CIS 5200 Fall 2025, Final Report

Classifying Spotify Music Popularity Based on Music Metadata

Team Name: SAM

Group Members: Meha Gaba, Arya Bansal, Sauman Das

- Meha Gaba; Email: mehag05@wharton.upenn.edu
- Arya Bansal; Email: abansal1@wharton.upenn.edu
- Sauman Das; Email: sauman@wharton.upenn.edu

Code: https://github.com/aryabansal1/cis5200

Abstract

This project investigates whether Spotify audio features and descriptive metadata can be used to reliably predict a song's popularity. Using a large dataset of tracks obtained from the Spotify Web API, we frame the task as a supervised regression problem and evaluate three machine learning models—Linear Regression, Random Forest, and XGBoost—across both the raw popularity score and an engineered target. Our primary contribution is the introduction of a *time-scaled popularity* metric that normalizes popularity by song age to correct for recency effects. This transformation substantially improves model stability and predictive performance. Across all experiments, XGBoost achieves the lowest MAE and RMSE and the highest R^2, indicating that boosted-tree models most effectively capture the nonlinear relationships present in musical attributes. Feature analysis shows that energy, valence, loudness, danceability, and tempo are among the strongest predictors of modern-day popularity. Overall, our results demonstrate that combining engineered temporal features with nonlinear models provides a robust framework for predicting contemporary Spotify track popularity.

1 Motivation

The goal of our project is to understand and model what makes a song popular on Spotify using measurable audio features. With the growth of streaming platforms, predicting song popularity is valuable for artists and producers. We aim to use machine learning to classify whether a song is likely to be popular" or "not popular" based on attributes such as tempo, energy, danceability, and valence. We have also decided to transform this problem into predicting a raw popularity score instead of making this a binary classification task. This allows for further flexibility in comparing the relative popularity of two songs which may both be "popular" or both "non-popular".

The three of us are personally very interested in music and work on releasing our own music as well. If we are able to develop a tool while can predict how popular a song will be, then it would be great way to assess prior to the release of a new song how well the song will do. Note that this implies that we are concerned with the current popularity of a song. We want to specifically know if the song will do well in 2025 which is why we introduced time-scaling in our project.

2 Related Work

A large body of prior work has shown that high-level audio features and descriptive metadata can be used to predict music popularity, but that nonlinear models are often required to capture the underlying structure. Gulmatico et al. propose the "SpotiPred" framework, where multiple machine learning models are trained on Spotify-derived audio features to predict popularity; tree-based models in their study outperform simple linear baselines, suggesting that popularity depends on nonlinear interactions between musical attributes.

This motivates the inclusion of ensemble methods such as Random Forest and, in this project, XGBoost, to better model complex relationships between features like energy, loudness, and danceability and the target popularity score.

Zangerle et al. study hit song prediction using both low- and high-level audio descriptors together with chart information, demonstrating that combining engineered features and nonlinear architectures yields better performance than using either type of feature alone. Their findings support this project's decision to jointly exploit Spotify's engineered audio features and additional temporal metadata, and to move beyond purely linear models when predicting continuous popularity values. In particular, the emphasis on mood- and energy-related characteristics in Zangerle's work aligns with this report's feature analysis, where energy, valence, loudness, danceability, and tempo emerge as some of the most informative predictors.

Other work has examined how low-level audio descriptors relate to listener preferences, reinforcing the value of platform-engineered features for predictive modeling. Ferwerda and Schedl show that users' music preferences correlate with audio descriptors derived from services like Spotify, including measures related to loudness and other acoustic statistics. This provides a conceptual basis for the present study's focus on readily available Spotify attributes—rather than raw waveforms—and justifies framing popularity prediction as a supervised regression problem over these structured features.

More recent research has incorporated deep learning on raw audio and metadata to forecast song success. Studies such as those by Kim et al. apply convolutional and recurrent neural networks to audio waveforms alongside metadata for pop-music popularity prediction, and often report that models using only metadata or high-level audio features can remain competitive with waveform-based approaches. This observation is consistent with the modeling choices in this project, which prioritize structured features and engineered temporal variables over computationally intensive raw-audio pipelines while still achieving strong predictive performance using XGBoost.

