ANALYZING THE IMPACT OF AUDIO FEATURES ON SPOTIFY SONG POPULARITY

A Machine Learning Approach to Predicting Song Popularity Based on Audio Attributes

A Project Report Presented in
DATA 443 – Statistical Machine Learning

By

Zoya Malik, Asjad Zubair, Yurii Bezborodov
University of Calgary

Date: December 5, 2025

Department of Mathematics & Statistics
University of Calgary

Contents

1 Introduction

1.1 Introduction and Objective

Music is an art form that connects people around the world. Songs from different cultures, soundtracks from movies and games, classical music, orchestral music, love and passion for music exists globally and is evident when looking at streaming services and concerts. According to Spotify user statistics [1] the streaming service has 713 million monthly active users and is one of the most popular music streaming services available. This shows the importance of music in our lives. Of course, not all of the music available on Spotify is listened to equally. Only so many songs can be considered "popular" and the list of popular songs is constantly changing. Knowing this, we set out to determine whether the audio features of a song can be used to predict its popularity.

It is clear to us that there are many external factors that influence how popular a song becomes. The popularity of the artist themselves, marketing of the music, collaborating projects, etc. all have a strong influence on how many people will listen to a song. The objective of our project is not to predict what songs will go viral, but to examine whether measurable audio properties allow meaningful prediction of how popular a song will become. This project aims to quantify the relationship between audio features and song popularity using linear and non-linear models, and determine if predictions can be reasonably made.

1.2 Dataset Description

For this project we are using a dataset that contains over 80 thousand tracks spanning across 125 different genres [2]. Before any data cleaning or preprocessing, the raw dataset has 114,000 rows and 21 columns. These columns include audio features (e.g. danceability, loudness, acousticness, etc.), popularity (0-100), and categorical metadata (e.g. artists, genre, key, etc.). The dataset also contains duplicates, extreme outliers, and heavily skewed data.

1.3 Research Question

The primary focus of this project is to answer the question: **How accurately can we predict a song's popularity using its audio attributes?** While we explore the answer to this question we will also be looking at which audio features are the strongest predictors of probability, and whether the strongest predictors differ across genres.

1.4 Data Cleaning

As mentioned earlier, the raw dataset has 114,000 rows. The obvious steps to clean the data and prepare it for exploratory data analysis is to check for missing values and duplicates as these add noise to our analysis. The dataset mentions that duplicate tracks can exist, but it is also possible for different songs to have the same name. To solve for this, duplicate tracks were detected using the track name, duration, and artists. After applying these filters, one row was dropped for having missing values, and over 30,000 rows were dropped for being duplicate tracks. Once this initial cleaning was done, the data was ready for initial exploratory analysis to help guide our preprocessing.

2 Exploratory Data Analysis

2.1 Summary Statistics and Correlations

We first inspected the summary statistics (Figure 1) to check for reasonable ranges and identifying potential outliers. The correlation analysis (Figure 2) revealed that no single feature strongly predicts popularity linearly; *instrumentalness* had the strongest negative correlation, while *loudness* showed a weak positive correlation.

```
Numerical columns for analysis: ['popularity', 'duration_ms', 'danceabilit
y', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'ins
trumentalness', 'liveness', 'valence', 'tempo', 'time_signature']
Summary Statistics:
                         mean            std       min            50%  \
popularity          34.720996      19.526807     0.000      35.000000
duration_ms       231276.522727  115830.598763  8586.000  215239.000000
danceability           0.559441       0.177296     0.000       0.573000
energy                 0.634864       0.258436     0.000       0.678000
key                    5.284096       3.557798     0.000       5.000000
loudness              -8.587417       5.284652   -49.531      -7.264500
mode                   0.633808       0.481766     0.000       1.000000
speechiness            0.088521       0.115810     0.000       0.049000
acousticness           0.329985       0.339976     0.000       0.191000
instrumentalness       0.182595       0.330123     0.000       0.000083
liveness               0.219429       0.197750     0.000       0.133000
valence                0.464983       0.263411     0.000       0.451000
tempo                122.149453      30.094001     0.000     122.027000
time_signature         3.897224       0.455136     0.000       4.000000

                          max
popularity            100.000
duration_ms       5237295.000
danceability            0.985
energy                  1.000
key                    11.000
loudness                4.532
mode                    1.000
speechiness             0.965
acousticness            0.996
instrumentalness        1.000
liveness                1.000
valence                 0.995
tempo                 243.372
time_signature          5.000
```

Figure 1: Summary statistics for numerical audio features.

