many more. The project is expected to improve the user engagement on music streaming platforms by providing personalized listening experiences to users. The rest of the report will detail the datasets we will use, the methodologies we will apply, and the evaluation metrics we will use to measure the effectiveness of our proposed techniques.

This is particularly crucial in the age of streaming music services, when consumers may not have the time or motivation to actively search for new songs and artists to listen to due to the overwhelming volume of music that is accessible. A strong system for music recommendations may boost user retention, engagement, and eventually income for streaming music services. It may also help musicians by exposing their work to a larger audience and thus boosting their cash streams. We hope our project will contribute to the development of music recommendation systems and provide valuable insights for future research.

## Related Work

J. Lui et al., conduct a survey of knowledge graph-based recommender systems and evaluate the main techniques used in these systems. The knowledge graph and user interest model have been used in a number of recent methods to enhance the system's interpretability and recommendation output. These methods are summarized in this study. Finally, in an effort to advance the growth of the recommendation fields, we suggest future research directions.[3]

The research paper discusses the use of knowledge graphs (KG) in recommender systems (RS) to overcome the limitations of traditional recommendation systems. It has been noted that consumers experience "cognitive overload" as a result of the internet's exponential rise in data and need recommendations to help them locate the pertinent facts. Traditional recommender systems, such those that rely on collaborative filtering and content, have trouble keeping up with users' shifting interests. Therefore, knowledge graphs are introduced to provide side information that can help in finding changes in the user interests and provide better explanations for recommendations.

The research done by Mok, L et al., adds to the body of knowledge on online music consumption by offering fresh perspectives on how consumers gradually seek for variation in their musical preferences. The authors demonstrate how humans are always discovering new information and how this process never ends. The article also emphasizes the significance of elements like platform design and seasonality in influencing user behavior. These findings have consequences for how recommender systems and other music discovery tools are designed, as well as for how we perceive online content consumption.[1]

Rishabh Mehrotra and Benjamin Carterette's research article "Recommendations in a Marketplace" (2019) [2] explores the difficulties in creating a framework for recommendations to drive a multi-stakeholder marketplace that includes both customers on the demand side and suppliers on the supply side. The authors stress the need for a multi-objective ranking/recommendation module that works to simultaneously optimize the various stakeholders' objectives while providing recommendations. The work is broken up into four sections, each of which goes into great detail into component-specific algorithmic techniques.

The writers analyze various case studies and emphasize current findings in phase IV, which is their final section. Overall, the study offers a thorough introduction to the difficulties in creating a framework for recommendations to drive a multi-stakeholder marketplace and underlines the research issues that must be resolved in this burgeoning field.

The authors[4] start out by going over how crucial recommender systems are in a variety of fields, such as e-commerce, healthcare, and education. They then give a thorough overview of the development and applications of knowledge graph-based recommender systems as well as how they differ from conventional recommender systems.

The authors explain how recommender systems built on knowledge graphs use the semantic connections between objects to provide tailored recommendations. They clarify that many techniques, including manual curation, automatic extraction, and crowdsourcing, can be used to generate knowledge graphs. The authors also go over the various varieties of knowledge graphs, including domain-specific, hybrid, and heterogeneous knowledge graphs, that can be applied in recommender systems.

Schedl, M concludes that DL has showed considerable potential in enhancing MRS's performance, notably in terms of accuracy and variety of suggestions. The lack of labeled training data, the requirement for accurate feature representation of musical elements, and the interpretability of DL models are some of the issues that the article also notes for DL-based MRS. Overall, the work demonstrates the potential of this technique to raise the caliber of music suggestions and offers a significant resource for scholars and practitioners interested in the application of DL in MRS.[6]

The authors draw attention to the limits of conventional music recommendation systems, notably in terms of their capacity to offer individualized suggestions. They contend that knowledge graphs, which encode music and user preferences[5] as entities in a graph and employ graph algorithms to provide suggestions, can offer a more practical approach.

The idea of knowledge graphs and their potential for music recommendation is introduced by the writers in the opening paragraphs. After that, they go through the many parts of a knowledge graph-based recommendation system, such as how music and user preferences are represented, how ontologies are used to store domain information, and how graph algorithms are used to provide suggestions.