Finally, Pons and collaborators use deep learning architectures that jointly process raw audio and feature representations to distinguish hit from non-hit tracks, highlighting the importance of rhythmic and energy-related characteristics for success on the charts. The present work parallels this emphasis by identifying energy, tempo, and related descriptors as key predictors, but extends prior literature through the introduction of a time-scaled popularity target that explicitly accounts for song age. By normalizing popularity with a logarithmic function of song age and pairing this engineered target with boosted-tree models, this project builds on earlier findings about feature importance and nonlinearity to produce more stable and temporally meaningful predictions of modern-day Spotify popularity.

3 Dataset

This study utilizes a curated Spotify music dataset originally sourced from the Spotify Web API. The dataset was constructed using two complementary extraction pipelines designed to retrieve both popular and non-popular tracks, enabling a balanced representation across the popularity spectrum. Each entry corresponds to a unique track and includes two broad categories of information: *descriptive metadata* and *audio features*. The audio features are generated through Spotify's internal audio analysis algorithms, while the metadata describes the track, album, and playlist context.

3.1 Descriptive Metadata

The descriptive features capture high-level contextual information relevant to each track. These include the track name, primary performing artist(s), album title, and album release date. Unique identifiers provided by Spotify (track ID, album ID, playlist ID) allow for reproducibility and cross-referencing with external datasets. Additionally, the dataset includes playlist-level information - playlist name, genre, and subgenre - which facilitates genre-aware analyses. A key outcome variable, *track popularity*, ranges from 0 to 100 and represents Spotify's proprietary popularity score based on cumulative and recent user streaming behavior.

3.2 Audio Features

Spotify's audio analysis engine generates a suite of quantitative musical attributes for each track. These features, widely used in computational musicology and music information retrieval (MIR), include:

- **Energy**: A measure of perceived intensity and activity; high-energy tracks are typically fast, loud, and rhythmically dynamic.
- **Tempo (BPM)**: The estimated beats per minute, capturing the speed of the track.
- **Danceability**: A composite score describing rhythmic stability, beat strength, and temporal regularity.
- **Loudness (dB)**: Overall loudness of the track, averaged across the audio signal.
- **Liveness**: Probability that the track was recorded with a live audience.
- **Valence**: A measure of musical positiveness, with higher scores corresponding to happier affect.
- **Speechiness**: The degree to which the track contains spoken words.
- **Instrumentalness**: Probability that the track contains no vocals; values near 1.0 indicate instrumental music.
- **Mode**: Indicates whether the track is in a major or minor key.
- **Key**: Encoded as an integer from 0 to 11 representing the musical pitch class.
- **Duration (ms)**: Total length of the track in milliseconds.
- **Acousticness**: A confidence measure indicating whether the track is acoustic.

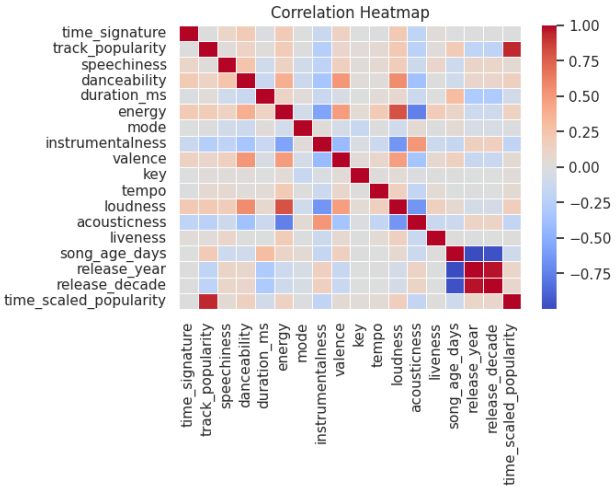
3.3 Dataset Summary

The combination of descriptive metadata and fine-grained audio features enables multi-dimensional modeling of musical properties, popularity trends, and genre-specific patterns. We did an initial analysis of the dataset to understand correlations between variables as well as identifying outliers. First, we split the data into its numerical and categorical variables.

