

BI2025 Experiment Report - Group 02

Miha Prah*
TU Wien
Austria

Abstract

This report documents a machine learning experiment conducted by Group 02 following the CRISP-DM process model. Using the Spotify 1 Million Tracks dataset, the goal of the experiment was to assess whether the future popularity of a music track can be predicted using only intrinsic audio features and metadata available at ingestion time. The business motivation is to support early-stage playlist curation and promotional decisions before user engagement data becomes available.

The experiment covers all CRISP-DM phases, including business and data understanding, data preparation, modeling, evaluation, and deployment considerations. A Random Forest classifier was selected as the primary modeling approach due to its robustness and ability to capture non-linear relationships among heterogeneous audio features. Model training involved systematic hyperparameter tuning, followed by retraining of the selected configuration on the full training data.

The final model was evaluated on a held-out test set and compared against a majority-class baseline. Results show a substantial improvement over the baseline, particularly in Macro-averaged F1-score, indicating reliable performance across all popularity classes. The findings demonstrate that meaningful early popularity estimates can be derived from audio characteristics alone, while also highlighting limitations related to bias, temporal drift, and deployment considerations.

CCS Concepts

- Computing methodologies → Machine learning

Keywords

CRISP-DM, Provenance, Knowledge Graph, Machine Learning

ACM Reference Format:

Miha Prah and Jakov Mutvar. 2026. BI2025 Experiment Report - Group 02. In *Proceedings of Business Intelligence (BI 2025)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

*Student A, Matr.Nr.: 12434660

†Student B, Matr.Nr.: 12440619

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

BI 2025, -

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YYY/MM

<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

Jakov Mutvar[†]
TU Wien
Austria

1 Business Understanding

1.1 Data Source and Scenario

The dataset used in this project is the Spotify 1 Million Tracks collection obtained from Kaggle, containing roughly one million songs with detailed metadata—such as artist name, track title, release year, genre, and an engagement-based popularity score—alongside Spotify's engineered audio features, including danceability, energy, loudness, acousticness, instrumentality, valence, tempo, and duration. These attributes describe the intrinsic characteristics of each track independently of user behaviour and therefore allow the construction of models that attempt to estimate popularity solely from acoustic and contextual properties. The business scenario motivating this analysis is that of a music-streaming platform seeking to evaluate newly ingested tracks before any substantial listening history exists: because early playlist placement and promotion strongly influence long-term performance, the platform requires a data-driven mechanism to identify promising tracks based only on their audio profile and metadata. The dataset aligns directly with this need, providing a large, diverse basis for analysing whether a track's inherent musical properties can serve as reliable predictors of its eventual popularity.

1.2 Business Objectives

The primary objective of the streaming platform in this scenario is to strengthen its early decision-making for newly released or newly ingested tracks, for which no meaningful user engagement data is yet available. By predicting the likelihood that a track will achieve above-average popularity using only its intrinsic acoustic and metadata attributes, the platform aims to improve the efficiency of its playlist-curation and recommendation processes. More accurate early assessments enable the platform to allocate promotional exposure more selectively, reduce dependence on manual curation, and increase listener engagement by prioritizing content with high potential impact. Ultimately, the objective is to support more effective catalogue management in a context where the volume of incoming tracks exceeds the platform's capacity for human evaluation.

1.3 Business Success Criteria

Business success in this context is defined by measurable improvements in how the platform identifies and promotes promising tracks before user engagement signals accumulate. Success would be reflected in higher downstream listener engagement for tracks selected through the predictive system compared with those promoted under existing heuristics, as well as reductions in manual curation effort due to increased automation of early-stage selection. Additionally, successful deployment would lead to more efficient allocation of promotional resources, observable through improved performance of curated playlists or early-exposure campaigns. These outcomes must be attributable to the predictive system's ability to

surface high-potential tracks earlier and more consistently than current operational processes.

1.4 Data Mining Goals

The central data mining goal is to construct and evaluate a predictive model that estimates a track's future popularity class based solely on the attributes available at the time of ingestion, namely its audio features and metadata. This involves identifying which features contribute most strongly to popularity outcomes, determining whether popularity can be reliably inferred from a track's intrinsic characteristics, and quantifying the model's ability to generalize across diverse genres and time periods. Beyond predictive accuracy, the analysis also seeks to generate interpretable insights into the relationship between musical properties and commercial performance. The overarching goal is to determine whether such a model can meaningfully support the platform's early-stage decision-making process.