# System Design & Implementation

## Algorithms Selected

### 

### ANNOY

A library called ANNOY (Approximate Nearest Neighbors Oh Yeah) offers a quick technique to find the closest neighbors in high-dimensional data. ANNOY and matrix factorization are frequently used in music recommendation systems to create customized suggestions based on user preferences.

Due to the high dimensionality of music data, ANNOY is particularly helpful in music recommendation systems. For instance, the Spotify 1.2M+ songs dataset has over 700,000 distinct track titles and over 300,000 unique artist names. This implies that there might be hundreds of thousands of variables to take into account when making user suggestions. With the help of ANNOY, it is simple to search through these high-dimensional data points and identify the closest neighbors, which are then utilized to provide tailored user suggestions.

When applied to music recommendation systems, ANNOY has been proven to significantly outperform other methods like brute-force search. For instance, ANNOY was used with matrix factorization to produce music suggestions for consumers in a research that was published in the Proceedings of the 13th ACM Conference on Recommender Systems. The findings demonstrated that ANNOY-based solutions performed better in terms of suggestion accuracy and efficiency than other cutting-edge methods.

In conclusion, ANNOY is a crucial tool for music recommendation systems since it offers a quick and effective technique to find close neighbors in large-scale musical data. Users can receive accurate and effective tailored suggestions by utilizing ANNOY in conjunction with matrix factorization.

### 

### Knowledge-Based Graph

Music recommendation systems that employ knowledge graphs can better describe complicated relationships between items and offer consumers recommendations that are more tailored to their needs. Additionally, they can use a variety of data sources, including user behavior, social network data, and audio features, to produce more accurate recommendations.

According to a Nielsen survey, the overall amount of audio consumption will rise by 14.6% in 2020 as a result of the popularity of music streaming. Forecasts indicate that by 2027, streaming music sales would reach $23.1 billion, continuing the current pace.

Users find it challenging to explore and find new music on streaming services since there is so much music accessible there. As a result, music recommendation algorithms are crucial for assisting users in discovering new music and building custom playlists.

Additionally, research demonstrates that tailored recommendations can boost user retention and engagement. According to Deloitte research, tailored recommendations can boost user engagement by as much as 60%. To boost user engagement and retention on music streaming services, it is crucial to have an accurate and effective song recommendation system.

The accuracy and personalization of music suggestions have improved thanks to knowledge graph-based recommendation systems. For instance, Zhao et al.'s (2020) study suggested a knowledge graph-based recommendation system for music that performed better in terms of suggestion accuracy than conventional collaborative filtering techniques.

As a result, knowledge graph-based recommendation systems are a useful tool in the music recommendation space because they can provide customers with precise and individualized suggestions, which raises user engagement and retention on music streaming services.

### Deep Neural Net

In the field of music recommendation today, DL is frequently used for three tasks: (1) automatic feature learning from audio signals and producing corresponding embeddings for CBF; (2) modeling item/track sequences for automatic music playlist continuation; and (3) extracting latent factors from user-item rating data to incorporate into CF models. The techniques particular to the music industry are covered in this review article. We avoid adding broad domain-independent research on the application of deep learning in wholly CF-based techniques (3) and instead restrict the scope to (1) and (2).