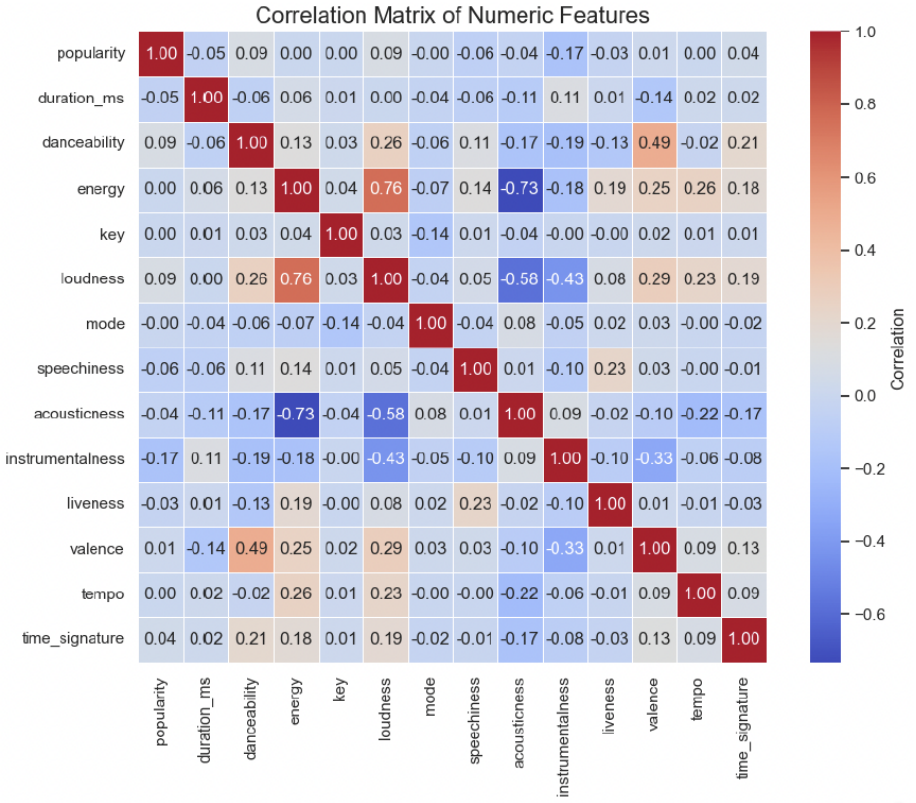

Figure 2: Correlation matrix heatmap showing weak linear relationships with popularity.

2.2 Distributions and Outliers

Boxplots of the audio features (Figure 3) highlight the presence of outliers, particularly in *duration*, and the skewness of features like *acousticness*. The popularity distribution itself (Figure 4) is heavily clustered between 20 and 60, with very few tracks achieving "hit" status (> 80).