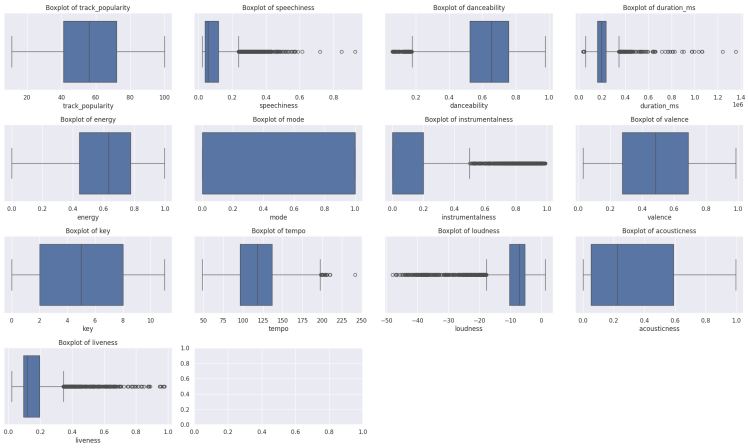
Numeric Features: time_signature, track_popularity, speechiness, danceability, duration_ms, energy, mode, instrumentalness, valence, key, tempo, loudness, acousticness, liveness, song_age_days, release_year, release_decade, time_scaled_popularity.

Categorical Features: playlist_name, track_artist, playlist_genre, playlist_subgenre, track_href, track_name, uri, type, track_album_release_date, analysis_url, id, track_album_id, playlist_id, track_id, track_album_name.

Here are the correlations between the numerical variables:



Note that we can see features "valence" and "tempo" being highly correlated with track_popularity which is the feature we are attempting to predict. We also analyzed the boxplots for all the numerical features which is displayed below:



From the box plot, we can see that certain features have many outliers such as "instrumentalness" because of the way the data is stored. That is, most values are concentrated between 0 to 0.2. With further analysis we found that there were 1039 outliers in this feature. As a result, we dropped this feature along with other features with a large number of outliers.

Because the dataset includes both popular and non-popular tracks, it supports research questions involving popularity prediction, audio-based clustering, genre classification, and temporal or stylistic analysis. The dataset is suitable for both supervised and unsupervised machine learning methods and provides an interpretable, real-world test bed for studying relationships between musical structure, metadata, and listener engagement.

In this project, we will be using supervised learning to analyze the data and understand which features are most important in determining song popularity.

4 Problem Formulation

The objective of this project is to model and understand the factors that influence a song's popularity on Spotify. Given a dataset containing descriptive metadata and audio-derived features, we formulate the task as a **supervised regression problem** in which the goal is to predict a track's popularity score - a continuous value between 0 and 100 computed by Spotify based on streaming behavior.

Let $x \in R^p$ denote the feature vector for a given track, consisting of:

- audio features such as `danceability`, `energy`, `loudness`, `valence`, `liveness`, and `tempo`
- engineered temporal features such as `song_age_days`, `release_decade`, and `time_scaled_popularity`
- selected categorical metadata encoded through one-hot or ordinal encodings.

The target variable is:

$$y = \text{track\_popularity} \in [0, 100].$$

Thus, the learning problem is to approximate a function

$$f : R^p \to [0, 100]$$

such that

$$\hat{y} = f(x)$$

minimizes predictive error on unseen tracks.

Since the outcome is continuous, we evaluate all models using regression-appropriate loss functions, primarily mean squared error (MSE) and mean absolute error (MAE). MSE provides a smooth, differentiable objective that penalizes large errors more severely, while MAE offers robustness to outliers—a useful property given the skewed distribution of popularity scores. Cross-validation is used for hyperparameter tuning and model selection to avoid overfitting.

