

# Milestone 2 - Project Proposal and Plan

DSC 412 Machine Learning - *Author: Isaac Davis*

## Introduction

Music recommendation systems utilized by media companies, such as Spotify and Apple Music, rely heavily on industrial-scale algorithms to recommend music to their listeners. Spotify has the most accurate music recommendation algorithm, which uses a combination of content-based and collaborative filtering. These algorithmic approaches employ data about the song and artist being recommended alongside the listening habits of "similar" users, and they often prioritize business goals, such as capitalizing on popular tracks and maximizing the time spent streaming.

While these methods are effective, many automated playlists feel impersonal and neglect subtle differences in listener preference. Further, Spotify has yet to be able to account for in-the-moment listening preferences - with its AI DJ and Enhanced Shuffle projects that attempt to bridge these gaps but have ultimately missed the mark for many users.

My argument is for a more tailored, human-centered music recommendation experience that puts the power in the hands of the listener. By capturing the listener's current mood alongside more traditional recommendation metrics, we can create a highly personal and engaging experience. Incorporating direct user input, focusing on smaller artists, and making the platform interactive and adaptable are key elements that set this project apart from other music recommendation systems.

## The Proposal

This project seeks to build a recommendation engine using the Spotify API that is:

1. **User-Centered:** After logging in using their Spotify account credentials, users will take a quiz or survey that allows users to input specific preferences, moods, and current feelings.
2. **Multi-Dimensional** Data will come from three primary sources: survey data, existing data on user preferences, and metadata associated with specific songs. These sources enable this algorithm to assess multiple dimensions of what makes a song "recommendable."
3. **Interactive and Adaptable:** Feedback from users will help the system refine its recommendations as preferences change.

This project will benefit multiple groups. Music listeners dissatisfied with current recommendation platforms (especially those from niche communities) and independent artists who struggle to find visibility could benefit from personalized music recommendations.

## Machine Learning Algorithms

This project will likely need to employ several machine learning algorithms, such as:

- **Matrix Factoring:** Mathematical models such as Alternating Least Squares (ALS) or Singular Value Decomposition (SVD) can help the model discover hidden patterns and relationships about users' deeper preferences.
- **Content-Based Filtering:** Here, the model will use Spotify's API to analyze song metadata, such as tempo, energy, danceability, and genre. This ensures that the recommendation aligns with explicit and implicit preferences of users.
- **Categorization:** A sturdy and lightweight categorization algorithm, such as K-Means and K-Nearest-Neighbors, will help this model recommend songs and artists based on similarity to other songs in the data set.
- **Quiz-Based Feedback:** A supervised learning model, such as logistic regression, can help the model provide initial recommendations based on survey responses.

This project will not utilize collaborative filtering, a core component of Spotify's recommendation algorithm. Collaborative filtering uses the data of similar users to predict listening habits and make recommendations. (If user A and user B both like artist Z, and user B likes artist Y, then user A might like artist Y.) This algorithm relies on a vast amount of data that is outside the scope of this project and may reinforce existing biases based on the filtering algorithm.

## Potential Obstacles

There are a number of potential obstacles that could interfere with the accuracy and personalizability of this model:

- **Availability of data:** If insufficient data is publicly available online or accessible through Spotify's API, designing this model will be incredibly difficult. Additionally, there will not be enough data to implement collaborative filtering.
- **Accuracy vs. Personalization:** This project will need to balance making an accurate recommendation and overfitting it to user preference, which reduces diversity in recommendations.
- **Scalability:** This project may be outside the scope of this course, and scaling the system to serve a larger number of users may compromise the personalized experience.

## The Plan

### Data Sources

This model will need data from a variety of sources:

1. A large, neutral music dataset from which the model will learn and recommend. The Million Song and Million Playlist datasets are often used when training music-related models. Each entry in the dataset should include the track, artist, album, and many metadata points. The Spotify API can also interface with their vast music datasets.
2. Personalized data on a user, such as listening habits and existing playlists. Spotify API calls can help gather this data.
3. Survey data on the user's current mood and subconscious preferences.

### Data Organization

Different sources of data will require different organization methods.

- **User Data:** User data should include an identifier, demographic information, quiz responses, listening history, and feedback
- **Song Metadata:** Song data is the basis of recommendation and includes artist, album, genre, acoustic features, popularity, and release year.
- **User-Item Interaction Data:** This matrix will contain data on how users interact with different songs. It could include a song's replay count and whether or not it was skipped by a particular user.

## Data Analysis

This model includes several features, including user preferences, listening patterns, and implicit feedback. Spotify's API allows for the assessment and communication of other metrics about individual listening habits.

## Testing

This project would need to evaluate how well the model predicts music and how users interact with the system. The SciKitLearn library can be used to perform industry-standard ways of testing model accuracy. Collecting user feedback and engagement will help improve the model after deployment. Other testing frameworks may be required to tune the recommendation.