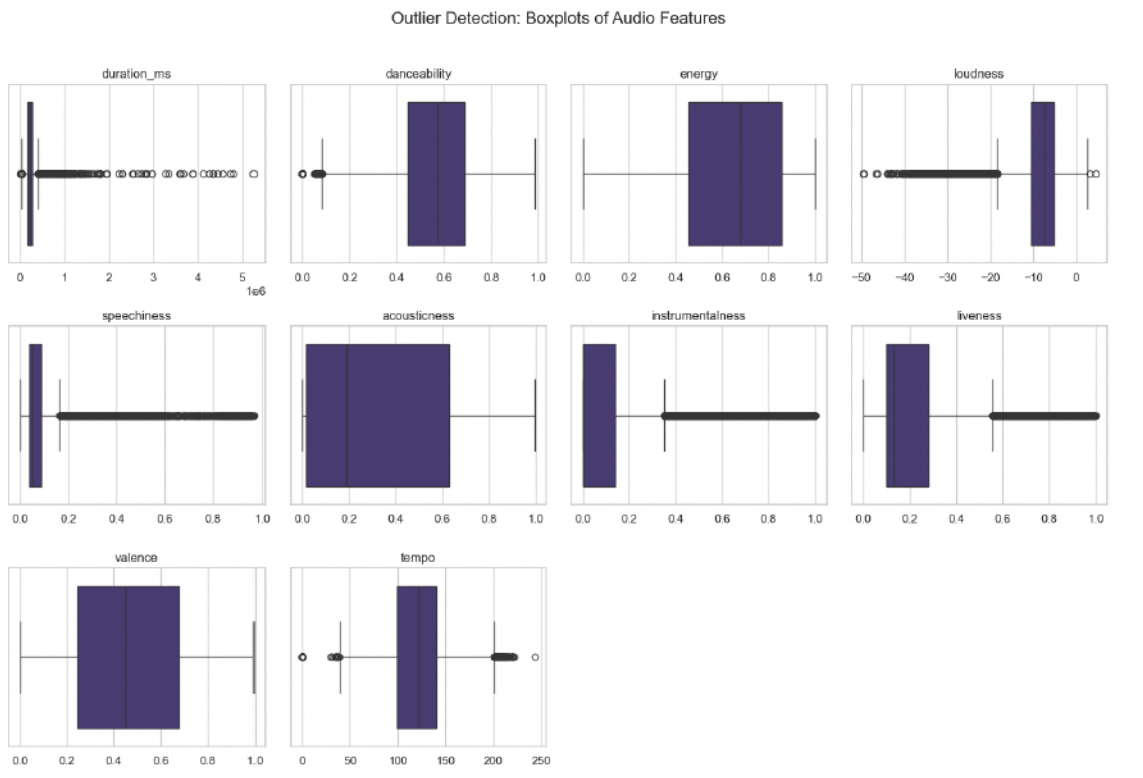

Figure 3: Boxplots of key audio features illustrating skew and outliers.

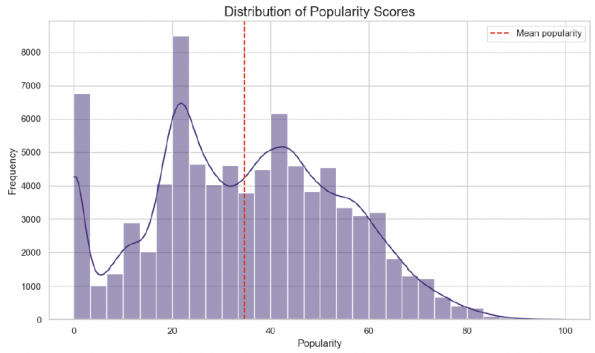

Figure 4: Distribution of Popularity Scores.

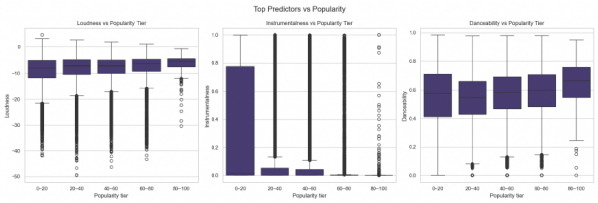

Figure 5: Top Predictors vs. Popularity Tier.

2.3 Genre Effects and Interactions

Genre plays a significant role in baseline popularity (Figure 6), with pop and metal scoring higher on average. Furthermore, we observed an interaction effect: the relationship between *loudness* and popularity changes direction depending on the genre (Figure 7), motivating the use of interaction terms in our models.

