Predicting Spotify Hits:

A Multimodal Classification Modeling Approach

15.095 Machine Learning under a Modern Optimization Lens - Final Report

Brian Lu
luki90@mit.edu

Annie Liu
yil297@mit.edu

1. Introduction

Spotify receives tens of thousands of new songs every day, yet only a small fraction achieve significant commercial success. Understanding what drives this success remains an important question for both industry practitioners and researchers. Inspired by the vision of a Holistic Artificial Intelligence for Medicine (HAIM), which brings together diverse data sources through a unified multimodal approach, this project addresses the challenge through a framework that integrates multiple sources: structured metadata, text descriptions generated from the metadata, audio previews, lyrics, and album covers. By integrating all these modalities, we can examine how each perspective captures different aspects of a song and how these perspectives together enhance the model's expressive ability.

2. Data

Our tabular dataset originates from a collection of 1.16 million Spotify Tracks Dataset, each accompanied by a set of audio descriptors (danceability, energy, loudness, valence, tempo, etc.), track- and artist-level metadata (including track name, artist name, release year, and album information), as well as a raw popularity score ranging from 0 to 100. Detailed definitions of each feature can be found in the dataset documentation linked with the original source. More details on the tabular preprocessing are provided in Section 3.

Using the tabular data as baseline, we construct a multimodal dataset by adding four additional sources of information: short text descriptions generated from the metadata, 30-second audio previews, song lyrics, and album cover images. Each modality is later converted into an embedding representation for classification modeling. The full multimodal collection pipeline is described in Section 4.

3. Tabular Data Processing

Before modeling, we need to clean, normalize, and engineer features from the raw data to address issues such as structural biases, class imbalance, and variation across artists. The following sections describe these processing steps and explain the reasoning behind them.

3.1. Year-Specific Popularity Labels

Spotify's popularity score is strongly time-dependent. Because the dataset ends in 2023, newer songs benefit from recent streaming activity and playlist exposure, which makes their popularity scores appear higher even if their impact is unknown in the long term. In contrast, older songs naturally decline in score over time. Using the raw popularity value would therefore introduce a strong bias and make older songs look unpopular regardless of their actual historical success.

To reduce this distortion, we convert the continuous popularity score into a three-level label (flop, regular, and hit) computed within each release year. All songs with zero popularity score (13.66% of the dataset) are labeled as flops (Fig. 1), and we extend the flop category to include the bottom 20% of nonzero songs for each year. The top 0.5% of songs within each year are labeled as hits, and the remaining songs form the regular class. By normalizing popularity within each release year, we ensure that songs are compared only to others released in the same year, so each track has a similar chance to accumulate streams.

After applying this method, the hit class remains extremely rare (0.55% of all tracks), while the flop class contains a large share of low-visibility songs, which matches real-world streaming patterns (Fig. 2).

