Spotify Track Popularity Predicition

Kevin Alvarez Will Burns Jenna Jabourian Connor Perrone
Gabe Sanchez

December 2, 2025

Abstract

Blah Blah Blah

1 Introduction

The purpose of this project is to analyze the Spotify track data set and develop a model capable of predicting the popularity of the track using various features. The data set contains numerical and categorical attributes that describe musical characteristics such as danceability, loudness, energy, tempo, key, and valence. Our goal is to explore which features are the most strongly correlated with popularity and to compare different predictive methods in terms of accuracy, interpret-ability, and robustness.

2 Results

Blah Blah Blah

3 Discussion

Blah Blah Blah

A Methods

A.1 Exploratory Data Analysis

For our exploratory data analysis (EDA), we began by examining the structure of the Spotify dataset, including the number of observations, feature types, and any missing or duplicated entries. We generate summary statistics for numerical features to understand their ranges, distributions, and potential outliers.

As part of our exploratory data analysis, we generated histogram plots for several of the major audio features in the dataset, including danceability, energy, loudness, valence, and tempo. These histograms (Figure 1) helped us understand how each feature is distributed across the dataset and revealed that many musical characteristics have non-uniform or skewed distributions. Understanding these distributions is important for constructing a similarity-based recommendation system, as features with narrow or concentrated ranges may have less discriminative power when comparing tracks.

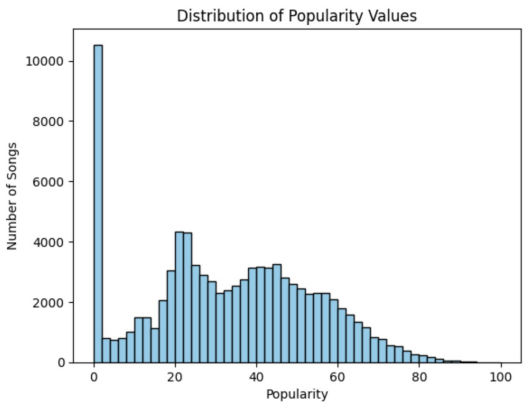

Figure 1: Histogram distributions of selected numerical audio features in the Spotify dataset.

To better understand how listener preferences vary across musical categories, we examined the relationship between genre and popularity. The bar chart in Figure 2 shows the average popularity for each genre represented in the dataset. This visualization highlights which genres tend to receive higher user engagement on Spotify and also illustrates the uneven distribution of popularity across categories. Observing this imbalance further supports our decision to exclude popularity as a feature for recommendation, as it reflects platform-wide trends rather than individual user taste.

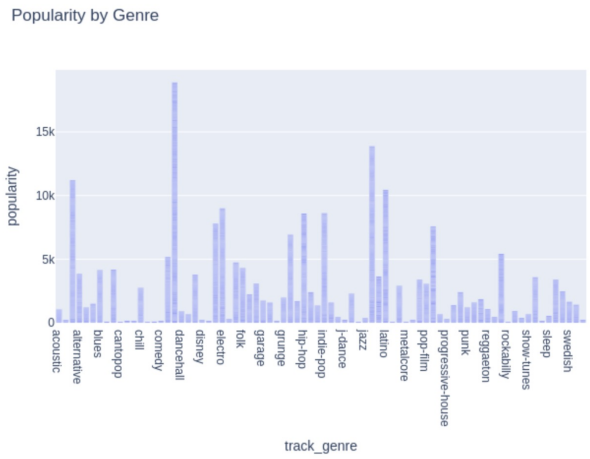

Figure 2: Relationship between Genre and Popularity.

To investigate how the audio features relate to one another, we generated a correlation heatmap, shown in Figure 3. This visualization highlights the strength and direction of pairwise relationships between numerical features in the dataset. Several strong correlations are immediately apparent, such as the positive relationship between loudness and energy, as well as between valence and danceability. These patterns indicate that certain musical characteristics tend to co-occur across tracks. Understanding these relationships is important for a similarity-based recommendation system, since highly correlated features contribute similar information when comparing songs.