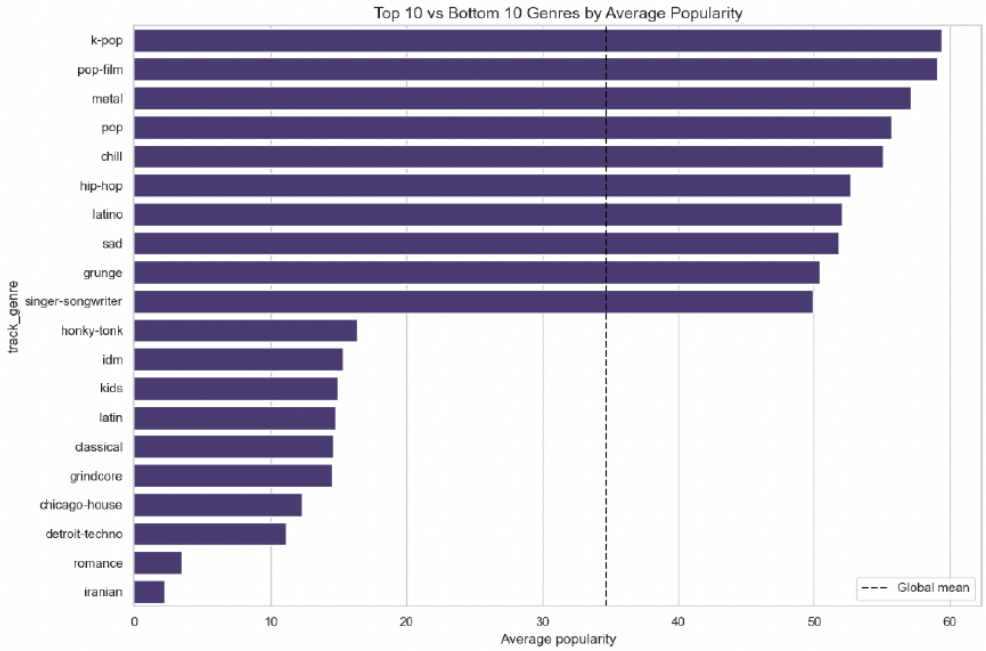

Figure 6: Top and Bottom 10 Genres by Average Popularity.

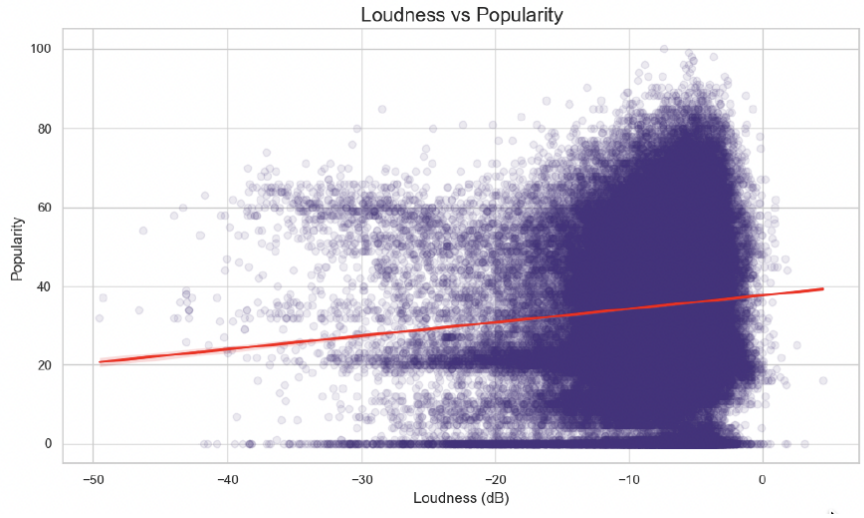

Figure 7: Interaction: Loudness predicts popularity differently across genres (Acoustic vs Dance vs Metal).