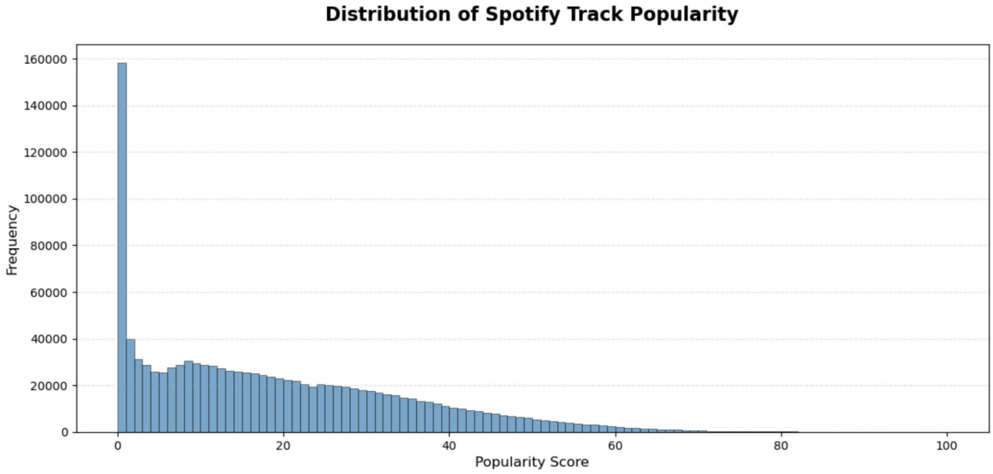


Figure 1. Distribution of overall spotify track popularity

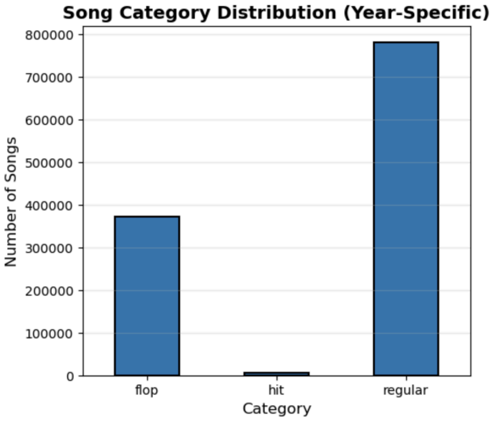


Figure 2. Distribution of the three popularity classes

3.2. Tabular Features Selection

From the full Spotify metadata, we keep only the variables that are most directly related to musical content or artist identity. This results in a 20-dimensional tabular feature set that includes familiar audio descriptors such as danceability, energy, valence, and loudness, as well as basic musical structure information like key, mode, and time signature. We also retain identifiers (track_ids) needed to merge the tabular data with other modalities. A small number of rows with missing artist or track names (fewer than twenty) are removed to ensure reliable joins in later steps.

3.3. Artist Tier

Predicting a song's performance depends not only on the song itself but also on who the artist is. Well-known artists usually receive more marketing support, better playlist placement, and larger fanbases, which give their songs an advantage even before people listen.

To capture this "artist effect," we use an external dataset containing global listener statistics from Last.fm (up to 2019). From this dataset, we calculate each artist's percentile rank in terms of listeners_lastfm, which serves as a proxy for long-term popularity and audience size. We then convert these percentiles into an ordered tier system ranging from S-tier (top 0.5%) to F-tier (bottom half). In addition, U-tier stands for artists without listener data; they are new or emerging artists after 2019. We can see that even though S tier artists only consist of 0/5% of total artists, they are well over-represented in terms of songs produced, and F-tier artists, despite consists of bottom half artists in terms of popularity, only produces a small percentage of music in our song dataset (Fig. 3)

After merging these tiers into the dataset, we see wide variation across artists. This feature gives the model a straightforward way to account for differences in artist exposure and fanbase size.