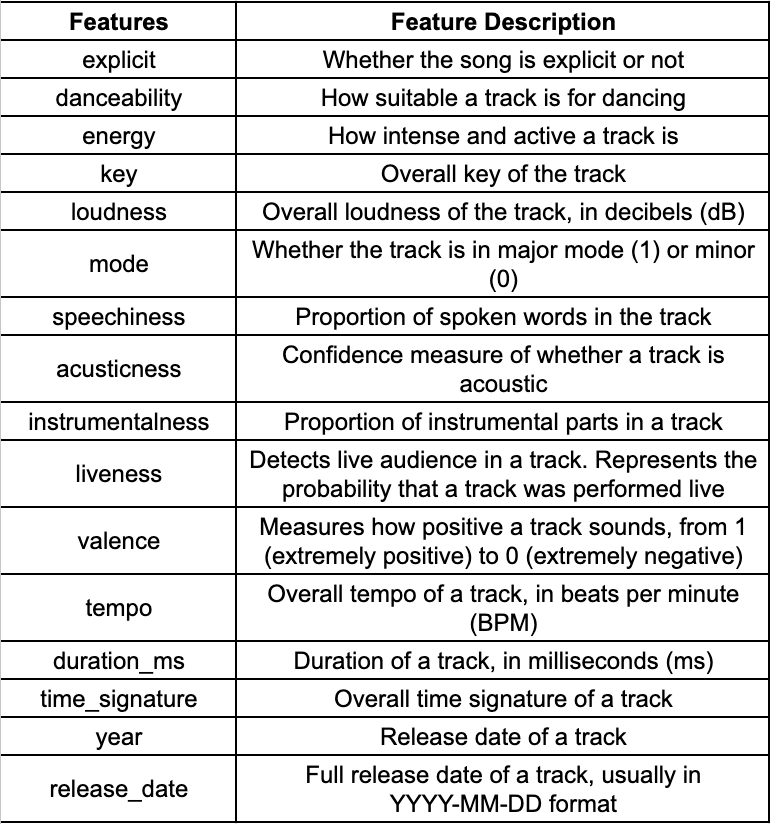
### Hybrid Model

In order to overcome the shortcomings of individual recommendation algorithms and give customers better recommendations, hybrid models are utilized in music recommendation systems. These models integrate two or more algorithms to produce a recommendation system that is more precise and varied.

A hybrid model that combines collaborative filtering and content-based filtering algorithms outperformed each method when employed separately, according to a study by Koren et al. Comparing the hybrid model to collaborative filtering or content-based filtering alone, the hybrid model was able to increase suggestion accuracy by 10–20%.

On the Last.fm dataset, Hidasi et al. conducted a different study to compare the effectiveness of various recommendation algorithms. The research discovered that the most accurate recommendations were made using a hybrid approach that combined collaborative filtering, content-based filtering, and popularity-based filtering. With an accuracy of 10 of 0.346 compared to 0.323 for collaborative filtering, 0.287 for content-based filtering, and 0.212 for popularity-based filtering, the hybrid model beat each technique applied separately.

## B. Technologies and Tools Used

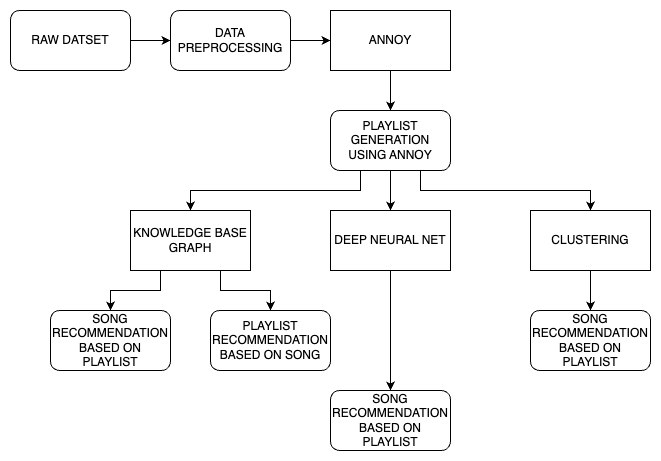


We have made use of the Cloud Python notebook made available by the Google Colaboratory which provides free access to the high-end GPUs and high RAM.

We have made use of the several libraries available in Python for the project. The libraries used include NumPy, Pandas, Matplotlib, Seaborn, Plotly Express, TensorFlow, Keras, Annoy, and Scikit-Learn.

## C. System Design

The workflow for the system follows a simple pattern where we start by taking a look at the dataset and visualize the similarities and the correlation between the features of the songs. We Preprocess the dataset and use the ANNOY algorithm on the initial normalized dataset to generate 500 playlists with random songs between 5 and 20. We then use these playlists and run several algorithms to recommend songs based on the playlist and also use the playlist to recommend songs based on it.



# Experiments

## Dataset Used

We are using the [Spotify 1.2M+](https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs) dataset available on Kaggle for this project. This dataset contains Audio Features for over 1.2M songs which are obtained from the Spotify repository using the Spotify API. The dataset is collected by downloading the entire MusicBrainz catalog and querying each album’s UPC(Universal Product Code) via Spotify API. Eventually the tracks for each album were obtained using the Spotify API, again.