2.4 Preprocessing Steps

As mentioned, our data analysis showed that the audio features were skewed to various degrees, but the popularity data was not heavily skewed. Due to this, we decided to bin the popularity data into tiers and let the models auto balance classes, allowing for a more stable target variable. We also found that there were tracks with extreme values for duration and unrealistic values for tempo. Filtering outliers that are shorter than 30 seconds or longer than 20 minutes, or tracks with a tempo of 0, would help normalize our data and allow us to stick to songs while avoiding tracks such as “rain noise”. Applying these filters led to an additional 228 rows of data being dropped. Now that our dataset contained

"normalized" songs, we had to deal with skewed distributions to avoid our models training with bias. For features that were highly skewed (instrumentalness, acousticness, speechiness, liveness) the best way to achieve a normal distribution was to use a 'PowerTransformer' with the Yeo Johnson method [3] to approximate a normal distribution. Other features that are still bounded between 0 and 1 but are less skewed (danceability, energy, valence) could be normalized using the MinMaxScaler. These features do not possess an extreme enough skew to bias our models, but are skewed enough to add meaning to the data. Using a PowerTransformer here could potentially cause us to lose context for our models as the MinMaxScaler preserves shape, skew, and relative distances. Features that are unbounded (loudness, tempo, duration) were normalized using a StandardScaler to achieve normal distribution. Since "track_genre", "key", and "time_signature" are categorical features, we used OneHotEncoder to encode them which resulted in a significant increase in the number of binary features. Our exploratory analysis also showed that the relationship between "loudness" and "popularity" varied by genre which is something we want to explore further. However, machine learning models do not usually create or see these interactions automatically. Instead, we explicitly defined some notable interactions (loudness x acoustic, heavy metal, dance) which allows our machine learning models to learn the slopes of these interactions. The result of these preprocessing steps was a feature matrix with over 82,000 rows and 16 columns, ready to be split and used for training and validating machine learning models.

3 Machine Learning Modeling

As our exploratory data analysis shows, this dataset has weak correlations and no clear linear structure in our plots. Given this we thought it would be best to use a logistic regression model, to see how a linear classifier handles non-linear data. To have a comparison, we decided to also use a support vector machine (SVM) with an RBF kernel. Logistic Regression assumes decision boundaries between classes are linear, is fast to train, and easy to interpret, making it an ideal baseline for our project. On the other hand, a SVM with an RBF kernel allows complex, curved decision boundaries. This model is more appropriate for our project given what our initial analysis revealed, and is expected to perform better than the logistic regression model. Using both models allows us to definitively determine whether there is linearity in the relationship between audio features and popularity. If there is not much linearity in these relationships, then comparing the two models will tell us whether flexible decision boundaries result in a significant increase in prediction accuracy. Our primary indicator of model performance is the macro-f1 score because it computes the f1 score per class and averages them, allowing all classes to be represented equally. This is critical for our project due to the number of songs in the higher popularity bins being noticeably less than the number of songs in other bins. Upon running our models the most immediate and noticeable difference was runtime. While the logistic regression model completed in under a minute, the SVM model took over 5 hours to complete. This is partly due to the fact that the RBF kernel computes similarity between every pair of training samples. Over 82,000 songs split into a test size of over 66,000 means the kernel has over 4 billion pairwise computations to make. Upon using GridSearch to identify the best parameters for each model, we found that for logistic regression the best parameters are {`'model__C': 0.1, 'model__penalty': 'l2'`}. The low C value means the model required stronger regularization, suggesting the data may be noisy, while the l2 penalty helps with generalization. The SVM model performed best with the parameters {`'model__C': 10.0, 'model__gamma': 'scale'`}. The high C value in this case means more complex boundaries, indicating that the data is not linearly separable and requires a more flexible boundary for increased performance.

4 Results and Interpretation

4.1 Overall Performance

The SVM with RBF kernel consistently outperformed the Logistic Regression baseline across all aggregate metrics (Table 1 and Figure 8). The improvement confirms the presence of non-linear boundaries, although the overall accuracy remains moderate ($\approx 62\%$).

