Music Recommendation System: Final Project Report

Introduction:

In modern life, music streaming services have been an integral part, with platforms like Spotify delivering millions of songs to users worldwide. However, the huge quantity of available music creates a challenge here: how could users discover music that genuinely resonates with their current emotion state? The project I am doing addresses the challenge by developing a personalized music recommendation system based on user behavior analysis and mood prediction.

This system leverages the connection between user's behavior patterns on the platform and the emotional state and then recommends music that could align with the emotional state. This project's approach is quite different from traditional recommendation system that focus solely on genre preferences, as it is actually communicating with the psychological aspect of music consumption, especially for how users' moods can influence their music choices and preferences.

This project actually is inspired from research in affective computing and music psychology, which suggests that people use music to enhance their emotional states. According to Yang and Chen (2012), communicating emotional factors in music recommendation systems can strongly and significantly improve user satisfaction by delivering more contextually relevant suggestions. Also, by predicting user's current mood from their behavior and then do music recommendation that matches the mood, my system aims to provide more personalized and satisfying recommendations.

Methods:

The music recommendation system includes four main components. Each addresses a specific aspect of the process of recommendation.

1. User behavior mood prediction: This component analyzes user behavior data in order to predict user's current emotional state. I chose to use the Spotify User Behavior Dataset from Kaggle for these crucial reasons:
    a. This dataset contains comprehensive information about users' demographics, listening habits, and preferences that could be really difficult to collect.
    b. This dataset includes users' self-reported information about how music influences their mood, delivering truth labels for training the mood prediction model.
    c. The data captures authentic Spotify user interactions, which make it a classic and representative of the real-world music streaming behavior.
    d. Although the dataset shows concentration in some demographics, such as younger adults, female users, it still provides sufficient diversity to build and test a generalized and comprehensive model.

    The features I used for prediction:

    a. Demographic information – age, gender.
    b. Platform usage pattern – subscription plan, using period, device usage for listening.
    c. Rich contents – preferred listening content, favorite music genre
    d. Listening habits – music time slot, exploration method
    e. Previous interaction with recommendations – music recommendation rating

    Dataset Demographics:

Analysis of the Spotify user behavior dataset revealed these important characteristics: The age distribution shows that large majority of users (81.2%) are between 20-35 years old, followed by 12–20-year-olds (13.7%), with minimal representation from other age groups. This concentration in younger demographics influenced my model design to ensure it can best serve the primary user base. The gender distribution indicates that females constitute 75.2% of users, males 21.9%, and other genders 2.9%. This gender imbalance was taken into account during model training to ensure fair recommendations through all gender categories.

I am using two prediction models:

a. Custom regularized logistic regression model
b. Random forest classifier

In order to address class imbalance in the mood labels, the RandomOverSampler technique was applied to make sure that the model learns effectively from all mood categories. Using RobustScaler for feature scaling makes sure to handle potential outliers, and polynomial features have been created to capture interaction effects between variables.

User Preferences Analysis:

The Spotify user behavior dataset also provides valuable insights into music preference and content pattern: Analysis of preferred music genres shows a distinct preference for Melody (over 250 users), followed by Rap and Pop (approximately 100 users each), and Classical & Melody/Dance genres. This distribution informed the understanding of how genre preferences might correlate with emotional responses to music. The huge majority of users (78.8%) prefer music content over podcasts (21.2%), confirming the focus on music recommendation for our system. However, the significant podcast audience suggests potential for future expansion into mood-based podcast recommendations. User ratings of previous recommendations show a concentration in the middle range (ratings 3-4), with fewer users giving extreme ratings (1 or 5). The pattern helped inform the evaluation metrics, particularly in understanding how users perceive recommendation quality. In the correlation heatmap for user behavior features, it reveals these important relationships: Strong correlation (0.43) between favorite music genre and preferred listening content. Moderate negative correlation (-0.46) between music time slot and preferred listening content. Moderate correlation (0.39) between music influential mood and preferred listening content. Correlation (0.38) between music time slot and listening device. Correlation (0.39) between subscription plan and premium subscription willingness. These relationships tell the feature selection and interaction modeling in the user behavior prediction system.

2. Music clustering and mood labeling

   Since the Spotify dataset does not have explicit mood labels for songs, I used these to create pseudo-labels:

   a. K-means for clustering to group songs with similar audio features. There are 12 distinct clusters here.
   b. The audio features are used for clustering: Valence, Energy, Danceability, Duration, Instrumentalness, Acousticness, Liveness, Speechiness, Loudness, Tempo.

Each cluster was assigned a mood label based on the feature: Happy: High valence, high energy. Melancholy: Low valence, high Acousticness. Party: High energy, high danceability. Relax: High acousticness, high instrumentalness. Talkative: High speechiness. Live: High liveness. Uplifting: High valence, high tempo. Calm: Low valence, low energy. Deep: High instrumentalness, low valence. Motivating: High tempo, high energy. Fun: High danceability, high valence. Other: Default label when no specific criteria are met. To determine how many mood clusters would work best, I used the elbow method, plotting how tightly packed the clusters were (using WCSS) against different numbers of clusters. While the graph suggested 4-5 clusters might be sufficient (where the line bent most sharply), I decided to use 12 clusters instead. This choice wasn't just technical - as a daily music listener, I know emotions in music are nuanced, and more clusters helped capture differences between moods like 'melancholy' vs. 'calm' that might otherwise be grouped together. To see if these clusters made sense visually, I used Principal Component Analysis (PCA), which helped me transform the complex audio features into a simple two-dimensional map where I could actually see how the different mood categories separated from each other.