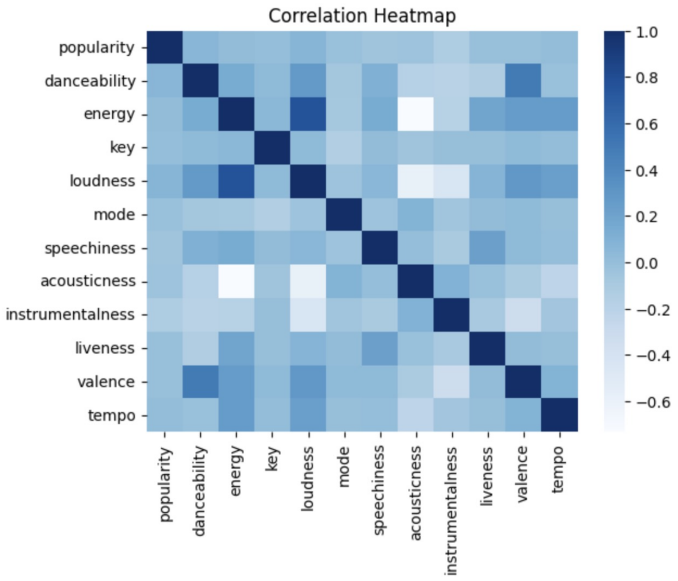

Figure 3: Correlation heatmap of numerical Spotify audio features.

To further explore the relationships identified in the correlation heatmap, we created scatterplots for pairs of features that exhibited strong positive correlations. Figures 4 and 5 show two such examples: loudness versus energy, and valence versus danceability. Both plots reveal clear upward trends, indicating that tracks with higher loudness generally have higher energy levels, and songs with greater valence tend to be more danceable. These visual patterns reinforce the idea that certain audio characteristics naturally cluster together, which is valuable information when measuring similarity between songs in a recommendation system.

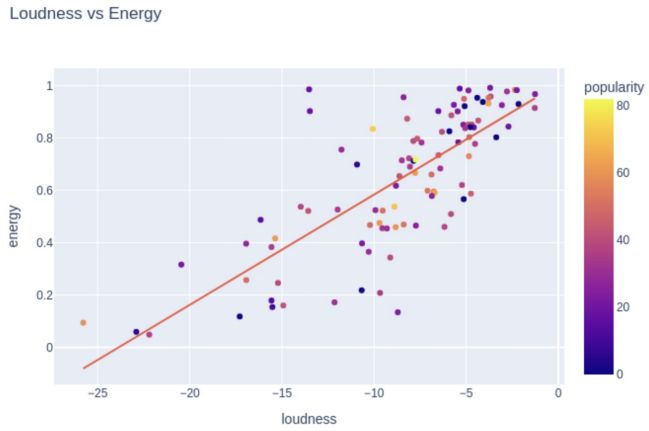

Figure 4: Scatterplot of Loudness versus Energy.

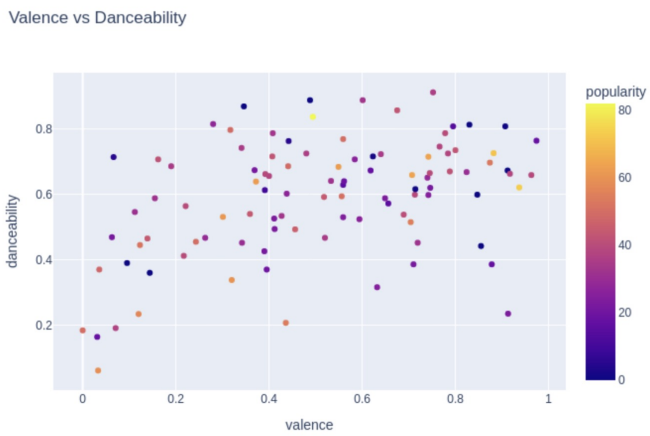

Figure 5: Scatterplot of Valance verses Danceability.

A.2 Data Pre-processing and Feature Engineering

Before building our recommendation system, we performed several preprocessing steps to ensure that the data set was clean, consistent, and suitable for feature-based similarity comparisons. We started by checking for duplicated songs using the `track_id` column. The data set contained a substantial number of duplicates, so we removed all repeated entries by keeping only the first occurrence of each unique `track_id`. This reduced the data set from 114,000 rows to 89,741 unique tracks, ensuring that no song was overly tagged during model development.