Table 1: Overall Performance Metrics

Model	Accuracy	Macro F1	Weighted F1	Macro Recall
Logistic Regression	0.570	0.471	0.598	0.589
SVM (RBF Kernel)	0.624	0.518	0.636	0.552

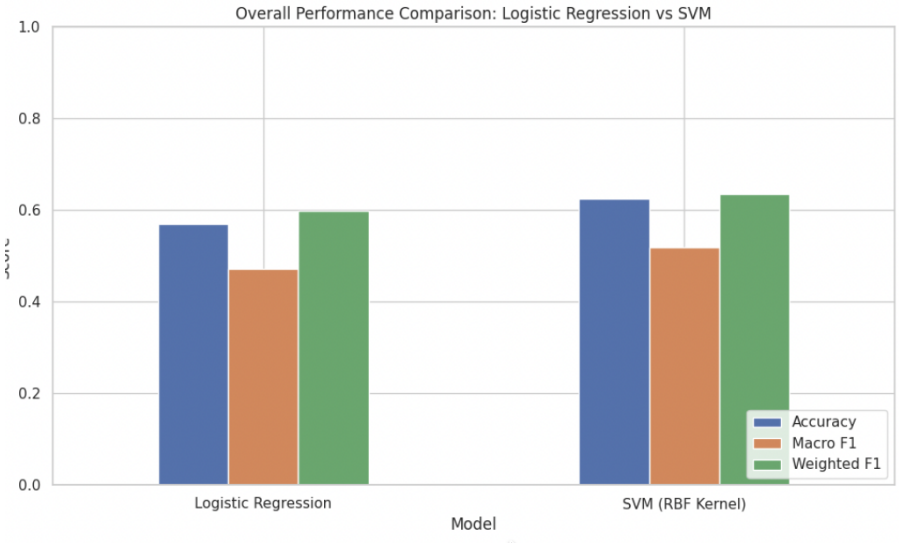

Figure 8: Overall performance comparison: SVM vs. Logistic Regression.

4.2 Class-wise Analysis

The confusion matrices (Figure 9) and class-wise F1 scores (Table 2) reveal that both models struggle significantly with the highest popularity tier (80–100). While SVM improves performance in the mid-range tiers, the F1 score for the top tier remains low (0.166), indicating that audio features alone are insufficient to distinguish global hits.

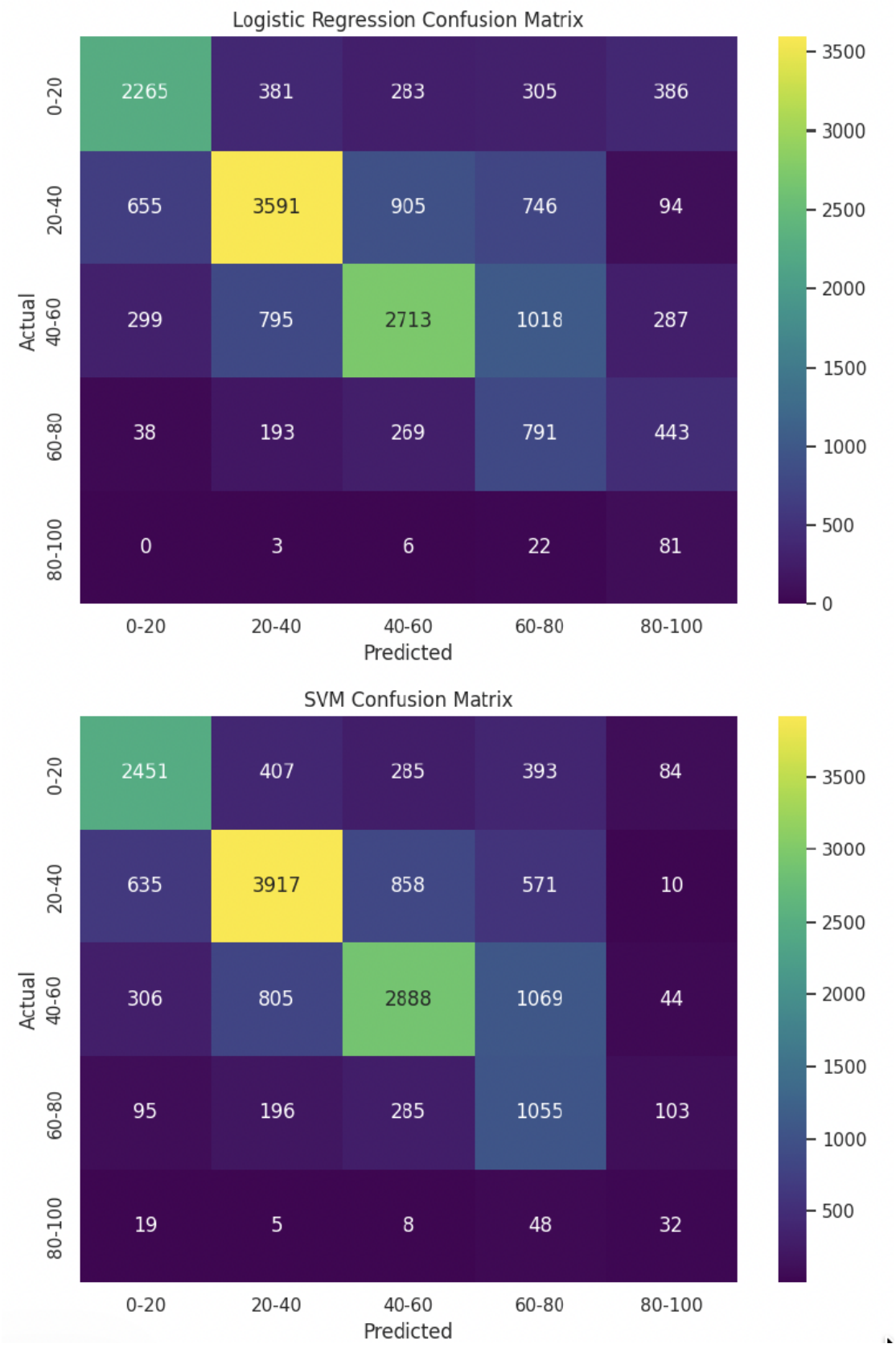

Figure 9: Confusion Matrices. Note the high misclassification in the 80–100 column.

Table 2: Class-wise F1 Scores

Popularity Tier	LogReg F1	SVM F1
0–20	0.6587	0.6879
20–40	0.6556	0.6919
40–60	0.5841	0.6121
60–80	0.3427	0.4332
80–100	0.1154	0.1662

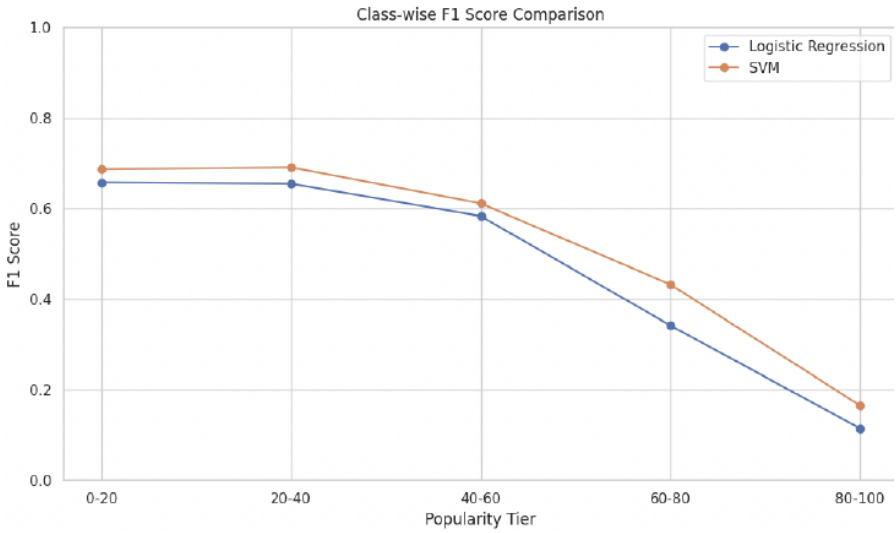

Figure 10: Class-wise F1 Score Trend.

4.3 Error Analysis

To better understand where the models fail, we examined the misclassifications shown in the confusion matrices. Both models tend to make "near miss" errors, where songs are predicted into an adjacent popularity tier rather than an entirely unrelated one. This pattern indicates that the models are capturing a coarse structure in the data but struggle with finer distinctions.

A more important observation is that misclassification rates increase sharply for the 60–80 and 80–100 tiers. Songs in these ranges share similar audio profiles with mid-tier songs, making them difficult to separate based on acoustic data alone. Even the SVM, which can learn flexible nonlinear boundaries, fails to recover a distinct decision boundary for top-tier tracks. This aligns with the intuition that the highest popularity tier is governed more by external factors (artist fame, marketing, virality) than by measurable audio properties.

Overall, the error structure reinforces that audio features provide moderate predictive power for broad groupings, but not for distinguishing hit songs.

4.4 Feature Importance Discussion

Although SVMs do not provide traditional feature importances, we can examine Logistic Regression coefficients and EDA trends to understand which features contribute most to the predictions. Features

such as loudness, energy, and danceability show weak but consistent positive associations with popularity, while instrumentalness displays a strong negative association. These trends align with the summary statistics and correlation heatmaps observed earlier.

However, the weak magnitude of all coefficients highlights the same conclusion reflected in model performance: no single audio feature exerts a dominant influence on popularity. Patterns are subtle, diffuse, and nonlinear, which explains why the SVM achieved higher performance than Logistic Regression.

4.5 Model Limitations

The overall performance of the models suggests inherent limitations in predicting popularity directly from audio features. Even with substantial preprocessing, normalization, and the use of nonlinear models, the predictive ceiling remains low for the highest popularity tier. This limitation arises from two factors: (1) the imbalance of the dataset, with far fewer hit songs than typical tracks, and (2) the absence of external metadata that strongly influences real-world popularity.

These limitations clarify that while audio-based models can detect general popularity structure, audio alone is insufficient to capture the complex mechanisms behind highly successful songs.

5 Future Steps

Based on our current findings, several ideas could further improve our ability to model and understand song popularity. First, we would extend the feature space by incorporating metadata such as artist popularity, release year, playlist placement, label information, and social/virality signals where possible (e.g., from charts or external APIs). This would allow us to explicitly test how much incremental predictive power comes from non-audio context compared to strictly acoustic features.

Also, future work could deepen our treatment of time and user behavior. Popularity is dynamic, so incorporating temporal features (e.g., weekly stream counts, release-to-peak time, decay curves) and experimenting with time-aware models could help capture rise-and-fall patterns that static snapshots miss. If feasible, we could also explore aggregate listener behavior (skips, saves, playlist adds) to better approximate engagement rather than relying solely on a single popularity score. Finally, we could prototype simple interfaces or dashboards, so the model isn't just for analysis, but also helps people make decisions and gets better over time with real user feedback.

6 Conclusion

"Can audio features be used to accurately predict a song's popularity, and which features contribute most?" Our results show that audio features do contain useful information, especially for predicting broad popularity tiers. SVM, which can capture nonlinear patterns in the data, performed the best and were able to classify songs in the 0–60 popularity range with moderate accuracy. This suggests that combinations of basic audio characteristics (such as energy, danceability, and tempo) are meaningfully related to whether a song is generally unpopular, moderately popular, or somewhat successful. However, our models struggled with the highest popularity tier (80–100). Even the best model could not reliably separate "hit" songs from more typical tracks. This indicates that audio features alone are not enough to explain extreme popularity. Factors such as artist fame, marketing, social media trends, and playlist placement likely play a much larger role when it comes to breakout hits. Overall, we can say that audio-only models are helpful for coarse-grained prediction and for understanding broad patterns, but

they are limited when we care about the very top of the charts. To better model top-tier success, future work should include extra metadata (artist, release context, exposure signals) and possibly multimodal approaches that combine audio with external information.

References

[1] Backlinko Team. 2025. *Spotify User Stats (Updated September 2025)*. Backlinko. Retrieved December 1, 2025 from https://backlinko.com/spotify-users

[2] MaharshiPandya. 2022. *Spotify Tracks Dataset*. Kaggle. Retrieved December 1, 2025 from https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset/data

[3] *PowerTransformer — scikit-learn 1.7.2 documentation*. scikit-learn. Retrieved November 27, 2025 from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html