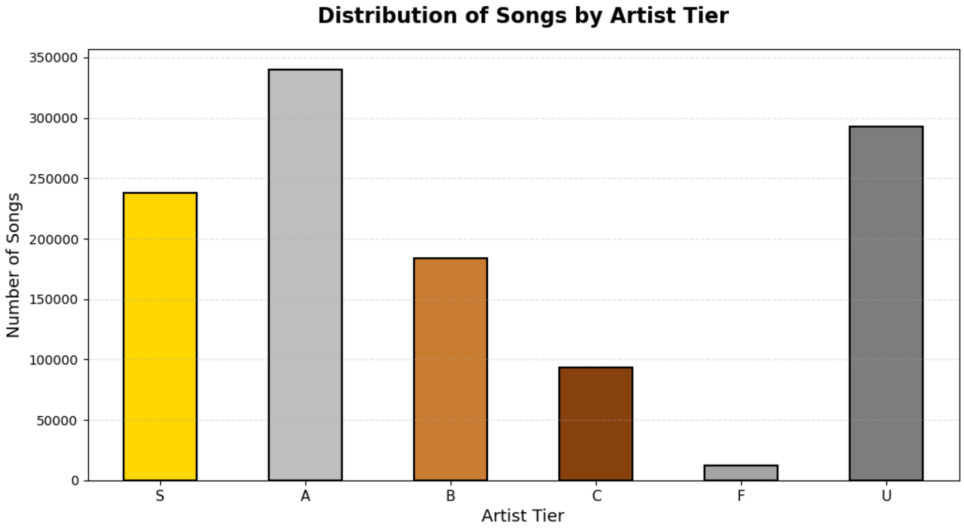


Figure 3. Distribution of songs produced by artists from each tiers

3.4. Artist Historical Popularity

Artist tier gives a broad sense of an artist's reputation, but it does not reflect how an artist has performed on Spotify in recent years. Two artists with the same tier may have very different streaming patterns. To capture these differences, we create an additional feature, artist_prev_pop (i.e., artist previous popularity), which summarizes an artist's averaged past performance up to 2021.

To build this feature, we first select all songs released on or before 2021 (later becoming our train set), which represents the information available when predicting tracks from 2022–2023. For each artist, we compute the average popularity of their earlier songs. This gives us a simple measure of how well the artist has performed historically on Spotify. Artists with a strong past record are more likely to release successful songs again, so giving the model this information provides a meaningful prior before it looks at any audio or visual features.

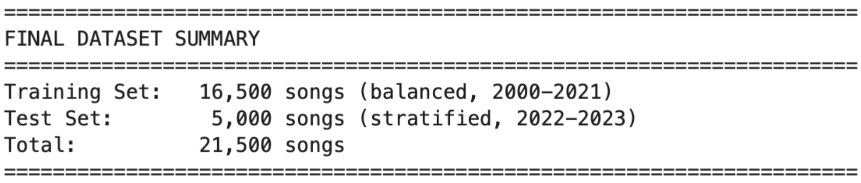
For artists who debut after 2021 and have no history, we do not leave this feature empty. Instead, we assign them the average artist_prev_pop of other artists in the same tier (S/A/B/C/F/U). This gives new artists a reasonable starting point based on their peer group while avoiding missing values in the model. In practice, artist_prev_pop becomes one of the strongest predictors in the tabular model (as shown later in Section 5).

3.5. Train/Test Construction and Balanced Sampling

To align a real forecasting setting, we create the train and test sets using a time-based split rather than random sampling. All songs released between 2000 and 2021 are used for training. The 2022–2023 releases form the test set, which gives us a true out-of-sample evaluation.

This split also shows the rarity of hit songs. Even after applying our year-based popularity labels, hits remain far less common than regular or flop tracks in our training set. Training directly on this skewed distribution would cause the model to learn very little about the minority class, and multimodal data collection adds additional limits on the total number of samples we can process.

To address class imbalance and stay within resource constraints, we build a balanced training set by sampling 5,500 songs from each class (i.e., flop, regular, and hit), resulting in 16,500 training examples. In contrast, for testing, we keep the natural class proportions by drawing a stratified sample of 5,000 songs from the 2022–2023 pool. This preserves real-world frequency patterns and gives us a realistic measure of model performance (Table 1).



```
============================================================
FINAL DATASET SUMMARY
============================================================
Training Set:   16,500 songs (balanced, 2000–2021)
Test Set:        5,000 songs (stratified, 2022–2023)
Total:          21,500 songs
============================================================
```

Table 1. Train/test data summary

4. Multimodal Data Collection Pipeline

In addition to tabular attributes, we enrich each song with several other forms of content: short text descriptions generated from metadata, 30s audio previews, lyrics, and album cover images. These modalities capture information that is not reflected in standard Spotify features such as danceability or valence. The collection process is designed to run reliably at scale while handling API limits and missing data.

4.1. LLM-Generated Text Descriptions

Before collecting the traditional multimodal signals, we first add a textual modality created directly from the tabular data. For each track, we use a large language model to generate a short natural-language template that summarizes the key metadata. This mirrors the idea introduced in the HAIM framework from lecture, where transforming structured data into contextualized text can give models a richer way to understand the information.