Listing 1: Duplicate detection and cleanup

```
dupe_mask = df['track_id'].duplicated(keep=False)
dupes = df[dupe_mask].sort_values('track_id')
df = df.drop_duplicates(subset='track_id', keep='first')
```

Next, we removed columns that did not contribute meaningful musical information. Specifically, we dropped the `track_id` and `Unnamed: 0` columns, since they are identifier fields rather than audio or metadata features and therefore cannot be used to compute similarity between songs.

Listing 2: Column Removal

```
df = df.drop('track_id', axis=1)
df = df.drop('Unnamed: 0', axis=1)
```

After cleaning the dataset, we evaluated the remaining features for their suitability in a recommendation setting. Since popularity does not reflect a user's personal preference and is computed using an opaque algorithm specific to Spotify, we excluded this column from all downstream analysis. This ensures that recommendations are based purely on the audio characteristics of songs rather than global listening trends.

Although our project did not require creating new engineered features, the preprocessing steps above established a consistent and reliable feature set that will later be standardized.

A.3 Regression Analysis

In order to understand how individual audio features contribute to the overall musical characteristics of a track, we applied regression analysis using `energy` as the target variable. Energy is a continuous numeric feature that reflects the perceptual intensity of a song, making it a good candidate for examining linear relationships in the dataset.

We first computed the correlation between energy and the other numerical features to identify potential predictors. Loudness showed the strongest correlation with energy, followed by danceability and valence. These correlations provided an initial indication that a linear model might capture part of the relationship between audio features and energy.

To quantify these relationships, we trained a multiple linear regression model using seven audio features as predictors. The model was trained on 80% of the data and evaluated on the remaining 20%. The resulting validation metrics indicated that the linear model explained a moderate portion of the variance in energy, with R^2, MAE, and RMSE values suggesting that the model was capturing general trends but not all complexities of the feature space. Figure 6 shows a scatterplot comparing the model's predicted energy values to the actual values in the validation set. The clustering around the diagonal line indicates reasonable predictive performance, though with visible deviation reflecting non-linear structure in the data.

We also examined the learned regression coefficients to interpret feature importance. The magnitude and sign of the coefficients aligned with the earlier correlation analysis: loudness had the strongest positive effect on energy, supporting the idea that louder tracks tend to feel more energetic.

To evaluate whether regularization was necessary, we trained a Ridge regression model with an α value of 2.5. Ridge regression slightly reduced overfitting, bringing the training and validation R^2 scores closer together. However, the performance improvements were small, indicating that multicollinearity among our selected features was present but not severe. This suggests that while regularization can stabilize the model, the linear feature set did not require heavy penalization to perform effectively.

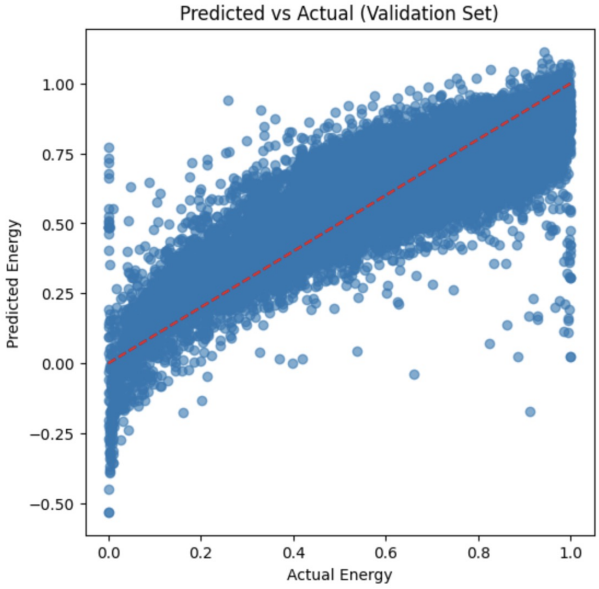

Figure 6: Predicted versus actual energy values for the linear regression model on the validation set. The diagonal dashed line represents perfect predictions.

A.4 Logistic Regression

To further analyze classification patterns within the dataset, we applied logistic regression to predict whether a song contains explicit content. The target variable `explicit` is binary, making logistic regression a natural choice for this task. We used eight audio and metadata features, including popularity, danceability, loudness, energy, instrumentalness, speechiness, valence, and tempo, as predictors.

A key challenge in this task is that explicit songs make up a small minority of the data set, resulting in a class imbalance. To address this, we assigned higher weights to the minority (explicit) class during training, allowing the model to better identify explicit tracks without being overwhelmed by the majority class.

