# Predicting Spotify's Song Popularity


This project uses a Kaggle dataset of Spotify tracks https://www.kaggle.com/datasets/karthikgangula/spotify_songs.csv to predict song popularity based on rich audio, metadata features and popularity scores. The aim is to understand what makes a song popular and build a machine learning model to forecast popularity scores (0–100). The Key Question is: **"**Can we predict a song's popularity using Spotify-provided features."

The goal is to **predict a song's popularity** using its available metadata and audio features. Success means building a machine learning model that can accurately predict the popularity score (regression task), ideally minimizing prediction error (e.g., RMSE or MAE) and maximize $R^2$ (R-squared) while also offering insight into what makes a song more likely to be popular in addition to providing reusable code.

Previous studies and projects have shown that audio features alone can give moderate predictive power for popularity, but including contextual factors like release year, playlist genre, or artist information often improves performance.

Several known factors may impact a song's popularity:

- **Audio features**: e.g., energy, danceability, valence, loudness.
- **Playlist genre**: the type of music can impact exposure and audience size.

- **Release date**: newer songs may be favored in Spotify's popularity algorithm.
- **Artist popularity**: some artists have a large fanbase that boosts popularity.
- **Playlist inclusion**: being added to popular playlists significantly increases visibility.

While many models focus solely on numerical audio features, we plan to experiment with:

- **Feature engineering** manipulate and extract data from text columns
- **One-hot encoding playlist genres** to capture the influence of music type.
- **Feature interactions**: e.g., how tempo interacts with danceability.
- **Feature selection** to understand feature importance.
- **Ensemble models**: combining models like XGBoost, Random Forest, and Linear Regression for better performance.

The stages of the project:

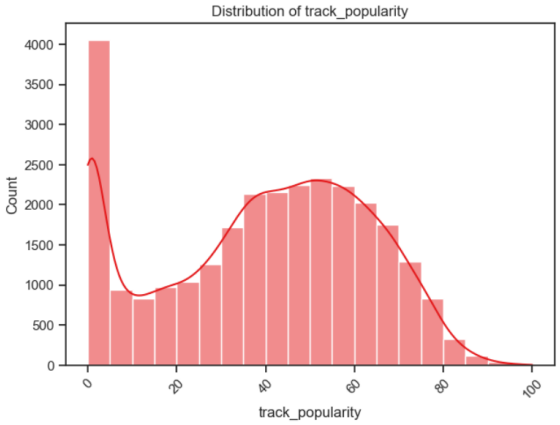
**Data Preparation:**

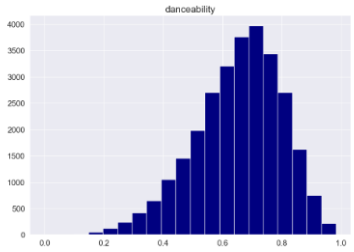
In this stage we aimed to form a “flat file” preparing the dataset for further analysis and model training. Key steps included:
Loading and checking the structure of the spotify_songs.csv file.
Removing Bias contributing column “subgenre”. Converting datatypes into the appropriate ones so they will be valid for processing. Creating a new ‘unique_playlist_count' column that sums the number of playlists a song has to replace the duplicates of a single “track_id” found in multiple playlists.
creating dummies for “playlist_genre” in order to aggregate “track_id” into a single row. normalize text by lowercasing, removes stop words, punctuation, and special characters, but keeps numbers and parentheses.
Text based columns were extracted into a separate DataFrame to facilitate feature engineering.

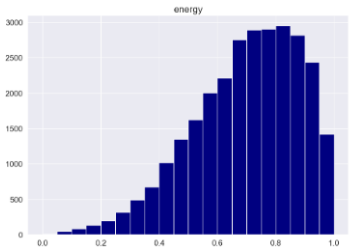
**EDA – Explanatory Data Analysis**

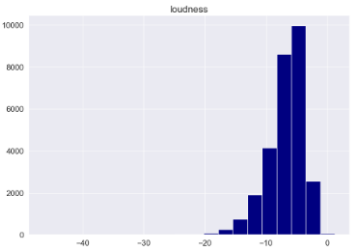
The dataset was explored to understand key distributions, correlations, and relationships between features. Main steps included:

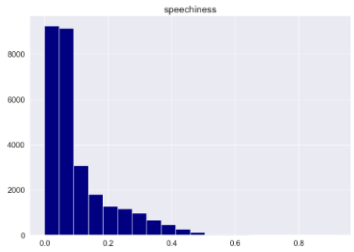
- **General inspection** of feature types, unique values, and basic statistics.
- **Visualization of numerical and categorical distributions** using histograms and boxplots to detect skewness and outliers.

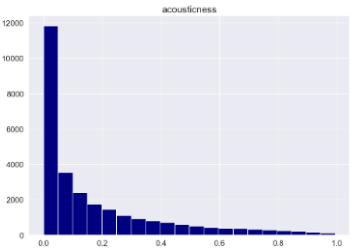

Distribution of track_popularity
Count
track_popularity

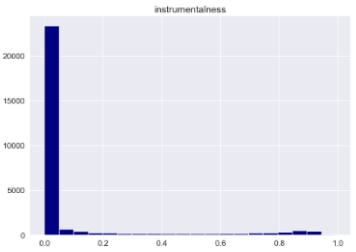

danceability

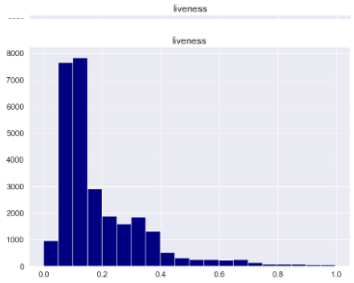

energy

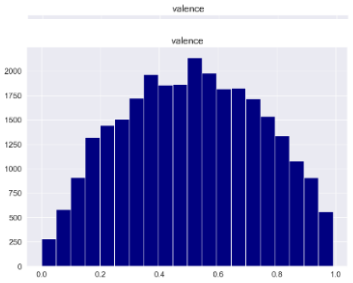

loudness

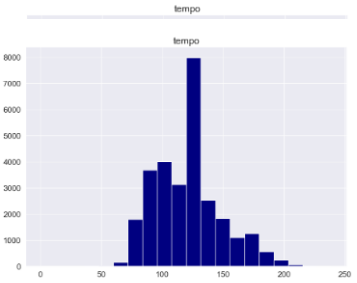

speechiness

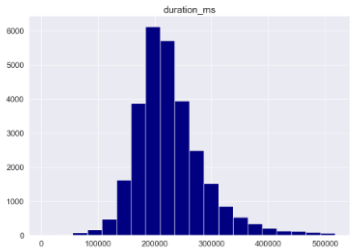

acousticness

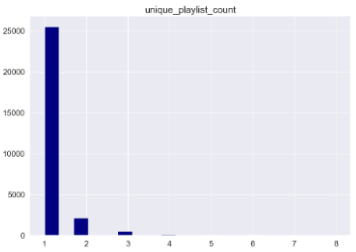

instrumentalness

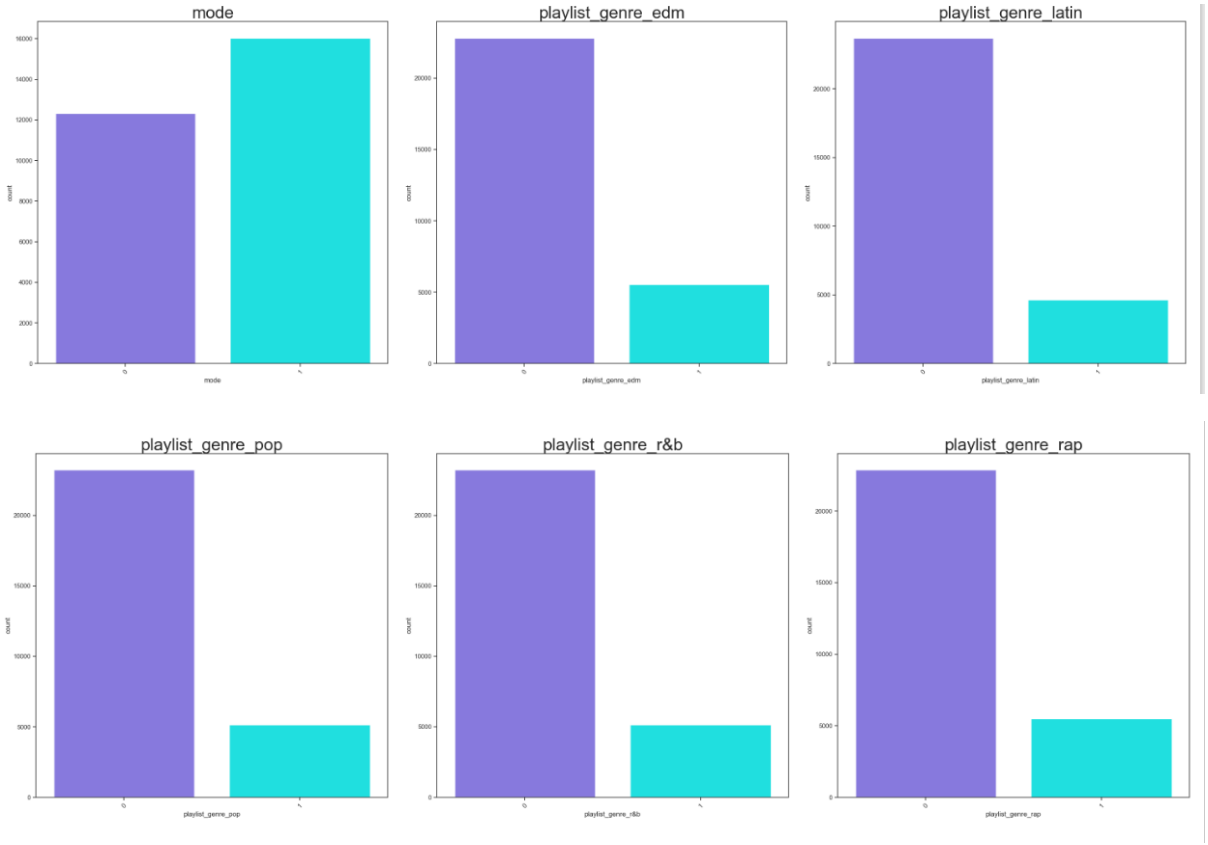

liveness
liveness

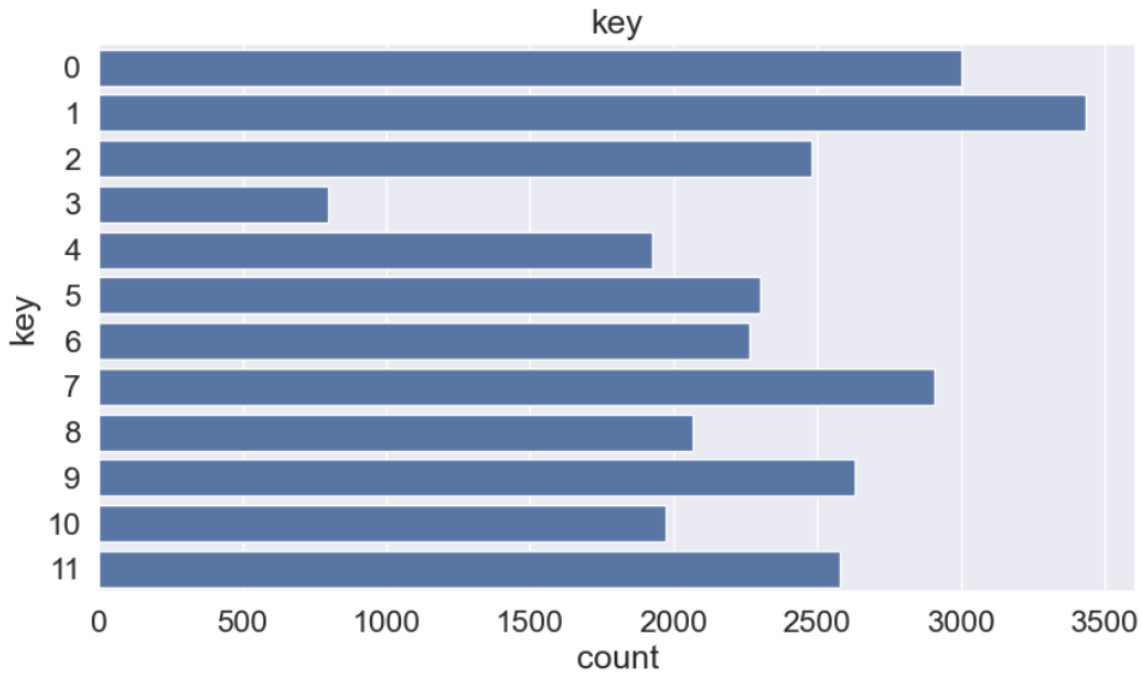

valence
valence


tempo
tempo

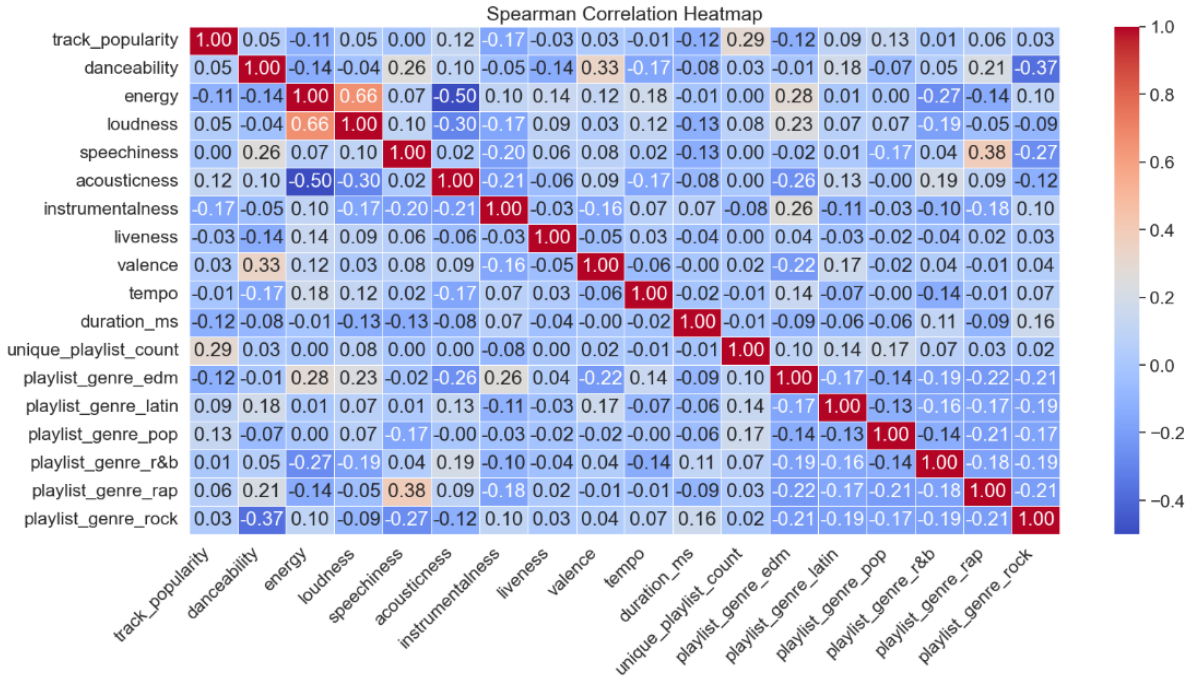

duration_ms


unique_playlist_count




- **Correlation analysis** using a heatmap to identify relationships between audio features.




- **Genre and popularity insights**, including bar plots and boxplots to explore how potentially important features relate to popularity.

1. **Data Cleansing – Outliers and Missing Values**
   by visualizing popularity extremes across features, This stage provided essential insights into feature behaviour and their potential impact on predicting track popularity. Outliers were detected and replaced by MICE algorithm for handling missing values

2. **Feature engineering**
   In this stage, the dataset was transformed to creatively extract more valuable data from the dataset and support proper modelling. Key steps included: Adding a new column, 'five popular words' which contains the five most frequently occurring words from the 'track name' column across all rows. scaling time values from milliseconds to seconds and extracting year and month from dates.

3. **Model Selection and finetuning**
   This step focused on identifying the most relevant features to improve model performance and reduce overfitting. Key actions included:
   Performing few regression models and analysing their results with the appropriate metrics finishing with the best model for production.
   This process helped simplify the model, reduce redundancy, and retain features most predictive of track popularity.

4. **Model**

This part contains the implementation, comparison, and fine-tuning of multiple regression models to predict Spotify song popularity using engineered features. several regression models were trained to predict track popularity, including Linear Regression, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, SVM, and XGBoost. The models were evaluated using four key metrics: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and $R^2$ (R squared). The results indicated that Random Forest outperformed all other models, achieving the lowest MAE, making it the most accurate model overall. RandomizedSearchCV was employed to search for best hyperparameters . There was a small Improvement of 0.24% on Finetuning, we will proceed with the improved model. Take into consideration that the $R^2$ value low meaning the model explains small portion of the variance in the target. There's likely room for improvement by Feature engineering and/or Hyperparameter tuning.

**How Will We Deploy the Machine Learning?**

The machine learning system developed in this project is designed to **predict the popularity score of a song** based on its audio features and metadata. The trained machine learning model ( Random Forest) will be saved as a serialized object using tools like joblib or pickle.

1. **Backend Integration**
   The model can be wrapped in a lightweight **API using FastAPI or Flask**, which accepts song features as input and returns a predicted popularity score. This API can be hosted on a cloud platform such as AWS, Google Cloud, or Heroku.

2. **User Interface (Optional)**
   A simple web dashboard or app can be built to allow users (e.g., producers or artists) to input song parameters and view predictions. This UI can also display feature importance to provide explainability.

3. **Batch Predictions and Insights**
   For larger-scale use, the model can be integrated into data pipelines to score batches of songs, allowing organizations to continuously monitor trends and predict future hits.

4. **Continuous Improvement**
   As more song data becomes available, the model can be retrained periodically to adapt to evolving music trends and listener preferences.

This machine learning system can provide value to several stakeholders in the music industry:

**Artists and Music Producers** Get early feedback on new tracks during the production phase which Helps make data-driven decisions about which songs to prioritize or promote.

**Record Labels and Marketers Use Case**: Evaluate a large catalogue of unreleased tracks to identify potential hits and Allocate marketing resources efficiently toward high-potential songs.

**Streaming Platforms (e.g., Spotify) Use Case**: Enhance recommendation systems or curate dynamic playlists based on predicted popularity by that Improve user engagement and satisfaction.

**Data Analysts and Researchers** Explore trends across genres and time periods to Generate insights for content strategy and long-term planning.