3. Music mood prediction model:

   Now the songs were labeled with mood categories, a Random Forest classifier was trained in order to predict the mood of a song based on the audio features. There are two purposes for the model: It can validate the cohesiveness of the K-means cluster results. It can also enable prediction of mood scores for songs, which can be really crucial for the recommendation process. I split the data using 80/20 approach - training the model on 80% of the songs and testing it on the remaining 20%. I was careful to use stratified sampling here to make sure the rarer mood categories were represented properly in both sets. This would help ensure the model could identify all mood types, not just the common ones.

4. Recommendation System Integration:

   The full recommendation process follows the following steps:

   a. Predict the user's current mood using User Behavior Mood Prediction Model.
   b. Map the predicted user mood to corresponding music mood labels using predefined mapping table as this: mood_label_mapping = { 'relax': ['relax', 'calm', 'deep'], 'sad': ['melancholy'], 'party': ['party', 'fun'], 'motivation': ['motivating', 'uplifting'], 'happy': ['happy', 'fun', 'uplifting'], 'talk': ['talkative'], 'live': ['live'], 'other': ['other'] }
   c. Filter the song database to include only songs with matching mood labels.
   d. Use music mood prediction model to calculate the score for each filtered song, delivering how well it matches the user's predicted mood.
   e. Sort the songs by the calculated scores and recommend the top 10 songs the highest scores.

Data sources and preprocessing

In my project, I used two main datasets:

1. Spotify User Behavior Analysis: It contains information about users' interaction with the platform, preferences, and self-reported mood influences. The Kaggle dataset is valuable since it provides real-world user data that could be really challenging to collect independently.

2. Spotify dataset: It contains audio features for over 170,000 songs, providing a really comprehensive selection of music with detailed characteristics of audio for accurate mood classification.

For the data preprocessing, I have the following: normalization of numerical features using MinMaxScaler, encoding of categorical variables by using LabelEncoder, handle missing values by removing incomplete records, feature scaling by using RobustScaler for user behavior model, create polynomial features to capture interaction effects.

Results

User behavior mood prediction model:

The random forest classifier did great performance for predicting user moods based on behavior. The model correctly classified most instances of mood categories, with just one single misclassification for category 1. The training output looks like this: User Behavior Prediction Mood Model Performance: Accuracy: 0.8000 F1 Score: 0.8000 Precision: 1.0000 Recall: 0.8000. The classification report showed good precision (1.0) across all mood categories, with only "Relaxation and stress relief" having lower recall (0.0), likely due to limited examples in the dataset. The weighted average metrics show the model's strong overall performance with an accuracy of 80%, which is impressive considering the subjective nature of mood classification.

Music clustering results:

The k-means clustering method successfully grouped songs into distinct mood categories. For the PCA, it reveals: The mood clusters are distributed across the two-dimensional space created by the principal components, with some clear separations between mood categories. Clusters like "happy," "fun," and "motivating" appear grouped in the lower left quadrant, which suggest similarity in the audio profiles. The "talkative" cluster appears in two distinct regions, indicating potential subgroups within this mood category. The "deep" and "melancholy" clusters have more central positions, with some overlap with other categories. Some clusters show significant overlap, reflecting the continuous but not discrete nature of mood in music.

When I dug into how the audio features relate to each other, I found some really interesting patterns. No surprise, the more energetic songs are typically louder too - these two features moved together strongly (correlation of 0.79).

I also noticed that energy and acousticness really don't get along (correlation of -0.76). This basically confirmed what I suspected - those raw, stripped-down acoustic songs tend to be more chill and less energetic than their electronic counterparts.

One finding I found particularly useful was how valence (basically the musical happiness) connects with danceability (0.57). Those upbeat, happy tunes really do make you want to move! This relationship helped me understand why certain mood clusters were forming the way they did in my analysis.

The acousticness-loudness relationship (-0.58) wasn't shocking either - acoustic tracks are typically more intimate and quieter. And tempo? It had some connection to energy and loudness, but weaker than I expected (just 0.26 and 0.22). This told me that fast songs aren't necessarily always the most energetic or loud - tempo is more independent than I initially thought.

Looking at the individual audio features in the dataset revealed several interesting patterns that helped shape my approach. Most songs fell into that sweet spot of 2.5 to 5 minutes long. The liveness values told a clear story too - the large majority of songs were definitely studio recordings, with very few live performances in the mix. This makes sense for a streaming platform focused on polished productions.

I found the tempo distribution particularly useful for mood classification. Most songs clustered around 100-130 beats per minute - that mid-tempo range where so much pop, rock, and hip-hop lives. But the full spread gave me plenty of examples at both extremes for identifying very slow, melancholy pieces and high-energy dance tracks.