For example, a track may be described as follows:

"This song, titled [ʻtrack_name'], is performed by [ʻartist_name'], and was released in the year [ʻyear']. [sentence describing ʻartist_tier']. The track has a danceability score of [ʻdanceability'] (on a scale form 0 to 1, where higher means easier to dance to)..."

The full description will describe the whole tabular data of a son. Each description is then encoded with a DistilBERT encoder, producing dense text embeddings that capture contextual relationships among the tabular features. We then took a step further to use the hit song classification task to fine tune the last layer of DistilBERT, and ended up using the last layer's representation as the embedding for textual description data.

4.2. Audio Previews

To collect audio content for each track, we obtain 30-second previews through the Deezer API. Since Spotify and Deezer have different catalogs, a Spotify track ID does not always match directly to a Deezer track. To increase our success rate, we use a two-step matching approach. We first try direct track-ID matching. When a known mapping exists between a Spotify track ID and a Deezer track, we download the preview associated with that Deezer entry. If the ID match fails, we fall back to title-artist matching, searching Deezer by track name and artist name and selecting the closest metadata match.

Using this strategy, we are able to retrieve audio previews for most tracks in our 21,500-song sample. Tracks for which both matching methods fail simply remain without audio data. All retrieved audio clips are then standardized and encoded with MERT-330M (a SoTA music transformer), producing high-dimensional embeddings that capture features such as timbre, instrumentation, rhythm, and vocal that are not present in the tabular data.

4.3. Lyrics

Lyrics are collected via the AudD API. Coverage is uneven because many songs are instrumental or non-English, and these cases are not always supported. As a result, about 48% of tracks do not have collected lyrics. For the songs without collected lyrics, in order to avoid having missing values, their lyrics is simply the placeholder text *"lyrics: [NO_LYRICS]*", so that the later embedding representation of all non-lyrics songs are still the same non-NA value.For all the song lyrics, we clean and encode the text using DistilBERT encoder producing compact embeddings that capture sentiment, themes, and writing style. Similar to text description data, we tuned the last layer of DistilBERT using the very hit song classification task, and used the fine-tuned last DistilBERT layer as the encoded representation of the lyrics of a song.

4.4. Album Cover Images

Album covers capture useful visual cues related to branding, marketing, and genre identity. We download the corresponding 640×640 cover image from the Spotify API. These images are then processed with a CLIP ViT-B/16 vision transformer, which converts them into visual embeddings that summarize style, color, mood, and other design patterns that may relate to audience targeting.

4.5. Dataset Alignment

All multimodal data are mapped to the tabular dataset using each song's unique track_id. For every song in our initial 21,500-track sample, the pipeline attempts to collect all available modalities: generated text, audio, lyrics, and album-cover images. Lyrics may be missing not randomly for many of them are purely instrumental or non-English. However, audio previews and album covers are required for consistent multimodal modeling, so songs missing either of these are removed.

After this filtering and applying the time-based split, the final multimodal-aligned dataset contains 15,652 training tracks and 4,706 test tracks, including only 28 hits in the test set. Each remaining track has complete tabular, text, audio, and visual embeddings, with lyrics included whenever available.

5. Methodologies, Experiments, and Results

The purpose of this project is to evaluate whether multimodal features improve prediction beyond what can be achieved using tabular data alone. All experiments follow a strict time-based split, training on songs released from 2000 to 2021 and testing on songs released in 2022–2023. Because only twenty-eight test-set tracks are labeled as hits, we also focus on metrics specific to the hit class, which in this case is hit-F1 and hit-AUC, in addition to overall accuracy and AUC.

We begin by training several models using only the tabular features. These include Spotify's audio descriptors as well as engineered variables such as artist tier and artist_prev_pop, which captures each artist's historical performance. We compare four models: Elastic Net, hyperparameter-tuned Random Forest, hyperparameter-tuned XGBoost, and a hyperparameter-tuned feed-forward neural network (FFNN). Their performance on the test set is shown in Fig.4.