Several characteristics of the dataset influence this formulation. First, audio features differ in scale and distribution, necessitating normalization to stabilize training. Second, the popularity variable is moderately imbalanced, with many tracks clustered at low popularity values; the regression model must therefore account for nonlinear relationships. Third, high-cardinality categorical features (e.g., `track_artist`, `playlist_genre`, `playlist_subgenre`) require encoding strategies that balance expressiveness and sparsity.

Under this framing, we compare multiple regression approaches—including linear models, tree-based ensembles, and neural models—to determine which best captures the complex relationships between musical structure, metadata, and popularity. We additionally investigate whether engineered temporal features (such as age-adjusted popularity) improve predictive accuracy, reflecting the intuition that older tracks accumulate streams differently from new releases. In the following section, we describe in detail what the temporal feature we predict is.

4.1 Time-Scaled Popularity

We decided to experiment with two different output variables. First, is the raw popularity. However, we have observed that an older song with a popularity score of 90 is likely less popular than a newer song with a popularity score of 90. Our motivation for this project is to develop a model which can help us determine whether a song will be popular in the modern day. For this reason, we introduced a new time-scaled target variable called "time_scaled_popularity". As one can tell by looking at the correlation heatmap presented in Section 3.3, there is a 100% correlation between the original track popularity and the time scaled popularity.

Although the two variables are perfectly correlated in a monotonic sense, the purpose of our transformation is not to alter the ordering of songs but to re-weight popularity in a way that corrects for differences in song age. We define the time-scaled popularity as:

$$\text{time\_scaled\_popularity} = \frac{\text{popularity}}{1 + \log(\text{song\_age\_days})}.$$

This transformation is motivated by the observation that popularity naturally *decays over time* and does so in a *nonlinear* fashion. In the first few weeks after a song is released, its streaming counts typically drop sharply; listeners rapidly move on to new music, playlists update frequently, and algorithmic recommendations shift. Later in a song's lifecycle, however, the rate of decay slows considerably. For example, the change in popularity between days 10 and 20 is typically much larger than the change between days 100 and 110. A logarithmic adjustment captures this intuition, as the logarithm grows quickly for small values and slowly for large ones. As a result, the transformation heavily down-weights very young songs (whose popularity may be inflated by recency effects) while applying relatively mild adjustments to older catalog tracks.

The denominator term $1 + \log(\text{song\_age\_days})$ therefore serves as a *recency correction factor*. Two songs with identical raw popularity but different ages are treated differently under this transformation: a newly released song with popularity 90 is likely benefiting from immediate release-cycle attention, whereas an older song with popularity 90 has demonstrated more persistent listener engagement. This distinction is central to our project, as our goal is to predict whether a song would be popular *in the modern day*, not simply whether it accumulated streams at some point in the past.

Additionally, music popularity is highly temporal: different genres, styles, and production aesthetics dominate at different points in time. By incorporating a decay term tied to song age, the time-scaled metric normalizes popularity across release eras, allowing the model to learn relationships driven by the audio and metadata characteristics themselves rather than short-term streaming surges.

In this way, the time-scaled popularity variable preserves the ranking structure of the raw popularity score while providing a target that is more temporally meaningful and better aligned with the predictive goals of our study.

5 Methods

Our modeling pipeline consists of three major stages: data preprocessing, model training, and model evaluation. Each stage was designed to ensure that the models were trained on clean, well-structured representations of the Spotify dataset and that performance comparisons across models were fair and reproducible.

5.1 Preprocessing

Before training, we performed several preprocessing steps to improve data quality and ensure compatibility with the learning algorithms:

Duplicate removal. Multiple entries corresponding to the same track were removed to avoid giving certain songs disproportionate influence during training.

Handling missing values. Missing numerical values were imputed using feature-wise means. This prevented unnecessary removal of samples and ensured that all models received the same complete feature matrix.

Normalization. All continuous numerical features were transformed using z-score normalization,

$$z = \frac{x - \mu}{\sigma},$$

to prevent models sensitive to feature scale (such as linear regression and XGBoost) from being biased toward high-magnitude variables.