1.5 Data Mining Success Criteria

The success of the data mining effort is assessed through model-based performance metrics that quantify how reliably popularity can be predicted from the available features. Suitable criteria include achieving a classification performance that clearly exceeds a trivial or random baseline, maintaining stable results across validation folds, and demonstrating adequate sensitivity to tracks in the higher-popularity classes, as these are the cases of greatest business interest. The model should show consistent behaviour across genres and release years, indicating that predictive patterns are not confined to narrow subsets of the data. In addition, the resulting feature-importance patterns or model explanations should be coherent with domain understanding and provide actionable insights into the drivers of popularity.

1.6 AI Risk Aspects

Several AI-related risks must be considered in this scenario. Because popularity is strongly influenced by prior exposure and historical preference patterns, a model trained on such data may inadvertently reinforce existing biases—for example, favouring established artists or mainstream genres while disadvantaging niche or underrepresented categories. The dataset lacks demographic or contextual interaction data, making it difficult to detect or mitigate such systemic effects. There is also a risk of temporal drift, as musical tastes and platform dynamics evolve, potentially degrading model performance over time if not monitored. Finally, deploying a popularity-prediction model introduces the possibility of creating self-fulfilling feedback loops, where the system boosts tracks it predicts to be successful, thereby influencing the very outcome it attempts to measure. These risks necessitate careful evaluation and ongoing monitoring before operational use.

2 Data Understanding

2.1 Dataset Description

The dataset used in this study contains metadata and audio features for approximately one million Spotify tracks. It combines descriptive identifiers, temporal information, categorical labels, and engineered acoustic features that characterize each track independently of user interaction.

An overview of all available attributes is provided in Table 1. The dataset includes basic identifiers such as a unique track index (ID), track and artist names, and a Spotify-specific track identifier. Temporal context is captured through the release year, while genre information provides a coarse categorical representation of musical style. The target variable, *popularity*, is an integer score between 0 and 100 reflecting aggregated user engagement on the platform.

The majority of features consist of continuous audio descriptors engineered by Spotify, including danceability, energy, acousticness, instrumentalness, valence, tempo, loudness, and speechiness. These attributes quantify perceptual and structural properties of the audio signal and form the core predictive input for the modeling phase. Additional musical metadata, such as key, mode, and time signature, encode harmonic and rhythmic characteristics that may further influence listener preferences.

2.2 Dataset size and basic data quality

The dataset contains 1,159,764 rows and 20 columns, indicating a large-scale collection of Spotify tracks suitable for robust statistical analysis and model training. No duplicate records were identified in the dataset. A small number of missing values were observed, specifically 15 missing entries in the *artist_name* column and 1 missing entry in the *track_name* column. Given the very low proportion of missing data relative to the dataset size, these records are not expected to bias the analysis and will be removed during the data preparation phase.

In addition to missing-value checks, all numerical attributes were validated against their expected value ranges. Most features fully comply with their defined bounds. Two exceptions were identified: 13,888 tracks exhibit *time_signature* values below the expected minimum of 3, and 1,198 tracks have *loudness* values exceeding the expected upper bound of 0 dB. All other attributes, including *popularity*, *year*, *tempo*, and normalized audio features, show no out-of-range values. These deviations will be considered explicitly in subsequent data preparation steps.

2.3 Skewness Analysis

The skewness analysis indicates that a large number of numerical features in the dataset deviate from symmetric distributions. Several audio attributes, including *duration_ms*, *speechiness*, and *liveness*, exhibit strong positive skewness, meaning that most tracks have relatively low values while a small number of tracks show very high values. Other features, such as *instrumentalness*, *acousticness*, and *popularity*, display moderate skewness, reflecting long-tail effects that are typical for streaming platforms. In contrast, features such as *tempo*, *valence*, and *danceability* are more evenly distributed. Negative skewness observed for *energy* and *loudness* suggests a concentration of tracks at higher values, which is consistent

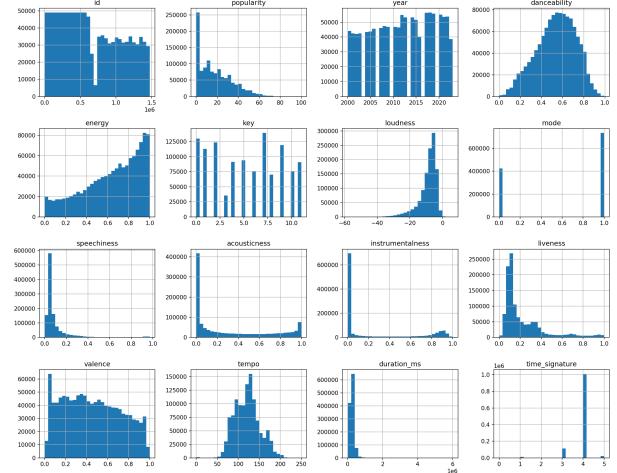
Table 1: Raw Data Features

| Feature Name | Data Type | Description |
|------------------|-----------|--|
| ID | integer> | Unique index of the track in the dataset |
| acousticness | double> | Confidence measure of whether the track is acoustic (0.0 to 1.0) |
| artist_name | string> | Name of the artist associated with the track |
| danceability | double> | Suitability of a track for dancing |
| duration_ms | integer> | Track duration in milliseconds |
| energy | double> | Perceptual measure of intensity and activity (0.0 to 1.0) |
| genre | string> | Genre label assigned to the track |
| instrumentalness | double> | Likelihood the track contains no vocals (0.0 to 1.0) |
| key | integer> | Estimated musical key of the track, encoded as integers 0–11 |
| liveness | double> | Probability the track was recorded live |
| loudness | double> | Overall loudness of the track in decibels (approx. -60 to 0) |
| mode | integer> | Modality of the track: Major (1) or Minor (0) |
| popularity | integer> | Spotify popularity score from 0 to 100 |
| speechiness | double> | Presence of spoken words in the track |
| tempo | double> | Tempo of the track in beats per minute (BPM) |
| time_signature | integer> | Estimated time signature indicating the number of beats per bar (e.g., 3 to 7) |
| track_id | string> | Spotify identifier of the track |
| track_name | string> | Title of the track |
| valence | double> | Musical positiveness conveyed by the track |
| year | gYear> | Release year of the track |

with contemporary music production practices. Overall, the results show that assumptions of normality do not hold for many variables, highlighting the importance of considering distributional properties in later analysis steps.

Figure 1 visualizes these distributional characteristics using histograms for all numeric features, making the presence of skewed

and long-tailed distributions explicit. The figure confirms that assumptions of normality do not hold for many variables, highlighting the importance of considering distributional properties in later analysis steps.

**Figure 1: Numeric Feature Histograms used for skewness analysis**

In addition to uneven distributional properties, pairwise relationships between numeric features were examined. Figure 2 shows the correlation heatmap for all numerical attributes. Most features exhibit only weak to moderate correlations, indicating limited redundancy among audio characteristics. Notable associations appear between energy and loudness, as well as between acousticness and instrumentalness, which are consistent with domain expectations. Absence of strong multicollinearity suggests that the feature set is well suited for tree-based modeling approaches such as Random Forests.

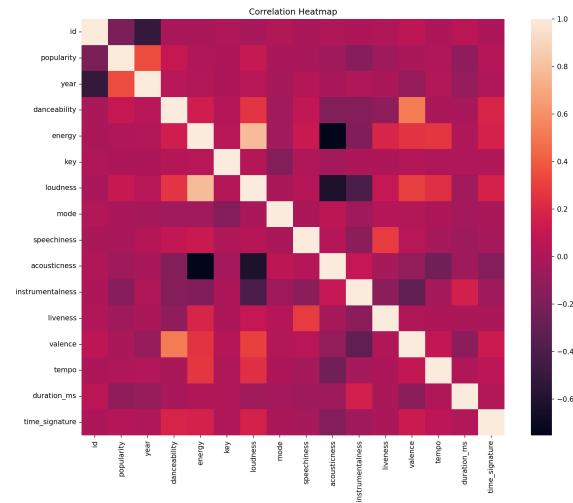
**Figure 2: Correlation heatmap of numeric audio features**

Figure 3 complements the histogram-based skewness analysis by showing kernel density estimates for all numeric features. These smooth density curves make long-tailed behavior and multimodality more apparent, further confirming that many audio attributes deviate from normal distributions.

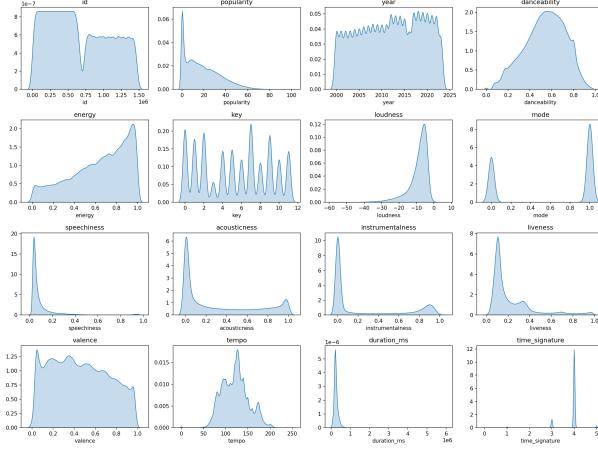


Figure 3: Numeric Feature Distributions (KDE plots)

2.4 Outlier Analysis

Outliers were identified using an IQR-based approach to assess the presence of extreme values across continuous audio features. The analysis was intended to characterize distributional properties rather than to remove observations, as many extreme values represent valid but infrequent musical characteristics. Consequently, outliers were documented but not excluded at this stage.

Figure 4 visualizes the distribution of selected numerical features using boxplots after excluding a small number of variables with extreme scales. The figure highlights the presence of long-tailed distributions and isolated extreme values across multiple audio attributes, confirming the findings from the skewness analysis. Importantly, most outliers appear systematically rather than as isolated anomalies, supporting the decision to retain them for modeling rather than applying global outlier removal.

3 Data Preparation

3.1 Applied pre-processing steps

The data preparation step was performed as a single processing activity. First, all rows containing missing values were removed. The number of such rows was very small compared to the overall dataset size, and the affected attributes are required for later analysis, making removal preferable to imputation.

Second, records containing values outside predefined valid bounds were excluded to ensure semantic correctness and consistency of the data. These bounds were derived from domain knowledge of Spotify audio features and metadata.

Identified outliers were not removed at this stage. The outlier analysis showed that extreme values mostly occur in features with naturally skewed or long-tailed distributions and likely represent

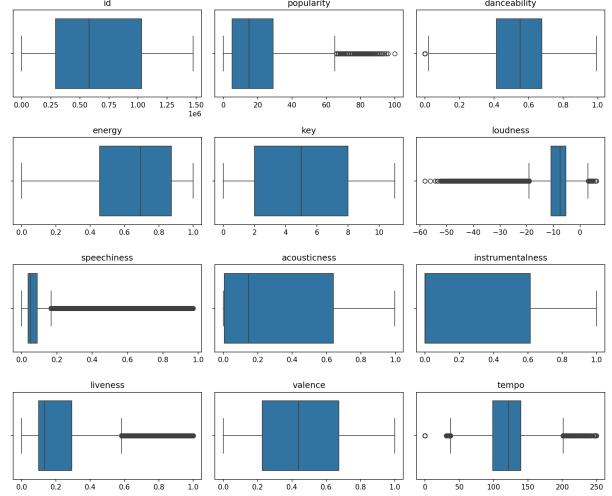


Figure 4: Filtered numeric feature boxplots for outlier inspection

valid but infrequent musical characteristics rather than data errors. Removing them globally could distort the underlying data distribution and reduce representativeness.

Finally, a new categorical attribute was created by discretizing the popularity score into five ordered classes (Very bad, Bad, Average, Good, Very good). This derived attribute supports downstream classification and interpretation while preserving the original numerical popularity measure.

As a final step of the data preparation process, a reproducible random subset of the prepared dataset was created for subsequent modeling experiments. From the fully cleaned and enriched data, a uniform random sample of 100,000 records was drawn using a fixed random seed to ensure reproducibility. This subset was persisted as a separate dataset artifact and is used consistently throughout the modeling phase in order to reduce computational effort while maintaining a representative distribution of the underlying data.

3.2 Non-applied pre-processing steps

Several additional preprocessing steps were considered but not applied. Feature scaling and normalization were not used because the chosen model type (Random Forest) does not depend on feature scaling. Techniques such as oversampling or undersampling to address class imbalance were also considered, but instead class weights were handled directly during model training.

3.3 Derived attributes

The creation of further derived attributes was also considered. Possible options included combining existing audio features or creating summary indicators. However, these were not applied because the model can already learn complex relationships between features, and the expected benefit was limited.

3.4 External data sources

Finally, potential external data sources were considered at a conceptual level. Examples include artist popularity, early streaming counts, or social media signals. These sources were not included in this analysis in order to keep the focus on audio features available at the time of track ingestion and to limit the scope of the project.

4 Modeling

4.1 Define Algorithm

The modeling task is formulated as a supervised multi-class classification problem, where the goal is to predict an ordered popularity class (Very bad, Bad, Average, Good, Very good) based on Spotify audio features and related metadata.

Several candidate classification algorithms were considered. Multinomial logistic regression was identified as a simple and interpretable baseline, but its assumption of linear decision boundaries limits its ability to capture complex relationships between audio characteristics. Support Vector Machines were also considered due to their strong theoretical properties, however they require careful feature scaling and can be computationally expensive for larger datasets. Decision Trees offer non-linear modeling capabilities and interpretability, but are known to be prone to overfitting when used as single estimators.

The Random Forest classifier was selected as the primary algorithm for the experiments. As an ensemble method that combines multiple decision trees trained on bootstrap samples with randomized feature selection, Random Forests effectively reduce variance while preserving the ability to model non-linear relationships and interactions between features. This is particularly suitable for the Spotify dataset, where popularity is influenced by complex combinations of audio attributes such as energy, danceability, tempo, and acoustics.

In addition, Random Forests are robust to noise and outliers, require minimal assumptions about the underlying data distribution, and perform well with heterogeneous feature types. These properties make the algorithm well suited for exploratory and predictive modeling in a Business Intelligence context, where robustness, stability, and reproducibility are prioritized over highly specialized model tuning.

Based on these considerations, the Random Forest classifier was chosen as the most suitable data mining algorithm for subsequent modeling and evaluation.

4.2 Define Data Split

The prepared dataset was split into three disjoint subsets for supervised learning: a training set, a validation set, and a test set. Because the target variable consists of five popularity classes (Very bad, Bad, Average, Good, Very good), stratified sampling was applied to preserve the class distribution across all subsets.

The split was defined as 60% training, 20% validation, and 20% test. A fixed random seed (`random_state=42`) was used to ensure reproducibility. No temporal or sequential dependencies between instances were assumed for this task, therefore a random stratified split was appropriate.

Resulting subset sizes are: training=60000, validation=20000, test=20000.

4.3 Identify Hyperparameters

The RandomForestClassifier exposes several hyperparameters that influence model complexity, generalization behavior, robustness to noise, and computational cost. Prior to selecting a parameter for tuning, multiple hyperparameters were examined.

The parameter `n_estimators` controls the number of trees in the ensemble. Increasing this value generally improves prediction stability but also increases training time, while performance improvements typically plateau beyond a certain number of trees. The `max_features` parameter determines how many features are considered at each split and influences the diversity of trees within the ensemble; although relevant in high-dimensional settings, it is often secondary once sufficient feature interactions are captured.

The parameters `min_samples_split` and `min_samples_leaf` act as regularization mechanisms by limiting tree growth. Higher values enforce smoother decision boundaries and can reduce overfitting, particularly in the presence of noise or outliers. The `class_weight` parameter was also considered to address potential class imbalance by re-weighting misclassification penalties, which can improve macro-averaged evaluation metrics. The `bootstrap` parameter controls whether bootstrap sampling is applied and is typically left at its default value, as it rarely serves as the primary tuning dimension.

Among these options, `max_depth` was selected as the most relevant hyperparameter for tuning. The `max_depth` parameter directly controls the complexity of individual decision trees: shallow trees may underfit complex, non-linear relationships between audio features, while very deep trees increase the risk of overfitting and computational cost. Tuning `max_depth` therefore provides a transparent and effective way to balance predictive performance, generalization, and training efficiency.

The `max_depth` parameter is evaluated over a small set of discrete values representing increasing model complexity (e.g., 5, 10, 15, 20, and an unconstrained setting). This interval-based approach allows transparent analysis of the bias-variance tradeoff while keeping computational effort manageable.

Table 2 summarizes the Random Forest hyperparameters that were identified as relevant for this experiment, together with their functional role in controlling model complexity, robustness, and computational cost. While several parameters influence different aspects of the learning process, only a subset was considered for explicit tuning, with the remaining parameters either fixed to reasonable defaults or used for regularization.

As shown in Table 2, `max_depth` was selected as the tuning dimension because it directly governs the bias-variance tradeoff of individual trees and provides a transparent mechanism for controlling model complexity.

4.4 Train And Finetune Model

Random Forest training and hyperparameter tuning were performed using a fixed train/validation split. Multiple model runs were executed by varying the hyperparameter `max_depth` over a predefined

Table 2: Identified Random Forest Hyperparameters

| Hyperparameter | Description |
|-------------------|--|
| bootstrap | Controls whether bootstrap samples are used when building trees; typically left at default. |
| class_weight | Weights associated with classes; useful for handling class imbalance and improving macro-F1. |
| max_depth | Maximum depth of individual trees; primary tuning parameter controlling model complexity and generalization. |
| max_features | Number of features considered at each split; affects tree diversity and ensemble robustness. |
| min_samples_leaf | Minimum samples required at a leaf node; acts as regularization to reduce overfitting. |
| min_samples_split | Minimum samples required to split an internal node; limits tree growth and overfitting. |
| n_estimators | Number of trees in the forest; improves stability but increases computational cost. |

grid ($\text{max_depth} \in \{5, 10, 15, 20, \text{None}\}$). All other model parameters were held constant across runs to ensure comparability and reproducibility.

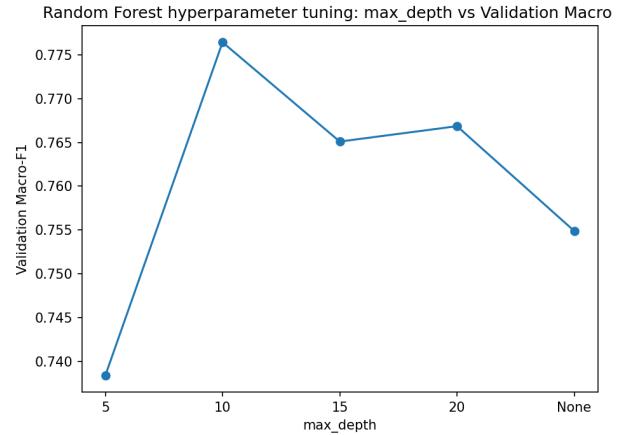
To enable experimentation with computationally more demanding configurations, the hyperparameter tuning experiments were conducted on stratified subsets containing 25% of the original training set and 25% of the original validation set. This subsampling strategy reduced computational overhead while preserving the class distribution and relative data characteristics. The reduced dataset size allowed the use of a larger ensemble size ($n_{\text{estimators}} = 1000$) to improve model stability without prohibitive runtime costs.

The following Random Forest parameters were fixed for all training runs: $n_{\text{estimators}} = 1000$, $\text{random_state} = 42$, $n_{\text{jobs}} = -1$, $\text{max_features} = \text{"sqrt"}$, $\text{class_weight} = \text{"balanced"}$, $\text{min_samples_leaf} = 2$, $\text{min_samples_split} = 4$, $\text{bootstrap} = \text{True}$, and $\text{criterion} = \text{"gini"}$. Only the max_depth parameter was varied during tuning.

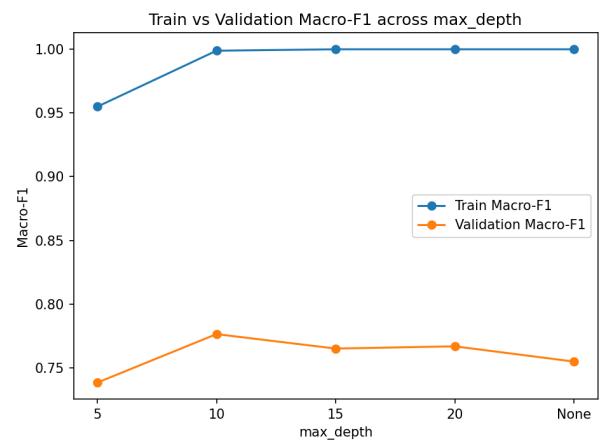
For each run, a preprocessing-and-model pipeline was trained on the (subsampled) training set. Categorical attributes (e.g., genre) were one-hot encoded (`handle_unknown="ignore"`), while numeric attributes were passed through unchanged. High-cardinality identifiers (artist_name, track_name, track_id) were excluded from the feature space. Although artist popularity is known to influence song success, artist identifiers were excluded to avoid identity-based memorization and to focus the model on generalizable musical characteristics. Track name and track identifiers were removed as they do not represent intrinsic musical properties relevant for popularity prediction.

Model performance was evaluated on both the training and validation sets using Accuracy and Macro-averaged F1-score. Macro-F1 on the validation set served as the primary model selection criterion due to its robustness under potential class imbalance across the five popularity classes.

Figure 5 shows the tuning curve of validation Macro-F1 as a function of max_depth . The results indicate a clear performance improvement when increasing model complexity from shallow trees to moderately deep trees, followed by diminishing returns for larger depths. The highest validation Macro-F1 is achieved at $\text{max_depth} = 10$, which was therefore selected as the optimal hyperparameter setting.

**Figure 5: Tuning curve: max_depth vs validation Macro-F1**

To assess potential overfitting, Figure 6 compares training and validation Macro-F1 across the same max_depth values. While training performance continues to improve with increasing depth, validation performance stabilizes and slightly declines for deeper trees, indicating overfitting beyond the selected depth. This divergence supports the choice of $\text{max_depth} = 10$ as a balanced trade-off between model expressiveness and generalization.

**Figure 6: Train vs Validation Macro-F1 across max_depth**

4.5 Retrain Final Model

The final model was retrained after hyperparameter selection using the full available training data, i.e., the union of the training and validation sets. Retraining was performed using the same preprocessing pipeline and identical hyperparameters as the best configuration identified during tuning ($\text{max_depth} = 10$). Categorical features (e.g., genre) were one-hot encoded with `handle_unknown="ignore"`, numeric features were passed through unchanged, and identifier/high-cardinality attributes (`artist_name`, `track_name`, `track_id`) were excluded from the feature space.

Fixed Random Forest parameters during retraining were:
`n_estimators = 1000`, `random_state = 42`, `n_jobs = -1`, `max_features = "sqrt"`, `class_weight = "balanced"`, `min_samples_leaf = 2`, `min_samples_split = 4`, `bootstrap = True`, and `criterion = "gini"`.

5 Evaluation

The final Random Forest model was evaluated on a held-out test set to assess whether it meets the business and data mining objectives defined in the Business Understanding phase. Model performance was compared against a majority-class baseline, which always predicts the most frequent popularity class ("Very bad"). This baseline represents a simple heuristic that reflects the limitations of early-stage decision-making when no user engagement data is available.

5.1 Baseline model comparison

The majority-class baseline achieved an accuracy of 0.605 but a very low Macro-F1 score of 0.151, showing that it largely fails to identify tracks with above-average popularity and ignores minority classes. While such a heuristic may appear acceptable when considering accuracy alone, it does not support the business goal of reliably surfacing promising tracks early.

In contrast, the final model achieved substantially higher performance across all evaluation metrics, as summarized in Table 3. The model reaches an accuracy of 0.996, a Macro-averaged F1-score of 0.920, and a Weighted F1-score of 0.996 on the test set. The strong improvement in Macro-F1 indicates that the model distinguishes well between all popularity classes rather than defaulting to the majority class. The confusion matrix shows that remaining errors mostly occur between neighboring popularity levels, which is acceptable for early-stage recommendation and curation support.

5.2 Success criteria

These results satisfy the data mining goal of predicting popularity using only intrinsic audio features and metadata available at ingestion time. The model clearly outperforms a trivial baseline, demonstrating that meaningful early popularity estimates are possible and supporting the business objective of reducing manual curation and improving early promotional decisions.

5.3 Bias assessment

To assess potential bias, musical genre was treated as a protected attribute proxy, as it defines relevant subgroups of the catalogue that could be unfairly favored or disadvantaged. Genre-wise accuracy on the test set was consistently high, with no systematic performance degradation for any genre. Minor variations are explained by stylistic diversity rather than structural bias. Overall,

no evidence of biased behavior toward specific genre groups was identified.

Table 3: Final Model Performance on Test Set

| Metric | Value |
|-------------------------|--------|
| Accuracy | 0.9959 |
| Macro-averaged F1-score | 0.8881 |
| Weighted F1-score | 0.9958 |

6 Deployment

6.1 Business Objectives Reflection and Deployment Recommendations

The final model clearly outperforms a trivial majority-class baseline and meets the data mining objective of predicting popularity classes from audio features and metadata alone. This indicates that the model can support early-stage decisions before user engagement data is available.

6.2 Ethical Aspects and Risks

The model does not use explicit sensitive attributes, but genre was treated as a protected or proxy attribute because it reflects cultural groupings. Performance across genres was consistently high, without evidence of systematic bias.

Nevertheless, there is a risk that popularity predictions could reinforce existing mainstream trends. To mitigate this, predictions should be used as guidance rather than as absolute decisions, and diversity considerations should remain part of the deployment process.

6.3 Monitoring Plan

After deployment, model performance should be monitored using overall accuracy, macro-F1 score, and per-class performance on newly ingested tracks. Changes in prediction distributions or sustained drops in macro-F1 should trigger investigation or retraining.

Comparing early predictions with later observed popularity outcomes is recommended to ensure the model remains aligned with real user engagement over time.

From a business perspective, the results are sufficient to assist playlist curation and promotional prioritization, but not to fully automate such decisions. A hybrid deployment is recommended, where the model is used to rank or filter tracks for further human review rather than to replace editorial judgment. Additional validation on newly released tracks would further strengthen deployment readiness.

6.4 Reproducibility Reflection

The workflow is largely reproducible, as preprocessing steps, hyperparameters, random seeds, and model choices are documented and linked through provenance information. This allows the experiment to be retraced with minimal ambiguity.

Remaining risks include dataset sampling choices and dependency versions. These could be further reduced by fixing dataset snapshots and explicitly versioning all external libraries.

7 Summary and Reflection

7.1 Additional comments beyond provenance

While the provenance graph captured the core CRISP-DM steps and the technical configuration of the pipeline, several practical considerations influenced the final setup.

First, we intentionally removed high-cardinality identifiers (`artist_name`, `track_name`, `track_id`) from the feature space. Even if these identifiers can correlate with popularity, they can lead to identity-based memorization rather than learning generalizable relationships between audio features and popularity. In the intended business setting (new tracks without listening history), such memorization would not translate into robust early predictions.

Second, the hyperparameter search was performed on stratified 25% subsamples of the training and validation sets. This decision was pragmatic: it reduced runtime sufficiently to allow more computationally expensive settings (e.g., larger ensembles), while still preserving the relative class distributions and main data characteristics. The final model was then retrained on the full training+validation data to maximize the use of available labeled information.

Third, the data understanding plots highlight that many audio features are strongly skewed and long-tailed (e.g., speechiness, liveness, `duration_ms`). Because such distributions are typical for real-world music catalogues, we documented outliers rather than removing them globally. This kept rare but valid musical characteristics in the dataset and avoided distorting the population.

Finally, we note that the extremely high performance values observed on the split used in this experiment may partly reflect characteristics of the dataset and the label discretization. In real deployment, continuous monitoring and periodic re-validation on newly released tracks would be necessary to ensure the model remains stable under distribution shift (new genres, production trends, or changes in user behavior).

7.2 Overall findings and lessons learned

The Random Forest approach proved highly effective for predicting popularity classes from intrinsic audio features and basic metadata. Compared to a majority-class baseline, the model achieved a large improvement in Macro-F1, indicating that it learned meaningful distinctions across all popularity classes rather than optimizing for the dominant class only. Remaining errors were mostly between neighboring classes, which is acceptable in a decision-support scenario where the model is used for ranking or filtering tracks rather than making irreversible decisions.

A key lesson learned is that evaluation with accuracy alone is insufficient when classes are imbalanced; Macro-F1 is essential to ensure minority classes are not ignored. In addition, the tuning curves were useful for diagnosing overfitting behavior and selecting a `max_depth` that balances complexity and generalization. Another lesson is that provenance-based documentation is valuable not only for reproducibility but also for structuring the report: once the graph is consistent, large parts of the report can be generated automatically with minimal manual effort.

7.3 Optional assignment feedback

The provenance workflow was useful for enforcing a structured CRISP-DM process and for keeping track of decisions (splits, parameters, and generated artifacts). The most challenging part was debugging endpoint/network issues (e.g., temporary HTTP errors) and ensuring consistent naming of entities (especially when small URI typos can prevent later queries from retrieving information). A substantial amount of time was also devoted to setting up and debugging queries for retrieving data from the server. This effort could arguably have been better spent on other aspects of the project, such as deeper analysis or further model improvements. While we clearly recognize the long-term benefits of automated, provenance-based documentation, especially for larger or recurring projects, in the context of a single assignment it introduced additional complexity and friction. In this case, the overhead of managing and querying the provenance graph outweighed the immediate benefits it provided.