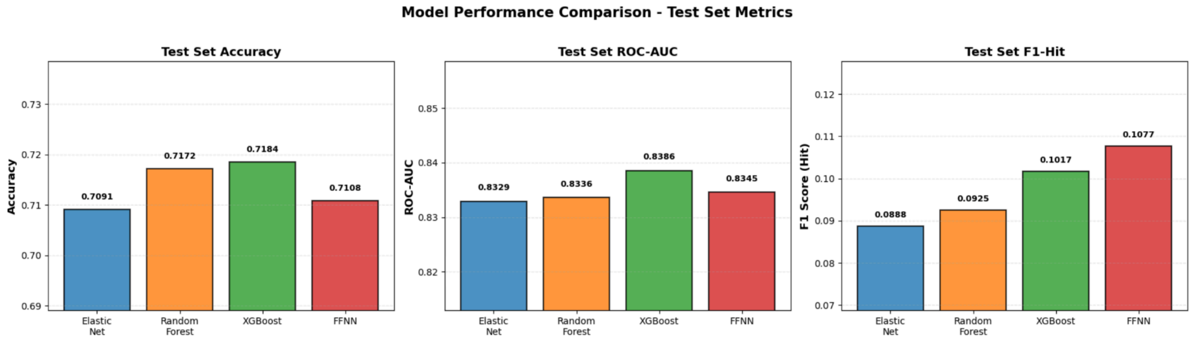


Figure 4: Comparison of four different models on baseline data

Based on the baseline model performance across all evaluated metrics, XGBoost and the Feed-Forward Neural Network (FFNN) emerged as the strongest performers, with XGBoost showing a slight overall advantage. However, we selected the FFNN as our baseline model, and as the foundation for multimodal expansion, for two key reasons. First, the FFNN architecture is GPU-accelerated, enabling substantially faster training and reducing computational burden relative to XGBoost. Second, our multimodal framework requires incorporating additional embedding-based feature sets, and neural networks typically handle high-dimensional embeddings more effectively than tree-based models. As the proportion of embedding features increases with each added modality, the FFNN becomes the more scalable and performance-aligned choice for our extended modeling pipeline.

It is also worth noting that XGBoost identifies artist_prev_pop as the most important feature (Fig. 5), confirming that historical artist popularity is the dominant predictor in the tabular setting. This observation becomes essential in later experiments, where we show how strongly this feature can overshadow multimodal information.

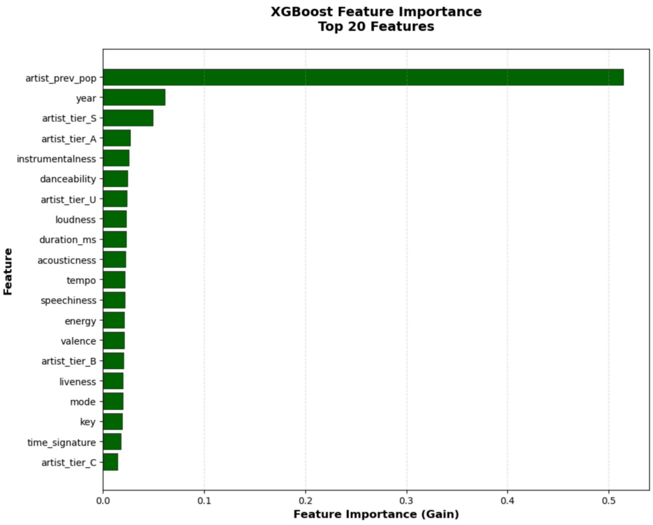


Figure 5. Tabular feature importance from XGBoost model

5.1 Experiment A: Multimodality With Full Tabular Features

Experiment A evaluates whether multimodal features add predictive power when the model already has access to the complete set of tabular variables, including artist_prev_pop. This setup reflects a scenario where the model has strong prior knowledge about an artist's historical performance and can rely on this information when making predictions.