Encoding categorical variables. Categorical musical attributes—`key`, `mode`, and `time_signature`—were one-hot encoded. Although these features appear ordinal, treating them numerically would incorrectly imply meaningful ordering; one-hot encoding preserves neutrality between categories.

Correlation-based feature reduction. We inspected inter-feature correlations to identify highly redundant predictors. When pairs of features exhibited near-perfect correlation, one feature was removed to improve model stability and reduce multicollinearity, particularly for linear regression.

5.2 Model Training

We trained and compared three regression models, chosen to represent a balanced mix of linear, ensemble-based, and boosted methods:

- **Linear Regression (Baseline).** Serves as an interpretable benchmark and enables evaluation of whether relationships between features and popularity are approximately linear.
- **Random Forest Regressor.** A nonlinear, tree-based ensemble method capable of modeling complex interactions. It is robust to noise and feature scaling and provides feature-importance estimates.
- **XGBoost Regressor.** A gradient-boosted tree model that captures nonlinear structure with strong predictive performance. Its regularization mechanisms help reduce overfitting.

Feature selection. Using feature-importance scores derived from the tree-based models, we selected a reduced set of the most influential variables to improve efficiency and mitigate overfitting.

Hyperparameter tuning. Each model was tuned using `GridSearchCV` with 5-fold cross-validation. Consistent with project guidelines, tuning and model selection were conducted exclusively on the training split, with the test split reserved for final evaluation.

5.3 Model Evaluation

To evaluate predictive performance, we computed the following metrics:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Coefficient of Determination (R^2)

These metrics reflect average predictive deviation (MAE), sensitivity to larger errors (RMSE), and overall explanatory power (R^2). All final evaluations were performed on a held-out 20% test set.

We evaluated each model on two target variables:

1. **Raw popularity**, the original Spotify popularity score.
2. **Time-scaled popularity**, our engineered recency-adjusted metric described in Section 4.1.

Comparing performance across these two formulations allowed us to assess how adjusting for song age influences model fit and whether modern-day popularity is more predictable when temporal decay is explicitly incorporated.

6 Experiments and Results

6.1 Experimental Setup

All models were evaluated under two complementary frameworks: (1) 5-fold cross-validation, used to estimate generalization performance and stability, and (2) a standard 80–20 train–test split, following CIS 5200 guidelines for maintaining a held-out evaluation set. Each experiment was conducted twice: once using the raw popularity score as the target, and once using the engineered time-scaled popularity variable introduced in Section 4.

Performance was measured using three regression metrics: mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R^2). These metrics jointly characterize average predictive deviation, penalization of large errors, and the proportion of variance explained by the model.

6.2 Cross-Validation Results

XGBoost consistently demonstrated the strongest performance across both target variables, while Random Forest showed the weakest performance, suggesting limited ability to capture the smooth structure of the data. Linear Regression performed moderately well and provided a strong baseline.

Table 1: Cross-Validation Results (5-Fold)

Model	**MAE**	**RMSE**	$\boldsymbol{R^2}$
	Raw Popularity		
Linear Regression	9.85 ± 1.00	13.59 ± 0.98	0.53 ± 0.06
Random Forest	15.58 ± 0.18	18.73 ± 0.19	0.10 ± 0.02
XGBoost	$\mathbf{7.88 \pm 0.29}$	$\mathbf{10.90 \pm 0.47}$	$\mathbf{0.70 \pm 0.02}$
	Time-Scaled Popularity		
Linear Regression	1.21 ± 0.09	1.66 ± 0.08	0.53 ± 0.04
Random Forest	1.82 ± 0.02	2.32 ± 0.03	0.10 ± 0.02
XGBoost	$\mathbf{0.97 \pm 0.03}$	$\mathbf{1.35 \pm 0.05}$	$\mathbf{0.69 \pm 0.02}$

