Melody Metrics: Predicting Popular Songs Based on Audio Features of Spotify

Group members (Group 09)

- Prabhleen Kaur (0857194)
- Rajwinder Kaur (0831280)
- Vindya Senadheera (0857437)
- STM Chathurangi (0850982)
- Ajay Haji Korbe (0852660)

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Streaming music platforms like Spotify have changed how people find music and connect with it. Understanding the factors that contribute to a song's popularity is essential for artists, producers, and marketers to stay competitive. They can improve playlists, enhance marketing campaigns, and provide tailored suggestions by determining the main factors that contribute to a song's success. This will ultimately improve user experience and retention rates.

This project examines the relationship between song popularity and various audio features, such as tempo, energy, valence, and danceability etc. The study uses exploratory data analysis and machine learning approaches to estimate the likelihood of a song's success based on 20 audio-related features (15 numerical and 5 categorical) that were taken from the Spotify API and a dataset of 2,409 songs.

1.1. Objectives

- To analyze the relationship between audio features and the popularity of songs.
- To develop a machine learning model that predicts a song's popularity based on its audio features.

1.2. Problem Statement

As the music industry becomes more data-driven, it is important to understand what makes a song popular is crucial for artists, music producers, marketers, and streaming platforms. Predicting songs' popularity can have a significant influence on decision-making processes, increasing user engagement and boosting marketing initiatives.

1.3. Scope

This project focuses on exploring patterns behind the audio features that influence a song's popularity. The findings can help producers and artists make more educated creative choices, help marketers better target consumers, and help streaming platforms' recommendation engines. The ultimate objective is to support a music industry that is more user-focused and data-driven.

2. Related Work

The use of audio characteristics to predict song popularity has been the subject of numerous studies using a variety of statistical models and machine learning approaches.

From the study of Saragih (2023) , it focuses on the Indonesian market, applying both regression and classification machine learning algorithms to forecast the song popularity based on Spotify's audio features. In addition to highlighting the importance of characteristics like energy, loudness, and instrumentality, they concluded that the Random Forest Classifier and Extra Trees Regressor were the most accurate models, achieving an accuracy of 69.74%, an F1-score of 69.44%, an R^2 score of 68.57%, and an RMSE of 12.33. This study highlighted the potential impact of regional preferences, such as those in Indonesia, on the models' predictive accuracy. From their study, we emphasized audio features like energy and loudness in our feature selection and validated our choice of Random Forest as a strong predictive model for "Melody Metrics."

From the study of Jung and Mayer (2024), they utilized a dataset of 30,000 songs spanning different genres from 1957 to 2020 and employed algorithms such as Regression Splines (MARS), Multivariate Adaptive Random

Forest, Ordinary Least Squares (OLS), and XGBoost. They found that using the average track popularity score gave a baseline MAE of 17.55. The full OLS model reduced MAE to 16.64 (5.2% improvement), XGBoost achieved 16.47 (6.16% improvement), and Random Forest performed best with an MAE of 16.31, marking a 7.08% improvement. Genre, particularly Electronic Dance Music (EDM), played a significant role in prediction accuracy. From their study, our "Melody Metrics" project drew inspiration from using a wide range of audio features and applying multiple machine learning models to predict song popularity. Their emphasis on genre importance, along with features like danceability and energy, also guided our feature selection and analysis process.

Furthermore, Li (2024) used a dataset of songs from 1986 to 2022 to examine the connection between audio characteristics and song popularity. According to his research, attributes including energy, danceability, and track length were positively connected with greater levels of popularity. His optimized model, after refining variables through transformations and model selection, achieved an adjusted R-squared of 0.282 and an R-squared of 0.284 on training data, and 0.308 (adjusted R-squared) and 0.311 (R-squared) on test data, indicating a moderate strength in explaining song popularity based on the selected audio features. Li's research also examined how listener tastes changed over time, emphasizing recent release years and the increasing significance of danceability. His study influenced our project by emphasizing the importance of danceability and track duration, which we incorporated into our feature selection for "Melody Metrics.

3. Methods

The steps taken to prepare the dataset, perform exploratory data analysis (EDA), preprocess the data, choose features, train and assess the model, and adjust hyperparameters are detailed in this section.

3.1. Data Source

The dataset selected in this study was obtained from the Spotify Web API. The dataset consists of 2,409 songs, each with 20 audio-related features. Of these, 15 are numerical features, and 5 are categorical. The dependent variable is “Track Popularity”, a numerical score ranging from 0 to 100, representing how popular the track is on Spotify.

3.2. Exploratory Data Analysis (EDA)

EDA was performed to understand the distribution and relationships in the dataset. The following steps were conducted:

- **Descriptive Statistics**: The mean, median, standard deviation, and range of numerical features were calculated to understand their distributions.
- **Removed the Outliers**
- **Dropped an unnecessary “Time Signature” column.**
- **Visualizations**:
  - ✓ The distribution of artist popularity was illustrated through histograms and boxplots. Accordingly, the histogram reveals the overall trend in artist popularity, emphasizing the most frequently occurring popularity scores.
  - ✓ A correlation matrix was generated to understand relationships among numerical features.

3.3. Statistical Analysis (Hypothesis Testing)

To understand the relationships between features and the target variable (Track Popularity), hypothesis testing was performed. This included:

- **ANOVA (Analysis of Variance)** applied to compare the mean track popularity across different genres in the dataset. This method was chosen as it is well-suited for analyzing differences across multiple groups
- **Correlation Test** applied to assess the linear relationships between track popularity and selected audio features (loudness, danceability, energy, and acousticness).
- **T-Test** applied to analyze the Danceability Difference Between Popular and Unpopular Songs

3.4. Feature Selection

Three methods were used to select important features for model creation such as Filter Method (Chi-Square Test), Wrapper Method (RFE with Random Forest) and Embedded Method.

3.5. Model Building:

Compared models using PyCaret, chose Random Forest, tuned hyperparameters with RandomizedSearchCV, finalized optimized model.

3.6. Saving the Model and Preprocessing Steps:

After obtaining a well-performing model with an R^2 score of **0.7694** and MSE of **103.8957**, we saved the following components using joblib:

- best_rf_model.pkl – Optimized Random Forest model
- target_encoder.pkl – For encoding categorical variables
- power_transformer.pkl – For normalizing skewed features
- feature_scaler.pkl – For scaling input features
- target_scaler.pkl – For inverse-scaling predicted values

This step was essential to ensure that the same data transformations could be applied during prediction in the deployed environment.

3.7. Deployment:

Saved model and preprocessing tools (joblib), built Flask web app for user input and predictions, hosted locally, shared project on GitHub.

4. Results

4.1. Exploratory Data Analysis (EDA)

4.1.1. Unique Artists by Country - Top 10 countries with the most unique artists were identified (India, US, etc.), visualized using a bar chart.

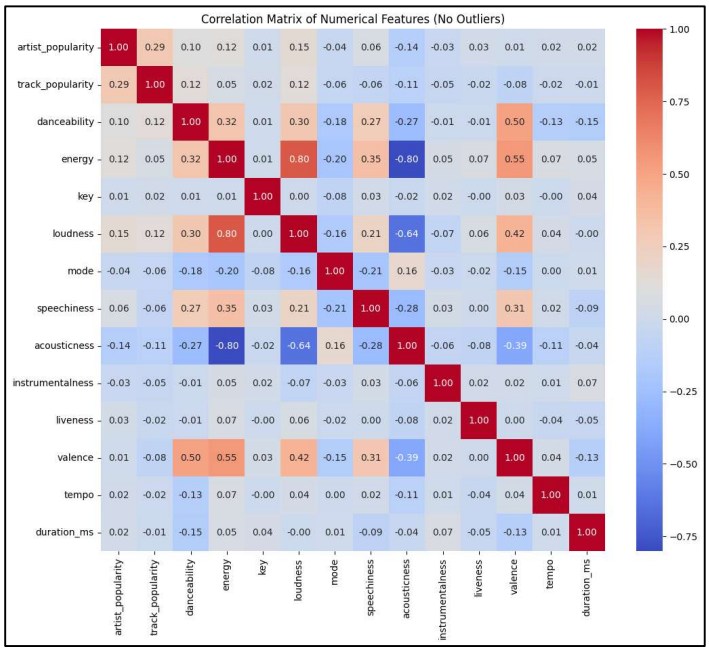
4.1.2. Distribution of Artist Popularity: Histograms and boxplots showed artist popularity trends, highlighting common scores and outliers.

4.1.3. Correlation Analysis of Track Popularity:

The heatmap showed:

- Moderate positive correlation with artist popularity.
- Weak positive correlations with loudness, danceability, acousticness, and valence.
- Very weak/negligible correlations with energy, instrumentalness, liveness, key, tempo, and duration.

Figure 3.1 - Heatmap of the Correlation Matrix for Numerical Features**.**



track_popularity	1.000000
artist_popularity	0.290006
loudness	0.120844
danceability	0.120702
acousticness	0.107754
valence	0.081506
speechiness	0.057225
mode	0.055120
energy	0.053824
instrumentalness	0.047217
liveness	0.020520
key	0.020356
tempo	0.018604
duration_ms	0.012904

4.2. Statistical Analysis (Hypothesis Testing)

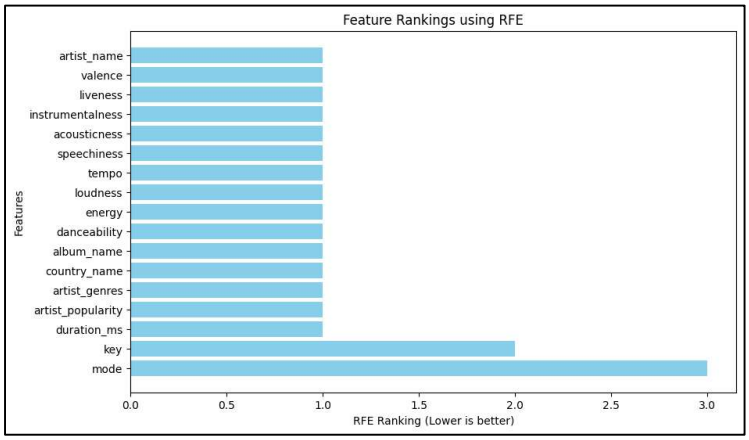
- **ANOVA Test (Genres vs. Track Popularity)**: Track popularity significantly differs across genres ($p < 0.05$). Genre influences listener engagement.
- **Pearson Correlation Test**:
  - ✓ **Loudness** and **Danceability**: Weak but significantly positive correlation with popularity.
  - ✓ **Acousticness**: Weak but significant negative correlation.
  - ✓ **Energy**: No significant correlation.
- **T-Test (Danceability in Popular vs. Unpopular Songs)**:
  - ✓ Popular songs have significantly higher danceability ($p = 0.0085$).
  - ✓ Supports the idea that danceability influences track popularity.

4.3. Feature Selection

We employed three main feature selection methods:

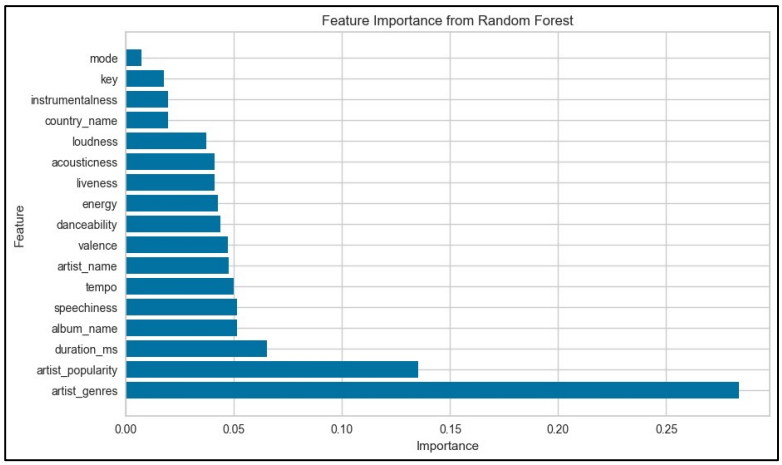
- **Filter Method**: Using the Chi-Square test, we identified the categorical features most significantly related to track popularity (e.g., artist genres, country, album name, artist name). Features like track name was excluded as they showed no significant relationship.
- **Wrapper Method**: Using Recursive Feature Elimination (RFE) with Random Forest, we selected the top 15 most important features for predicting track popularity.

Figure 4.1 - Visualization**:** Wrapper Method



- **Embedded Method**: Using Random Forest Regressor for feature importance, we finalized a set of key features for the model, excluding overfitting features like artist popularity and underfitting features like key and mode.

Figure 4.2 - Embedded Method Top 15 features by importance



Top 15 Features by Importance:

	Feature	Importance
2	artist_genres	0.283802
1	artist_popularity	0.135119
16	duration_ms	0.065330
4	album_name	0.051526
10	speechiness	0.051279
15	tempo	0.049827
0	artist_name	0.047438
14	valence	0.047122
5	danceability	0.043566
6	energy	0.042539
13	liveness	0.040959
11	acousticness	0.040941
8	loudness	0.037128
3	country_name	0.019355
12	instrumentalness	0.019323

4.4. Model Building

The Random Forest Regressor model, after preprocessing and hyperparameter optimization, achieved the following results:

- **Training R^2 Score**: 0.8873
- **Testing R^2 Score**: 0.7694
- **Mean Squared Error (MSE)**: 103.90

These results indicate that the model explains approximately 89% of the variance in song popularity on the training set, and around 77% of the variance on the testing set. The optimized model's performance is considered satisfactory for the given task, as it provides reliable predictions of song popularity based on the selected audio features.

4.5. Model Deployment

To make our model accessible, we developed a web application using **Flask**, a lightweight Python web framework. The application allows users to enter specific audio features of a song and receive a predicted popularity score.

Key Features:

- **Front-end Interface**: A clean and interactive form where users input values like tempo, energy, danceability, etc.
- **Backend Integration**: The server receives the inputs, applies the same encoding, transformation, and scaling steps using the saved files, and then feeds the data into the trained model for prediction.
- **Output Display**: The predicted popularity score is inverse transformed to its original scale and displayed to the user in real-time.
- **Template Pages Added**: The web application includes three core templates for seamless navigation:
  - ✓ **Home Page** – Offers an introduction and overview of the tool.
  - ✓ **About Page** – Describes the purpose, goals, variable descriptions, and team behind the project.
  - ✓ **Prediction Page** – Hosts the input form and displays the predicted result.

4.6. GitHub Repository

To support collaboration, transparency, and reproducibility, we created a **GitHub repository** for this project.

Link for the GitHub Repository: https://github.com/vindy92/DAB422-Capstone-Group-Project---Group-09.git

5. Discussion

The optimized Random Forest Regressor demonstrated strong performance, with the model achieving a high R^2 score on both training and test data. This suggests that the selected audio features, along with the preprocessing steps and feature engineering techniques, played a significant role in capturing the factors that influence song popularity.

One key takeaway from this analysis is that interaction features, such as danceability , energy , valence and tempo contributed positively to the model's performance. These interactions allowed the model to capture more complex relationships between features that single-feature models might miss.

However, there were challenges related to handling categorical features, especially with high-cardinality columns such as artist name and genre. Target Encoding provided a practical solution, but further experimentation with other encoding techniques could further improve performance.

6. Conclusion

The project successfully developed a predictive model to forecast song popularity based on audio features using Random Forest Regressor. The model's optimization through preprocessing, feature engineering, and hyperparameter tuning led to satisfactory performance. The findings provide valuable insights into which audio features are most predictive of popularity. Future work could involve exploring other machine learning models, such as gradient boosting or deep learning, to further improve accuracy.

7. Contributions

- **Prabhleen Kaur (0857194)**

Prabhleen initiated the project and introduced the idea of "Melody Metrics" to the team. She led the core technical work, starting with creating the initial machine learning model and handling data cleaning and feature engineering. In statistical analysis, she used the F-test and Filter method to select key features. Prabhleen conducted thorough exploratory data analysis (EDA) to understand the dataset, identifying patterns and relationships within the audio features. She also played a major role in building and validating the models, especially in optimizing the final Random Forest model. She analyzed feature importance to better understand the model's decisions.

- **Rajwinder Kaur (0831280)**

Rajwinder played a major role in documentation and visualization throughout the project. She supported early-stage dataset exploration and was primarily responsible for crafting effective visual narratives in both the EDA phase and final report. Moreover, she used Wrapper method for feature selection and also did Pearson correlation test. Her contributions to writing large sections of the report and presentation ensured that insights were communicated clearly and engagingly. She maintained consistency in design and language across deliverables, enhancing the overall quality of the team's output.

- **Vindya Senadheera (0857437)**

Vindya was the key force behind deployment. While also contributing to data preparation and feature selection, Her main task was developing and implementing the Flask web application that enabled real-time song popularity predictions, ensuring seamless user interaction and functionality. In addition to her deployment efforts, Vindya also handled hyperparameter tuning using RandomizedSearchCV and applied feature scaling techniques such as the Yeo-Johnson transformation and StandardScaler to optimize model performance. Her contributions extended to data preparation, where she conducted statistical analysis using T-tests and applied embedded methods for selecting important features. Vindya also created and maintained the project's GitHub repository, ensuring code clarity and facilitating effective team collaboration.

- **STM Chathurangi (0850982)**

Chathurangi contributed during the data cleaning and validation stages and worked closely with Vindya during the deployment phase, supporting both data and deployment tasks effectively. She also worked on T-test hypothesis testing and applied embedded methods for feature selection. Her testing and debugging assistance during Flask integration ensured the stability of the app structure. Chathurangi created the final presentation, showcasing her adaptability and team-oriented mindset throughout the project.

- **Ajay Haji Korbe (0852660)**

Ajay played a strong supporting role in testing, model validation, and documentation. He contributed by evaluating models, running comparison tests, and helping select the best-performing algorithm. Moreover, did hypothesis testing and assisted in wrapper method. He took the initiative to record meeting minutes and summarize progress across stages. Ajay also helped structure the final presentation and contributed feedback that helped polish the team's delivery and narrative.

8. References

- ✓ Spotify Web API – Source for audio feature data - https://developer.spotify.com/documentation/web-api/
- ✓ Pandas – Data manipulation and analysis - https://pandas.pydata.org/
- ✓ NumPy – Numerical operations and array handling - https://numpy.org/
- ✓ Scikit-learn – Machine learning algorithms, preprocessing, and model evaluation - https://scikitlearn.org/
- ✓ Category Encoders – Encoding techniques for categorical features (e.g., TargetEncoder) - https://contrib.scikit-learn.org/category_encoders/
- ✓ Joblib – Model serialization and efficient object storage - https://joblib.readthedocs.io/
- ✓ Flask – Lightweight web application framework (if used for front-end/backend integration)- https://flask.palletsprojects.com/
- ✓ PyCaret -Low-code machine learning library for quick prototyping and deployment- https://pycaret.org/
- ✓ Statista, 2024. Spotify and music streaming in the Netherlands - statistics & facts. Statista. Available at: https://www.statista.com/topics/4699/spotify-and-music-streaming-in-the-netherlands/#topicOverview
- ✓ Saragih, H.S., 2023. *Predicting song popularity based on Spotify's audio features: insights from the Indonesian streaming users*. Journal of Management Analytics, 10(4), pp.693-709. Available at: https://doi.org/10.1080/23270012.2023.2239824
- ✓ Jung, N.S. and Mayer, F., 2024. *Beyond Beats: A Recipe to Song Popularity? A machine learning approach*. University of Innsbruck, pp. 2-15. Available at: https://doi.org/10.13140/RG.2.2.22260.16007
- ✓ Li, K., 2024. *Predicting song popularity in the digital age through Spotify's data*. Theoretical and Natural Science, 39(1), pp.68–75. Available at: https://doi.org/10.54254/2753-8818/39/20240600
- ✓ Breiman, L., 2001. RANDOM FORESTS. Machine learning, 45(1), pp.5–32.Available at: https://link.springer.com/article/10.1023/A:1010933404324
- ✓ Yeo, I.K. and Johnson, R.A., 2000. A NEW FAMILY OF POWER TRANSFORMATIONS TO IMPROVE
- ✓ NORMALITY OR SYMMETRY. Biometrika, 87(4), pp.954–959 Available at: https://academic.oup.com/biomet/article/87/4/954/252073