The emotional valence (musical happiness) was surprisingly balanced across the spectrum, giving me good coverage for both uplifting and melancholy mood categories. I used all these patterns to set realistic thresholds when defining my mood categories.

Music mood prediction model:

The random forest classifier for predicting song moods based on audio features have strong performance: Music Prediction Mood Model Performance: Accuracy: 0.9573, F1 Score: 0.9573, Precision: 0.9573, Recall: 0.9573. The classification reports show consistent performance through all mood categories, with precision and recall values ranging from 0.95 to 0.97. "Melancholy" and "talkative" categories achieved the highest F1-scores (0.97), while "deep" had the lowest recall (0.89) but still maintained high precision (0.95). With over 42,000 test samples, these results demonstrate the model's robustness and reliability for music mood classification.

Recommendation system evaluation:

The recommendation system successfully demonstrated the ability to predict user moods based on behavior pattern, map those moods to appropriate music mood labels and recommend songs that match the predicted mood with high confidence scores. When running the recommendation system, it first predicts user moods from behavior data. The output array below shows divers range of predicted moods: The predicted mood is: ['Sadness or melancholy' 'Social gatherings or parties' 'Relaxation and stress relief' 'Relaxation and stress relief, Social gathering...' 'Relaxation and stress relief, Social gathering...' '...' 'Uplifting and motivational, Sadness or melancholy' 'Relaxation and stress relief, Uplifting and mo...' 'Relaxation and stress relief, Sadness or melan...' 'Relaxation and stress relief, Uplifting and mo...' 'Relaxation and stress relief, Sadness or melan...' '...' '...' 'Relaxation and stress relief, Sadness or melan...' 'Relaxation and stress relief, Sadness or melan...' 'Relaxation and stress relief, Social gathering...' 'Relaxation and stress relief, Social gathering...' 'Relaxation and stress relief, Uplifting and mo...' 'Sadness or melancholy' 'Sadness or melancholy' 'Social gatherings or parties' 'Social gatherings or parties' 'Uplifting and motivational, Sadness or melancholy' 'Uplifting and motivational, Sadness or melancholy']

For the example of a user with a predicted mood of “Sadness or melancholy”, the system recommends the following songs, melancholy score of 1.0 which is perfectly matched:

```
The mood is: Sadness or melancholy
Music recommendation:
                                                 name  ... melancholy_score
3                                           Danny Boy  ...              1.0
94692  La boutique fantasque, P. 120 (after Rossini):...  ...              1.0
94671                                 Anniversary Waltz  ...              1.0
94643  Violin Sonata in G Minor, L 140: III. Finale. ...  ...              1.0
94639                                         Ay-ay-ay  ...              1.0
94637                      Girl Crazy: Bidin' My Time  ...              1.0
94632  Twelve Poems of Emily Dickinson: II. There cam...  ...              1.0
94617  I due Foscari: Notte! Perpetua notte che qui r...  ...              1.0
94616  Romeo and Juliet, Op. 75: 9. Dance of the Girl...  ...              1.0
94613                      Bhalobasi Bole Sarati Jiba  ...              1.0
```

In this example (using the predicted emotion of the first user as input), it demonstrates that the system is effective in identifying songs, which precisely match user’s emotional state, with all recommended songs having the max possible melancholy score.

Conclusion:

What worked well:

One thing I'm particularly proud of is how my system actually considers what users might be feeling, rather than just throwing songs at them based on what's popular or what they've listened to before. Looking at the user demographics (mostly young women between 20-35), I was able to fine-tune the recommendations for the people who would actually be using the system.

Breaking the system into two separate models - one for figuring out the user's mood and another for matching songs to moods. It made the whole system easier to understand and adjust, and I could fix problems in one part without messing up the other.

The K-means clustering is efficient. Without existing mood labels for all these songs, I would've been stuck. I was also happy with how I combined different types of information - everything from basic user demographics to detailed audio characteristics like valence and tempo. Finding those connections (like how favorite genre relates to content preferences) really helped improve the quality of recommendations.

Room for improvement:

Be honest - mood is really subjective. A song that sounds melancholy to me might be relaxing to someone else. My system doesn't yet account for these personal differences, which is definitely a limitation.

The user dataset I worked with wasn't as diverse as I'd like. It skewed heavily female and younger, which might make the recommendations less accurate for other groups. A bigger, more varied dataset would definitely make the model more robust.

Right now, the mapping between user moods and music moods is pretty rigid - I basically hard-coded which music moods correspond to which user emotional states. A smarter approach would be to let this mapping evolve based on user feedback. I also stuck to just audio features, but lyrics obviously play an important role in a song's emotional impact. Adding some natural language processing to analyze lyrical content could make the mood detection better. Lastly, with only about a third of users willing to pay for premium subscriptions, it might make sense to develop different recommendation approaches based on whether someone's a free or premium user. Premium users might appreciate more complicated or personalized mood-based recommendations.

Work Cited:

Yang, Y. H., & Chen, H. H. (2012). Machine recognition of music emotion: A review. ACM Transactions on Intelligent Systems and Technology, 3(3), 1-30.