To test the incremental value of multimodal data, we add each modality step by step. We begin with the generated text descriptions created from the tabular metadata. We then add audio embeddings, followed by lyric embeddings for songs with available text. Finally, we append album cover embeddings. This creates a sequence of models that gradually expand the feature set: (1) tabular only; (2) tabular + text; (3) tabular + text + audio; (4) tabular + text + audio + lyrics; (5) tabular + text + audio + lyrics + images.

Even though these multimodal features capture rich information about the sound, meaning, and visual presentation of each track, we find that the model's performance remains almost unchanged (Fig. 6). Accuracy, overall AUC, and hit-AUC fluctuate only slightly across all multimodal configurations. This result suggests that, when artist_prev_pop is included, the model already has a highly informative prior about expected performance. The artist's historical streaming strength dominates the prediction, leaving little room for multimodal content to influence the outcome. In this setting, multimodality adds very little because the tabular features already explain most of the variance.

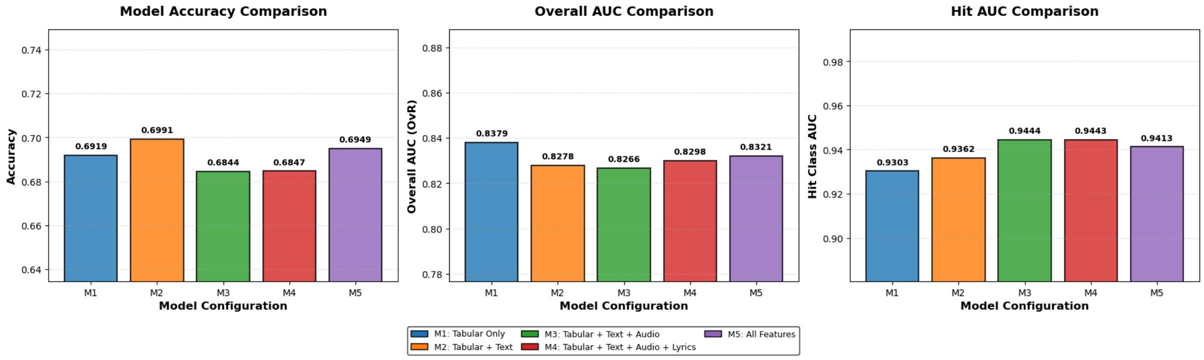


Figure 6. Accuracy, overall AUC, and hit AUC remains level despite multiple modes added

5.2 Experiment B: Removing artist_prev_pop to Reveal Multimodal Value

Experiment B investigates whether the weak performance of multimodal models in Experiment A is caused by the dominant influence of artist_prev_pop. To test this idea, we remove artist_prev_pop from the tabular features and repeat the entire modeling pipeline under the same conditions. Without this feature, the model can no longer rely on past streaming momentum and must instead judge each song based on its content and the remaining metadata.

Once artist_prev_pop is removed, the performance of the tabular-only model drops noticeably across all metrics. This confirms that historical artist popularity was a major source of predictive power in the baseline model. In this more challenging setting, multimodal inputs matter. For every mode we include in the training data, we can see gradually better out-of-sample performance (Fig. 7). The growths are significant for most of the modes, with adding text description, audio, and lyrics having the most significant effect. After adding these multimodal features, the model's performance improves significantly and recovers much of what was lost by removing artist_prev_pop.