The dataset contains 1204025 tracks along with their features. It has 24 variables which include the track names, album names and the features of the current track.

The features available for each track can be seen listed below.

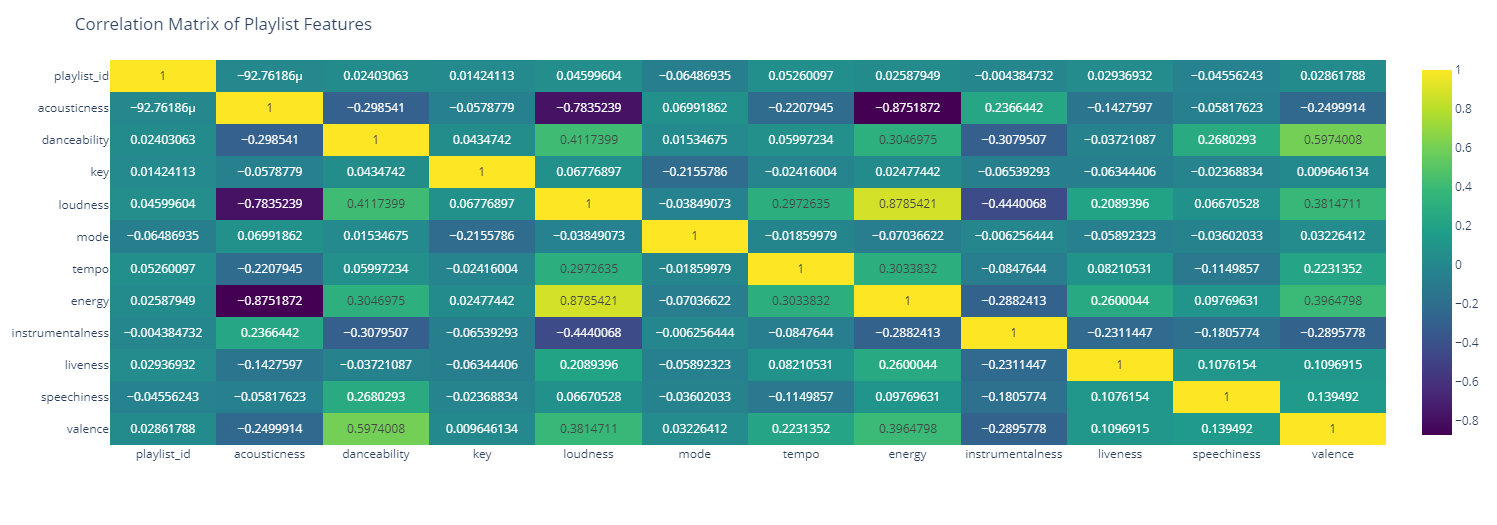
## 

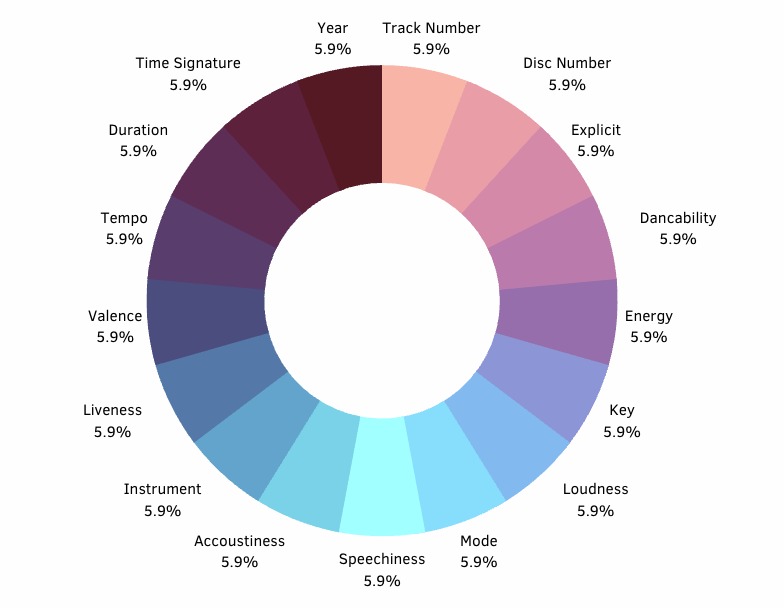
## Data Preprocessing

The dataset we used did not contain any kind User-item-Rating pairs which can be used for recommendation. We had to choose an approach where we generated playlists from the dataset and stored them as a new dataset.

