Spotify Popularity Prediction

Kaidi Chen (kc3819), Tiffany Cheng (yc4554), Hasong Cho (hc3523), Hailey Chung (hc3539), Jiayun Yin (jy3445)

I. Introduction

Spotify is a leading global music streaming platform operating on a “Freemium + Premium” model. Free users access music with ads and limited controls, while premium subscribers enjoy ad-free, high-quality audio with offline playback, generating revenue through subscriptions. With a 30-33% global market share as of 2024, Spotify dominates the digital music streaming industry, particularly in North America and Europe. Its vast music library, built through direct agreements with labels and artists, caters to diverse preferences. Our research explores how key musical features, such as tempo, energy, and danceability, have evolved over time, comparing trends across different eras. By analyzing these attributes and their relationships, we aim to uncover shifts in production styles and listener preferences. This research can help Spotify refine song promotion, enhance user engagement, and optimize playlist curation, aligning with its vision of delivering a personalized and superior music experience.

II. Data Collection

Our research uses the 30,000 Spotify Songs dataset from Kaggle, pre-cleaned by the authors of the spotify package. The data, sourced via the Spotify API, includes attributes that provide insights into each track’s musical structure. It categorizes the attributes into four key aspects. **Song Information** includes details such as track_id, track_name, track_artist, and track_popularity, capturing the identity and popularity of each track. **Album Information**, including track_album_id, track_album_name, and track_album_release_date, provides context about the albums the songs belong to. **Playlist Information** features attributes like playlist_name, playlist_id, playlist_genre, and playlist_subgenre, reflecting how tracks are grouped on Spotify. Lastly, **Music Features** encompass technical and perceptual characteristics such as danceability, energy, key, loudness, and more, offering deeper insights into each song’s structure and mood. We cleaned the data by removing null values, duplicates, and entries marked as “N/A” to ensure completeness and consistency. Additionally, we filtered out songs with a duration of less than 60 seconds, focusing on full tracks suitable for analysis.

III. Exploratory Analysis

<u>Music Popularity Trend Analysis:</u>

For the exploratory analysis, we examined the comparison of popular music genres and the evolving trends in music feature popularity over time. Popularity was defined as music with a popularity score in the top 25th percentile. Given that most of the dataset are from 2015 to 2019, we focused our analysis on this period. The trends observed in the data highlight notable changes across 11 music features. Features such as danceability, valence, speechiness, tempo, and energy showed an upward trend, indicating a growing preference for music that is lively, upbeat, and engaging. In contrast, features like duration, acousticness, instrumentalness, and mode displayed a downward trend, reflecting a shift from music that is longer, more acoustic, or instrumental in nature (Appendix: Figure A. Popularity Trend).

We also analyzed music genres based on their popularity, comparing the least popular and most popular categories. Least popular music was identified as having a popularity score below the 25th percentile. A bar chart visualization showed that EDM is considered as the least popular genre, while pop music stood out as the most popular (Appendix: Figure B. Popularity Bar Chart). When these findings were combined with the feature trends, it became evident that EDM generally has lower valence compared to pop music, on average. Valence, which measures the positivity of a song, has shown an increasing association with popularity, further reinforcing the dominance of pop music (Appendix: Figure C. Genre Comparison Box Plot). These insights highlight the relationship between genre characteristics and their popularity, offering valuable implications for artists, producers, and the music industry.

Music Feature Correlation Analysis:

There is a relatively strong positive correlation between energy and loudness (0.68), which indicates that songs with higher energy tend to be louder. Valence and danceability also have a positive correlation (0.33), which highlights that more positive or “happy” songs tend to be more danceable. On the other hand, there is a relatively strong negative correlation between energy and acousticness (-0.54). This aligns with our common knowledge that electronic or amplified instruments are more common in energetic music. With this information, we decided to combine the categories “energy” and “loudness” together to improve the accuracy of our models’ predictions since including two redundant features may introduce collinearity. One key conclusion that we drew from this exploratory data analysis is that there is no single feature that rules out popular songs from unpopular songs, as the correlation values between individual features and track_popularity variable are small. (Appendix: Figure D. Correlation Heatmap)

Word Cloud:

In the word cloud, we see that “love” is the most frequently used word in song titles regardless of the time period, which indicates its central role in popular song themes. This shows the enduring popularity of love, as it is a relatable music theme (Appendix: Figure E. Word Cloud).

IV. Modeling

Data preparation and preprocessing

Feature selection: Since we observed that both energy and loudness were correlated with track popularity, we calculated their weighted combination based on their absolute correlations with popularity. Categorical variables such as key and mode were one-hot encoded using the pd.get_dummies() function without dropping the first category to retain all information. Therefore, the final set of features used for modeling included both engineered and raw features: danceability, energy_loud, key, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration_normalized.

Popularity Classification: For our regression model, we started off by treating popularity as a continuous variable. Then for the classification models, track popularity was treated as a binary classification task, and the classification thresholds were determined based on popularity percentiles. Given the absence of strict criteria for defining popularity, three thresholds were considered to examine their effects on model performance: (a) Percentile > 0.75: Popular tracks classified as 1. (b) Percentile > 0.85: Popular tracks classified as 1. (c) Percentile > 0.95: Popular tracks classified as 1. These thresholds provided flexibility in defining the target variable, allowing exploration of model performance across different levels of popularity.

Train-Test Split: The dataset was split into training and testing sets using a 70:30 ratio. This ensured sufficient data for training the model while retaining a representative sample for evaluation. The train-test split was performed randomly to prevent any potential biases and to ensure the generalizability of the model on unseen data.

Regression models

Linear regression with OLS on training data: To evaluate the relationship between audio features and track popularity, we applied Ordinary Least Squares (OLS) regression using the training dataset as it allowed us to evaluate the significance of each feature and assess the model's fit. Key metrics, such as coefficients and p-values, provided insights into the direction, strength, and statistical significance of the features. However, the R-squared value was 0.099, indicating that the model did not explain much of the variability in track popularity (Appendix: Figure F. OLS Summary).

Linear regression with scikit-learn on test data: After applying OLS on the training data, we used scikit-learn's LinearRegression to fit the model on the training data and predict track popularity on the test data. This model also helped evaluate the prediction performance through key metrics such as R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE). However, the R-squared value for the test data was 0.086, suggesting that the model still explained only a small portion of the variance in track popularity. Given this low value from both the training and test sets, we concluded that linear regression may not be the best fit for this task, and we moved on to exploring classification models.

Classification models

Logistic regression: We applied Logistic Regression using scikit-learn's LogisticRegression and evaluated its performance on the training and test sets at the 75th, 85th, and 95th popularity percentiles, using key metrics such as accuracy, precision, recall, F1 score, and the confusion matrix. For the 75th percentile, logistic regression performed moderately, with coefficients indicating the significance of features like danceability, speechiness, and acousticness. However, at the 85th and 95th percentiles, all metrics except accuracy were 0. This occurred because the model likely predicted all true negatives and struggled to correctly predict the minority class (true positives) due to class imbalance. (Appendix: Table A.)

Decision tree: To build the Decision Tree model, we used DecisionTreeClassifier and optimized its performance through hyperparameter tuning with RandomizedSearchCV and 20-fold cross-validation. We defined a hyperparameter grid to explore a variety of configurations. This included parameters such as the splitting criterion, splitter strategy, maximum tree depth, maximum number of leaf nodes, minimum samples required to split a node, minimum samples required to form a leaf node, and class weight to address class imbalance.

Random forest: We started by defining a hyperparameter grid to explore various model configurations. The grid included several important parameters: the number of trees in the forest (n_estimators), the number of features to consider for splits (max_features), the maximum depth of each tree (max_depth), the maximum number of leaf nodes (max_leaf_nodes), the splitting criterion, the minimum number of samples required to split a node (min_samples_split), the minimum number of samples required to form a leaf node (min_samples_leaf), and the class weight (class_weight). To find the best configuration, we used RandomizedSearchCV to evaluate multiple hyperparameter combinations and performed a 10-fold cross-validation to ensure robust performance evaluation. The Random Forest model was then trained using the fit method on the training dataset.

Models evaluation and selection

Based on Appendix Table A showing the model results, the models were evaluated on three criteria:

1. Consistency Between Training and Testing Performance: A key measure of a model's robustness is its ability to perform consistently on both training and testing datasets. Large discrepancies, where the model performs well on the training data but poorly on the test data, indicate overfitting. For instance, the Decision Tree model at the 0.75 threshold shows signs of overfitting, with a train F1 score of 0.99 compared to a much lower test F1 score of 0.68.

2. Recall as the Primary Metric: Since the task prioritizes identifying popular songs (reducing false negatives), recall is the primary evaluation metric. Misclassifying popular songs as not popular is more critical than misclassifying non-popular songs as popular. For example, the Decision Tree at the 0.95 threshold achieves a high test recall of 0.93, indicating strong performance in identifying popular songs. Similarly, the Random Forest at the 0.75 threshold achieves a recall of 0.87, highlighting its effectiveness in capturing popular songs despite lower overall precision.

3. Predicting Popular Songs Effectively: The ideal model balances consistency and high recall while maintaining reasonable performance in other metrics such as precision and F1 score. The Decision Tree at the 0.95 threshold and the Random Forest at the 0.85 threshold provide strong performance, with high recall values and relatively balanced train-test consistency, making them more suitable for identifying popular songs compared to other configurations.

V. Conclusion and Future Improvements

In summary, while recall is the main focus for identifying popular songs, ensuring consistent train-test performance is crucial to avoid overfitting and maintain reliability. The Decision Tree at the 0.95 threshold stands out as the most robust model for this task. Through hyperparameter tuning, the optimal hyperparameters for the Decision Tree model at the 0.95 popularity classification level are as follows: 1) Splitter: best (chooses the best split for each node); 2) Criterion: gini (uses the Gini impurity to evaluate splits); 3) Max Depth: None (no maximum limit for tree depth); 4) Max Leaf Nodes: None (no limit on the number of leaf nodes); 5) Min Samples Split: 2 (minimum number of samples required to split an internal node); 6) Min Samples Leaf: 1 (minimum number of samples required to be at a leaf node); 7) Class Weight: balanced (addresses class imbalance by adjusting weights inversely proportional to class frequencies). Based on the selected model, the feature importance values then indicate the contribution of each feature to the model's decision-making process. Features with higher importance values have a stronger influence on the model's predictions. Based on Table B in the Appendix, we can conclude that the top contributing features are speechiness (12.7% importance), danceability (12.0% importance), energy_loud (11.5% importance), instrumentalness (10.7%), and liveness (10.4%).

Popularity classification relies on arbitrary percentile thresholds (0.75, 0.85, 0.95), which might not align with real-world perceptions of popularity. Additionally, the class imbalance (more non-popular songs than popular ones) can bias the model toward predicting non-popular songs more often. To improve this, instead of fixed thresholds (e.g., 0.75, 0.85, 0.95), data-driven threshold optimization, such as using ROC-AUC analysis or precision-recall trade-offs, could yield more meaningful popularity classifications.

Appendix

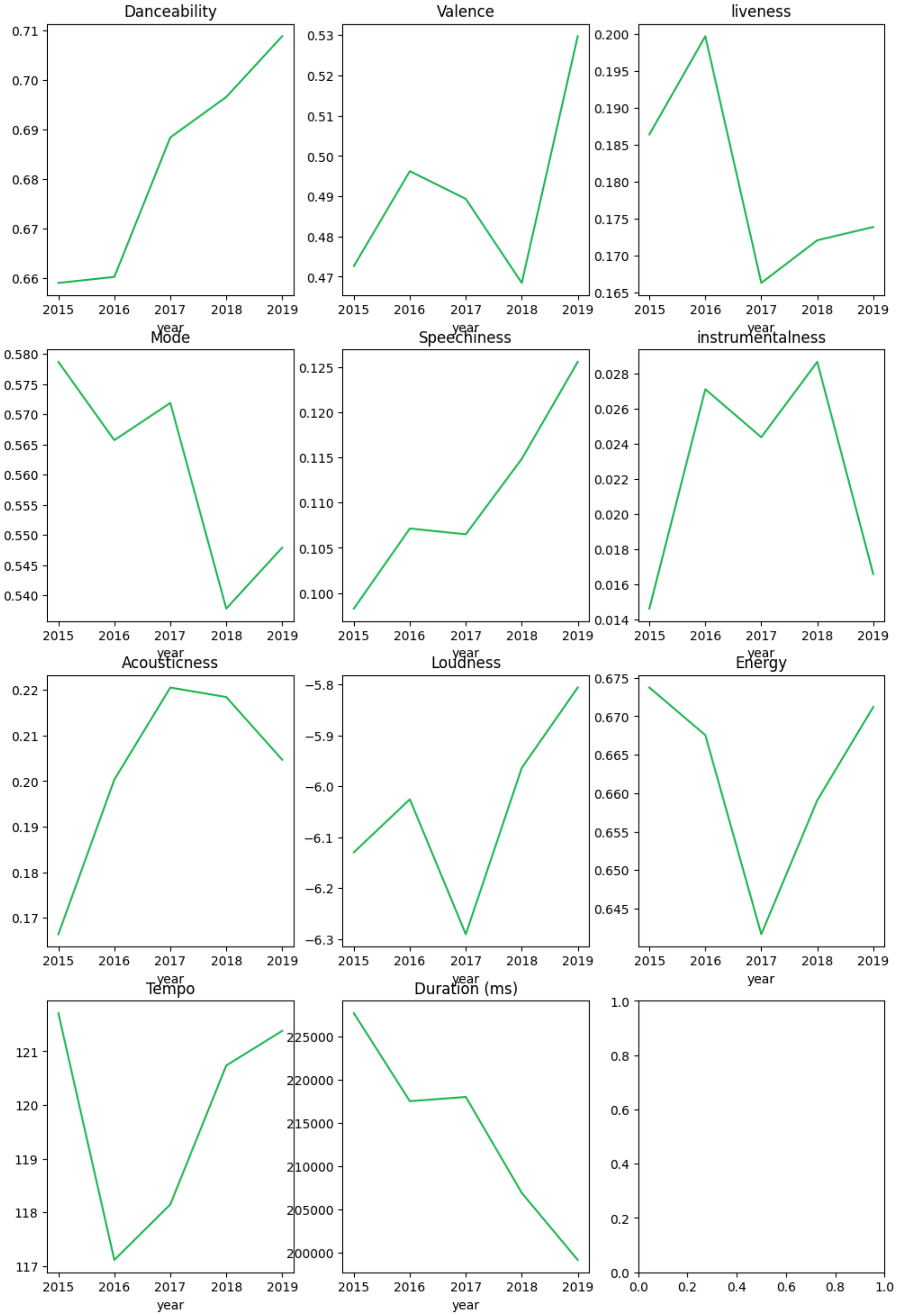


Figure A. Popularity Trend

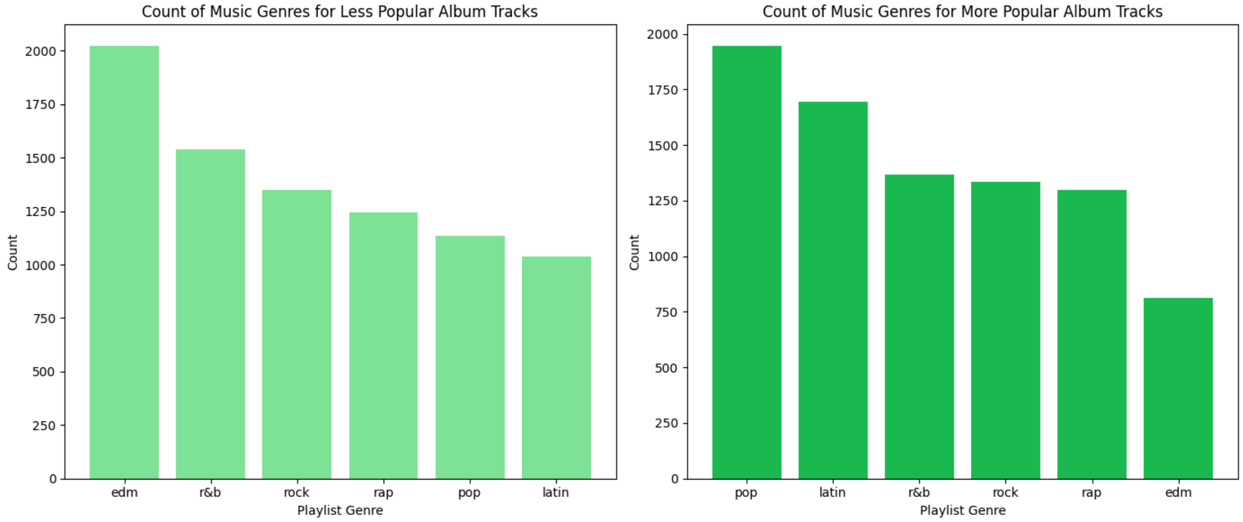


Figure B. Popularity Bar Chart

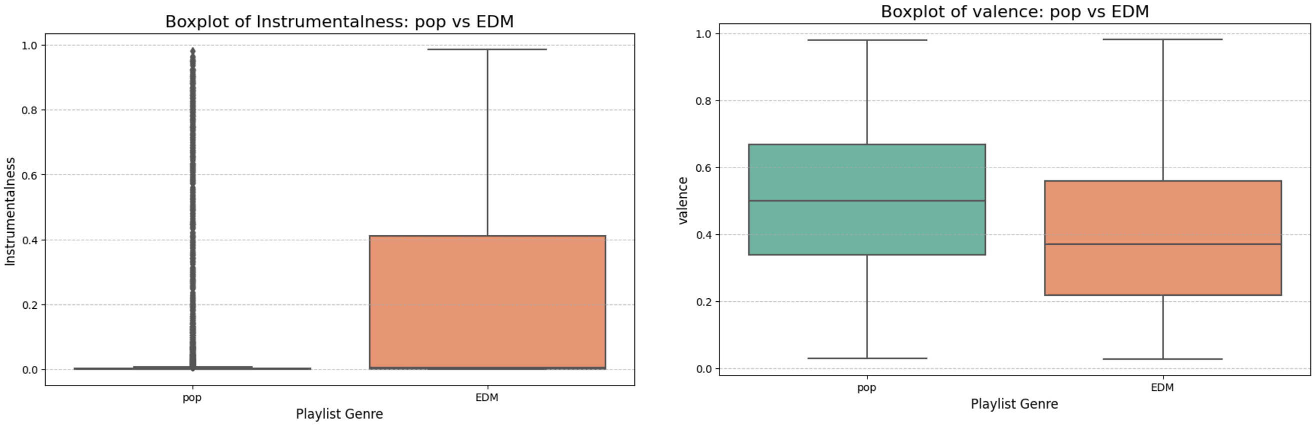


Figure C. Genre Comparison Box Plot

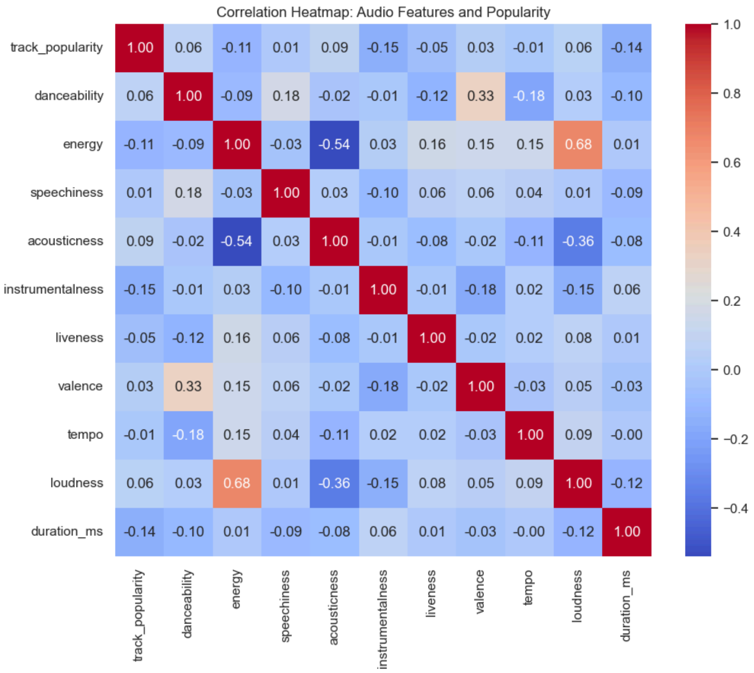


Figure D: Correlation Heatmap

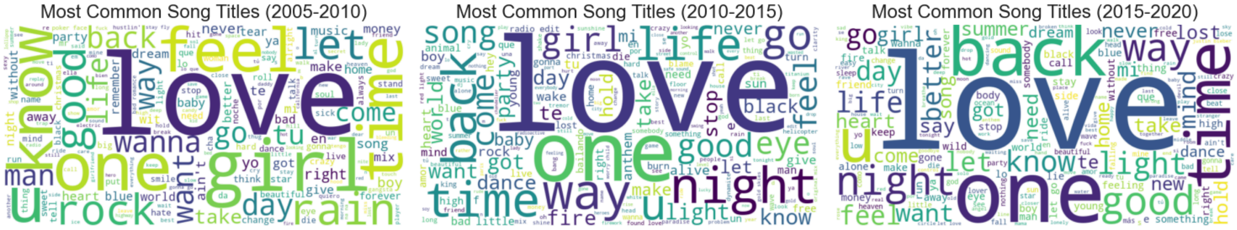
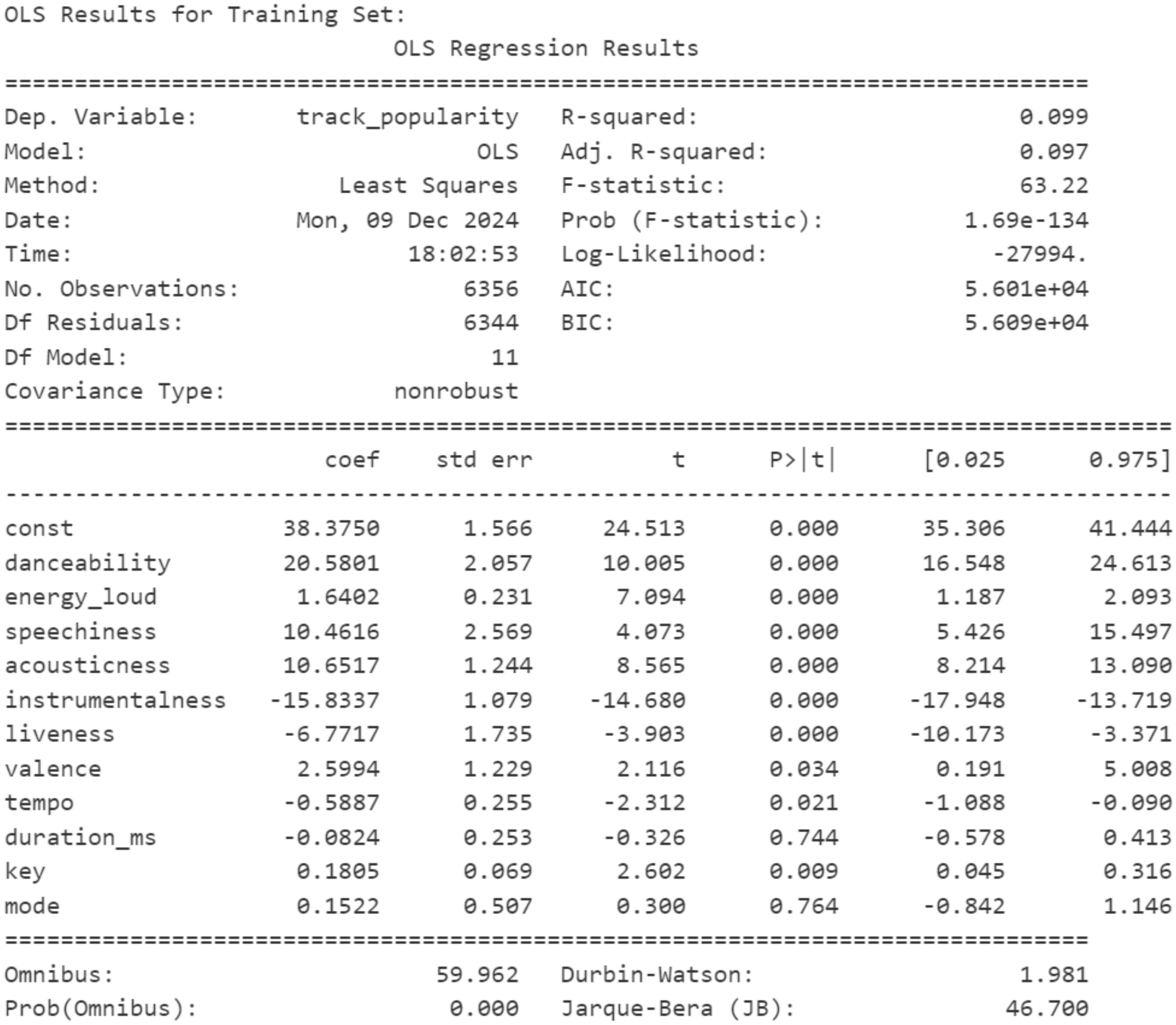


Figure E: Word Cloud



```
OLS Results for Training Set:
                            OLS Regression Results
==============================================================================
Dep. Variable:       track_popularity   R-squared:                       0.099
Model:                            OLS   Adj. R-squared:                  0.097
Method:                 Least Squares   F-statistic:                     63.22
Date:                Mon, 09 Dec 2024   Prob (F-statistic):          1.69e-134
Time:                        18:02:53   Log-Likelihood:                -27994.
No. Observations:                6356   AIC:                         5.601e+04
Df Residuals:                    6344   BIC:                         5.609e+04
Df Model:                          11
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               38.3750      1.566     24.513      0.000      35.306      41.444
danceability        20.5801      2.057     10.005      0.000      16.548      24.613
energy_loud          1.6402      0.231      7.094      0.000       1.187       2.093
speechiness         10.4616      2.569      4.073      0.000       5.426      15.497
acousticness        10.6517      1.244      8.565      0.000       8.214      13.090
instrumentalness   -15.8337      1.079    -14.680      0.000     -17.948     -13.719
liveness            -6.7717      1.735     -3.903      0.000     -10.173      -3.371
valence              2.5994      1.229      2.116      0.034       0.191       5.008
tempo               -0.5887      0.255     -2.312      0.021      -1.088      -0.090
duration_ms         -0.0824      0.253     -0.326      0.744      -0.578       0.413
key                  0.1805      0.069      2.602      0.009       0.045       0.316
mode                 0.1522      0.507      0.300      0.764      -0.842       1.146
==============================================================================
Omnibus:                       59.962   Durbin-Watson:                   1.981
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               46.700
```

Figure F. OLS Summary

model	train_f1	train_accuracy	train_precision	train_recall	test_f1	test_accuracy	test_precision	test_recall
Decision Tree (75)	0.99	0.99	0.98	0.99	0.68	0.82	0.63	0.74
Decision Tree (85)	0.98	0.99	0.96	1.0	0.73	0.91	0.66	0.83
Decision Tree (95)	0.96	0.99	0.92	1.0	0.84	0.98	0.76	0.93
Random Forest (75)	0.57	0.66	0.41	0.90	0.56	0.65	0.41	0.87
Random Forest (85)	0.97	0.99	0.96	0.98	0.78	0.90	0.93	0.67
Random Forest (95)	0.94	0.98	0.99	0.91	0.89	0.73	0.91	0.6
Logistic Regression (75)	0.03	0.77	0.51	0.01	0.05	0.77	0.53	0.02
Logistic Regression (85)	0.00	0.86	0.00	0.00	0.00	0.86	0.00	0.00
Logistic Regression (95)	0.00	0.96	0.00	0.00	0.00	0.96	0.00	0.00

Table A. Model Results

	Feature	Importance		Feature	Importance
0	speechiness	0.1270753	**13**	key_2	0.00595488
1	danceability	0.1203429	**14**	key_1	0.005925435
2	energy_loud	0.1149683	**15**	key_10	0.005835663
3	instrumentalness	0.1070178	**16**	key_9	0.005666285
4	liveness	0.1035365	**17**	key_7	0.00372493
5	tempo	0.09990558	**18**	key_5	0.001884045
6	valence	0.09260155	**19**	key_11	0.001760967
7	acousticness	0.08787034	**20**	key_8	0.000322818
8	duration_normalized	0.07273751	**21**	mode_0	5.817933E-16
9	key_6	0.02019023	**22**	mode_1	0.0
10	key_4	0.008615891			
11	key_0	0.008024051			
12	key_3	0.006039016			

Table B. Feature Importance