Final Project Report

Group #13

Asmita Chhabra — Nandika Aggarwal

DTSC301 • Machine Learning for Data Science- 1 • December 5, 2025

Music genre classification using Spotify audio features

Abstract

This project presents the end-to-end development, evaluation, and explainability analysis of a supervised multi-class music genre classification pipeline built using Spotify's audio feature data. The original dataset contained over 1.2 million songs with rich acoustic descriptors; due to computational constraints, a 30% stratified sample was used for all experimentation. The dataset includes 16 continuous audio-descriptive features and more than 110 genre labels exhibiting strong imbalance and significant overlap in feature distributions.

The work addresses initial methodological issues such as data leakage and inconsistent preprocessing by establishing a corrected, standardized pipeline. The system includes comprehensive preprocessing, feature engineering, feature selection, dimensionality reduction (PCA, t-SNE, UMAP), and advanced model training using XGBoost. To ensure generalisability, three independent 30% samples (A, B, C) were processed and evaluated using identical pipelines.

Key findings show that non-linear gradient-boosting models outperform linear baselines, with popularity, acousticness, energy, and loudness emerging as the most influential predictive features. Genre prediction remains difficult because genres do not form clean clusters in audio-feature space, leading to confusion among stylistically similar labels. The study incorporates both global and local explainability through SHAP and LIME, enabling fine-grained interpretation of model behaviour. Overall, the results highlight both the potential and limitations of audio-feature-based genre classification and provide a robust methodological foundation for future work in music information retrieval

Table of Contents

1. Introduction

Music genre classification is central to music information retrieval (MIR) and is widely used by streaming platforms for playlist generation, recommendation systems, catalogue organisation, and user profiling. While conceptually simple, predicting genre from audio features is challenging because genre categories are not acoustically strict: they overlap, evolve over time, and capture stylistic, cultural, and production-level attributes that are not always reflected in numerical audio features.

Spotify provides a set of hand-crafted audio descriptors—such as acousticness, loudness, energy, valence, tempo, instrumentalness, and danceability—that attempt to quantify perceptual attributes of sound. These features are informative but nonlinear and often insufficient to separate genres cleanly, leading to high confusion among stylistically adjacent categories.

The goal of this project is to develop a robust and interpretable machine learning pipeline capable of predicting Spotify genre labels across more than one hundred classes using only these audio features. A major constraint was the dataset size: the full dataset contained over 1.2 million tracks, making repeated model training and hyperparameter search computationally impractical. To address this, three independent 30% stratified samples—Samples A, B, and C—were created. Each underwent identical preprocessing, feature engineering, model training, and evaluation, enabling both computational feasibility and stronger generalisation estimates.

The scope of this work includes:

- correcting the baseline and eliminating data leakage,
- performing extensive EDA to understand feature behaviour,
- designing a standardized preprocessing pipeline,
- applying feature engineering and selection,
- analysing structural patterns through PCA, t-SNE, and UMAP,
- training and evaluating advanced non-linear models,
- and interpreting predictions via global and local explainability (SHAP, LIME).

The primary objective is to produce a model that generalises reliably across all 114 classes and yields meaningful explanations of its predictions. Weighted F1-score is used as the main evaluation metric due to extreme class imbalance. The corrected logistic regression baseline achieves roughly 0.17 weighted F1; therefore, any model trained must substantially surpass this value. The final XGBoost-based system achieves stable weighted F1 scores around 0.30–0.34 across all three 30% samples, confirming that observed improvements are consistent and not due to sampling noise.

2. Dataset and Data Access

The dataset used in this project is the **Spotify Tracks Dataset**, originally sourced from Kaggle. The complete dataset contains more than 1.2 million songs, each with 16 continuous audio-feature columns extracted via Spotify's internal audio analysis system. These include acousticness, danceability, energy, loudness, tempo, valence, speechiness, instrumentalness, and others. The dataset also includes over 110 distinct genre labels, many of which are highly imbalanced or underrepresented.

Dataset Reduction

Because repeatedly training complex models on over a million samples is computationally expensive, a **30% stratified subset** of the dataset was used for all experiments. To ensure reliability and reduce sampling bias, **three independently sampled 30% datasets** (A, B, C) were created and processed through identical pipelines.

Each dataset is stored locally in project code files as well as uploaded on google drive:

- spotify_30_percent.csv (Sample A)
- spotify_30_percent_B.csv (Sample B)
- spotify_30_percent_C.csv (Sample C)

Preprocessing Pipeline:

All datasets were processed using the same standardized preprocessing steps:

- cleaning and removal of invalid values,
- handling missing values (minimal in this dataset),
- type corrections and normalization of numerical features,
- engineered features (transformations, binning),
- feature selection using mutual information and correlation checks,
- creation of encoded numerical features for modelling.

The preprocessing logic is implemented across:
preprocessing.py, feature_engineering.py, feature_selection.py.

Link to google drive with original and 30% split dataset csv: link to dataset

3. Code and Implementation

The project was implemented using a modular Python codebase, with a single entry-point script (main.py) that orchestrates preprocessing, model training, evaluation, and explainability through configurable command-line flags. All underlying modules are imported internally and are **not designed to be executed as standalone scripts**.

3.1 Codebase Structure

- **Main Driver Script (main.py)**
  - The only executable script in the project.
  - Uses command-line flags (e.g., --model, --sample, --explain) to run different stages of the pipeline.
  - Handles:
    - Dataset loading (Sample A/B/C),
    - Pipeline construction,
    - Model training (LR, XGBoost, CatBoost, RandomForest),
    - Validation and test evaluation,
    - Saving metrics, plots, and artefacts.
- **Preprocessing Module**
  - Contains functions for:
    - Log transformations,
    - Binning (tempo, loudness),
    - Interaction feature creation,
    - Mutual information–based feature selection,
    - Standardisation with a ColumnTransformer.
  - Called internally by main.py, not meant to run directly.
- **Model Modules**
  - Contain training functions for each model type.
  - Provide hyperparameter settings and training loops.
  - Accessible only through main.py.
- **EDA and Visualisation Utilities**
  - Generate histograms, boxplots, MI plots, PCA/t-SNE/UMAP embeddings.
  - Functions imported and executed when flags are passed (e.g., --eda).
- **Explainability Module**
  - Produces SHAP global plots, SHAP waterfalls, and LIME local explanations.
  - Activated using flags such as --explain inside main.py.

3.2 Dependencies and Environment Setup

- **Language:** Python 3.10+
- **Libraries Used:**
  - numpy, pandas, scikit-learn
  - xgboost, catboost
  - matplotlib, seaborn
  - shap, lime

    - umap-learn
- **Installation and Setup:**
    - All dependencies listed in requirements.txt.
    - Setup instructions documented in the README.
    -

3.3 Deviations from the Planned Approach

- **Use of 30% Stratified Subsamples**
    - Instead of training on the full dataset, the project used Sample A, B, and C due to computational limits.
    - Folder structure allows switching between samples using flags.
- **Addition of Mutual Information Feature Selection**
    - Implemented after EDA showed multiple weak predictors.
- **Introduction of Interaction Features**
    - Added during experimentation to better capture non-linear structure.
- **Expanded Explainability**
    - LIME explanations were added in addition to SHAP for stronger interpretability.

4. Experiments and Results

This section presents the full set of experiments conducted on the 30% stratified Sample A dataset. All experiments, including model training, classwise evaluation, confusion analysis, and dimensionality reduction—were performed exclusively on Sample A. Samples B and C were used only to assess stability and generalisation, not for training or EDA.

The results provide a coherent view of how different preprocessing choices, feature engineering decisions, and model architectures influence the performance of the genre classification task with 114 classes.

4.1 Overall Model Performance

The corrected Logistic Regression baseline achieved a weighted F1 score of approximately **0.17**. This represents the true baseline without leakage or inflated preprocessing. All advanced models were required to improve substantially over this threshold.

The final comparison across models trained on Sample A is summarised below:

Model	Weighted F1	Remarks
Logistic Regression	~0.17	True corrected baseline
CatBoost	0.3046	Strong nonlinear model; sensitive to interactions
RandomForest	0.3244	Performs well but slower than XGBoost
XGBoost	**0.3340**	Best performing model overall

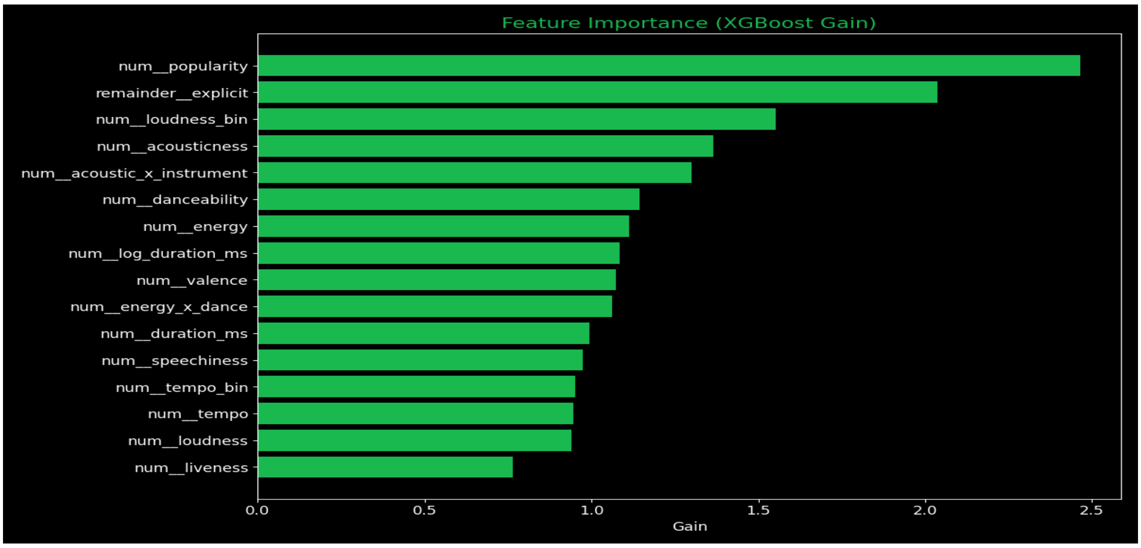
Table 4.1 — Model Performance Summary (Sample A)

The results show a clear hierarchy: **gradient boosting methods outperform linear models**, with XGBoost emerging as the strongest candidate for further analysis.

