Search

Home

Library

Our Music Playlists

Spotify®

Predicting Streams

Problem

- Difficulty in forecasting the success of a song.
- The music industry is increasingly data-driven. It’s critical to understand the impact of musical characteristics, artist popularity, exposure, and industry recognition on the success of songs.

Why it matters?

- Strategic decision-making for record labels and artists
- Revenue forecasting
- Reducing risk in a hit-driven industry

Data Structure

Spotify Top Songs and Audio Features (Kaggle)

Outcome Variable: Streams (Integer)

Key Predictors:

- **Audio Features**
  - Tempo, Danceability, Energy, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Loudness, Duration
- **Categorical Song Features**
  - Key, Mode, Time Signatures
- **Artist Popularity Features**
  - Artist Popularity, Number of Collaborators

Data Structure

Spotify Top Songs and Audio Features (Kaggle)

Outcome Variable: Streams (Integer)

Dataset:

- Rows: 6513 unique songs
- Columns: 25 variables (full dataset)
- Selected Features for Modeling: 20 variables (3 categorical, 17 numerical)
- Variable types: integer, string, decimal
- Training/Testing: 80/20

Feature Engineering

Added Popularity Features:

Number of collaborators

Artist Followers

Artist popularity

Total Grammys

Total Nominations

Tiktok Viral

Data Preprocessing

Data conversion:

Converted TikTok_Viral
Viral ⟶ 1, Not Viral ⟶ 0

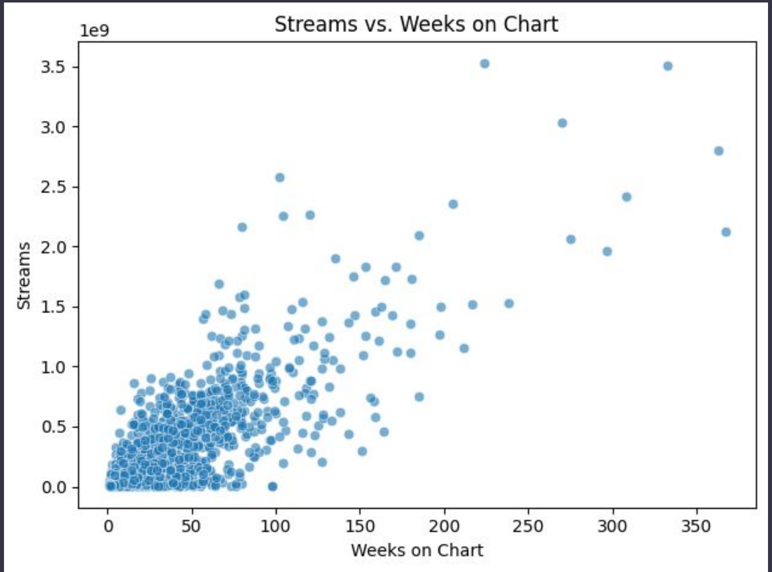
Removed identifiers:

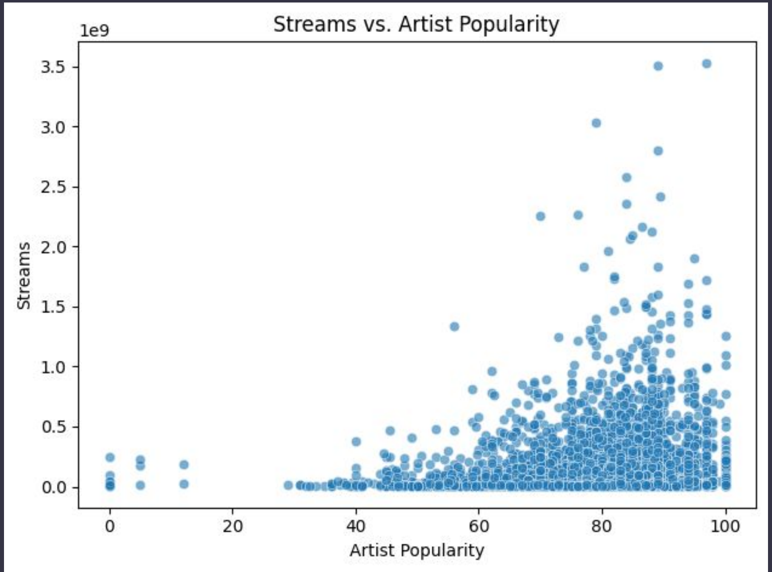
streams, id, artist_names, track_name, source

Dummy variables:

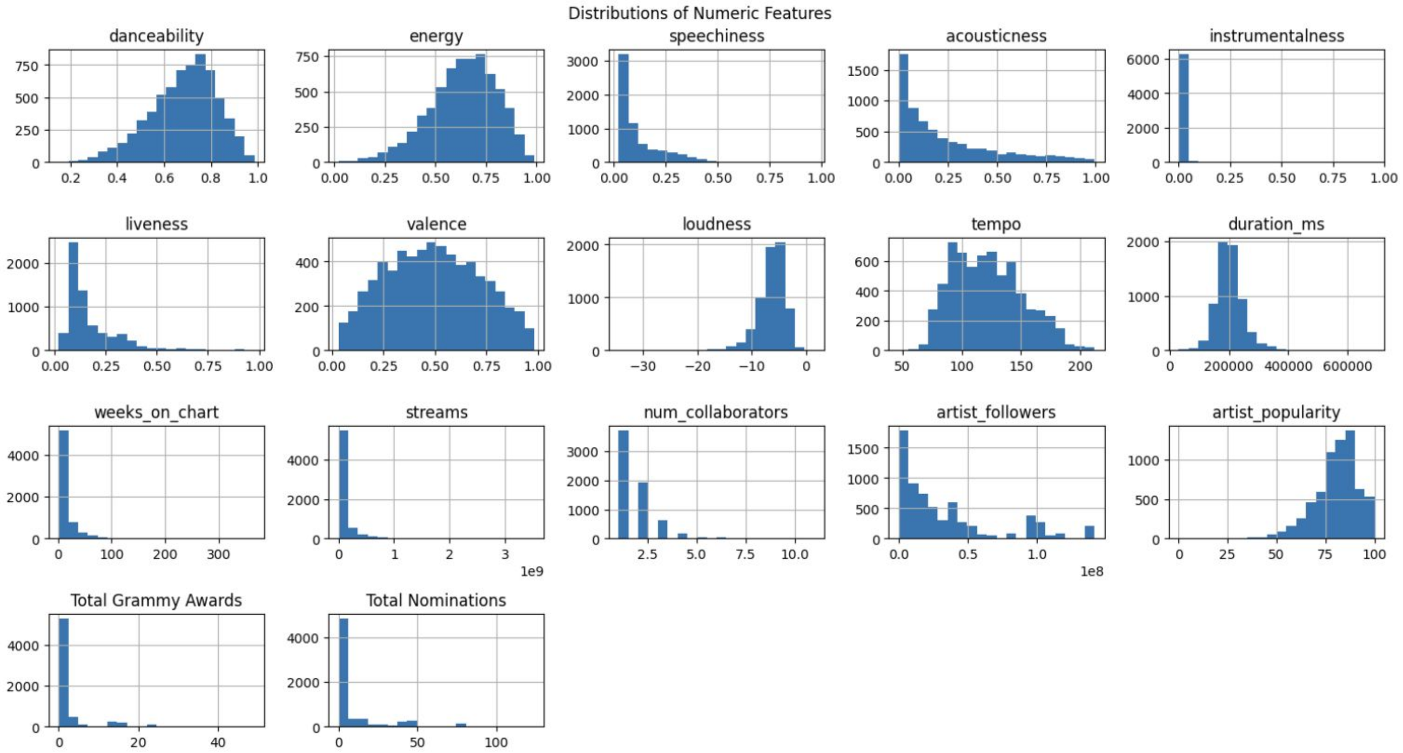
key, mode, time_signature

Data Exploration

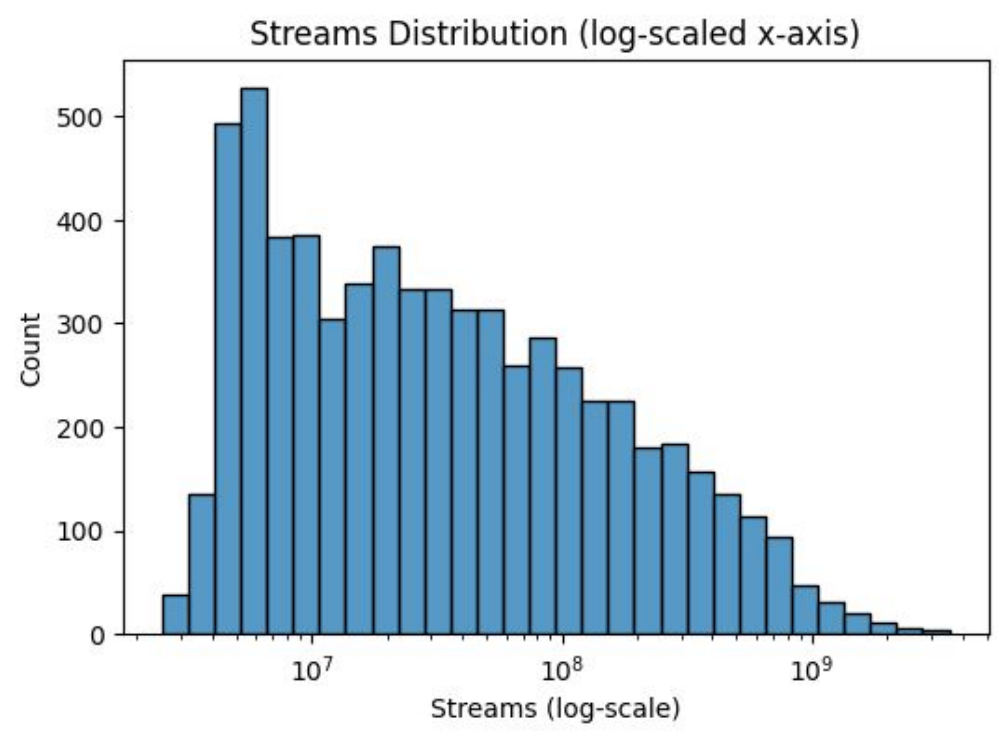




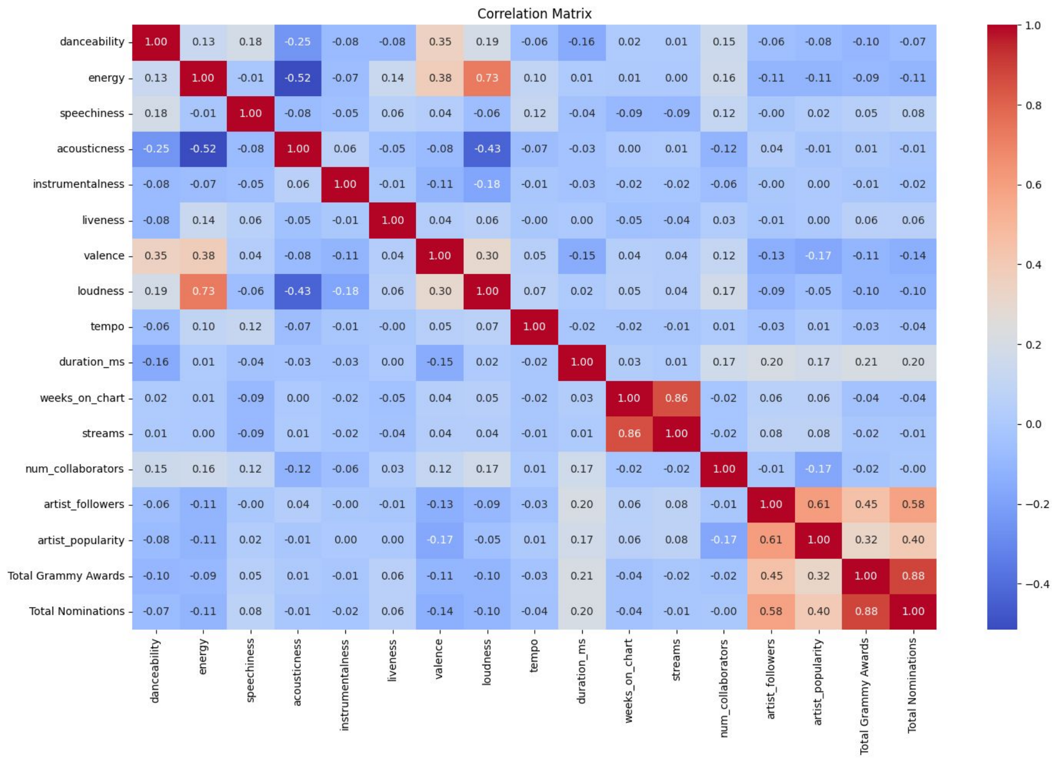
Data Exploration



Data Exploration



Data Exploration



Overall Approach

1. **Initial Model Exploration**
   - Ran 10 models to find promising models for predicting Spotify streams.
2. **Structured Model Evaluation with Nested Cross-Validation**
   - Used nested cross-validation to tune hyperparameters and select the best-performing model based on generalization performance.
3. **Hyperparameter Optimization and Final Model Assessment**
   - Optimized learning rate, number of estimators, and tree depth; evaluated final model performance using MSE and R^2 scores.

Models

- Baseline Model
- Linear Regression with CV
- Ridge Regression
- Lasso Regression
- Elastic Net
- K-Nearest Neighbor
- Decision Tree with pruning
- Random Forest
- AdaBoosting
- Gradient Boosting

Baseline Model

MSE: 2.2136
R^2: -9.7698e-10

Ridge

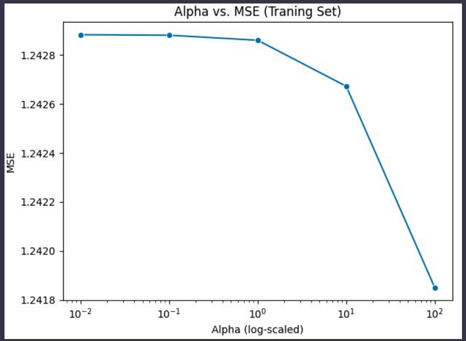
- alpha = 100
- Test MSE = 1.0758
- R^2 = 0.5140

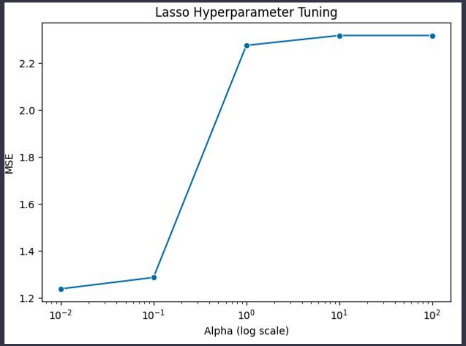
Lasso

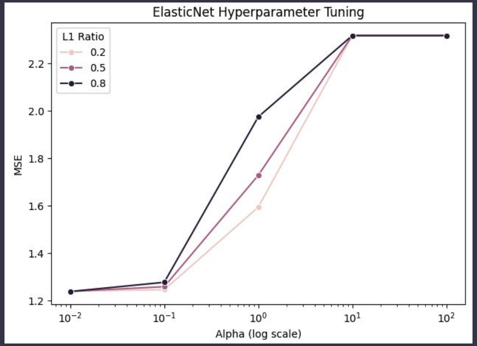
- alpha = 0.01
- Test MSE = 1.0707
- R^2 = 0.5163

Elastic Net

- alpha = 0.01
- l1 ratio = 0.8
- Test MSE = 1.0714
- R^2 = 0.5160







ADA Boosting

Parameter Grid:

# of Estimators

- 50, 100, 150

Learning Rate

- 0.01, 0.1, 0.5

Max Depth

- 2, 3 ,5

Test MSE

0.3664

Test R^2

0.8345

Gradient Boosting

Parameter Grid:

of Estimators
- 100, 200

Learning Rate
- 0.01, 0.1

Max Depth
- 3 ,5

Test MSE	Nested CV Test MSE
0.3654	0.4292

Test R^2	Nested CV Test R^2
0.8349	0.8126

Pruned Decision Tree

Test MSE

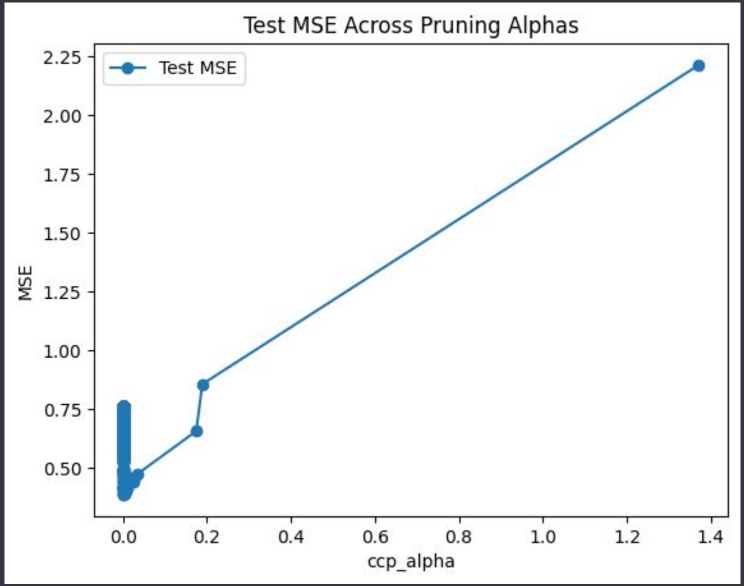
0.3860

Test R^2

0.8256

Alpha

0.002465



Random Forest

Parameter Grid:

# of Estimators
- 100, 200, 300

Max Depth
- None, 10, 20, 30

Test MSE

0.3760

Test R^2

0.8301

Comparing All Models

	Best Hyperparameter	Test MSE	R^2
Baseline Model		2.2136	-9.7698e-10
Linear Regression		1.0719	0.5158
Ridge Regression	alpha = 100	1.0758	0.5140
Lasso Regression	alpha = 0.01	1.0707	0.5163
Elastic Net	alpha = 0.01, l1 ratio = 0.8	1.0714	0.5160
K-Nearest Neighbor	k = 16	1.5263	0.3105
Decision Tree with pruning	ccp_alpha = 0.002465	0.3860	0.8256
Random Forest	max_depth: 10, n_estimators: 300	0.3760	0.8301
AdaBoosting	Max_depth: 5, learning rate = 0.01, n_estimators: 50	0.3664	0.8345
Gradient Boosting	Max_depth: 3, learning rate = 0.1, n_estimators: 100	0.3654	0.8349

Best Model: Gradient Boosting

With the lowest MSE and highest R^2 Gradient Boosting outperformed every other model we ran!

Best Parameters

- # of Estimators: 100
  - Enough trees to reduce bias without overfitting
- Learning Rate: 0.1
  - To improve steadily and avoid overshooting
- Max Depth: 3
  - Deep enough to capture complex relationships but shallow enough to avoid overfitting

Test MSE:

0.3654

Test R^2:

0.8349

Challenges

#1 Lack of Popularity Features
Added 7 new columns to predict the influence of artist popularity.

#2 Raw stream number were large and skewed
Log-transform stream to improve model stability and interpretation.

#3 Nested CV long runtime
Limited the hyperparameter grid, yet still experience long runtime.

What we Learned

Experimentation and Nested CV Matters

Ensemble Methods

Stream forecasting to drive decision making in the music industry

Our Music Playlists

Search

Home

Library

Thank you!

Questions?