6.3 Interpretation of Cross-Validation

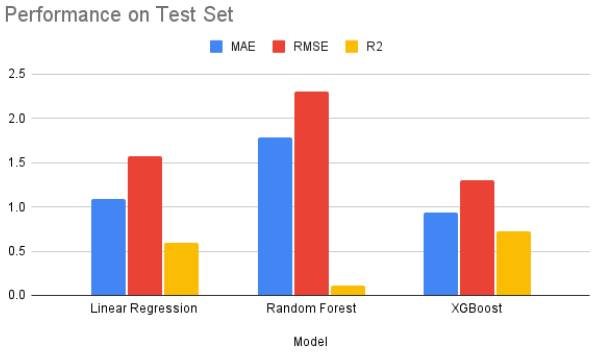
XGBoost exhibits the lowest MAE and RMSE and the highest R^2, indicating strong predictive power and robustness across folds. Linear Regression performs reasonably well, suggesting that much of the signal in the data is approximately linear. Random Forest performs significantly worse, likely due to the smooth, continuous nature of the target variables and the high cardinality of the features.

6.4 Train-Test Split Results

Held-out test performance mirrors the cross-validation pattern, confirming that the CV estimates were reliable and that the models generalize appropriately. Again, XGBoost outperforms the alternatives by a substantial margin.

Table 2: Train–Test Split Results (80–20)

Model	MAE	RMSE	R^2
Raw Popularity			
Linear Regression	8.57	12.30	0.61
Random Forest	15.32	18.54	0.11
XGBoost	**7.45**	**10.17**	**0.73**
Time-Scaled Popularity			
Linear Regression	1.09	1.57	0.59
Random Forest	1.79	2.30	0.11
XGBoost	**0.94**	**1.30**	**0.72**



6.5 Key Findings

- **XGBoost is the strongest model overall**, outperforming all other methods across every metric.
- **Time-scaled popularity is easier to predict**: all models show significantly reduced MAE and RMSE when predicting the engineered target.
- **Random Forest underperforms consistently**, suggesting that the underlying relationships are smoother and better captured by linear or boosted-tree structures.
- Agreement between cross-validation and held-out performance indicates that the models generalize well and are not overfitting to the training folds.

7 Conclusion and Discussion

This project demonstrates that Spotify's audio and descriptive features can meaningfully predict a track's popularity, but our primary contribution is the introduction of a **time-scaled popularity** metric that

adjusts for the natural decay in listener engagement as songs age. This engineered feature proved to be the most impactful aspect of our modeling pipeline: across all models, the time-scaled target yielded substantially lower MAE and RMSE and higher R^2, indicating that correcting for recency effects reduces noise in the target variable and produces more stable learning signals.

Among the evaluated models, XGBoost consistently achieved the best performance for both raw and time-scaled popularity, while Linear Regression provided a surprisingly strong baseline and Random Forest underperformed. Feature inspection further revealed that energy, valence, loudness, danceability, and tempo were among the strongest predictors of popularity, aligning with prior findings in music information retrieval.

Two main lessons emerged from this project: (1) thoughtful feature engineering—in particular, temporal normalization through our time-scaling transformation—can meaningfully improve predictive performance, and (2) boosted-tree models such as XGBoost are especially effective for capturing nonlinear relationships in musical data. Future extensions could incorporate richer audio embeddings, model temporal streaming dynamics, or frame popularity prediction as a ranking task.

Overall, our results show that combining Spotify's audio features with our novel time-scaling transformation yields a more robust and reliable framework for predicting modern-day music popularity.

References

1. Ferwerda, B., Schedl, M. (2016). Predicting music preferences using low-level audio features. In UMAP Workshops.

2. Gulmatico, J. S., Susa, J. A. B., Malbog, M. A. F., Acoba, A. G., Nipas, M. D., Mindoro, J. N. (2022). SpotiPred: A machine learning approach prediction of Spotify music popularity by audio features. In 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T).

3. Kim, J., et al. (2021). Deep learning models for pop music popularity prediction.

4. Maheshwari, Chhavi. "Music recommendation on spotify using deep learning." arXiv preprint arXiv:2312.10079 (2023)

5. Pons, J., et al. (2019). Neural approaches for hit song prediction.

6. Zangerle, E., et al. (2019). Hit song prediction: Leveraging low- and high-level audio features. In Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR).