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 along with 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) 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 favoured 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/numeric 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:

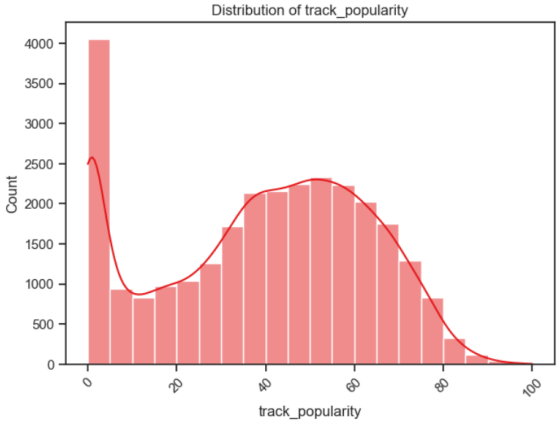
In this stage I aimed at forming a “flat file” preparing the dataset for further analysis and model training. Key steps included:

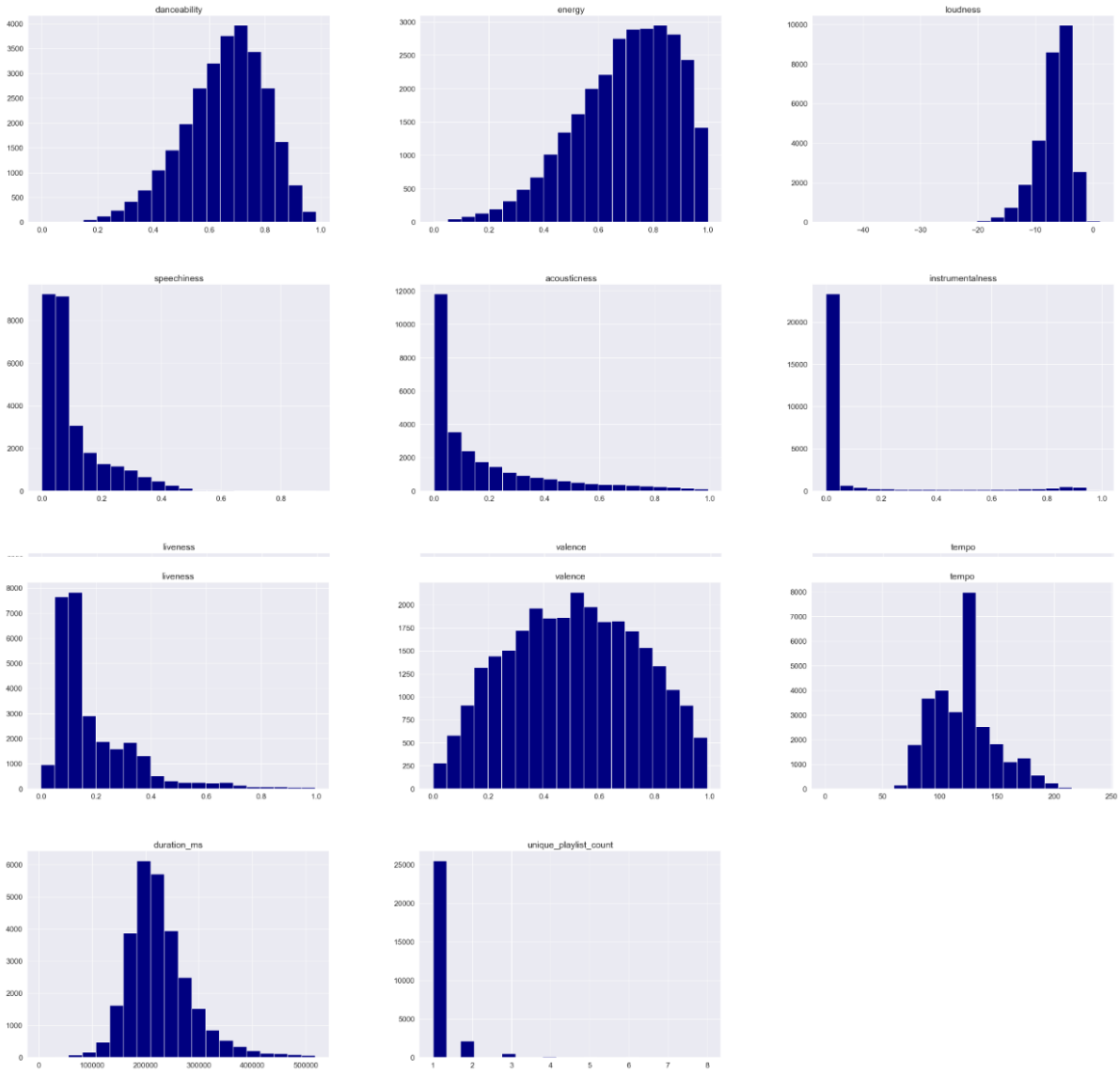
- Loading and checking the structure of the spotify_songs.csv file.
- Removing Bias contributing column “subgenre”.
- Converting datatypes into the appropriate ones so they will be valid for processing.
- Creating a new ‘unique_playlist_count' column that sums the number of playlists a song must replace 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 Data Frame to facilitate feature engineering.

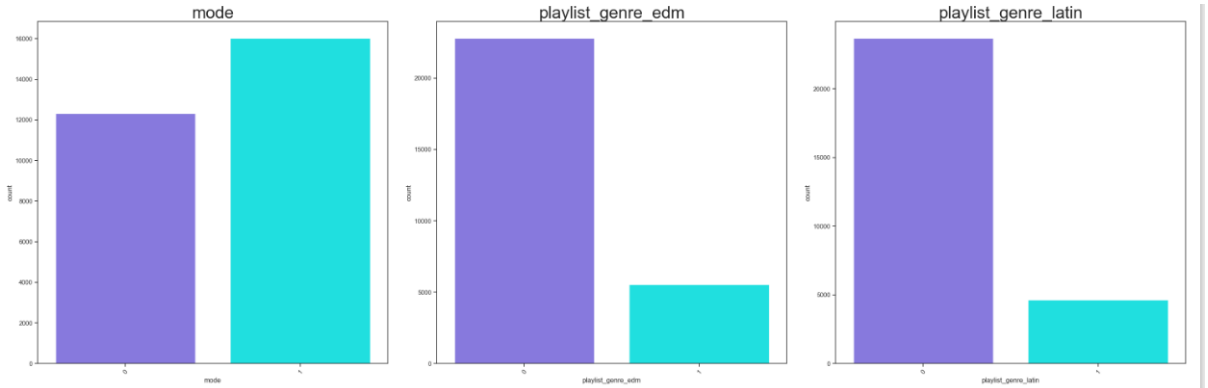
EDA – Explanatory Data Analysis

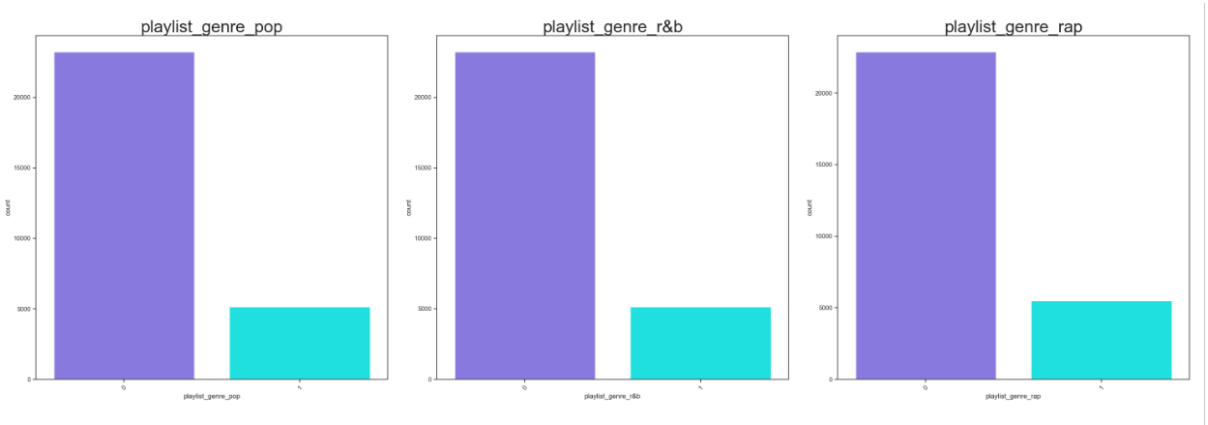
The dataset was explored to understand key distributions, correlations, and relationships between features. Main steps included:

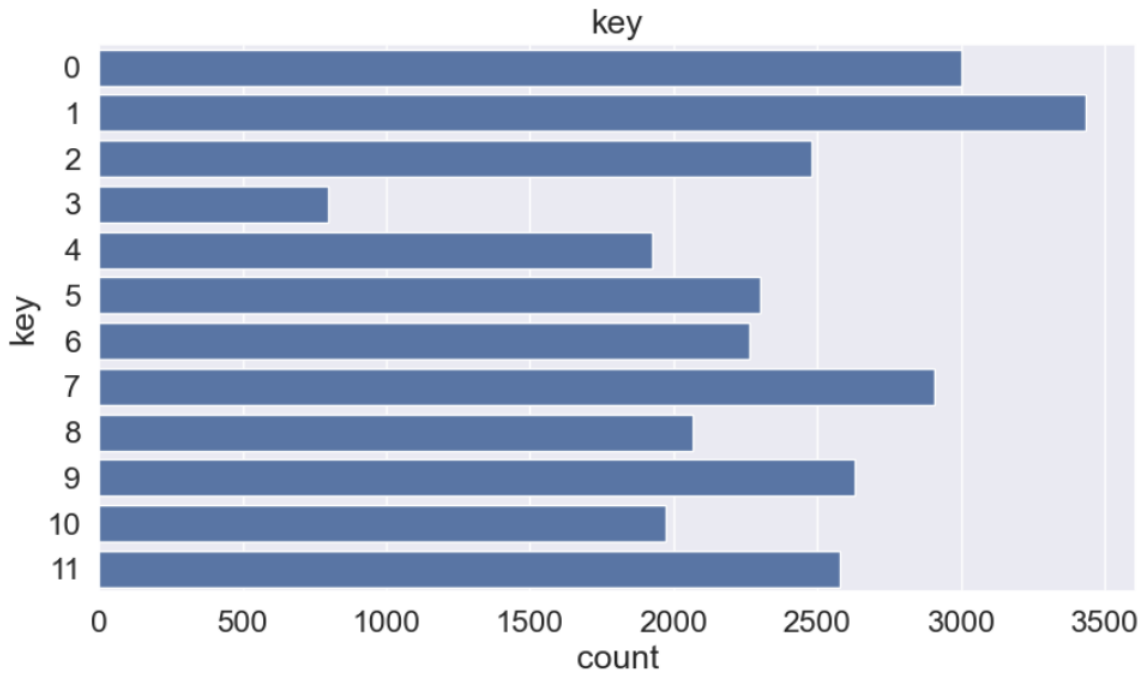
- **General inspection** of feature types, unique values, and basic statistics.
- **Visualization of numerical and categorical distributions** using histograms and boxplots to detect skewness and outliers.





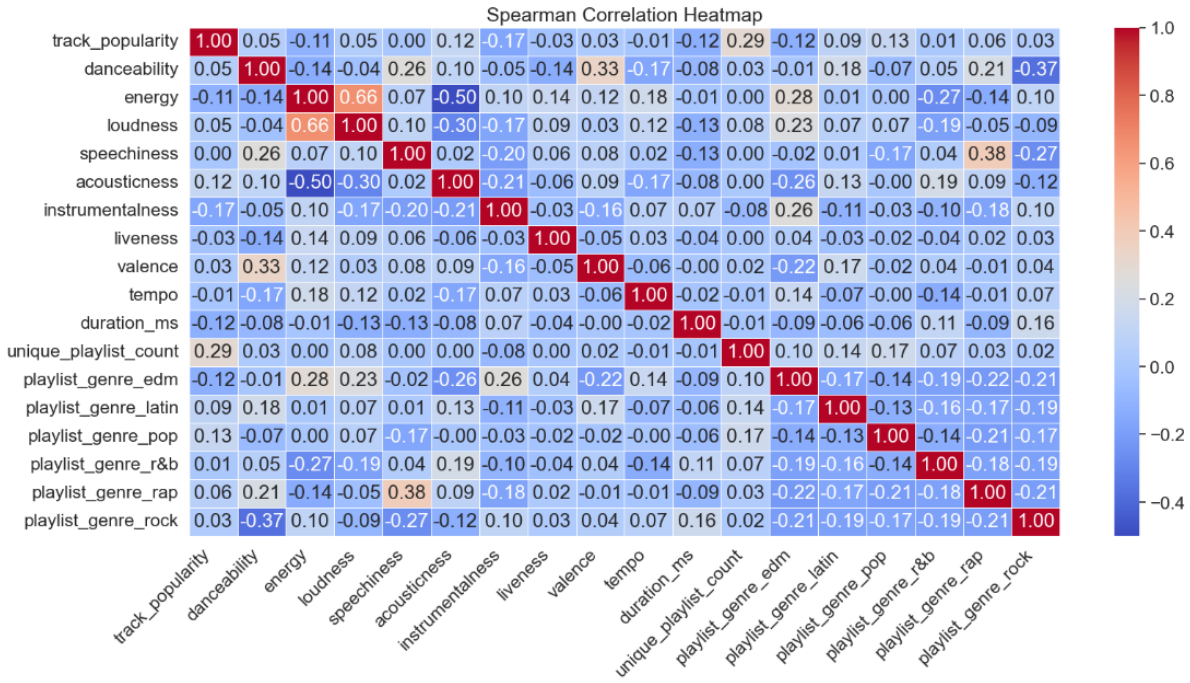

mode
playlist_genre_edm
playlist_genre_latin
count
mode
playlist_genre_edm
playlist_genre_latin


playlist_genre_pop
playlist_genre_r&b
playlist_genre_rap
count
playlist_genre_pop
playlist_genre_r&b
playlist_genre_rap


key
0
1
2
3
4
5
6
7
8
9
10
11
0
500
1000
1500
2000
2500
3000
3500
count

	skewness
unique_playlist_count	5.035169
instrumentalness	2.624985
liveness	2.081481
speechiness	1.964975
acousticness	1.576720
duration_ms	1.115164
tempo	0.513949
valence	-0.007334
danceability	-0.505789
energy	-0.645440
loudness	-1.359303

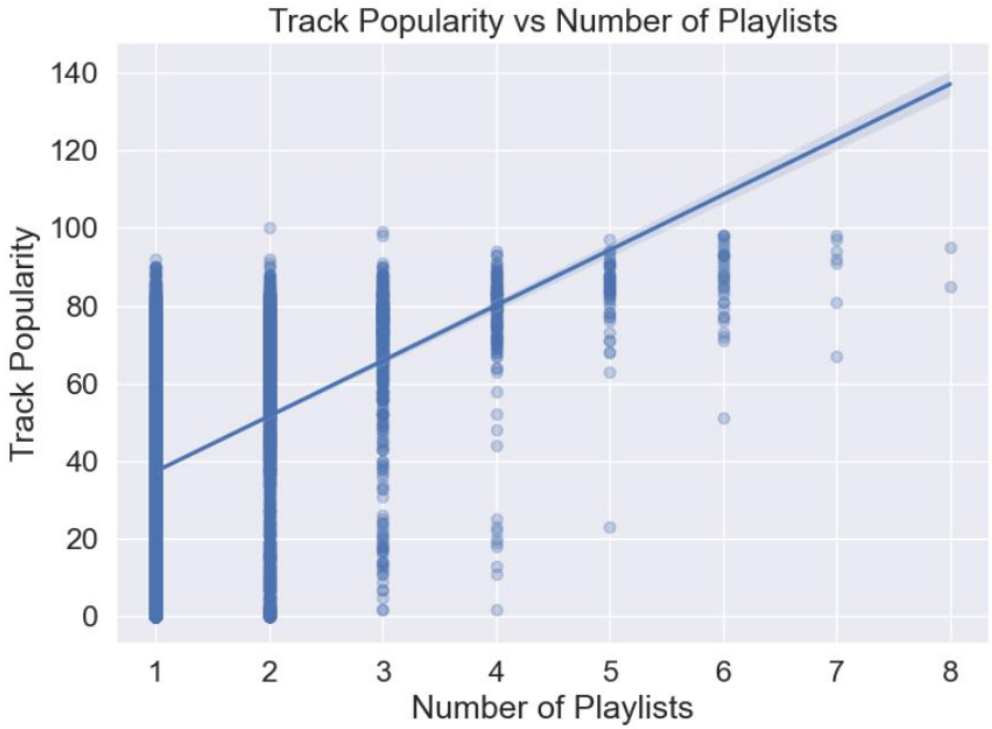
- **Correlation analysis** using a heatmap to identify relationships between audio features.



TTEST analysis revealed that the difference in track_popularity before and after 2010 is statistically significant. The main reason can be:

- Changes in Music Consumption (Streaming Era vs. Pre-Streaming) Before 2010 (Physical & Digital Downloads) CDs, MP3s, and iTunes dominated the market.
- Popularity was measured by radio airplay & album sales.
- Artists had longer-lasting hits because people bought albums instead of streaming single songs.

- After 2010 (Streaming Era) Spotify (2008), Apple Music (2015), YouTube, and TikTok changed music consumption.
- Popularity is now driven by streams, playlists, and social media trends. Songs have shorter lifespans because of streaming algorithms favoring fresh content.



The number of playlists a track is in influences its popularity — positively and significantly.

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

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.

- Adding a new column "before_2010" - Since the TTEST in the EDA showed a significant difference between songs that were released before 2010 and after, the column was added to the data frame for future modeling.
- Adding a new column "number_tracks_artist" – this can give hints about the artist's fame or visibility. If an artist has lots of tracks in the dataset, they're probably more well-known, more active, or more likely to be picked up by Spotify's algorithm. Famous artists often release more songs, get more playlist placements, and have a bigger fanbase leading to higher popularity scores.
- Adding new column "release_year" - computes the number of years since each song was released. It will capture a different angle: Instead of absolute time, it captures how old a song is — which is often more relevant to popularity modeling.

Model Selection and finetuning

This step focused on identifying the most relevant features to improve model performance and reduce overfitting.
Several regression models were performed, and results were analysed with the appropriate metrics finishing with the best model for production. 27 selected features continued to be modelling.

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 0.15% on Finetuning that The R^2 value was 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.

Production

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. It will connect to

the Spotify API and will be able to predict whether a song has the potential to become popular On a scale of 1-100.

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.