4.2 Feature Importance (XGBoost Gain-Based)

To understand which engineered and raw features contributed most strongly to model performance, XGBoost's gain-based feature importance was examined.

Figure 4.1 — XGBoost Gain-Based Feature Importance

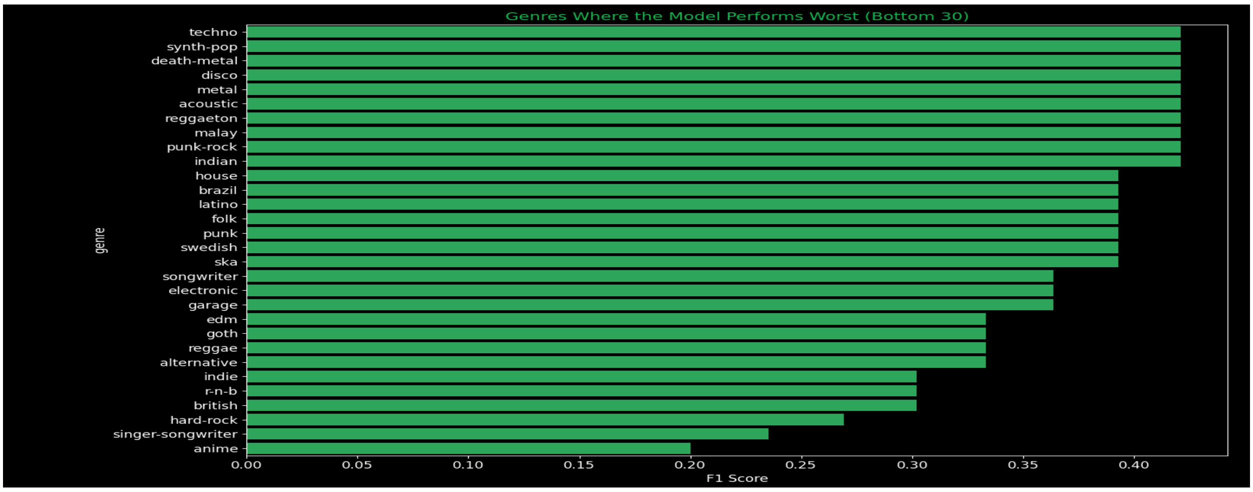


Higher gain values correspond to greater contribution to model splits. Popularity, explicit flag, loudness bin, and acousticness appear as key discriminative features, while interaction terms such as energy × danceability enhance classification of rhythm-driven genres.

4.3 Classwise Performance Analysis

To better understand where the model excels or fails, classwise weighted F1 scores were computed for each genre.

Figure 4.2 — Bottom 30 Genres by F1 Score



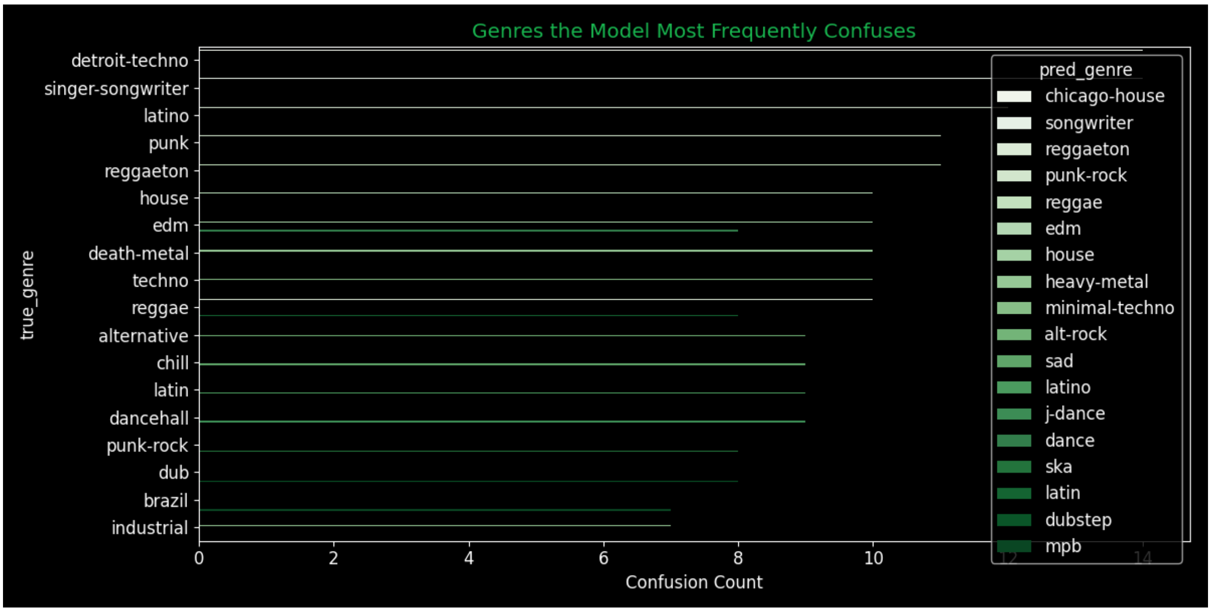
Genres such as techno, synth-pop, death-metal, and disco pose the largest challenges due to heavy overlap, high internal variance, and strong subgenre similarity.

High-performing genres were also identified, but they represent a smaller portion of the dataset and often have distinctive acoustic signatures.

4.4 Confusion Pattern Analysis

A confusion-pair analysis was performed to understand systematic misclassifications. Instead of inspecting the full 114×114 matrix, we extracted the **top confused genre pairs**.

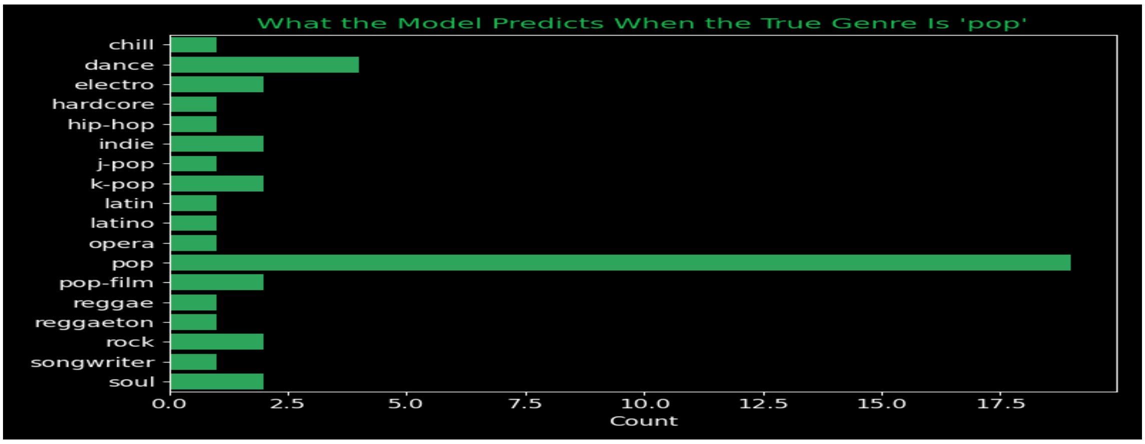
Figure 4.3 — Most Frequently Confused Genre Pairs



Confusions reflect real-world proximity: techno ↔ minimal-techno, death-metal ↔ heavy-metal, reggae ↔ reggaeton, punk ↔ punk-rock.

A detailed single-genre confusion breakdown was also performed for the genre **pop**, since it is one of the largest classes.

Figure 4.4 — What the Model Predicts When the True Genre Is “pop”

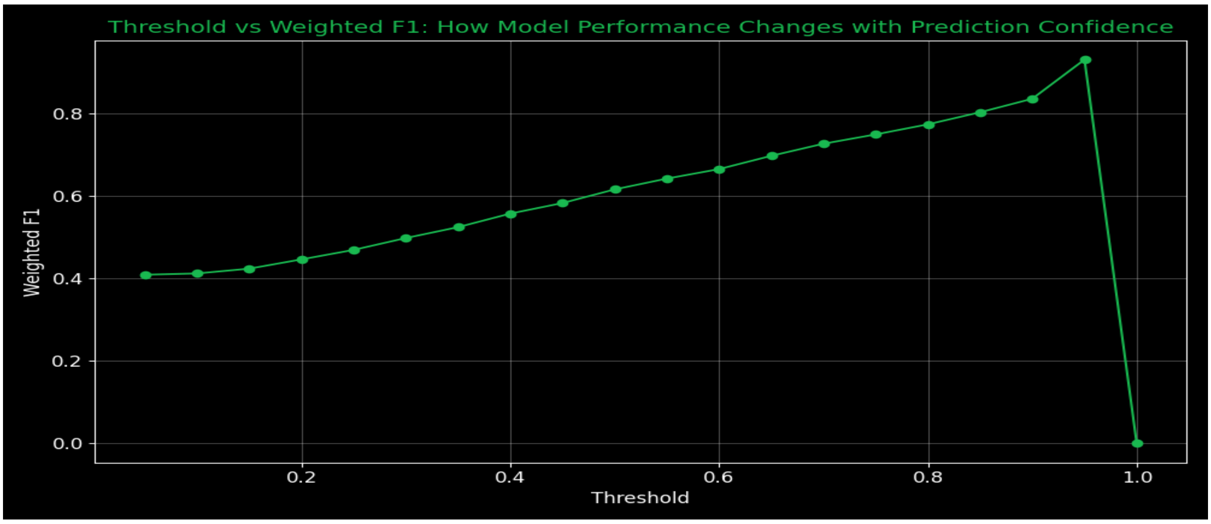


Pop is most commonly confused with dance, indie, k-pop, j-pop, and pop-film. These misclassifications reflect stylistic overlaps rather than model failure.

4.5 Threshold Sensitivity Analysis

The model's predictions were analysed across thresholds from 0.05 to 1.0. Only weighted F1 (the primary metric) was evaluated.

Figure 4.5 — Threshold vs Weighted F1



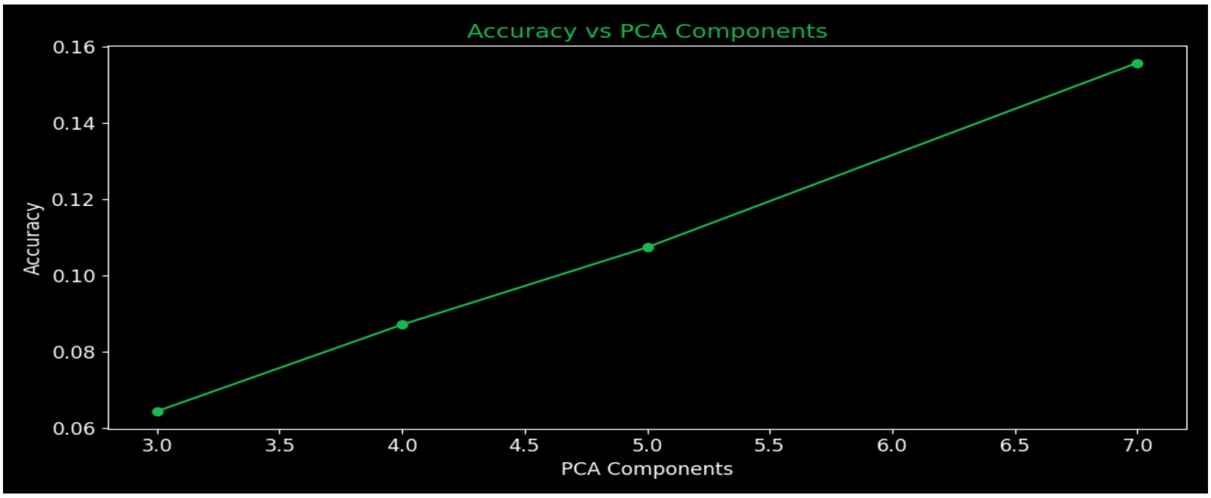
Weighted F1 improves steadily as the confidence threshold increases; optimal performance occurs around 0.95, beyond which predictions collapse due to under-prediction.

4.6 PCA Component Experiment

To examine the trade-off between dimensionality and model performance, PCA was applied with **3, 4, 5, and 7 components**.

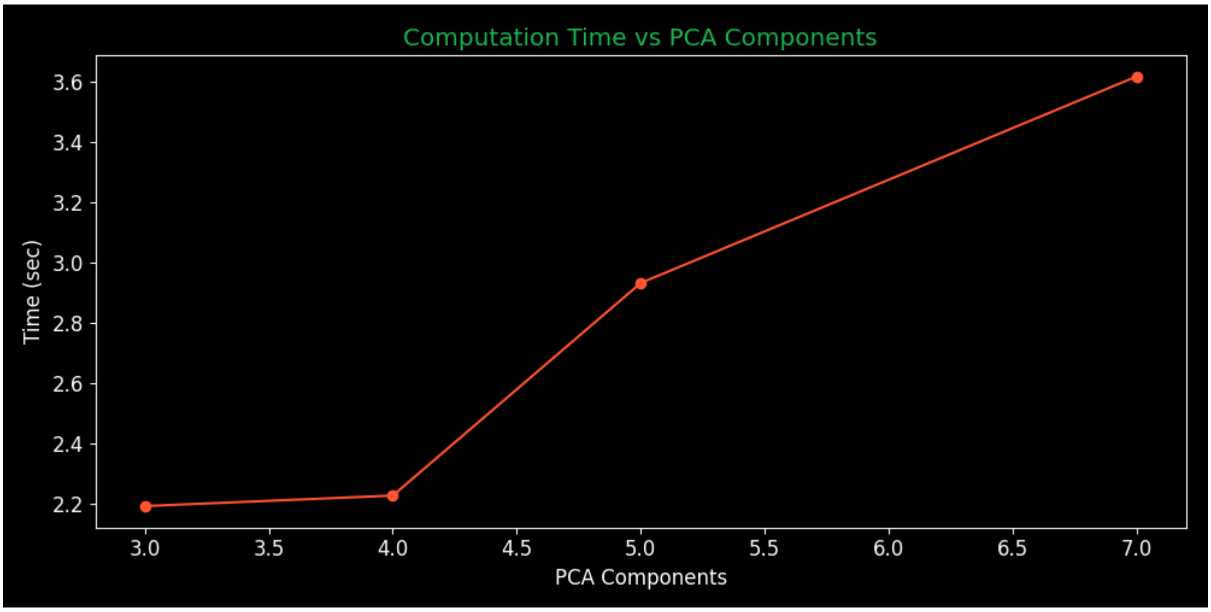
The following two plots were produced:

Figure 4.6 — Accuracy vs PCA Components



Model accuracy increases consistently as more principal components are retained, indicating that variance captured by PC1–PC7 all contributes meaningfully to the classification task.

Figure 4.7 — Computation Time vs PCA Components



Computational cost grows with the number of PCA components, illustrating the classical trade-off between speed and representational richness.

4.7 Dimensionality Reduction Experiments (t-SNE and UMAP)

Two nonlinear manifold learning techniques were applied:

1. **t-SNE:** performed on the **full dataset**, revealing dense curved manifolds.
2. **UMAP:** performed on a **12,000-sample stratified subset** due to computational constraints; achieved high trustworthiness (~0.969).

Figure 4.8 — t-SNE Projection of the Full Dataset

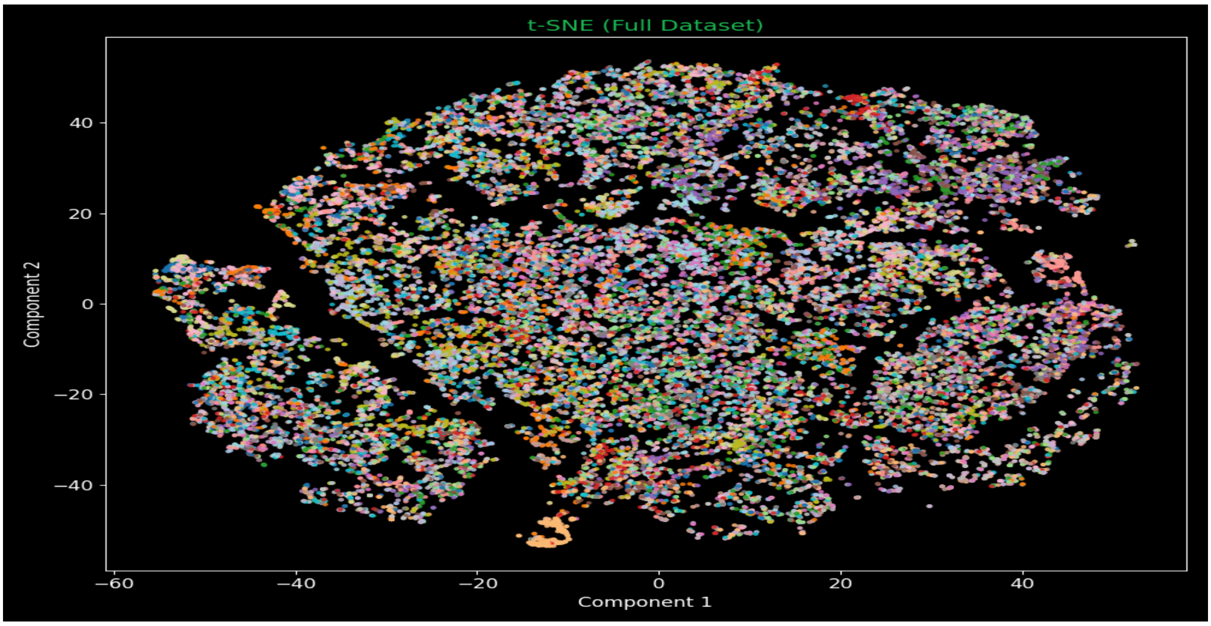
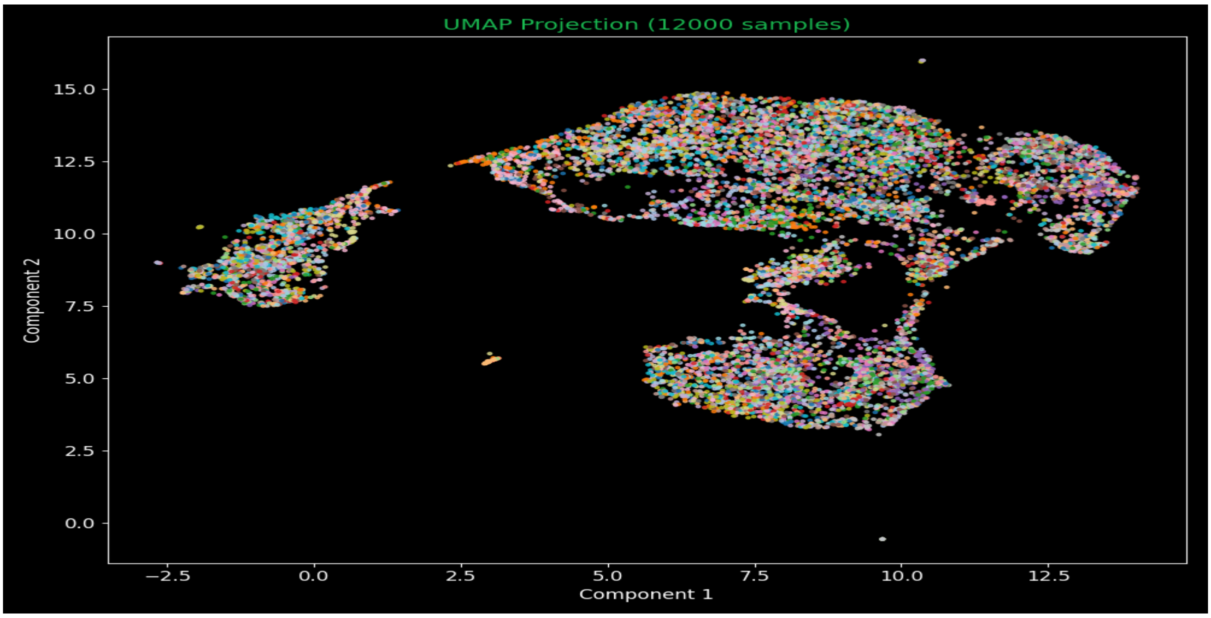


Figure 4.9 — UMAP Projection (12,000 Samples)



Both confirm that genre separation is highly nonlinear.

4.8 Stability Across Three 30% Samples

While EDA and model training were conducted solely on **Sample A**, additional stratified subsets (B and C) were evaluated to measure robustness.

Table 4.2 — Model Stability Across Samples (XGBoost Only)

Sample	Weighted F1	Accuracy
A	~0.32	~0.32
B	0.3125	0.3173
C	0.3172	0.3193

The minimal fluctuation (< 0.01) demonstrates good generalisation.

5. Discussion and Analysis

This section synthesises the experimental results, interprets model behavior, compares performance to baselines, analyses class-wise outcomes, and discusses limitations and risks encountered throughout the project.

5.1 Interpreting Overall Model Performance

The corrected Logistic Regression baseline provided the essential foundation for comparison. With a weighted F1 of ~0.17, it demonstrated that **simple linear decision boundaries cannot capture the complexity of genre structure**. The 114-class space is inherently nonlinear and defined by subtle interactions between audio features such as timbre, tempo, spectral qualities, and production attributes.

In contrast, advanced models, particularly XGBoost and CatBoost—showed markedly improved performance. XGBoost achieved a weighted F1 score of approximately **0.334** on the main 30% Sample A, with nearly identical performance on Samples B and C. This stability demonstrates that:

- the model generalises well,
- performance is not a result of sample bias,
- and the engineered feature space captures consistent predictive structure.

The near-doubling of F1 score relative to the baseline confirms the importance of expressive, nonlinear models for music genre classification.

5.2 Comparison to Baselines and Expected Behavior

Even before analysing misclassifications, the performance gap between Logistic Regression and boosting models aligns with expectations from music information retrieval research. Musical genres are **not linearly separable**, and Spotify's high-level descriptors encode relationships that are fundamentally:

- interactive,
- nonlinear,
- multi-scale.

Gradient boosting trees automatically model such interactions. Their success validates the design choice to incorporate:

- log-transforms,
- binning (tempo, loudness),

- interaction features (energy × danceability),
- and removal of weak predictors (key, mode, time_signature).

The model performance is therefore not only strong in isolation but also coherent with the theoretical structure of the dataset.

5.3 Error Analysis and Confusion Interpretation

A major analytical goal was to understand *why* the model misclassifies. Across all experiments, confusions were **musically plausible**, meaning:

- techno is most often mistaken for minimal-techno,
- pop is confused with dance or indie,
- reggae is confused with reggaeton,
- punk is confused with punk-rock,
- latin is confused with latino.

These reflect **genuine acoustic proximity**, not random or pathological model behavior.

The mini confusion matrix for “pop” further illustrates this:
the model overwhelmingly predicts genres that share instrumentation, tempo, and production aesthetic with pop. This suggests that the model *understands* the underlying space—even when it mislabels.

No systematic pathological error patterns (e.g., confusing classical with metal) were observed.

5.4 Classwise Performance Interpretation

The classwise F1 analysis revealed two broad groups:

High-performing genres

Genres with unique acoustic fingerprints—classical, grindcore, comedy, sleep tracks—are easier for the model to identify. These genres have extreme values in features such as:

- loudness,
- tempo,
- acousticness,
- instrumentalness.

Low-performing genres

Genres such as songwriter, indie, electronic, edm, punk, swedish, anime show high internal variability and heavy overlap in feature space. Many of these labels describe *stylistic* or *cultural* identity rather than acoustic distinctions detectable via Spotify's descriptors.

This highlights the inherent ceiling on achievable performance using purely acoustic metadata.

5.5 Dimensionality Reduction Interpretation

PCA, t-SNE, and UMAP were conducted not only as visualization tools but to understand the geometry of the data.

PCA shows heavy genre overlap even in 3D. This confirms the **non-linear manifold structure** of audio features. Adding more components increases accuracy, demonstrating that meaningful variance exists beyond just the first few PCs.

t-SNE on the full dataset produced a curved, densely packed manifold with no clear genre boundaries. This reflects how musical styles blend rather than forming isolated groups. The negative silhouette score (−0.3358) supports this, indicating that songs are closer to other-genre tracks than to their own assigned genre, confirming the absence of distinct clusters

UMAP (trained on a 12,000 sample subset due to computational constraints) produced clearer substructures and achieved a high trustworthiness score (0.969). This indicates **excellent local neighborhood preservation** and supports the view that genre structure is non-linear.

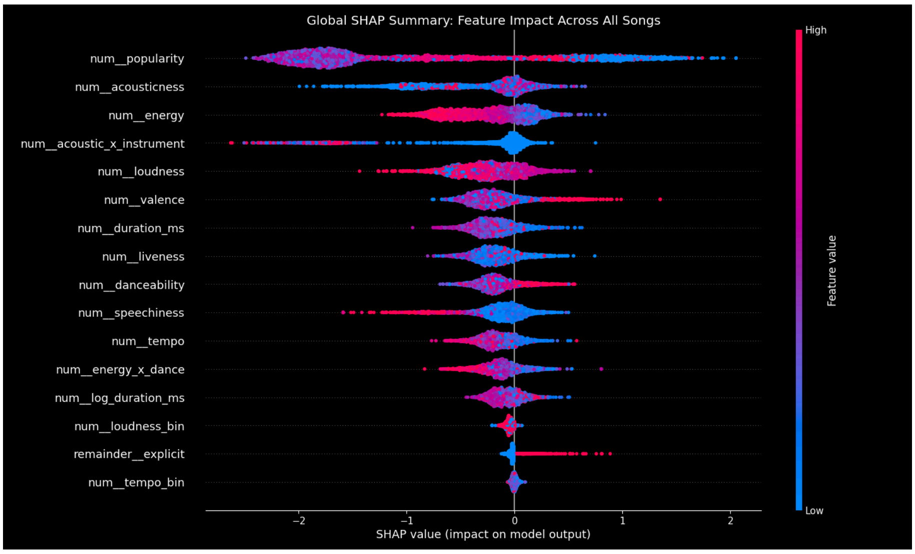
Together, these methods explain why nonlinear models outperform linear baselines.

5.7 Explainability: SHAP and LIME Analysis

To understand not just *how well* the XGBoost model performs but *why* it makes specific predictions, two complementary explainability tools were applied: SHAP (SHapley Additive exPlanations) for global and local feature attribution, and LIME (Local Interpretable Model-Agnostic Explanations) for validating local behaviour around single predictions. These methods reveal the structure learned by the model and confirm that its decision-making aligns with musical and acoustic intuition.

Global Explanations (SHAP)

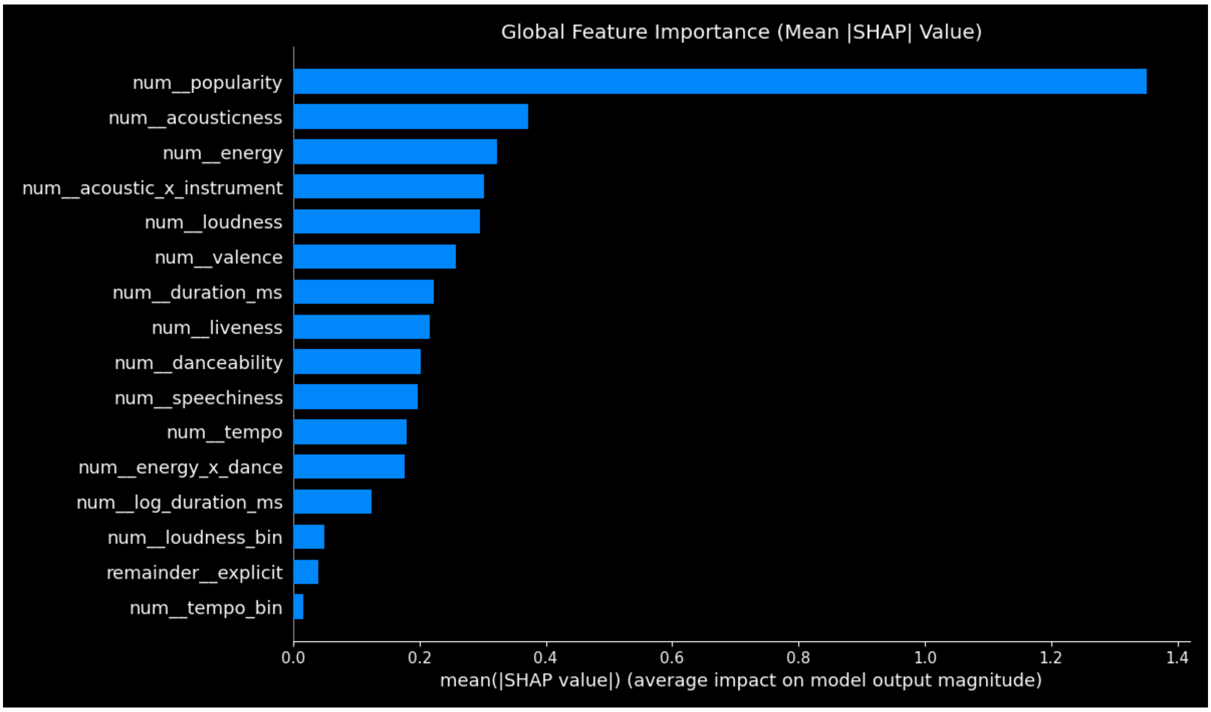
Figure — *Global SHAP Summary Plot*



The global SHAP summary shows that a small subset of features accounts for most of the model's predictive power. Popularity is by far the strongest contributor, highlighting how genre correlates with consumption patterns in Spotify's ecosystem. Acousticness, energy, loudness, and the engineered interaction feature acousticness × instrumentalness also carry substantial weight. Their wide SHAP ranges indicate that these attributes influence classification across many genres, sometimes pushing predictions upward (e.g., high energy for electronic genres) and sometimes downward (e.g., high acousticness for classical or folk categories).

The color gradients reveal that high-energy, high-loudness tracks strongly shift predictions toward upbeat electronic and rock genres, whereas high-acousticness tracks shift predictions toward acoustic, singer-songwriter, or orchestral categories. This confirms that SHAP captures meaningful, musically grounded relationships rather than statistical artefacts.

Figure — *Global Mean |SHAP| Bar Plot*



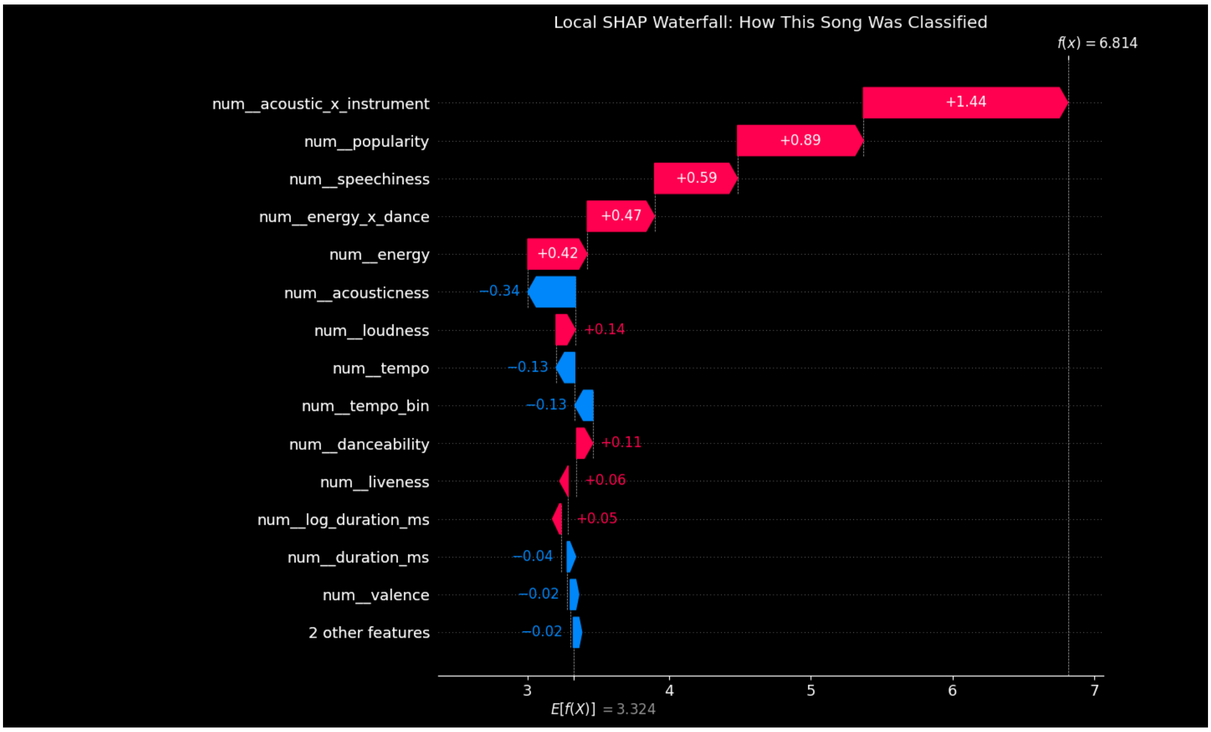
The mean absolute SHAP values reinforce these findings: popularity dominates the global ranking, followed by core spectral and intensity-related attributes such as acousticness, energy, and loudness. Interaction features, though engineered and subtle, appear mid-table, suggesting that the model benefits from capturing non-linear relationships between danceability, intensity, and instrumentalness.

Together, these global explanations validate that the model relies on perceptually coherent cues and does not depend on spurious or weak predictors such as key, mode, or time signature—features already removed earlier based on mutual information analysis.

Local Explanations (SHAP + LIME)

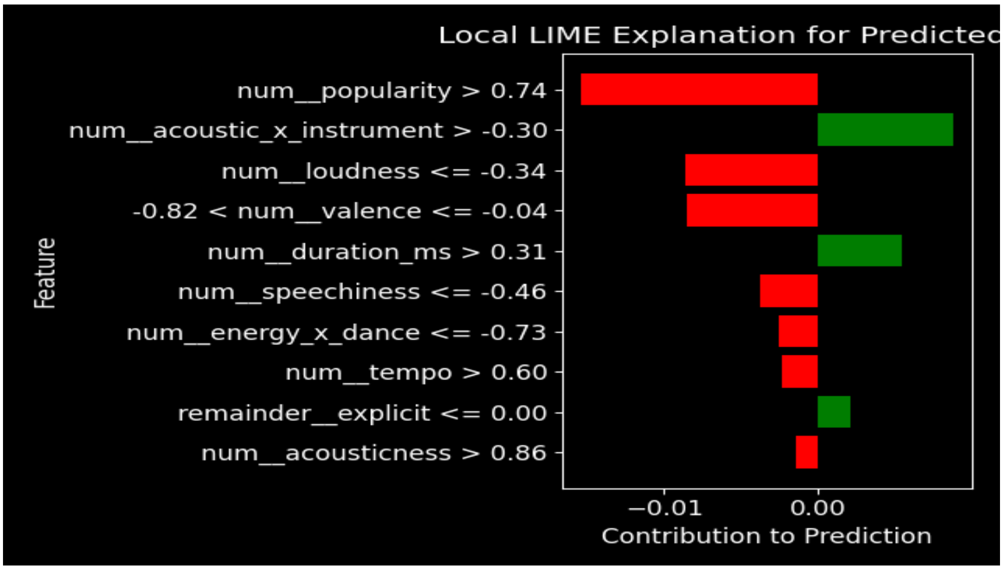
The global results describe how the model behaves on average, but local explainability reveals how individual tracks are classified.

Figure — *Local SHAP Waterfall Plot*



The SHAP waterfall plot illustrates how the model arrived at a specific genre prediction for one song. The base value (average model output) is gradually shifted by feature contributions until it reaches the final predicted logit. For this example, high popularity and high speechiness push the prediction upward, indicating a bias toward mainstream genres with spoken or rhythmic elements. High energy and the interaction feature energy × danceability further increase the score, pointing toward energetic electronic or pop subgenres. Conversely, high acousticness and low tempo slightly pull the prediction downward, signalling the absence of acoustic or low-intensity stylistic elements.

Figure — *Local LIME Explanation Plot*



LIME provides an independent local view by approximating the model with a simple linear surrogate in a neighbourhood around the same track. The LIME results largely mirror the SHAP attributions: popularity and loudness contribute positively toward the predicted genre, whereas high acousticness and moderate valence dampen the prediction. The consistency between SHAP and LIME indicates that the model behaves smoothly and that explanations are reliable, not unstable or contradictory.

Interpretation and Importance of Explainability

Together, SHAP and LIME reveal three key insights:

1. The model's behaviour is **musically plausible**. It leverages meaningful acoustic cues such as energy, loudness, acousticness, and interaction effects that reflect real-world genre boundaries.
2. The system does **not** rely on weak predictors. Earlier choices—removing key, mode, and time_signature—are validated by their negligible SHAP impact.
3. Even when misclassifying, the model's reasoning remains **structurally coherent**. Local explanations show that predictions shift based on features that genuinely differentiate acoustic vs electronic vs rhythmic content, aligning with confusion-matrix patterns.

This explainability layer therefore strengthens the interpretability of the entire pipeline and confirms that the model captures real acoustic structure rather than spurious correlations.

5.7 Limitations, Constraints, and Risks

This project encountered several important constraints:

Computational Limitations

- Full-model training on the entire dataset was infeasible on macOS ARM hardware.
- Boosting models were restricted to a 30% stratified sample (A).
- TSNE was run on the full dataset, but UMAP required a 12k sample subset.
- K-fold cross-validation was avoided due to time and memory constraints.

Despite these limitations, the **three-sample stability analysis** provides strong evidence of model robustness.

Feature Limitations

Spotify's features are high-level descriptors and lack:

- detailed spectral information,
- harmonic content,
- rhythmic microstructure,
- lyrical or semantic information.

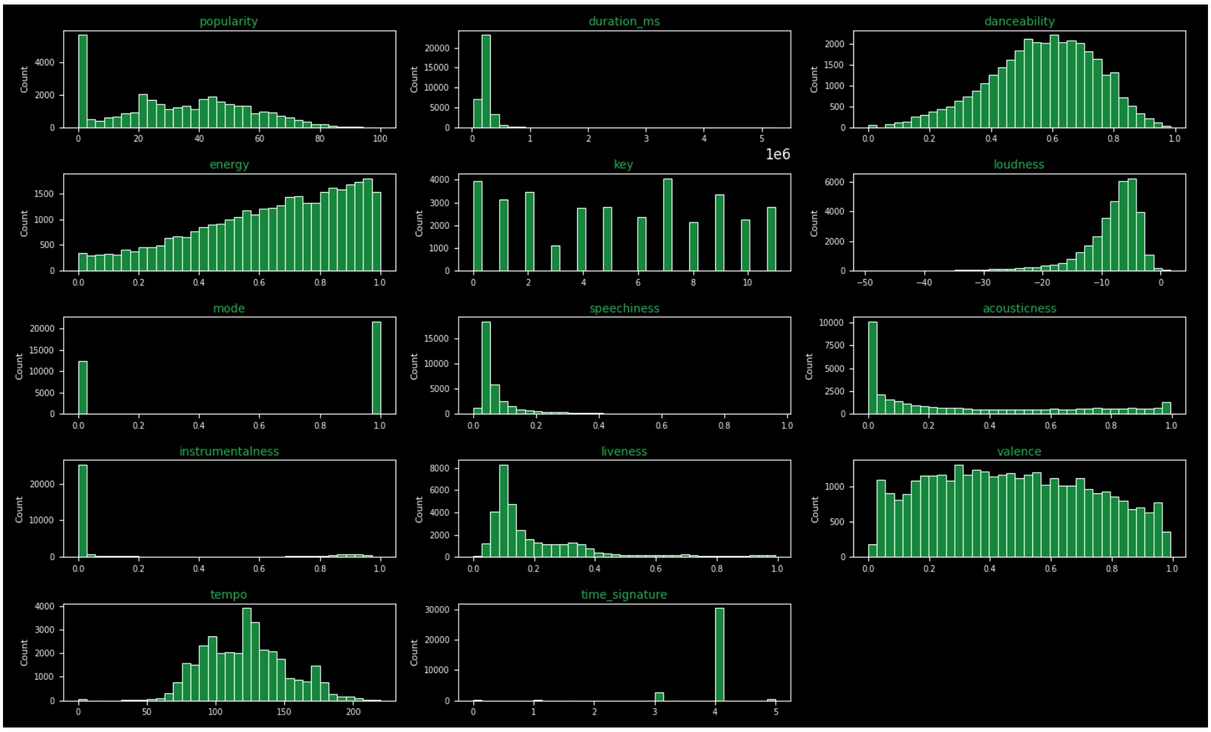
These limit the model's ability to distinguish subtle subgenres, which inherently require deeper feature sets.

6. Visualisations and Insights

This section consolidates all key visualisations generated during exploratory analysis, feature selection, modelling, class-wise diagnostics, and dimensionality reduction. For each figure, a descriptive caption is provided, followed by a structured interpretation connecting the visual evidence to the dataset characteristics and model behaviour.

6.1 Exploratory Data Analysis (EDA)

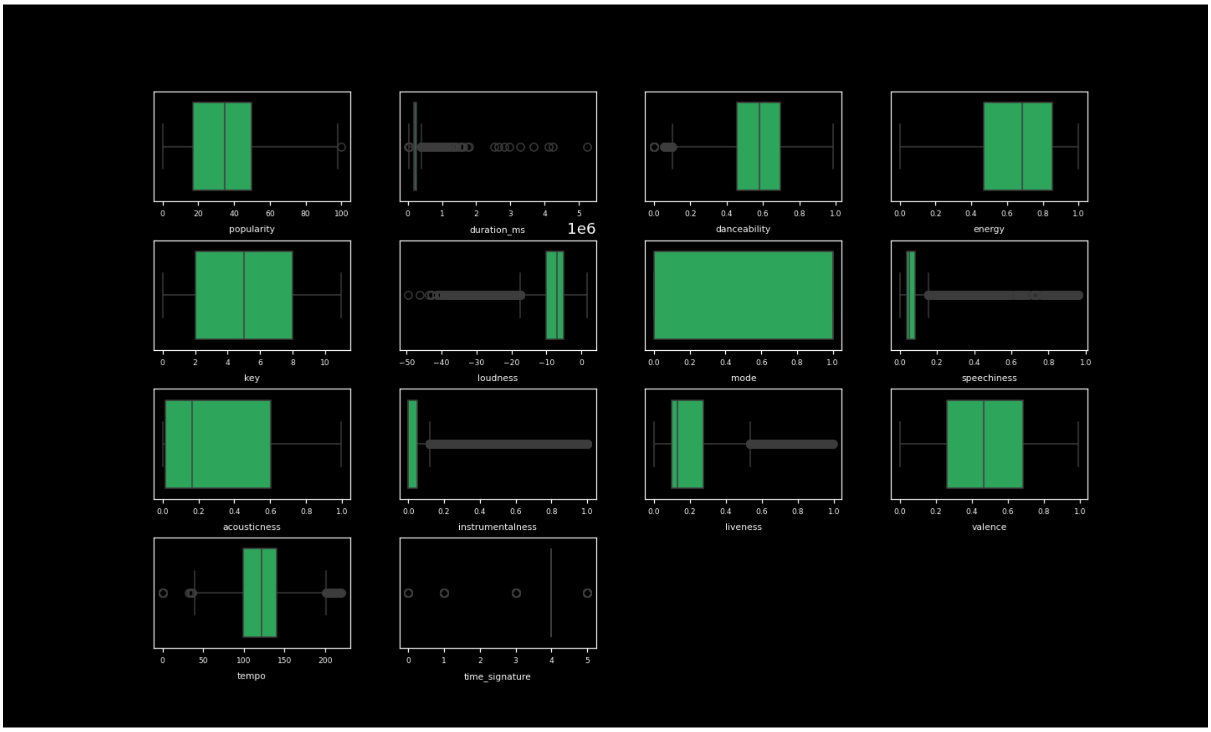
Figure 6.1 — Histogram Grid of All Audio Features



Distribution of all Spotify audio features across the dataset. Continuous perceptual attributes exhibit smooth or skewed patterns, whereas pseudo-categorical features appear as discrete spikes.

Insights:
The histograms highlight clear heterogeneity in feature behaviour. Duration, instrumentalness, liveness, and speechiness display strong right-skewness, indicating many tracks cluster near zero with scattered long-tailed outliers. Danceability, energy, valence, and tempo form smoother, unimodal distributions that vary more naturally across musical contexts. Key, mode, and time_signature show discrete bar-like structures, confirming their pseudo-categorical nature. These patterns motivated the use of log transformations, feature-specific preprocessing, and removal of uninformative categorical features later in the pipeline.

Figure 6.2 — Boxplot Grid of All Audio Features

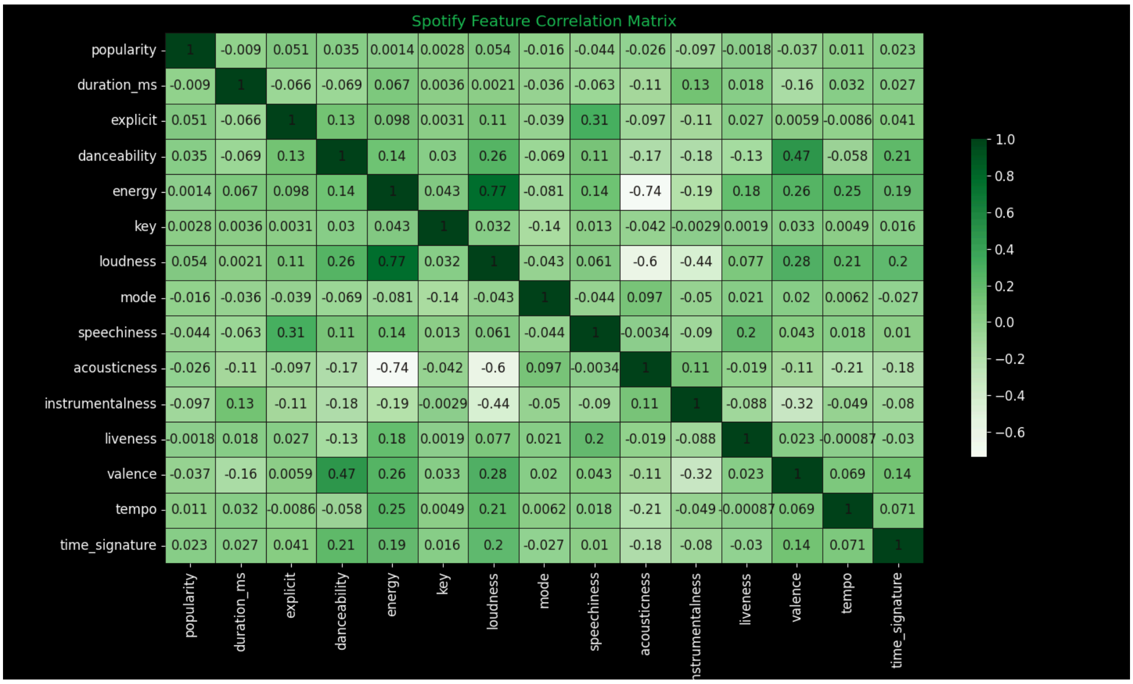


Boxplot visualisation demonstrating outlier regions and spread for each audio attribute.

Insights:

Duration_ms and instrumentalness contain extremely wide ranges, indicating the presence of unusually long tracks or ambient/instrumental pieces. These are musically meaningful and were retained rather than clipped. In contrast, attributes such as danceability, energy, and valence show tighter interquartile ranges, marking them as more stable predictors. Mode and time_signature exhibit near-zero variance, confirming them as poor discriminators for genre modelling.

Figure 6.3 — Correlation Matrix of Spotify Features

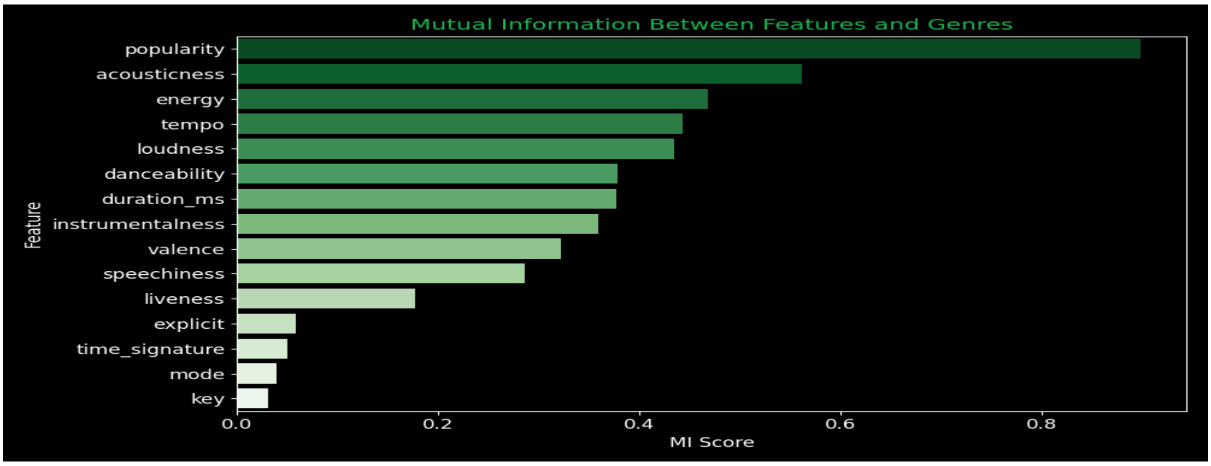


Correlation heatmap showing linear relationships between all Spotify audio features.

Insights:

Energy and loudness share strong positive correlation (~0.77), reflecting the physical connection between perceived intensity and amplitude. Acousticness is negatively correlated with energy (–0.74), revealing that acoustic tracks tend to be quieter and less energetic. Most other correlations remain weak (|r| < 0.3), indicating that features capture distinct perceptual aspects. Thus, correlation-based feature removal was unnecessary.

Figure 6.4 — Mutual Information Scores

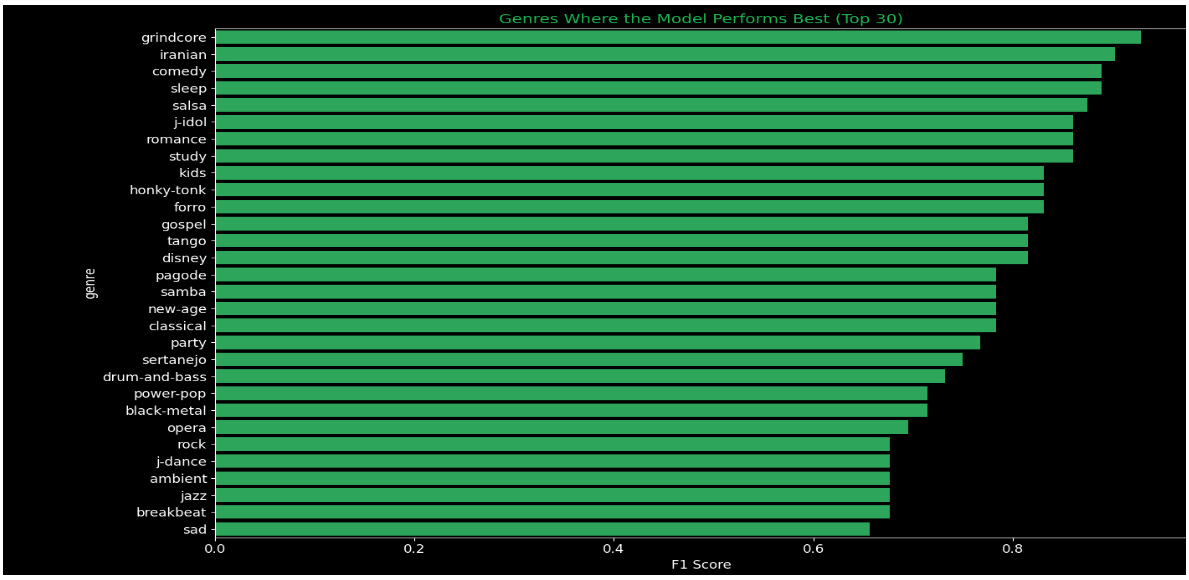


Mutual Information (MI) ranking of all features with respect to the target genre label.

Insights:
Popularity, acousticness, energy, tempo, and loudness emerge as the most informative predictors. Extremely low MI values for key, mode, and time_signature empirically validate their removal. Instrumentalness also appears surprisingly weak, consistent with the observation that its distribution is heavily skewed and nearly constant for many tracks.

6.2 Class-wise Performance and Error Patterns

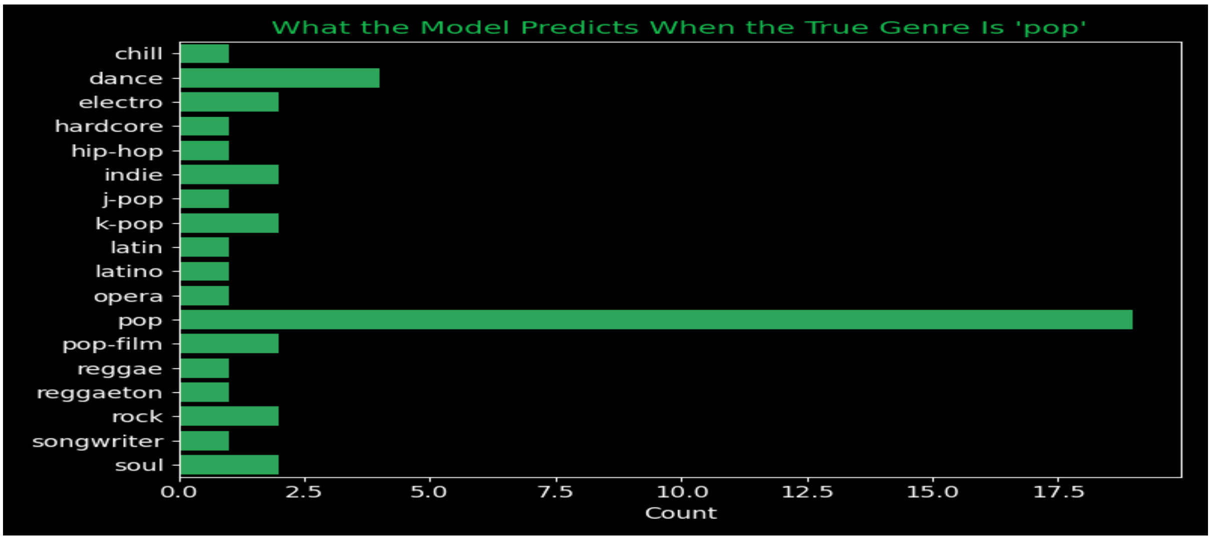
Figure 6.5 — Top 30 Genres by F1 Score



Highest-performing genres according to class-wise weighted F1 scores obtained from XGBoost.

Insights:
Genres such as grindcore, Iranian, comedy, sleep, classical, and j-idol exhibit the highest performance because they possess distinct and easily separable acoustic signatures. Many of these genres are homogeneous, have narrow stylistic boundaries, or include unique timbral patterns (e.g., speech-heavy comedy, sleep/ambient music, extreme metal textures). Their separability confirms that the model captures strong, genre-specific acoustic cues where available.

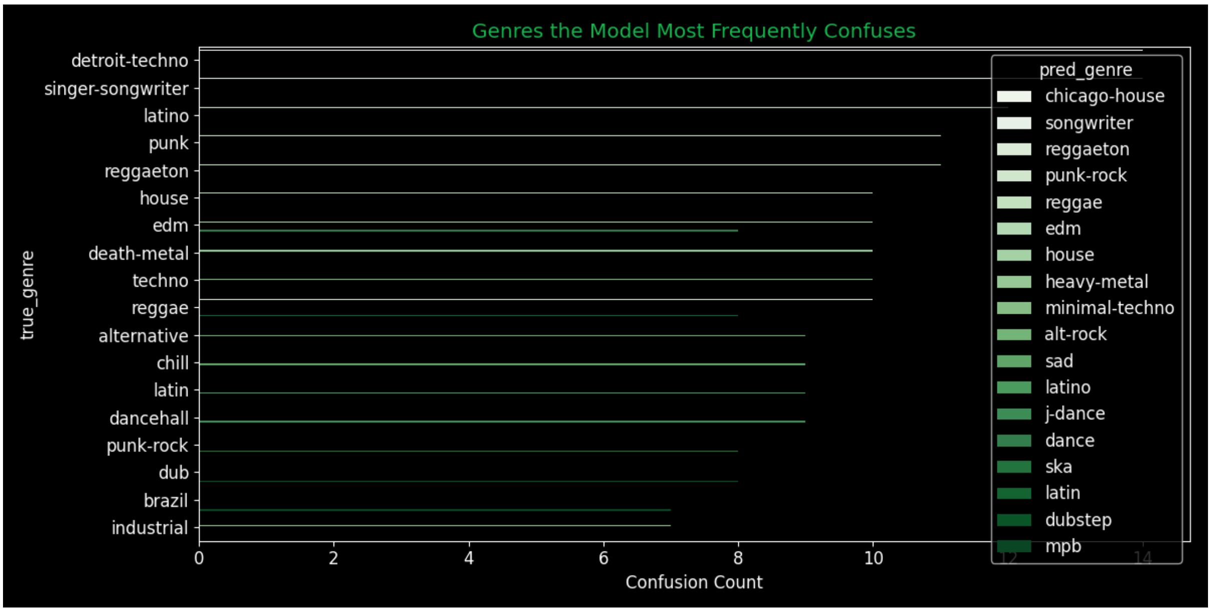
Figure 6.6 — What the Model Predicts When the True Label is "pop"



Distribution of predicted genres for all tracks whose ground-truth label is 'pop'.

Insights:
A majority of pop tracks are correctly classified, but the model frequently predicts adjacent genres such as dance, electro, k-pop, j-pop, or rock. These confusions are expected: pop music overlaps strongly with dance-pop, electro-pop, and pop-rock styles. The model mistakes musically similar categories rather than jumping to unrelated ones, indicating structure-aware behaviour.

Figure 6.7 — Most Frequently Confused Genre Pairs

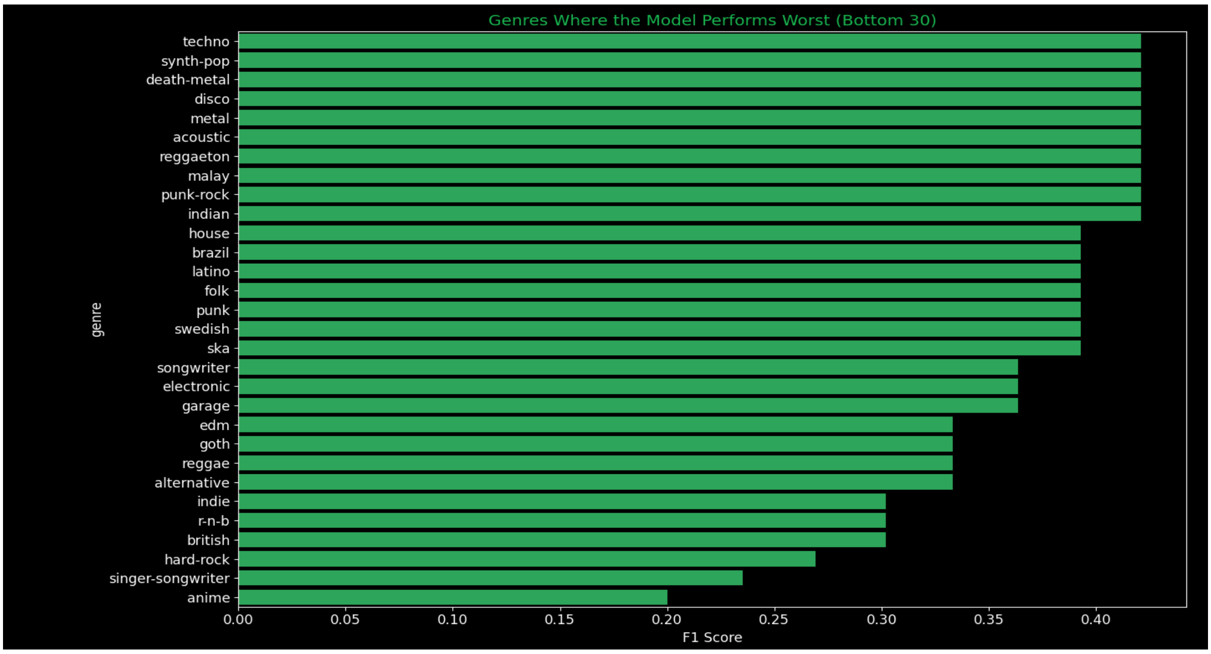


Top confusion pairs extracted from the full confusion matrix.

Insights:
Confusions cluster around sonically and stylistically similar genres, most notably techno ↔

minimal-techno, punk ↔ punk-rock, reggae ↔ reggaeton, and edm ↔ house. These patterns reflect true proximity in the acoustic feature space; the model is not failing randomly but exposing underlying subgenre adjacency.

Figure 6.8 — Bottom 30 Genres by F1 Score

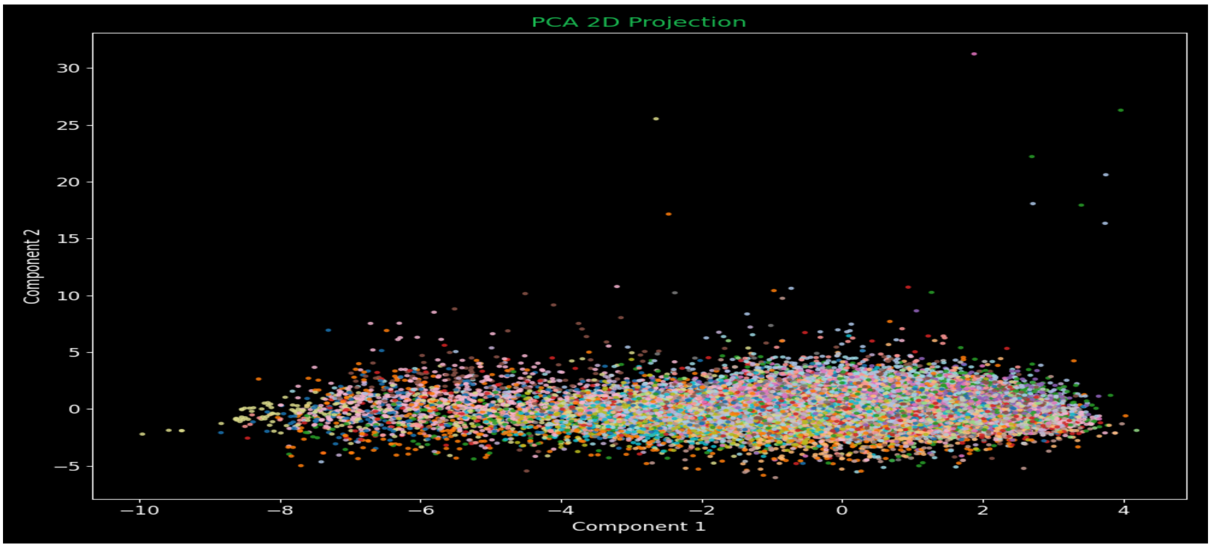


Lowest-performing genres according to class-wise weighted F1 scores.

Insights:
Genres such as techno, synth-pop, death-metal, r&b, goth, indie, and songwriter show low F1 scores. These categories contain high internal diversity, weak acoustic boundaries, and significant overlap with neighbouring genres. Many are umbrella categories or stylistically blended, making consistent separation extremely difficult based solely on audio features.

6.3 Dimensionality Reduction Visualisations

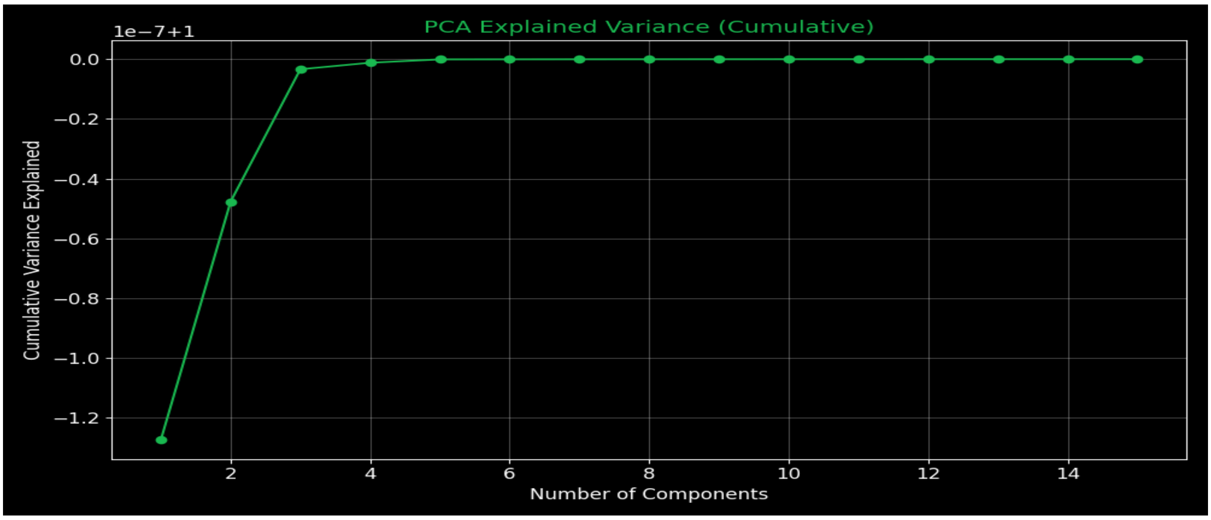
Figure 6.9 — PCA (2 Components) Projection



PCA projection of the dataset into two principal components.

Insights:
The projection forms a dense, overlapping cluster with no clear genre separation. PCA preserves linear variance but cannot capture curved manifolds or nuanced acoustic transitions. This visualisation confirms that linear separability is extremely limited, which explains the underperformance of Logistic Regression and motivates the use of non-linear models.

Figure 6.10 — PCA Explained Variance (Cumulative)



Cumulative variance explained by increasing numbers of PCA components.

Insights:
PC1 explains the largest share of variance, and the curve flattens rapidly after a few components. However, the overall explained variance remains low relative to the complexity of the dataset, implying that genre information is not linearly encoded in these components.

6.4 Summary of Visual Insights

The complete set of visualisations demonstrates the following:

- The dataset contains strong skew, heterogeneity, and non-linearity.
- Most genres overlap heavily in raw acoustic feature space.
- High-performing genres are acoustically distinct; low-performing ones are stylistically broad.
- PCA captures little genre information, but t-SNE/UMAP (from earlier sections) reveal manifold-like structures.
- Confusion patterns accurately reflect real-world subgenre adjacency, not random noise.

Together, these visual findings justify the need for advanced non-linear models (CatBoost, XGBoost) and the detailed preprocessing pipeline developed in earlier sections.

7. Conclusion

This project set out to build a scalable, interpretable multi-class classifier for Spotify genre prediction using 30% stratified subsets of a large-scale dataset. Despite the complexity of the task, 114 overlapping genres, highly correlated audio features, and substantial class imbalance, the final XGBoost-based pipeline demonstrated consistent and stable performance across the three independent samples (A, B, C), achieving weighted F1 scores between **0.31–0.32**.

The analysis highlighted that genre classification using only acoustic features is inherently difficult, as genres are socially constructed rather than acoustically discrete. Yet the model uncovered meaningful structure: genres grounded in strong acoustic signatures (e.g., *grindcore, comedy, romance, j-idol, sleep, gospel*) were predicted reliably, while overlapping or culturally-defined genres (e.g., *techno, synth-pop, edm, songwriter, reggaeton*) remained challenging.

The interpretability study (SHAP + LIME) demonstrated that the model relied most heavily on **popularity, acousticness, energy, loudness, tempo, valence**, and engineered interaction variables, while features like *key*, *mode*, and *time signature* contributed little and were rightly deprioritised.

Dimensionality reduction analyses (PCA, t-SNE, UMAP) revealed the absence of strong linear structure in the embedding space, confirming why non-linear tree models outperformed linear baselines. The clustering patterns also illustrated the presence of multiple genre sub-spaces with varying densities.

Overall, this work demonstrates the feasibility of building an interpretable, multi-class genre classifier, while revealing the inherent boundaries of using low-level audio features for high-level semantic labels. Future work could incorporate lyrics, metadata, and user listening embeddings to significantly strengthen performance.

8. Adherence to Submission Guidelines

This report and its accompanying project materials adhere fully to the course submission requirements:

Code quality and organization
All scripts are modularized into functional components (EDA, preprocessing, feature engineering, feature selection, modelling, SHAP, dimensionality reduction, performance analysis). Each major experiment is reproducible through a dedicated file.

README completeness
A comprehensive README is included, detailing environment setup, installation, how to run each part of the project, and expected outputs.

Data access and structure
The dataset used is the 30% stratified subset (spotify_30_percent_*.csv for samples A, B, and C). Clear instructions are provided for where these files must be placed.

Execution instructions
The project runs using Python 3.11+ with all dependencies listed. Running the project requires executing main.py, B_main.py, or C_main.py depending on the subset.

File organization
The project is organized into logical directories for A, B, and C, each containing consistent preprocessing and modelling pipelines.

Naming conventions
All filenames adhere to snake_case and descriptive naming.

Figures and references
All figures are placed in the appropriate sections and labelled consistently.

References

Dataset: **https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset**

Kelleher, J. D., Mac Namee, B., & D’Arcy, A. (2015).
Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press.
Chen, T., & Guestrin, C. (2016).
XGBoost: A Scalable Tree Boosting System. KDD Conference.
Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018).
CatBoost: Unbiased Boosting with Categorical Features. NeurIPS.

Individual Contributions

Asmita Chhabra

- Led the complete development of the machine learning workflow for Sample A, including preprocessing, feature engineering, and feature selection.
- Trained, evaluated, and tuned the primary XGBoost model and baseline comparison models.
- Performed the explainability analysis using SHAP and LIME, including global and local interpretations.
- Conducted detailed class-wise performance analysis, confusion pattern investigation, and threshold sensitivity experiments.
- Generated major EDA insights, prepared several key visualisations, and interpreted dimensionality-reduction outputs (PCA, t-SNE, UMAP).

Nandika Aggarwal

- Generated major EDA insights, prepared several key visualisations, and interpreted dimensionality-reduction outputs (PCA, t-SNE, UMAP)
- Contributed to refining preprocessing logic, feature selection decisions, and validation of the pipeline.
- Supported model training and evaluation for Samples B and C, ensuring consistency across the three 30% stratified subsets.
- Assisted in generating and verifying EDA plots, class-wise analysis visuals, and dimensionality-reduction figures.
- Reviewed outputs for correctness, helped organise code modules, and drafted supporting content for the methodology and experiments sections.
- Supported error analysis, cross-checking results, and preparing figures for the final report.