After training the model in an 80/20 train–validation split, we evaluated performance using a confusion matrix (Figure 7), classification report metrics and general prediction accuracy. The confusion matrix revealed that the model correctly identified a large proportion of non-explicit songs while achieving reasonable true positive performance on explicit songs, despite the imbalance. This demonstrates that class weighting improved the model's sensitivity to explicit content.

To further assess predictive quality, we examined the model's ROC curve and computed the area under the curve (AUC), shown in Figure 8. The resulting AUC score indicated that the logistic model

performed substantially better than random guessing and was effective at separating explicit from non-explicit songs based on the selected features.

Finally, we performed 5-fold stratified cross-validation to verify the stability of the model. The AUC and accuracy scores across folds remained consistent, indicating that the model generalizes reliably and is not overfitting. Regularization did not significantly impact performance, suggesting that multicollinearity among the selected features was limited, and the logistic model remained stable without heavy penalization, similar to Linear Regression.

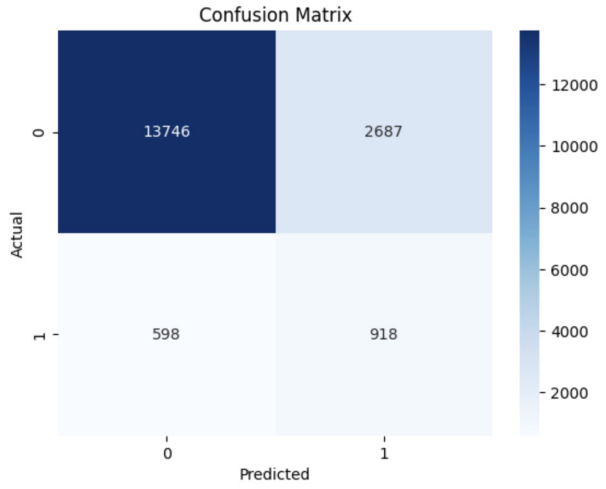

Figure 7: Confusion matrix for logistic regression model predicting explicit content.

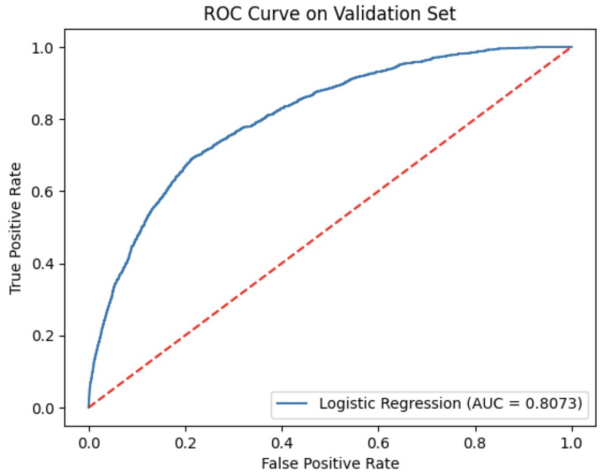

Figure 8: ROC curve for logistic regression on the validation set. The dashed line represents random guessing.

A.5 KNN, Decision Trees, and Random Forest

KNN + DECISION TREE + RANDOM FOREST

A.6 PCA and Clustering

To explore groupings and latent structure within the Spotify dataset, we applied Principal Component Analysis (PCA) and K-Means clustering. Since clustering algorithms are sensitive to feature scale, we first standardized all numerical predictor variables using a `StandardScaler`. This ensured that features measured on different scales contributed equally to the distance computations used by K-Means.

We then applied the K-Means algorithm to the standardized feature set. Because the optimal number of clusters is not known beforehand, we used two diagnostic tools: the elbow method and silhouette scores. The elbow plot (Figure 9) shows the distortion, or within-cluster sum of squares, for values of k ranging from 2 to 10. The point at which the curve begins to flatten—commonly referred to as the "elbow" suggests a suitable choice for k. Similarly, the silhouette score plot (Figure 10) measures how well-separated the clusters are for each value of k. Higher silhouette scores indicate more coherent and better-defined clusters.

Both diagnostics suggested that $k = 6$ provided a good balance between low distortion and high separation quality. After selecting $k = 6$, we fit the final K-Means model and computed cluster assignments for all standardized samples. Examining the cluster sizes confirmed that no cluster was excessively small or disproportionately large.

To visualize the clusters in two dimensions, we applied PCA to the standardized feature matrix and extracted the first two principal components, which captured the largest amount of variance. Figure 11 shows a 2D scatterplot of the samples in PCA space, colored according to their cluster label. Although PCA reduces the dimensionality and may not retain all information, the plot reveals meaningful separation between several clusters, indicating that the underlying audio features contain structure that can be grouped into musically distinct categories.

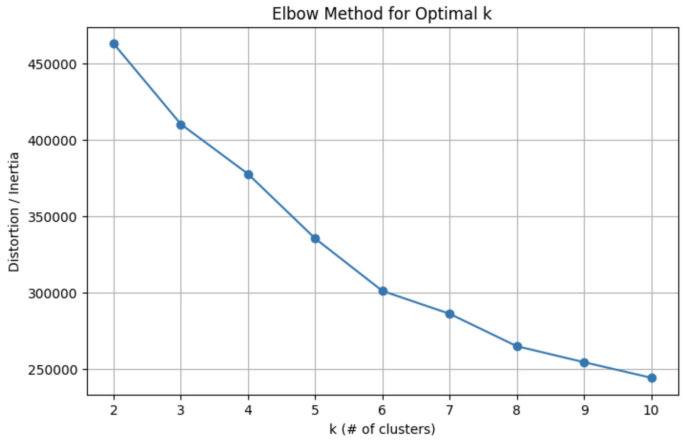

Figure 9: Elbow method showing distortion values for k between 2 and 10. The point where the curve begins to flatten suggests a suitable number of clusters.

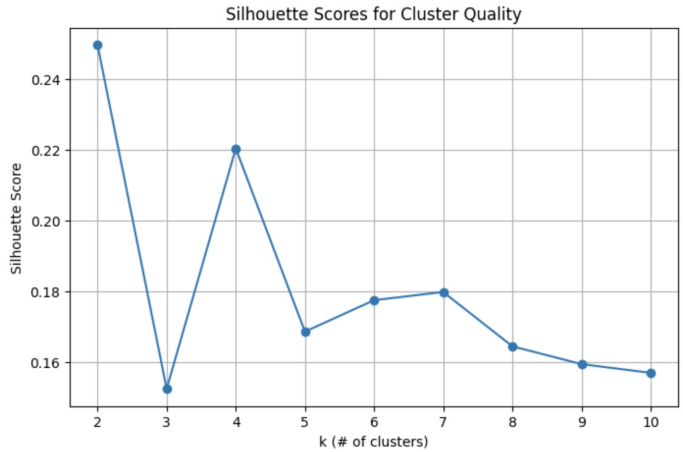

Figure 10: Silhouette scores for k between 2 and 10. Higher values indicate more well-defined clusters.

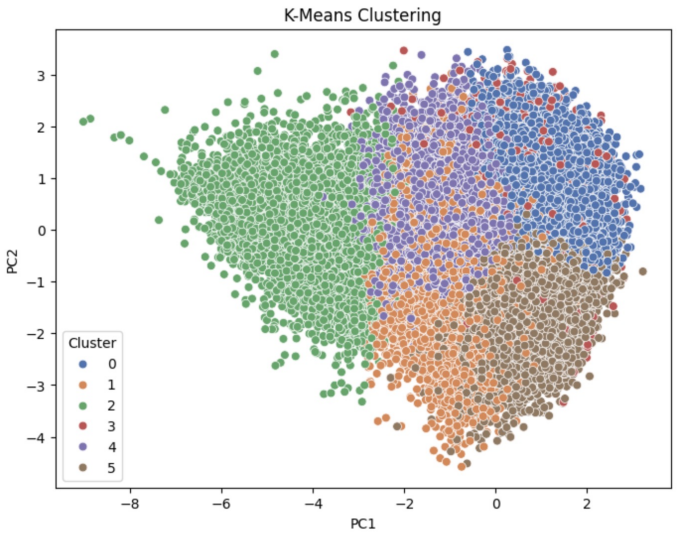

Figure 11: K-Means clustering visualized in the first two principal components. Colors represent the six cluster assignments.

A.7 Neural Network Experiments

NEURAL NETWORKS

A.8 Hyperparameter Tuning

HYPERPARAMETER TUNING