Spotify

Presented by Group 2

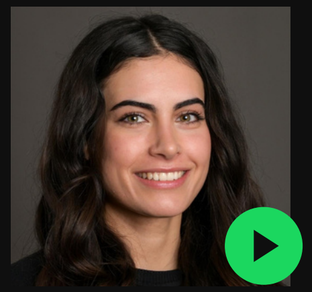
Predicting Song Popularity

with Spotify Data

Table of Content

#	Agenda Titles		
1	End-to-End Pipeline & Business Context	09:00 AM	
2	Data Preparation	10:00 AM	
3	Feature Engineering & Insights	11:00 AM	
4	Target Variable Binning	12:00 PM	
5	Model Training & Comparison	01:00 PM	
6	Overfitting & Generalization	02:00 PM	
7	Feature Importance	01:00 PM	
8	Final Model Recommendation	02:00 PM	
9	Key Takeaways & Next Steps	02:00 PM	

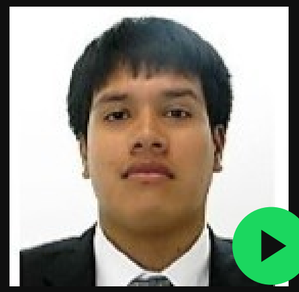
Our Team



Abril

Ana

Juan David



Juan Luis

Konstantin

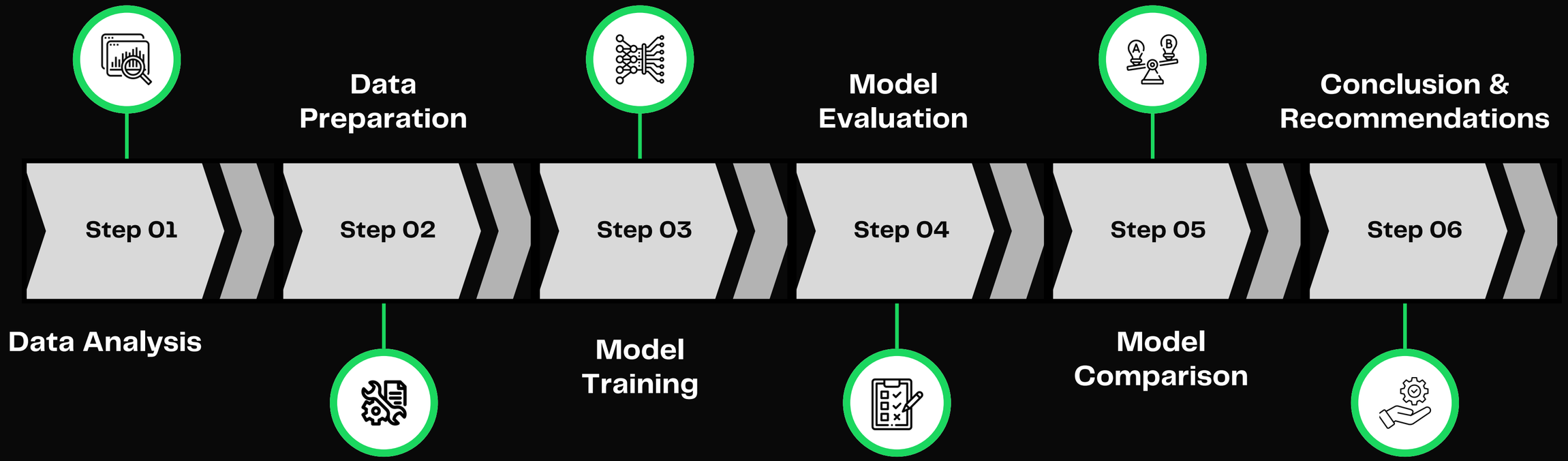
Mohammad

Machine Learning 2
Group Assignment – Group 2
Alvaro Jose Mendez Lopez

Predicting Song Popularity
with Spotify Data
23/11/2025

End-to-End Pipeline

Our Machine Learning Workflow to transform Data Into Decisions



Business Context & Objective

Why predicting song popularity matters

Predicting song popularity matters because it lets labels and producers invest money, time and marketing efforts in the tracks most likely to succeed, instead of guessing.

Dataset Overview

- BDataset: **2,200 songs**, combining `11` **acoustic features** (e.g. energy, danceability, tempo) + **metadata** (artist, decade, genres).
- Challenge: **Class imbalance** in popularity levels – **High ≈ 60%**, **Medium ≈ 25%, Low ≈ 15%**, making Low-class songs harder to predict.

Our Goal

- **Business goal:** Identify early potential “hits” so resources (budget, promotion, collaborations) can be focused on the most promising songs.
- **ML goal:** Build a model that classifies songs into Low, Medium, or High popularity based on their features.

Data Preparation

Data cleaning

Removed duplicates, handled missing values, and checked for outliers.

Transformations

Applied log transforms to highly skewed variables (e.g. duration, acousticness, instrumentalness).

Train/test split

80/20 stratified split to preserve class proportions.

Target definition

Fixed popularity bins: Low (0–40), Medium (41–65), High (66–100).

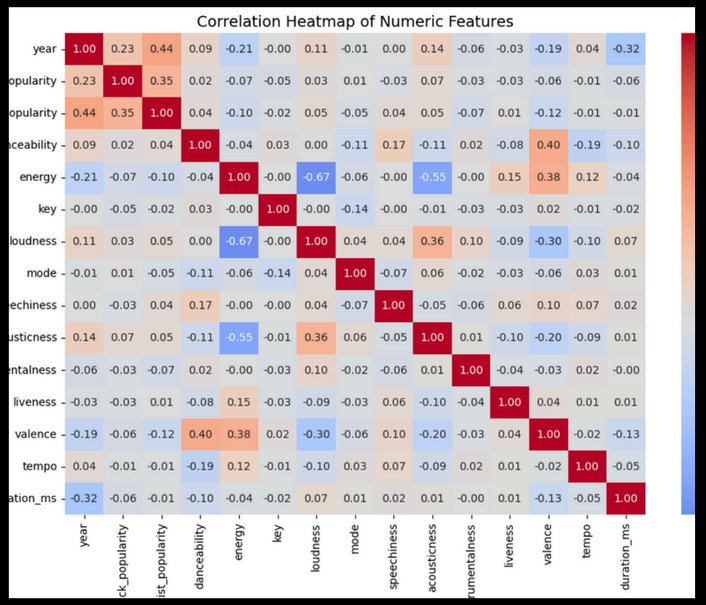
Feature Engineering

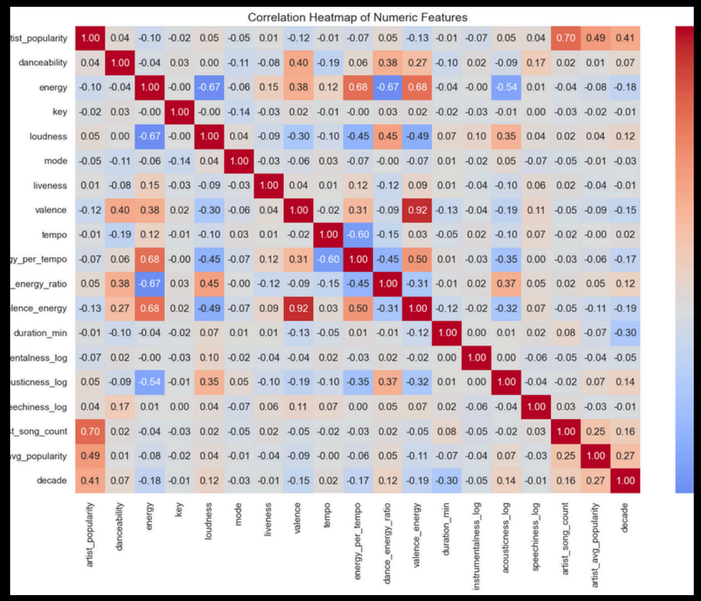
New variables	Encoding	Log Tx	Sign Inversion	Binning	Std.
• energy_per_tempo • dance_energy_ratio • valence_energy, • is_modern • decade • duration_min • artist_song_count	• artist_name • artist_genres	• instrumentalness_log • acousticness_log • speechiness_log	• loudness	• track_popularity	• For KNN

Outcome

68% improvement on model performance. Example: Boosting Accuracy increased from 0.508 to 0.854

Correlation:
Before vs After





After Feature Engineering:

- **Stronger and more meaningful correlations emerged:** New engineered features such as energy_per_tempo, dance_energy_ratio, and valence_energy show much stronger correlations with existing variables (e.g., energy_per_tempo correlates 0.68 with energy and 0.60 with tempo).

- **Artist-related engineered features show high predictive potential:** Features such as artist_song_count and artist_avg_popularity demonstrate strong positive correlations with track_popularity (e.g., 0.70 and 0.49).

- **Reduction of weak and noisy variables:** Several features had very weak correlations (close to 0) with most other features and especially with track_popularity. e.g. tempo energy_per_tempo

- **Relationships dominated by easy-to-understand musical interpretations:** strong correlation between dance_energy_ratio and danceability

Target Variable Binning

What binning was used

We didn't use quantile binning to maintain business meaning and model explainability.

Consequence

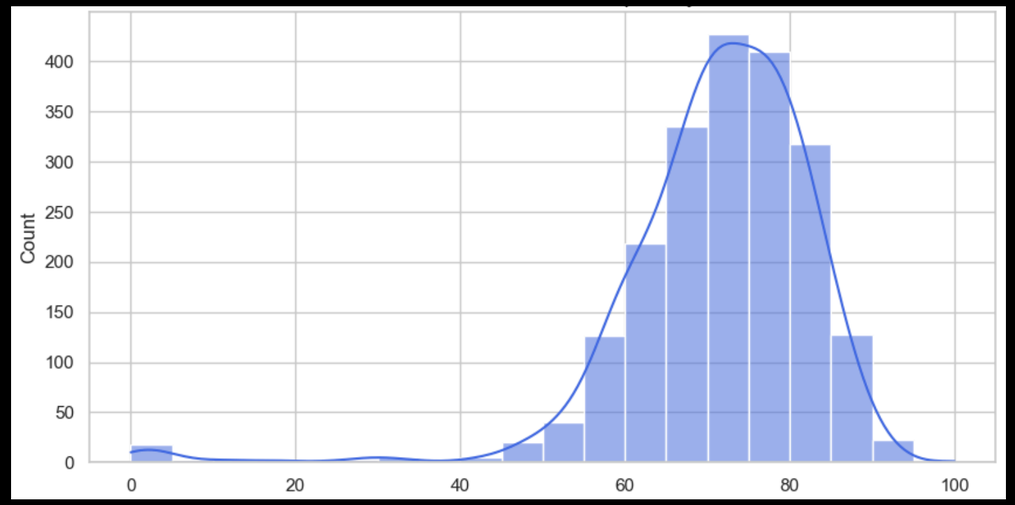
Scale binning resulted in imbalanced target classes.

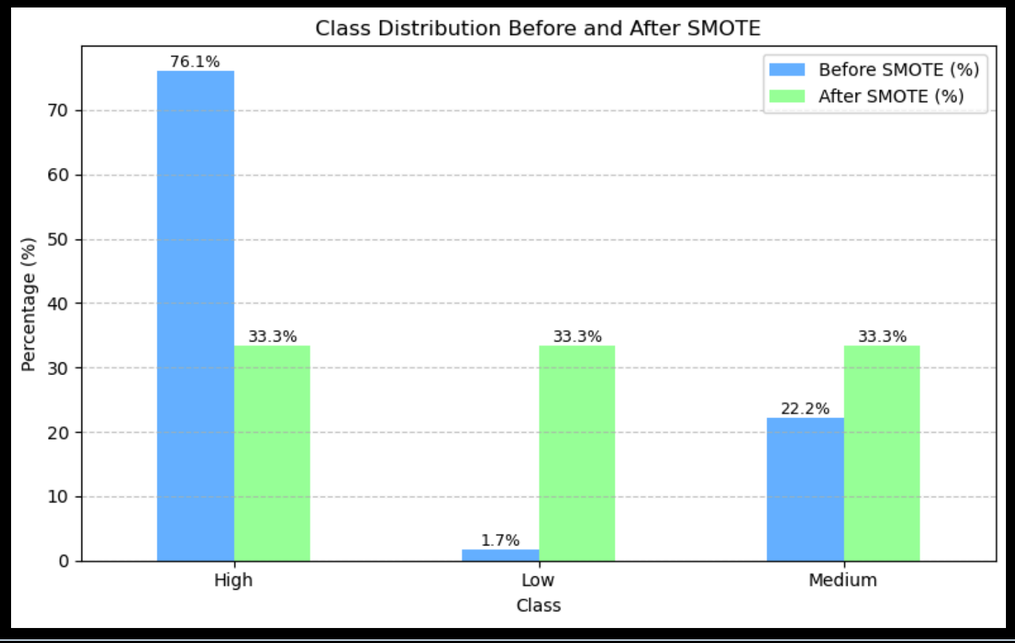
How did we fix it

Balanced Class Weight
SMOTE

Result

18.6% improvement on model performance. Example: Boosting Balanced Accuracy increased from 0.658 to 0.78.





Models Comparison

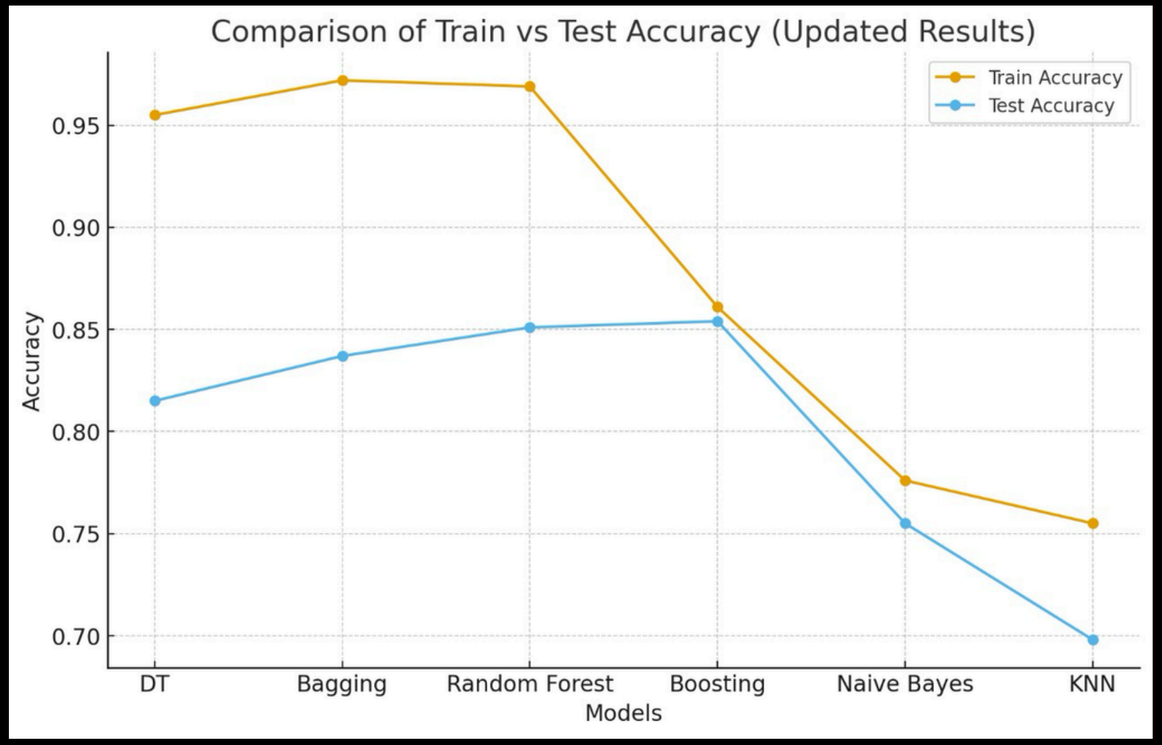
Model	Training Accuracy	Test Accuracy	Macro F1	Training Time (s)	Comment
Decision Tree	0.955	0.815	0.82	0.91	Fast, slight overfitting
Bagging	0.972	0.837	0.84	120	Limited benefit vs RF; slower
Random Forest	0.969	0.851	0.85	20.62	Best accuracy–speed balance
Boosting (XGBoost)	0.861	0.854	0.86	114	Highest accuracy overall
Naive Bayes	0.776	0.755	0.77	0.02	Fastest model; good baseline
KNN (Tuned)	0.755	0.698	0.72	9	Weakest performance

Metrics compared: Training Accuracy, Test Accuracy, Macro F1, Training Time

- Boosting (XGBoost): Highest Test accuracy (0.854) and best generalization (small train–test gap), but it is the second slowest model (114s)
- Random Forest: Nearly identical Test accuracy (0.851) to Boosting but ~5× faster; moderate overfitting but excellent practical choice
- Bagging: High accuracy (0.837) but noticeable overfits (Training 0.972 vs. Test 0.837) and is slower (120s), offering limited real-world value
- Decision Tree: Extremely fast (0.91s) and interpretable;; accuracy (0.815) with some severely overfitting. Acceptable for lightweight use cases
- Naive Bayes: Fastest model (0.02s), stable train–test performance, but lower accuracy (0.755)
- KNN (Tuned): Weakest test performance (0.698) and slow prediction

Overfitting & Generalization

Difference between Train & Test accuracy = main indicator of overfitting



Large Gap
model memorizes noise

Small Gap
strong generalization

Decision Tree, Bagging & Random Forest

- Very high Train Accuracy (≈ 0.95+)
- Lower Test Accuracy (≈ 0.82–0.85)
- Models capture some noise due to high complexity

Boosting → best generalization

- Train ≈ Test (mid-0.85s)
- Minimal gap → well-regularized and robust

Naïve Bayes → stable with low variance

- Moderate accuracy
- Small train–test gap due to simplicity

KNN → underfitting

- Lowest Train & Test accuracy
- Custom Distance function based on SME in Music is required

Feature Importance

Artist reputation and timing are the strongest drivers of song popularity.
Features like artist_popularity and decade show that well-known artists and modern production styles boost success. Energetic and emotionally positive songs (valence_energy, energy_per_tempo) also increase listener engagement and replay value.

Rank	Decision Tree	Bagging	Random Forest	Boosting	Naïve Bayes	KNN
1	decade	artist_popularity	artist_popularity	artist_popularity	decade	artist_popularity
2	artist_popularity	decade	decade	is_modern	artist_popularity	is_modern
3	main_genre_encoded	artist_song_count	is_modern	artist_song_count	duration_min	mode
4	speechiness_log	speechiness_log	artist_song_count	decade	is_modern	artist_song_count
5	energy_per_tempo	main_genre_encoded	duration_min	valence_energy	energy	valence

Final Model Recommendation:
Random Forrest

Metric	Result
Test Accuracy	0.851
Balanced Accuracy	0.732
Training Time	20.6s

Near-top predictive accuracy: achieves 97% of Boosting's accuracy with ~5× faster runtime.
Performance: Shows good generalization across popularity levels, though some imbalance remains between hit and non-hit predictions
Explainable & transparent: delivers clear feature importance, enabling actionable insights for music teams.
Operationally efficient: retrains quickly, scales easily, and integrates seamlessly into existing workflows.
Business reliable: consistent performance across retraining cycles, ideal for continuous monitoring of new releases.
Strategic fit: bridges accuracy, interpretability, and efficiency, the three pillars of real-world ML deployment.

Takeaways & Next Steps

Key Takeaways

- **Boosting** achieved the **highest accuracy** (0.854) but with **high computation cost**.
- **Random Forest** offered the **best trade-off** between accuracy (0.851), speed (20 s), and explainability → **final model chosen.**
- **Naïve Baye**s gave a fast, reliable baseline for **low-resource or early-testing scenarios**.
- Artist popularity & era (decade/is_modern) emerged as **dominant predictors,** confirming that who and when matter more than how a song sounds.
- **Feature engineering and imbalance correction** (SMOTE + class weights) were **decisive** for reaching balanced accuracy > 0.8.

Next Steps

- Design a custom similarity metric for KNN or hybrid recommenders.
- Expand dataset to include playlist placements, user skips, and region, to predict streaming impact instead of raw popularity.

Short-term

- Integrate Random Forest into a dashboard or API to predict popularity for new releases.
- Automate weekly retraining as Spotify's data updates.

Medium-term

Long-term

- Build a decision-support tool to estimate "hit potential" pre-release and guide marketing investment.

Turning Insights into Action:
Managerial Recommendations

Objective

Translate model insights into practical actions for music marketing and artist management.

Key Recommendations

1– Boost Artist Presence & Collaboration

- Partner emerging artists with established ones to leverage popularity spillovers.
- Maintain audience engagement through digital campaigns and playlist features.

2– Modern Production Choices

- Prioritize songs with modern-era sonic characteristics (clean production, clarity, spatial mix).
- Re-master older tracks to fit contemporary sound standards.

3 – Optimize Energy & Valence

- High-energy and emotionally positive tracks (high valence) correlate with higher popularity.
- Aim for energetic but not chaotic compositions for wider audience appeal.

4 – Adapt Song Length to Digital Listening Trends

- Optimal track duration: 2.5–3.5 minutes for maximum completion and playlist retention.

Thanks !