Spotify

Analyzing The Relationship Between Genre Count And Artist Success

Objective

- Investigate whether genre diversity (number of genres an artist spans) is associated with higher artist popularity.
- Analyze and compare results across primary (API + charts) and secondary (Kaggle) datasets.
- Evaluate the strength of this relationship using exploratory analysis and regression models.

Spotify

Data Collection

1. Primary Dataset:

- Weekly Spotify Top-200 Charts from 20 global regions, downloaded as CSV files.
- Extracted unique artist names and enriched them using the Spotify Web API via Spotipy to retrieve:
  1) Artist ID
  2) Genres
  3) Popularity score

2. Secondary Dataset:

- A cleaned Spotify dataset from Kaggle, containing artist popularity, followers, genres, and metadata.

Spotify

Data Preprocessing

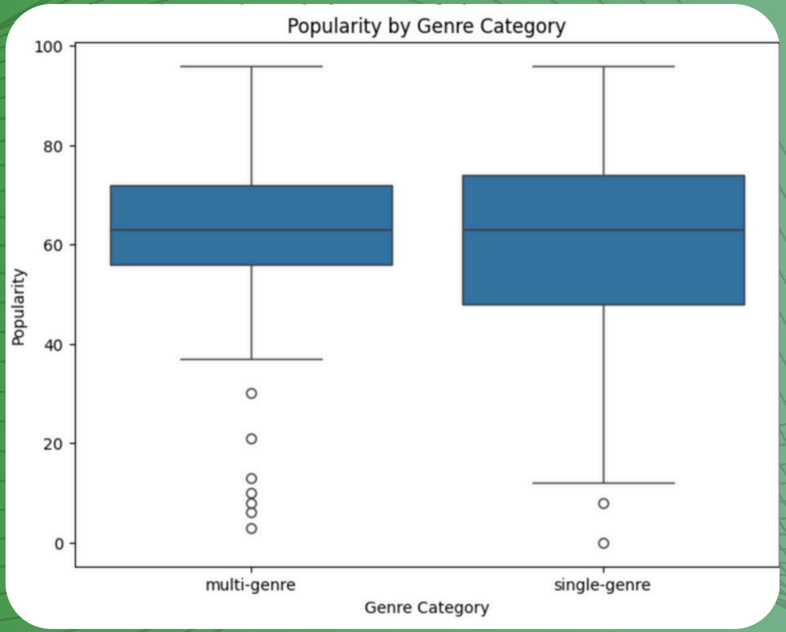
- Key steps applied to both datasets:
  - Removal of duplicates (artist-level).
  - Dropping artists with missing or empty genre information.
  - Cleaning API inconsistencies (missing IDs, rate-limit exceptions).
  - Transforming list-based genres into string format.

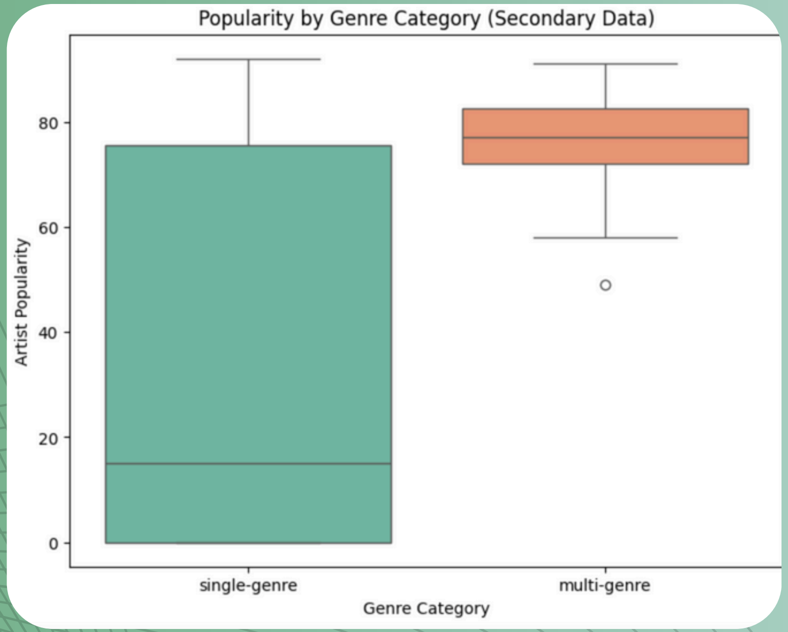
- Engineering two core features:
  - genre_count (number of genres an artist spans)
  - genre_category ("single-genre" vs "multi-genre")

Spotify

Exploratory Data Analysis

- Visualizations: heatmaps, scatter plots, histograms, boxplots.
- Relationship focus: Does genre diversity correlate with higher popularity?
- Initial findings:
  - Primary dataset shows a weak positive correlation (≈0.12).
  - Secondary dataset shows a much stronger correlation (≈0.60).





Modelling Approach

Since popularity is a continuous value (0–100), the modelling task is regression.

Process:

1. Build a baseline model using only genre_count.
2. Extend the model with additional features (enconded genres)
3. Train multiple regression models: Linear Regression, Random Forest, and XGBoost
4. Use cross-validation to evaluate MAE, RMSE, and R^2.
5. Compare models and select the best-performing one for interpretation.

Spotify

Algorithm Selection & Rationale

1.. Baseline model (genre_count only)

- Simple, interpretable line between genre_count and popularity.
- Serves as a benchmark to see whether extra features and more complex models are worth it.

2. Linear Regression

- Captures the combined effect of genre_count, followers, and specific genre indicators.
- Coefficients are directly interpretable and show which genres are most associated with higher popularity.

3. Random Forest Regressor

- Nonlinear, tree-based ensemble model.
- Handles interactions between features and skewed popularity values.
- Provides feature importances for understanding which genres and features matter most.

4. XGBoost

- Boosting model that focuses on reducing the remaining error step by step.
- Typically gives the best R^2 and lowest error among the tested models.

Why these models?

They represent a progression from simple to complex models:

- Baseline Model: establishes the minimum expected performance.
- Linear Regression: adds more features and provides interpretable coefficients.
- Random Forest: captures nonlinear patterns and interactions between features.
- XGBoost: optimized boosting method that usually delivers the strongest performance.

Spotify

Evaluation Metrics Used

1. Mean Absolute Error (MAE)

- Measures average magnitude of errors.
- Good for interpretability since it's in the same units as popularity (0–100).

2. Root Mean Squared Error (RMSE)

- Penalizes large errors more heavily.
- Helpful because popularity has extreme outliers.

4. Coefficient of Determination (R^2)

- Measures how much of the variance in popularity is explained by the model.
- Useful for comparing baseline vs. improved models.

Spotify

Evaluation Metrics Used

Why these metrics?

- Popularity is a continuous numeric target, so regression metrics (MAE,, RMSE) are the correct choice.
- The data contains outliers and skew, especially in popularity distribution → RMSE handles this better by penalizing large errors.
- MAE provides interpretability because it's in the same units as popularity (0–100).
- RMSE help capture bigger mistakes, which is important since genre_count alone is a weak predictor.
- Using different metrics ensures balanced evaluation instead of relying on a single measure.
- R^2 complements these metrics by showing how much variance the model explains, which is useful when comparing the baseline to improved models.

What the metrics revealed:

- The baseline model using only genre_count achieved a very low R^2 ($\approx$ 0.05) and relatively high error.
- Adding followers and encoded genre features clearly improved performance:

1. Linear Regression reached $R^2 \approx$ 0.32–0.33.
2. Random Forest improved R^2 further to around 0.37–0.38.
3. XGBoost achieved the best R^2 (around 0.40+) with the lowest MAE/RMSE.

- Even with these improvements, a large part of the variance remains unexplained, meaning popularity is only partially driven by genre-related features.

Spotify

Technical Hurdles & Solutions

1. Spotify API Rate Limits
Challenge: Frequent timeouts + API throttling while fetching hundreds of artists.
Solution: Added delays, used try/except, and saved partial results locally to avoid repeating API calls.

2. Missing or Empty Genre Data
Challenge: Some artists returned empty genre lists → breaking feature engineering.
Solution: Used safe_parse() to clean malformed genre fields and filtered out entries with empty genres.

3. Duplicate or Alias Artist Entries
Challenge: Same artist appearing multiple times under different names or IDs.
Solution: Removed duplicates by Spotify ID after merging all CSV files.

4. Representation Bias Across Regions
Challenge: Larger markets (US, UK, Global) dominated the dataset.
Solution: Extracted unique artists only and validated results using the secondary dataset.

Spotify

Technical Hurdles & Solutions

5. Collecting artist data
Challenge: Spotify's API doesn't give artist data directly
Solution: Used Spotify Charts to extract top regional charts and used artist's name from the track to get artist data

6. Weak Feature Predictive Power
Challenge: Having only num_genres made modelling unstable and limited.
Solution: Tested multiple models (Baseline Linear, Multiple Linear, Random Forest, XGBoost) and used MAE/RMSE/R^2 to confirm model limitations.

Spotify

Future Improvements

1. Extend the feature set beyond genres and followers to include track-level signals (streams, weeks on chart, peak rank) and temporal trends.

2. Explore more advanced ensemble methods (e.g., stacking models that combine linear, Random Forest, and XGBoost predictions).

3. Use more systematic hyperparameter tuning and repeated cross-validation to stabilize performance estimates.

4. Investigate non-tabular signals (e.g., lyrics or audio embeddings) to capture more of the unexplained variance in popularity.

Spotify

Thank You!

Course Instructor : Dr. Abeer Aldayel

Alya Almuqren — 444200610

Dana Alsalami — 443200945

IT362 | 80657
Group #5

Norah Aljayan — 444200832

Aljohara Alsultan — 444203635

Layan Alhowaimel — 444200969