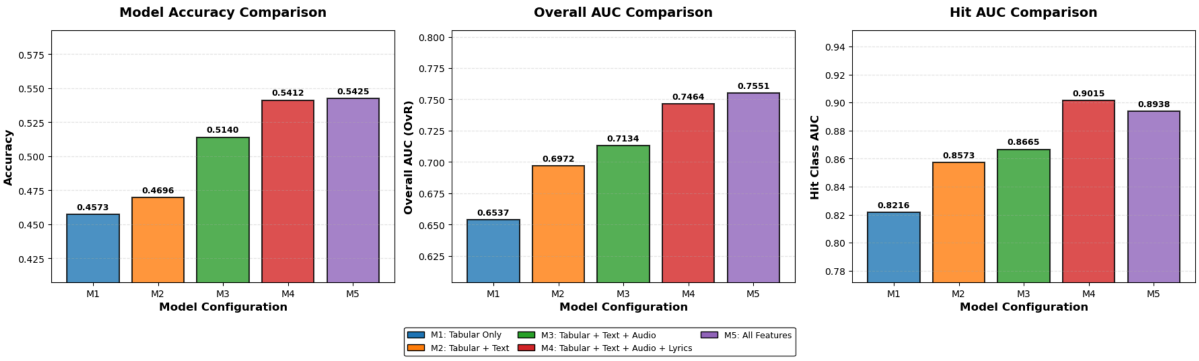


Figure 7. Steady growth in all three metrics with increasing modes for models without artist_prev_pop

An interesting outcome is that even with all five modalities included, the final multimodal model still does not outperform a single tabular baseline that contains the dominant artist_prev_pop feature. This suggests that song success depends heavily on artist reputation, and that multimodal features become most useful only when the artist information is weak or missing.

5.3 Summary of Findings

Together, Experiments A and B show that when the model has access to artist_prev_pop, the tabular features already capture most of the useful predictive signal, so adding multimodal information has almost no effect. However, once this strong history feature is removed, the value of multimodal data becomes much clearer. Generated text, audio, lyrics, and images embeddings help the model regain performance to different extent and better separate hits from non-hits. This suggests that multimodal modeling is most useful in cold-start or discovery settings, where an artist's past popularity is limited or unavailable. When artist history is present, it largely determines the prediction and leaves little room for multimodal features to influence the outcome.

6. Conclusion and Future Steps

This project builds a multimodal framework for predicting Spotify song popularity by combining tabular features with text, audio, lyrics, and album-cover embeddings. Inspired by HAIM's multimodal philosophy, we generate multiple views of each track and evaluate how these views contribute under different information settings. The time-based split and artist-history feature allow us to simulate a real world use case and clearly see when the model relies on artist identity and when it must depend on the content of the song.

Our results show that historical artist popularity is the dominant driver of predictions. When this information is available, multimodal features add little improvements. But when it is removed, multimodal inputs become essential for recovering predictive accuracy. This reflects real streaming dynamics that established artists succeed largely because of reputation, while new artists depend more on the intrinsic qualities of their music.

These insights also apply beyond music. They show that in any prediction task where identity or history plays a major role, multimodal models help most when those signals are weak or missing. This provides exceptional value in scenarios like cold-start recommendation, talent discovery, or early-stage content evaluation where relying on past identity would not be appropriate. Our framework provides a simple way to mix history-based and content-based features across different use cases.

Our multimodal approach provides a flexible way to combine history-based and content-based features across diverse use cases. Given the timeline of this project, there remains substantial room for further exploration. For example, we evaluated only 5 out of the 32 possible combinations of modalities; systematically testing all combinations could reveal which modes contribute the most predictive power while maintaining model efficiency. Another natural extension is to fine-tune additional embedding models. Due to computational constraints, we were able to fine-tune only the NLP model for text descriptions and lyrics, while relying on pretrained embeddings for audio and image data. Future work could meaningfully improve performance by applying transfer learning to music-specific audio embeddings and album-cover image embeddings as well.

Overall, these directions highlight promising pathways for expanding the model's capabilities, and we believe the foundation built in this project offers a strong starting point for deeper multimodal experimentation and real-world applications.

7. Contributions

This project was a collaborative effort. Brian Lu handled preprocessing of the tabular data, engineered features for the baseline model, generated text descriptions, and performed transfer learning for text description and lyrics data. Annie Liu collected and scraped all external data sources, including audio previews, lyrics, and album cover images, and converted them into modality-specific embeddings. Both team members jointly developed the multimodal models, conducted experiments, interpreted results, and co-authored the final report.