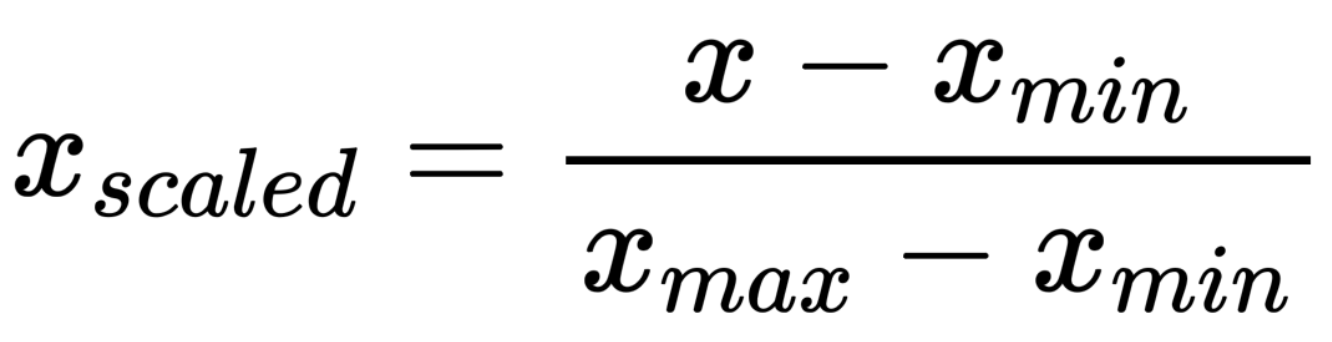
This was done using the ANNOY algorithm where songs were clubbed in playlist according to their similarities, i.e. Cosine Similarity which is used by the ANNOY algorithm. The features used for generating playlists of similar songs included features like ‘danceability’, ‘energy’, ‘key’, ‘loudness’, ‘mode’, ‘speechiness’, ‘acusticness’, ‘instrumentalness’, ‘liveness’, ‘valence’, and ‘tempo’.

We used 11 features to find similarity between the songs and club them together for a playlist. The playlists were generated randomly with each playlist containing between 5 and 20 songs and we made a subset of 500 playlists to keep the data big enough but not very big which causes heavy computation consumption.





All the features used were initially normalized using Min-Max Scaling technique to get them in the range of 0-1 which can be easily interpreted by the algorithms and the higher value terms do not get preference and the bias is avoided.



## Analysis of Results

There were many reasons which made ANNOY, an effective approach for music recommendation based on track features. High Dimensionality: There were many features to describe the audio quality of the tracks in the given dataset which can be represented as high dimensional vectors. ANNOY’s ability to handle nearest neighbour searches, especially in high dimension space makes it well-suited for such recommendation tasks. Scalability: Music Database can be extensive with millions of data entries, ANNOY can be scaled effectively on large datasets to provide real-time recommendations. Content-Based recommendation: By focusing on track features, the ANNOY-based nearest neighbours approach provides content-based recommendations, this can help recommend songs for the tracks that are not widely known or share no explicit collaborative information, such as user ratings or listening history.

The Knowledge Based Graph algorithm helps to solve an issue which we used to face in recommending the songs to playlist which are not seen during the creation of the annoyIndex. For doing that we require to again create the Trees using AnnoyIndex and recommend the songs, while using the Knowledge Based Graph we can directly connect the edge to the playlist with the weights of the edge on the basis of the cosine similarity of the track\_features and playlist\_features. Here we are actually trying to recommend the playlists which are kind of similar with the track\_features that particular user liked and addition to that we are trying to recommend the songs which can be added to the particular playlist on the basis of the cosine similarity matrix generated between the playlists\_id and all the track\_ids considering their features as the main aspect for recommendation. By using this particular algorithm we can add any number of nodes to the playlist and the graph would build on its own and no need to retrain/recreate the graph. So we will be not wasting time creating the representations of the playlists with the songs.

The advantage of using a Deep Neural Net is to reduce the memory usage that we are required to keep for Storing the Graph using Knowledge Based Graphs and making it run at lower computes which are available for normal local systems. A Flatten layer, several Dense layers, and then a few more Dense layers make up the model architecture. To obtain a dense representation of the input sequence, an embedding layer is applied after the input layer receives a sequence of track IDs. The convolutional layers, which are intended to extract local features from the input sequence, are then fed this representation. Convolutional layer output is transformed into a 1D tensor that can be fed into the dense layers using the Flatten layer. In order to determine whether or not the input sequence is similar to a specific playlist, the output from the dense layers is finally sent through a Softmax activation function. The binary cross-entropy loss function and the Adam optimizer are used to train the model. The evaluation metric for the model is the cosine similarity. GridSearchCV is used to fine-tune the model's hyperparameters in order to identify the ideal set of parameters that produce the highest performance.In conclusion, the offered code provides a deep learning model that can suggest playlists that are similar to an input track based on the attributes of the input track. Convolutional neural networks, dense layers, and dropout layers are used in the model architecture to extract local features from the input sequence and generate a sparse cross categorical entropy classification output. GridSearchCV is used to optimize the model's hyperparameters in order to boost its performance.

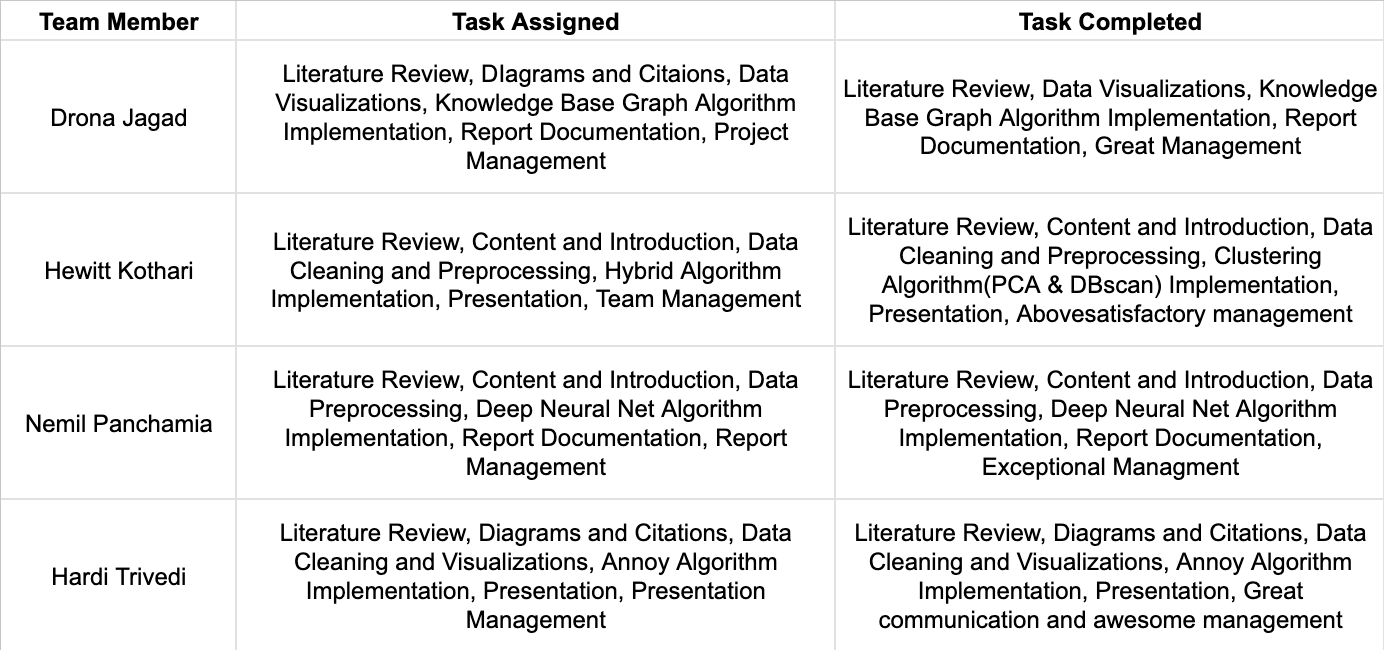
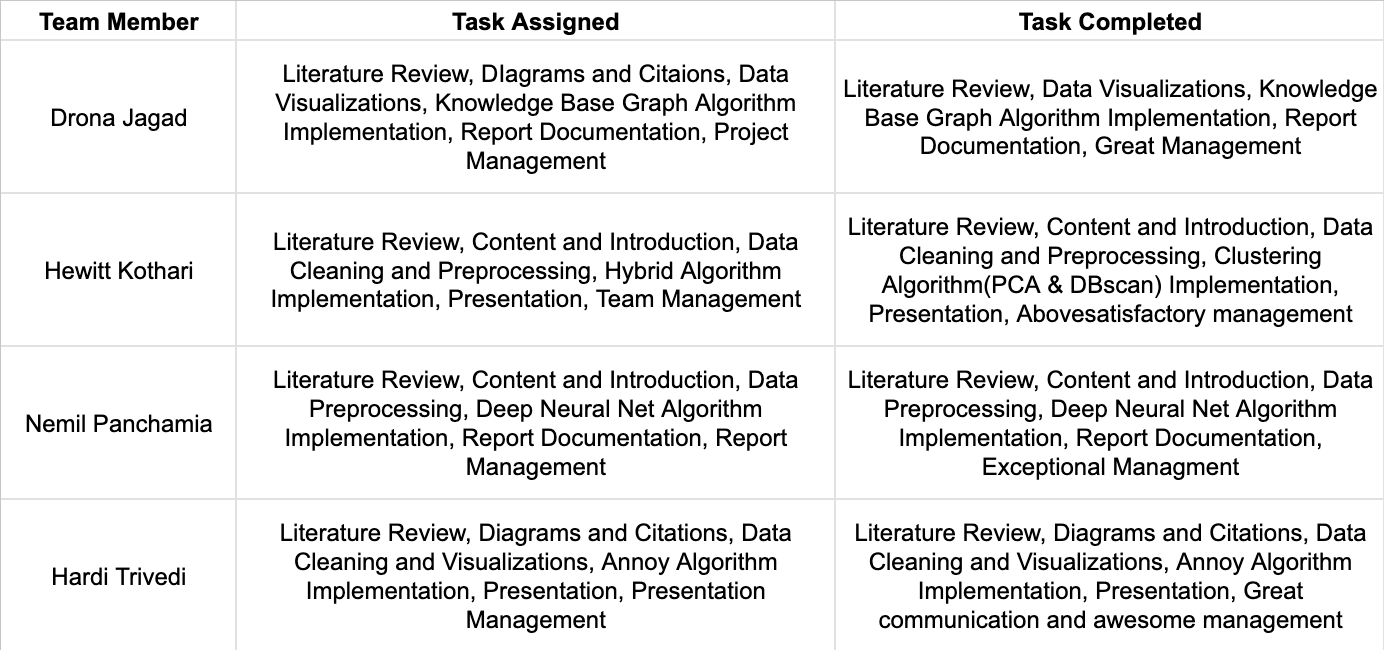
# Experiments

## Things that Worked

We used a number of strategies to construct playlists and assess their similarity when developing our model. To start, we made over 50K playlists from the 1.2M tracks database using ANNOY models and the distinctive qualities of the songs. Then, in order to save computational power, we used a Knowledge-Based Graph Model and a different method for computing cosine similarities. We also employed a deep neural network and a matrix function called cosine similarity.

We calculated the mean of all the attributes of the songs in each playlist in order to assess the similarity between them. As a result, we were able to compare playlist similarity using a knowledge graph-based approach. Overall, the combination of these methods enabled us to create a large number of distinctive playlists, compare their similarities in an effective manner using a more effective computational method, and ultimately develop our music recommendation system.

## Things that didn’t Work

One of the primary difficulties stemmed from ANNOY's lack of support for incremental index building. This limitation meant that we could not add new items to the index without completely rebuilding it from scratch. This constraint could lead to potential inefficiencies, as the entire index would need to be reconstructed whenever new data points were introduced. In order to overcome the aforementioned issue, we attempted to build ANNOY on top of other indexing structures, such as FAISS (Facebook AI Similarity Search) and HNSW (Hierarchical Navigable Small World) indexes. However, this approach resulted in a decrease in accuracy, which was an undesirable trade-off.

While working with Knowledge Based Graph we faced the RAM outage issues for creating the graph on the 1.2M songs i.e. creating the 1.2M songs and want to create edges between the songs by using the cosine similarity measure as the weight attribute for the edge. We resolved this issue by creating the playlists which also have the meaning of the features of all tracks in that particular playlist. By doing this we were able to reduce the number of nodes and edges. We stored the tracks\_ids of that playlist to store in the attribute which we can later access, update it anytime. While recommending the songs for the particular playlist we were not able to make that run on the Graph as it took a large amount of RAM to calculate the Cosine Similarity between the 1.2M songs and number of playlist we were having. So to resolve this issue we created the basic matrix between the playlist\_id and Number of songs and used that for recommendation.

When using the Deep Neural Net Algorithm the main issue was to create the loss function which was suitable for training the model which we later fixed with the cosine loss. Took more time to search through the Grid Search for finding the best suitable parameters for running the model with the best hyperparameters, used GPU for fastening the training through the GridSearch.

1. Conclusion

Music is a form of art that is being appreciated all around the world by people. There are millions of song available with different artist, genres etx such that it can get overwhelming for the users to find music that align their tastes. That is where music recommendation comes in. Here we have tried to recommend songs based on track features and audio quality. By leveraging four different approaches, including ANNOY nearest neighbours, Knowledge-based graph, Deep Neural Networks, and Density-based Clustering, we were able to train models on a large dataset of 1.2 Million songs and generate high-similarity recommendation playlists. Our ANNOY-based approach allowed us to efficiently represent our data as trees and achieve homogenous recommendations. The Knowledge-based graph approach enabled us to create highly interpretable graphs and gain insights into the structure of our data. The Deep Neural Network approach helped us uncover intricate correlations and patterns between features and labels. Lastly, through DBSCAN, we represented the data in the form of clusters and produced more diverse recommendations. Overall, our project highlights the potential of combining multiple approaches to tackle complex recommendation problems in the music domain.

# Project Plan

The project was executed end-to-end with all the members participating equally and having their input in all the aspects of the project to some extent. All the tasks were distributed equally so as to include all the members and let everyone contribute and have an hands-on experience in all the aspects.

# References

| [1] | Mok, L., Way, S. F., Maystre, L., & Anderson, A. (2022). The Dynamics of Exploration on Spotify. Proceedings of the International AAAI Conference on Web and Social Media, 16(1), 663-674. https://doi.org/10.1609/icwsm.v16i1.19324 |
| --- | --- |
| [2] | Rishabh Mehrotra and Benjamin Carterette. 2019. Recommendations in a marketplace. In Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19). Association for Computing Machinery, New York, NY, USA, 580–581. https://doi.org/10.1145/3298689.3346952 |
| [3] | J. Liu and L. Duan, "A Survey on Knowledge Graph-Based Recommender Systems," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2021, pp. 2450-2453, doi: 10.1109/IAEAC50856.2021.9390863. |
| [4] | J. Chicaiza and P. Valdiviezo-Diaz, “A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions,” Information, vol. 12, no. 6, p. 232, May 2021, doi: 10.3390/info12060232. |
| [5] | Sergio Oramas, Vito Claudio Ostuni, Tommaso Di Noia, Xavier Serra, and Eugenio Di Sciascio. 2016. Sound and Music Recommendation with Knowledge Graphs. ACM Trans. Intell. Syst. Technol. 8, 2, Article 21 (March 2017), 21 pages. https://doi.org/10.1145/2926718 |
| [6] | Schedl, M. (2019). Deep Learning in Music Recommendation Systems. Frontiers in Applied Mathematics and Statistics, 5. https://doi.org/10.3389/fams.2019.00044 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |