Final Report by Victoria Brigola

Tuning into Popularity: Predicting Spotify Track Success Through Audio Features

Why Predicting Matters: Introduction

Spotify isn't just a platform for streaming music, it's a personal refuge where users escape, connect, and immerse themselves in sound. With millions of tracks from diverse genres and artists, Spotify's library is constantly growing. As new music is released daily, accurately predicting which tracks will rise in popularity is vital for effective playlist placement, user engagements, and marketing strategy. Popular tracks not only drive listener retention and discovery but also play a key role in amplifying artist exposure. By Identifying the right song at the right time, Spotify strengthens its position as the top destination for music lovers all around the world.

Signal in the Noise: Problem at Hand

With the overwhelming volume of new music released everyday, Spotify faces a core challenge: efficiently identifying; which tracks are likely to become popular before the audience decides. Relying on historical metrics, overlooking breakout hits, especially lesser-known or emerging artists. The absence of scalable, predictive tools mean high-potential songs can get buried in the noise of the platform's massive catalog.

This project addresses the challenge by using machine learning to predict track's popularity based on audio features. By solving this problem, Sporify can enhance its recommendations system, flag emerging hits, and refine playlist curation to deliver a more engaging and personalized experience for every listener.

Approach: Data Collection & Wrangling

The first initial phase of the project began with preparing my first dataset ***spotifytracks.csv***, which columns such as t*rack_id, track_name, artist_name, album_name,* and *track genre*. Redundant and irrelevant fields like track_href and album_id were removed. The goal was to leverage the genre data to predict track popularity. As a result of the inconsistent labeling and overlap of genres, it made my initial approach unreliable. Therefore shifting my focus to audio features for more effective modeling.

The second dataset, ***audiofeatures_dataset.csv***, provided more structured data, including audio characteristics such as d*anceability, energy, loudness, speechiness, acousticness, instrumentalness, liveeness, valence, temp, duration_ms, key, mode, and time_signature.* After dropping any low-variance or sparsely populated columns, this dataset was ready to to be merged.

Both datasets were merged using a left join on the shared track_id column. The resolve duplicate column names to overlapping data such as track_name_x and track_name_y. I retrained the version with the most complete and consistent data. After merging, I dropped the remaining redundant null columns and remove duplicate track_id entries. The resulting dataset contained a cleaned combination of descriptive metadata and robust audio features.

Once the merged dataset was cleaned, two features were utilized to enhance modeling later on: *genre_category,* which grouped granular genre labels into broader categories, and *loudness_binned,* which categorized loudness values into discrete ranges. However, the core focus for modeling and analysis was on such as d*anceability, energy, valence, and loudness.* These features were consistently emphasized in both exploratory analysis and the final predictive modeling. The final dataset, saved as ***masterfinal.csv***, contained 1,567 records and 22 refined features.

Behind the Beats: Exploratory Data Analysis

Uncovering the underlying structure and relationship of my data, I conducted a comprehensive exploratory analysis focused on key audio features linked to track popularity. Higher popularity was associated with greater danceability, instrumentalness, loudness and valence, suggesting that listeners are more engaged by upbeat and emotionally positive music. Genres like pop and hip hop were more prevalent in higher popularity bins, while the raw popularity distribution was skewed, supporting the decision to use classification over regression. Correlation heat maps revealed strong interdependencies among features such as temp, loudness, and energy, which guided feature scaling and model selection decisions. These visualizations below helped me identify key decisions in feature selection and validated the importance of specific audio characteristics in predicting track success.

Behind the Beats: Exploratory Data Analysis, Visualizations

Figure 1 (Histogram): Shows skewed distribution and justification for binning/popularity classification.

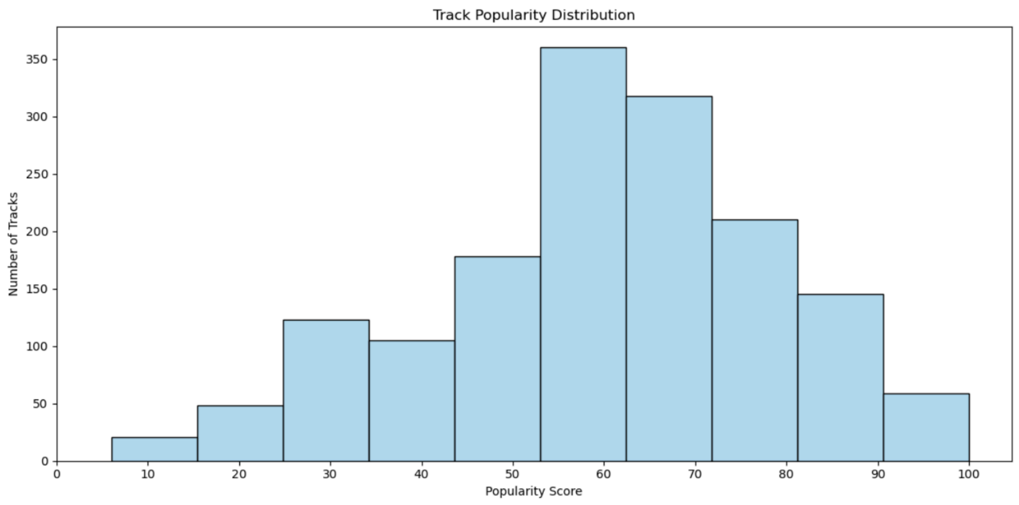
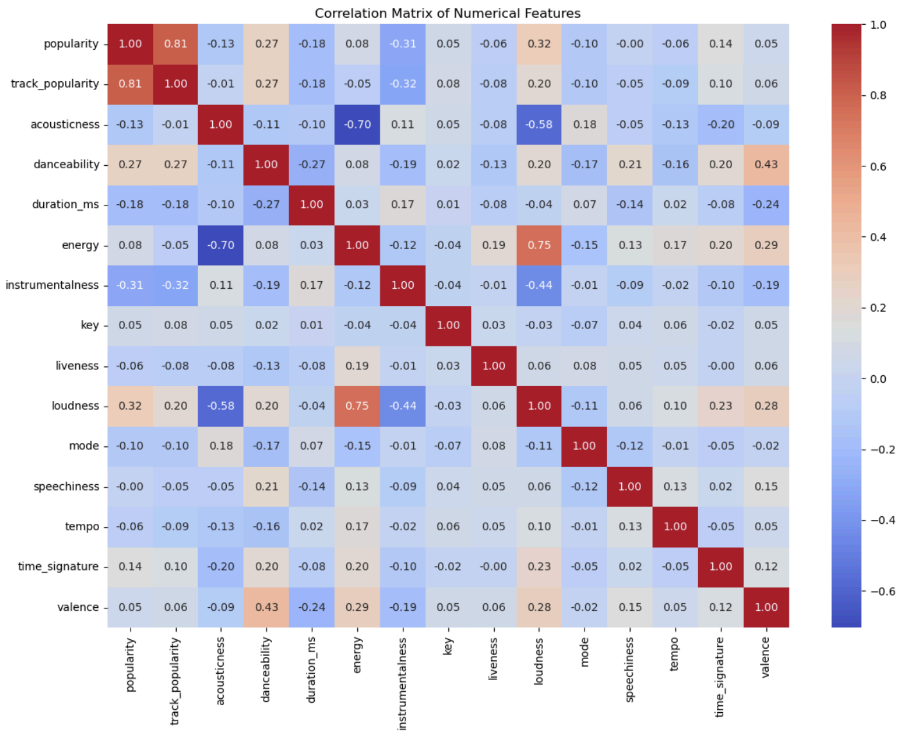


Figure 2 (Correlation Matrix): Identifies strong correlations (e.g., loudness with energy), multicollinearity issues.



	popularity	track_popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
popularity	1.00	0.81	-0.13	0.27	-0.18	0.08	-0.31	0.05	-0.06	0.32	-0.10	-0.00	-0.06	0.14	0.05
track_popularity	0.81	1.00	-0.01	0.27	-0.18	-0.05	-0.32	0.08	-0.08	0.20	-0.10	-0.05	-0.09	0.10	0.06
acousticness	-0.13	-0.01	1.00	-0.11	-0.10	-0.70	0.11	0.05	-0.08	-0.58	0.18	-0.05	-0.13	-0.20	-0.09
danceability	0.27	0.27	-0.11	1.00	-0.27	0.08	-0.19	0.02	-0.13	0.20	-0.17	0.21	-0.16	0.20	0.43
duration_ms	-0.18	-0.18	-0.10	-0.27	1.00	0.03	0.17	0.01	-0.08	-0.04	0.07	-0.14	0.02	-0.08	-0.24
energy	0.08	-0.05	-0.70	0.08	0.03	1.00	-0.12	-0.04	0.19	0.75	-0.15	0.13	0.17	0.20	0.29
instrumentalness	-0.31	-0.32	0.11	-0.19	0.17	-0.12	1.00	-0.04	-0.01	-0.44	-0.01	-0.09	-0.02	-0.10	-0.19
key	0.05	0.08	0.05	0.02	0.01	-0.04	-0.04	1.00	0.03	-0.03	-0.07	0.04	0.06	-0.02	0.05
liveness	-0.06	-0.08	-0.08	-0.13	-0.08	0.19	-0.01	0.03	1.00	0.06	0.08	0.05	0.05	-0.00	0.06
loudness	0.32	0.20	-0.58	0.20	-0.04	0.75	-0.44	-0.03	0.06	1.00	-0.11	0.06	0.10	0.23	0.28
mode	-0.10	-0.10	0.18	-0.17	0.07	-0.15	-0.01	-0.07	0.08	-0.11	1.00	-0.12	-0.01	-0.05	-0.02
speechiness	-0.00	-0.05	-0.05	0.21	-0.14	0.13	-0.09	0.04	0.05	0.06	-0.12	1.00	0.13	0.02	0.15
tempo	-0.06	-0.09	-0.13	-0.16	0.02	0.17	-0.02	0.06	0.05	0.10	-0.01	0.13	1.00	-0.05	0.05
time_signature	0.14	0.10	-0.20	0.20	-0.08	0.20	-0.10	-0.02	-0.00	0.23	-0.05	0.02	-0.05	1.00	0.12
valence	0.05	0.06	-0.09	0.43	-0.24	0.29	-0.19	0.05	0.06	0.28	-0.02	0.15	0.05	0.12	1.00

Figure 3 (Reg Plots for danceability, loudness, instrumentalness, energy): Shows positive trend supports its use as a predictive feature and quantifies their contribution to popularity across track features.

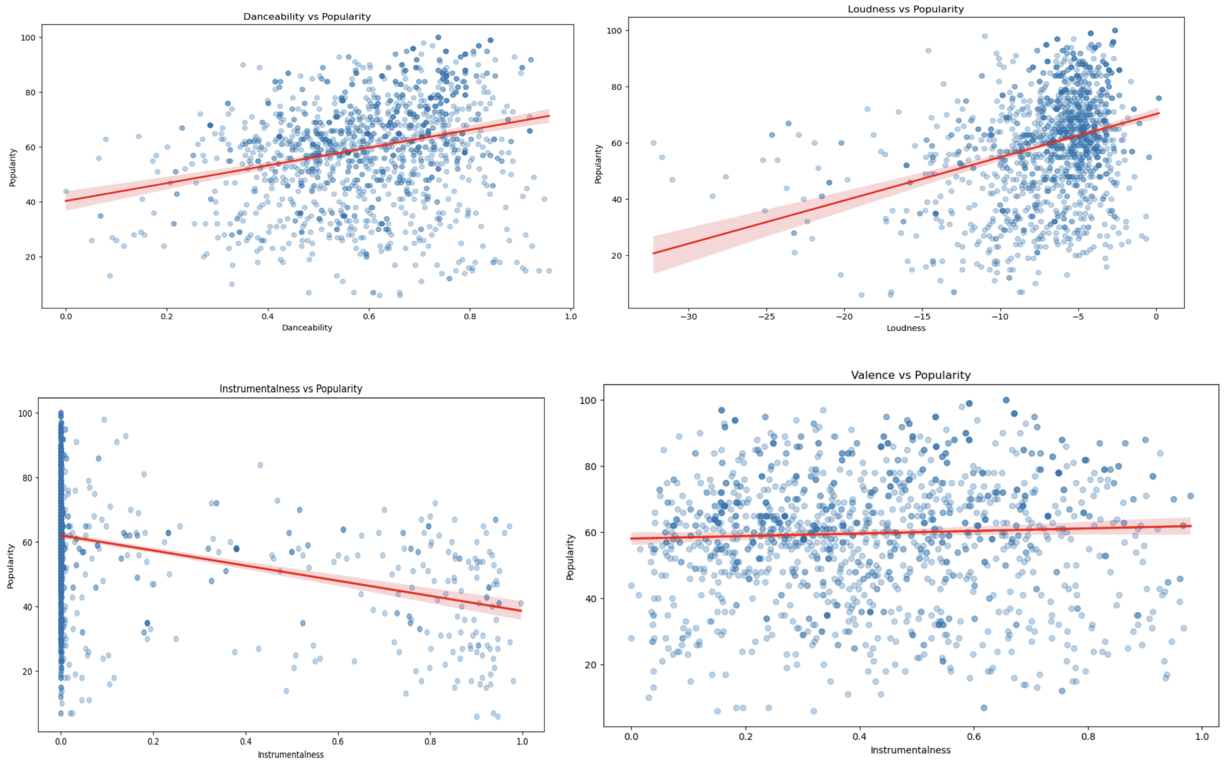


Figure 4 (Scatter plot grouped by loudness level): Visual support for engineered feature loudness_binned.

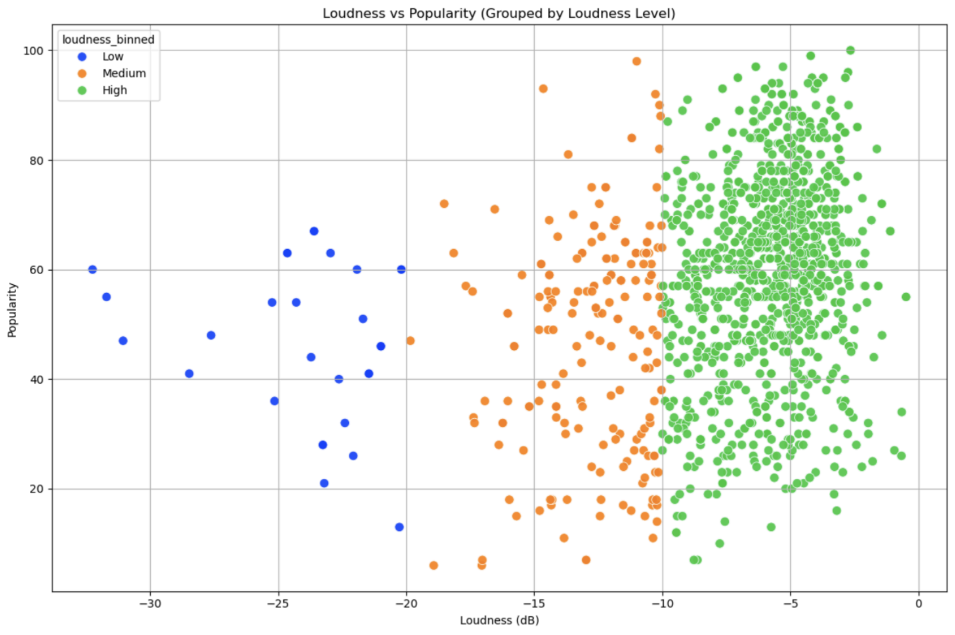
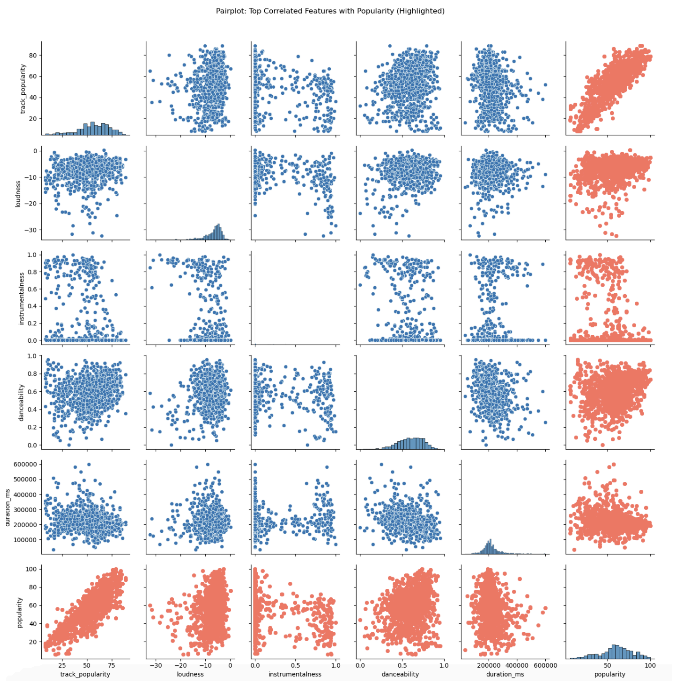


Figure 5 (Pair plot): Shows multidimensional relationships driving popularity.



Fine-Tuning the Features: Pre-Processing

I needed to make sure the dataset was in the best shape possible for training predictive models. I knew from experience that if the data wasn't cleanly formatted or consistently scaled, even the best algorithms would struggle. I started by normalizing all numeric audio features using ***`StandardScaler`***, which helped bring different value ranges like tempo and loudness into alignment. Without this, louder or faster songs might have unfairly influenced the model's learning process.

To prepare the target variable for classification, I used ***`LabelEncoder`*** to convert the binned popularity scores into integer values. I also one-hot encoded categorical features like *`genre_category`* and *`mode`* to remove any implied order and better reflect the nature of those variables. It was important to preserve the meaning of genre groupings without letting the model assume a hierarchy that didn't exist.

Once the features were ready, I split the dataset into training and testing sets with an ***80/20 split.*** This helped me train the models on the bulk of the data while reserving enough for a solid evaluation. The resulting files ***`X_train`, `X_test`, `y_train`, and `y_test`*** formed the backbone of the modeling phase.

To make sure everything was repeatable and reproducible later, I saved the preprocessing tools (`scaler.joblib` and `label_encoder.joblib`). This step was essential; it meant I could transform new data the exact same way and maintain consistency across predictions. This process felt like finalizing the groundwork before letting the models take over.

Training the Hits: Modeling

The original *track_popularity* score, which was a continuous variable ranging from 0 to 100, was transformed into a classification task by binning it into three categories: 0 for low, 1 for medium, and 2 for high popularity. The target distribution **(see Figure 6 below)** was noticeably imbalanced, with the majority of tracks falling into the high-popularity category.

I explored a range of classifications including *Bagging Classifier, Random Forest (both base and tuned), Decision Tree (Gini and Entropy), and Logistic Regression*. Performance was assessed by primarily using F1 macro scores across cross-validation folds **(see Figure 7 below).**

While *Logistic Regression* and basic *Decision Trees* performed poorly due to oversimplification and lack of generalization, even the tuned Random Forest struggled with the underrepresented Class 1 (medium popularity). Despite higher precision in some cases **(see Figure 8 below)**, recall for Class 1 remained low.

Model accuracy and class-wise breakdown were by a confusion matrix **(see Figure 9 below)** that reinforced its consistent performance across the test set. It was especially helpful in visualizing how well the model distinguished between three popularity classes, revealing patterns of misclassification. For example, it showed where the model had difficulty distinguishing between medium and high popularity tracks which is a valuable insight for refining future models.

Ultimately, I selected the ***Bagging Classifier*** as the final model. It demonstrated strong and balanced F1 performance across classes, particularly from Class 0 and Class 2, and handled noisy or overlapping features well. Its structure helped reduce variance, improving overall generalization without overfitting.

Figure 6 (Histogram): Target variable class distribution.

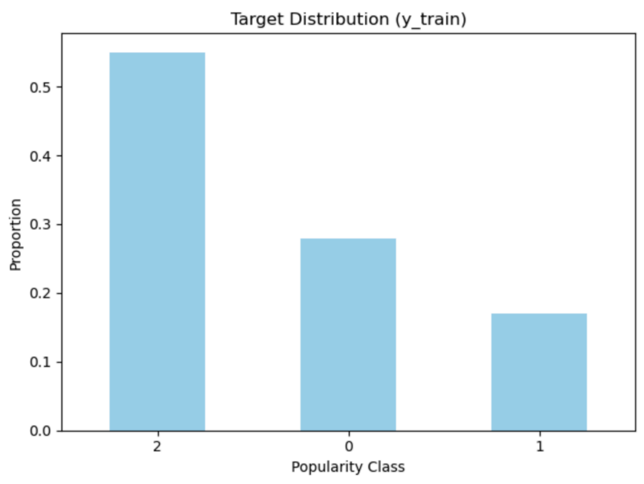
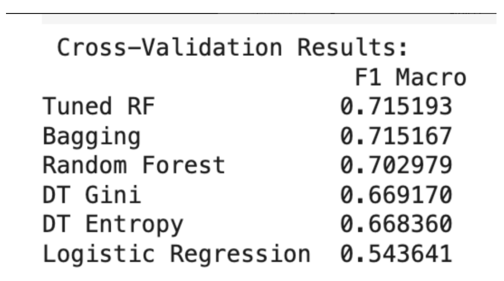
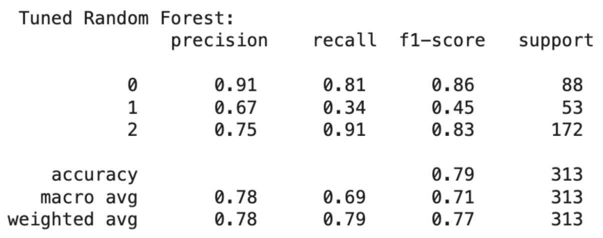


Figure 7: Cross-validation F1 scores for all models.



```
  Cross-Validation Results:
                      F1 Macro
 Tuned RF            0.715193
 Bagging             0.715167
 Random Forest       0.702979
 DT Gini             0.669170
 DT Entropy          0.668360
 Logistic Regression 0.543641
```

Figure 8: Classification report for Tuned Random Forest.

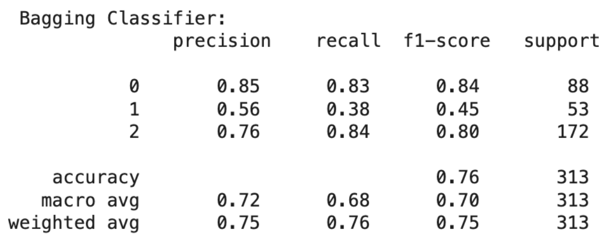


```
 Tuned Random Forest:
              precision    recall  f1-score   support

           0       0.91      0.81      0.86        88
           1       0.67      0.34      0.45        53
           2       0.75      0.91      0.83       172

    accuracy                           0.79       313
   macro avg       0.78      0.69      0.71       313
weighted avg       0.78      0.79      0.77       313
```

Figure 9: Classification report for Bagging Classifier.

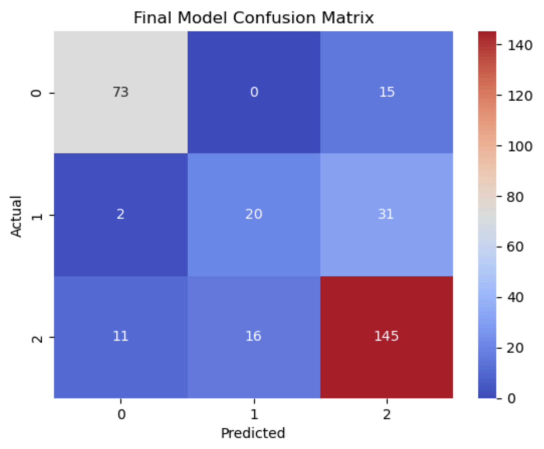


```
 Bagging Classifier:
              precision    recall  f1-score   support

           0       0.85      0.83      0.84        88
           1       0.56      0.38      0.45        53
           2       0.76      0.84      0.80       172

    accuracy                           0.76       313
   macro avg       0.72      0.68      0.70       313
weighted avg       0.75      0.76      0.75       313
```

Figure 9: Confusion matrix showing performance breakdown across classes



Performance Breakdown: Results and Evaluation

Amongst the models tested, Bagging Classifier demonstrated the best performance in terms of generalization and consistency. It showed robustness to noise and variance in the feature space, making it a reliable choice for Spotify. Evaluation metrics indicate strong results in accuracy and F1 macro score, with balanced performance across all three popularity classes. Confusion matrices and classification reports confirmed the model's ability to handle imbalance class distributions with maintaining predictive power.

Streaming Smarter: Recommendations for Spotify

Spotify should consider using models like the Bagging Classifier to make early predictions about unreleased tracks , helping identify potential hits before they go viral. The model's performance was most influenced by features such as danceability, energy, valence, and loudness. By emphasizing these audio characteristics during track evaluations, Spotify can better predict which songs are most likely to trend. For instance, high danceability and energy levels often align with high popularity, while valence captured emotional positivity and loudness contributed to listener intensity perception.

Wrapping the Set: Conclusion

My capstone project demonstrates the viability of using supervised machine learning techniques to predict track popularity on Spotify based solely on audio features and metadata. By integrating and cleaning two datasets, conducting thorough exploratory data analysis, and applying effective preprocessing, and modeling techniques, the project built a predictive pipeline capable of classifying tracks into popularity tiers. This will support Spotify's strategic goals, including content discovery, playlist management, and artist data, exploring regression-based approaches, and applying analytics to capture trend dynamics.