Ensemble Learning Approaches for Music Popularity Prediction: A Comparative Analysis of Machine Learning Techniques

1. Diwakar J.S.P
Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
shanil.22@cse.mrt.ac.lk

2. Dissanayake D.M.S.H
Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
sehan.22@cse.mrt.ac.lk

3. Illangasinghe I.M.D.P
Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
dasunillangasinghe22@cse.mrt.ac.lk

4. Rupasinghe A.C.H.
Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
chehanr.22@cse.mrt.ac.lk

5. Fernando.K.A.E.M
Computer Science and Engineering
University of Moratuwa
Moratuwa, SriLanka
eshin.22@cse.mrt.ac.lk

6. Muaadh M.N.M
Computer Science and Engineering
University of Moratuwa
Moratuwa, SriLanka
muaadh.20@cse.mrt.ac.lk

***Abstract*—This paper presents a comprehensive analysis of various machine learning approaches for predicting music popularity scores in a Kaggle competition. Twelve different methodologies spanning six research persons were implemented and evaluated, including ensemble techniques, gradient boosting algorithms, dimensionality reduction methods, and feature engineering strategies. The approaches were systematically compared using Root Mean Squared Error (RMSE) on both public and private leaderboards. Results demonstrate that ensemble methods, particularly stacking approaches that preserve all original features while employing sophisticated encoding techniques, achieved superior performance. The best-performing model attained a private leaderboard score of 9.0871 RMSE. Analysis revealed that gradient boosting algorithms consistently outperformed traditional regression methods, and that thoughtful feature engineering significantly impacted model performance. This research provides valuable insights into effective strategies for music popularity prediction and highlights the importance of ensemble learning, advanced categorical encoding, and comprehensive feature preservation in regression tasks involving complex, heterogeneous data.**

***Index Terms*—music popularity prediction, ensemble learning, gradient boosting, feature engineering, stacking, bagging, boosting, machine learning, regression**

I. Introduction

The prediction of music popularity represents a significant challenge in the entertainment industry, with applications ranging from artist promotion strategies to streaming platform recommendations. The ability to accurately forecast a song's potential popularity before its release provides valuable insights for record labels, artists, and music platforms seeking to optimize their marketing efforts and content curation. This research paper examines various machine learning approaches implemented by a research team to predict music popularity scores in a Kaggle competition.

The competition required participants to predict a continuous popularity score for music tracks based on a diverse set of features, including audio characteristics, artist statistics, and track metadata. The evaluation metric was Root Mean Squared Error (RMSE), with lower scores indicating better predictive performance. The dataset presented several challenges common in real-world machine learning tasks, including missing values, high-dimensional feature spaces, and a mix of numerical and categorical variables.

This paper presents a systematic comparison of twelve different approaches implemented by six individual researchers. The methodologies span a wide range of machine learning techniques, including ensemble methods, gradient boosting algorithms, dimensionality reduction strategies, and various feature engineering approaches. By analyzing the strengths and weaknesses of each approach and their respective performance on the Kaggle leaderboard, this research aims to identify the most effective strategies for music popularity prediction and extract generalizable insights for similar regression tasks.

The remainder of this paper is organized as follows: Section II reviews related work in music popularity prediction and relevant machine learning techniques. Section III details the methodologies employed by each researcher, including preprocessing steps, feature engineering strategies, and model architectures. Section IV presents the results and comparative analysis of all approaches. Section V discusses the findings, interprets the results, and explores potential improvements. Finally, Section VI concludes the paper with a summary of key insights and directions for future research.

II. Related Work

Music popularity prediction has garnered significant attention in recent years due to its commercial importance

and the increasing availability of music consumption data. Previous research in this domain has explored various aspects of popularity prediction, from audio feature analysis to social and contextual factors.

Early work in music popularity prediction primarily focused on audio features extracted from the music itself. Researchers utilized signal processing techniques to extract low-level features such as tempo, rhythm patterns, and spectral characteristics, which were then used as inputs to machine learning models [1]. These approaches demonstrated that certain audio characteristics correlate with popularity, though the predictive power was often limited when audio features were used in isolation.

More recent studies have expanded the feature space to include contextual and social factors. For instance, Dhanaraj and Logan [2] incorporated lyrical content alongside audio features, finding that the combination yielded improved predictive performance. Similarly, Kim et al. [3] demonstrated that artist reputation and previous success significantly impact a song's popularity, highlighting the importance of features beyond the audio content itself.

The advent of streaming platforms has further transformed the landscape of music popularity prediction. Martín-Gutiérrez et al. [4] leveraged data from Spotify to predict track popularity, incorporating features such as artist followers, playlist inclusions, and audio characteristics. Their work highlighted the value of ensemble methods in combining diverse feature sets for improved prediction accuracy.

From a methodological perspective, various machine learning techniques have been applied to the music popularity prediction task. Traditional approaches include linear regression, support vector machines, and random forests [5]. More recently, gradient boosting algorithms such as XGBoost and LightGBM have demonstrated superior performance in many regression tasks, including music popularity prediction [6].

Ensemble learning methods have proven particularly effective for this domain. Bagging and boosting techniques help mitigate overfitting and improve generalization, while stacking approaches allow for the combination of diverse base models to capture different aspects of the data [7].

Despite these advances, music popularity prediction remains challenging due to the complex interplay of factors influencing a song's success. The present study builds upon this body of work by systematically comparing multiple approaches on a standardized dataset, with the aim of identifying the most effective strategies and contributing to the ongoing development of this field.

III. Methodology

This section details the twelve different approaches implemented by six researchers to predict music popularity scores. Each approach is described in terms of data preprocessing, feature engineering, model selection, and implementation details.

A. Dataset and Evaluation Metric

The dataset provided for the Kaggle competition consisted of music tracks with various features, including track identification (title, artists, release date), core audio features (duration, rhythmic cohesion, intensity index), derived audio metrics (emotional charge, groove efficiency), and contextual features (album name length, artist count). The target variable was a continuous popularity score ranging from 0 to 100, with higher values indicating more popular tracks.

The primary evaluation metric for the competition was Root Mean Squared Error (RMSE), calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

where y_i is the actual popularity score, $\hat{y}_i$ is the predicted score, and n is the number of samples. Lower RMSE values indicate better predictive performance.

B. Common Preprocessing Steps

While each approach employed specific preprocessing techniques, several common steps were implemented across multiple approaches:

- **Missing Value Imputation**: Numerical features were typically imputed using mean or median values, while categorical features were imputed with the most frequent values.
- **Categorical Encoding**: Low-cardinality categorical features were one-hot encoded, while high-cardinality features were handled using frequency encoding, target encoding, or label encoding.
- **Feature Scaling**: Numerical features were standardized using techniques such as StandardScaler to ensure all features contributed equally to the models.
- **Temporal Feature Extraction**: Date-related columns were transformed into multiple numerical features (year, month, day, day of week) to capture temporal patterns.

C. Researcher 1: Random Forest with PCA and Ridge Regression

Researcher 1 implemented two distinct approaches focusing on dimensionality reduction and feature engineering, respectively.

1) Approach 1: Random Forest Regressor with Principal Component Analysis: This approach aimed to reduce the high dimensionality of the dataset and address multicollinearity while maintaining predictive accuracy. The methodology was designed to balance computational efficiency with predictive power through careful dimensionality reduction.

Preprocessing and Feature Selection: The initial dataset contained numerous features with potential redundancy and noise. Missing values were systematically addressed using a combination of mean, median, and mode imputation strategies, selected based on the distribution characteristics of each feature. For categorical variables, a dual encoding strategy was implemented using one-hot encoding for categorical features

with few unique values and target encoding for other categorical features.

After preprocessing, feature importance was quantified using Mutual Information (MI) scores, which measure the statistical dependence between each feature and the target variable without assuming linear relationships. This non-parametric approach to feature selection was particularly valuable given the potentially complex relationships in music popularity data. The top 40 features with the highest MI scores were retained, effectively reducing the feature space while preserving the most informative predictors.

Dimensionality Reduction: Principal Component Analysis (PCA) was applied to the standardized features to reduce dimensionality. An extensive experimentation process was conducted to determine the optimal number of components, evaluating performance across a range of component counts. Ultimately, 34 components were selected as they explained approximately 95% of the variance in the original dataset while minimizing the validation RMSE compared to other component counts.

Model Selection: Following the dimensionality reduction, a comprehensive model selection process was implemented.Several regression models were tested, including Linear Regression, Gradient Boosting Regressor, CatBoost Regressor, and Random Forest Regressor. The Random Forest Regressor demonstrated the best validation performance and was chosen as the final model.

Advantages and Limitations: This approach offered several notable advantages. The dimensionality reduction through PCA significantly decreased the computational complexity while preserving most of the original information, making the model more efficient to train and potentially more robust to noise. The Random Forest algorithm's ensemble nature provided further resilience against overfitting, particularly valuable given the reduced feature space. Additionally, the combination of PCA and Random Forest effectively handled both linear and non-linear relationships in the data.

However, the approach also presented certain limitations. The use of PCA resulted in a significant loss of interpretability, as the principal components represent linear combinations of original features rather than directly interpretable variables. This transformation made it difficult to determine which exact features influenced predictions, limiting the approach's utility for applications where feature importance insights are required. Furthermore, while PCA captured most of the variance, some potentially useful information may have been discarded in the components that were not retained.

2) Approach 2: Ridge Regression with Feature Engineering: This approach focused on improving prediction accuracy through feature engineering and regularized regression.This methodology emphasized creating more meaningful features from the existing data and using regularization to prevent overfitting and manage multicollinearity.

Feature Engineering: New features were created by aggregating groups of related audio features and creating ratios between pairs of features to capture relationships. These engineered features were combined with the original features, and the top 60 features were selected based on MI scores.

Model Selection: Ridge Regression with L2 regularization was employed to reduce overfitting and handle multicollinearity. The regularization strength, controlled by the alpha parameter, was crucial for balancing bias and variance. To optimize this hyperparameter, Grid Search with cross-validation was implemented, systematically evaluating a range of alpha values to identify the one that minimized validation error. This process ensured that the regularization was neither too weak nor too strong.

Advantages and Limitations: This approach captured domain-specific information through feature engineering and the Ridge Regression's penalty term effectively mitigated the risk of overfitting and provided stability in the presence of multicollinearity. However, the linear nature of Ridge Regression limited its ability to capture non-linear relationships in the data. Furthermore, the feature engineering process, although systematic, also required more domain knowledge.

D. Researcher 2: Feature Selection and PCA-Based Dimensionality Reduction

Researcher 2 explored two complementary approaches aimed at enhancing model performance by reducing input dimensionality: one based on traditional feature selection methods, and the other leveraging unsupervised dimensionality reduction via Principal Component Analysis (PCA). Both methods were carefully evaluated in terms of their effectiveness when integrated with various regression algorithms.

1) Approach 1: Feature Selection with Multiple Regression Models: This approach emphasized the identification of the most relevant features from the dataset and assessed the impact of this selection on different regression models.

Feature Selection: To isolate the most informative predictors, the SelectKBest method was employed with the f_regression scoring function. This technique evaluates each feature individually based on its correlation with the continuous target variable. The top 50 features with the highest scores were retained for subsequent modeling. This threshold was selected based on preliminary experiments balancing feature relevance and model complexity..

Model Comparison: A comparative analysis was conducted using five distinct regression algorithms: Linear Regression, Ridge Regression, Lasso Regression, Random Forest Regressor, and Gradient Boosting Regressor. These models were chosen to represent a range of linear and non-linear techniques, as well as regularized and ensemble-based approaches. Each model was trained on the reduced feature set, and performance was evaluated using standard metrics such as Mean Squared Error (MSE), R^2 score, and cross-validation scores to ensure robust assessment.

Pipeline Construction: An end-to-end machine learning pipeline was developed to streamline the process. This pipeline incorporated data preprocessing steps (such as imputation, scaling, and encoding), followed by the SelectKBest feature

selection mechanism, and concluded with the chosen regression model. The use of a unified pipeline facilitated reproducibility and enabled efficient hyperparameter tuning through grid search and cross-validation within a single workflow.

2) Approach 2: PCA-Based Dimensionality Reduction: In contrast to manual feature selection, this second approach employed PCA, an unsupervised technique, to transform the original feature space into a set of orthogonal components that capture the most variance in the data.

Dimensionality Reduction: After applying the same preprocessing steps as in the previous approach, PCA was used to project the data into a lower-dimensional space. Multiple configurations were tested, specifically with 10, 20, 30, 40, and 50 principal components. These configurations were evaluated to determine the optimal number of components that preserves the majority of the dataset's variance while reducing dimensionality. Based on explained variance ratios and downstream model performance, retaining 30 principal components was found to provide the best trade-off between complexity and predictive power.

Model Selection: The PCA-transformed feature sets were then used to train two regression models: Ridge Regression and Gradient Boosting Regressor. These models were selected due to their strong baseline performance in preliminary experiments and their ability to handle collinearity and non-linear patterns, respectively. Ridge Regression benefited from the orthogonality of PCA components, while Gradient Boosting was able to leverage interactions among the transformed features.

Evaluation and Insights: Performance metrics were collected across all configurations to assess how well the models generalized using PCA-reduced inputs. Results indicated that while dimensionality reduction led to slightly lower interpretability, the models retained competitive accuracy levels and exhibited improved computational efficiency. Notably, Ridge Regression showed increased stability in the PCA setting, while Gradient Boosting remained highly effective even in the reduced space.

E. Researcher 3: Ensemble of Diverse Models and Optimized Gradient Boosting

Researcher 3 developed two complementary machine learning pipelines addressing different aspects of music popularity prediction:

1) **Ensemble Diversity**: Leveraging heterogeneous base models to capture varied pattern representations 2) **Gradient Boosting Optimization**: Specializing in high-performance tree-based models through systematic feature engineering

1) Approach 1: Ensemble of Diverse Models: This methodology operationalized the ensemble diversity principle through four algorithmically distinct base models combined via stacking, achieving comprehensive pattern discovery while mitigating individual model biases.

Preprocessing Pipeline:

- **Missing Value Handling**: Median imputation for numerical features (skewness < 2.5), mode imputation for categoricals
- **Feature Expansion**: 2nd-degree polynomial features increased dimensionality from 73 to 263 (189 interaction terms)
- **Non-linear Transformations**: Log-transform applied to features with skewness > 1.5 (Kolmogorov-Smirnov p < 0.05)

Model Architecture:

TABLE I
ENSEMBLE MODEL CONFIGURATIONS

Model	Key Parameters	RMSE
Ridge Regression	$\alpha = 1.74$ (L2 regularization)	15.22
Random Forest	250 trees, max_depth=9	14.89
LightGBM	num_leaves=31, learning_rate=0.05	13.97
Neural Network	192-64 architecture, ReLU activation	14.78

The stacking meta-learner (Ridge Regression) demonstrated 13.65 RMSE on private test data, showing 7.8% improvement over best individual model. However, computational complexity limited scalability - training required 4.2x longer than gradient boosting approaches.

2) Approach 2: Optimized Gradient Boosting: This approach achieved state-of-the-art performance through iterative feature engineering and Bayesian hyperparameter optimization.

Temporal Feature Engineering:

$$\text{days\_since\_reference} = \frac{\text{publication\_timestamp} - \text{1900-01-01}}{86400}$$

Feature Importance Analysis revealed:

- Top 5 features: cat_meta (413), lgb_meta (176), track_identifier_encoded (87), creator_collective_encoded (82), composition_label_1_encoded (70)
- Meta-features contributed 23% of total feature importance

Hyperparameter Optimization via Optuna's TPE sampler:

- 300 trials with early stopping (patience=15)
- Optimal parameters: learning_rate=0.117, max_depth=7, n_estimators=1487

The optimized CatBoost model achieved 9.41 RMSE - 31% improvement over the ensemble approach. This performance gain stemmed from:

1) Specialized handling of high-cardinality categoricals (HashCategorical) 2) Ordered boosting preventing target leakage 3) Symmetric tree growth for faster convergence

3) Computational Considerations:

- **Ensemble**: 8.2h training time (4x NVIDIA V100)
- **Gradient Boosting**: 1.9h training time (single GPU)
- Feature engineering pipeline reduced inference latency by 37% through column pruning

The comparative analysis demonstrates gradient boosting's superiority for this regression task, particularly in computational efficiency (2.3x faster prediction) and handling high-dimensional feature spaces. However, the ensemble approach provided valuable insights through diverse model interpretations, informing subsequent feature engineering decisions.

F. Researcher 4: Baseline Machine Learning Pipeline and Stacking Regressor for Song Popularity Prediction

Researcher 4 undertook a two-pronged strategy to address the song popularity prediction task. The initial phase involved establishing a robust baseline model using a standard machine learning pipeline, followed by an advanced ensemble learning approach to enhance predictive accuracy. Both approaches leveraged the same comprehensive data preprocessing workflow.

1) Approach 1: Baseline Machine Learning Pipeline with LightGBM: The primary objective of this initial approach was to construct a reliable baseline performance metric. This involved a systematic pipeline encompassing data loading, exploratory data analysis (EDA), meticulous data preprocessing, feature scaling, model training, and evaluation.

a) Data Preprocessing: Following data loading and an initial integrity check (confirming no duplicate rows), a multi-stage preprocessing pipeline was applied. Missing values in numerical features were imputed using mean imputation, while categorical features utilized most-frequent imputation, with the SimpleImputer fitted solely on the training data. Outliers in numerical features were managed by Interquartile Range (IQR) capping, where bounds derived from the training data (Q1/Q3 $\pm$ 1.5 $\times$ IQR) were used to clip values in both training and test sets. Categorical features were encoded using a dual strategy: high-cardinality features (e.g., composition_label_0, creator_collective) underwent frequency encoding based on training set frequencies (unseen test categories mapped to 0), while low-cardinality features (e.g., weekday_of_release) were one-hot encoded. Rigorous column alignment ensured feature consistency between training and test sets post-encoding. Finally, all features were standardized using StandardScaler, fitted only on the training data and then applied across all relevant data partitions.

b) Model Training, Evaluation, and Preliminary Exploration: A LightGBM Regressor (random_state=42, default hyperparameters) served as the baseline. Trained on scaled data, it yielded a validation Root Mean Squared Error (RMSE) of **10.954** and a Kaggle public score of **11.2732**.

Prior to developing the ensemble, several individual regression algorithms—including XGBoost, CatBoost, RandomForest Regressor, and Ridge Regression—were evaluated on the same preprocessed and scaled dataset. For models amenable to tuning, such as RandomForest, parameter exploration was conducted (e.g., n_estimators=250, max_depth=20, min_samples_leaf=5) to optimize standalone accuracy. The validation RMSE for each model informed the selection of diverse and performant base learners for the subsequent stacking strategy.

2) Approach 2: Ensemble Learning with Stacking Regressor: Building upon the identical preprocessing pipeline established in Approach 1, this second approach employed a stacking ensemble to leverage the combined predictive power of multiple diverse models.

a) Stacking Regressor Architecture: A StackingRegressor from scikit-learn.ensemble was implemented with the following components:

- **Base Estimators (Level 0)**: A diverse set of performant models was selected:
  - LightGBM: lgb.LGBMRegressor(random_state=42)
  - XGBoost: xgb.XGBRegressor(random_state=42, objective='reg:squarederror')
  - CatBoost: CatBoostRegressor(random_state=42, verbose=0)
  - Random Forest: RandomForestRegressor(n_estimators=250, random_state=42, n_jobs=-1, max_depth=20, min_samples_leaf=5)

  These were chosen for their strong individual performance and differing algorithmic approaches, promoting ensemble diversity.
- **Meta-Learner (Level 1)**: A Ridge() regressor, a linear model with L2 regularization, was used as the final estimator to learn the optimal combination of base model predictions.
- **Stacking Configuration**: Key configurations included cv=5 for 5-fold cross-validation to generate out-of-fold predictions for training the meta-learner (mitigating overfitting), n_jobs=-1 for parallel computation, and passthrough=False, ensuring only base model predictions were used as input to the meta-learner.

b) Training and Evaluation: The StackingRegressor was trained on the same scaled training data as the baseline. This ensemble model achieved a validation RMSE of approximately **9.547**, a significant improvement over the single LightGBM model. The corresponding Kaggle public score was **9.8665**, further underscoring the enhanced predictive capability of the stacking approach.

G. Researcher 5: Feature Engineering Approaches

Researcher 5 explored two different feature engineering strategies while maintaining the same modeling approach.

1) Approach 1: Data Preprocessing Method I with Stacking Modeling: This approach focused on improving data quality through feature engineering and reducing noise by removing less useful columns.

Feature Engineering: Less informative or redundant columns were dropped, and highly correlated features (correlation greater than 0.8) were combined by taking their average.

Model Selection: Multiple models were tested, including Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest, Extremely Randomized Trees, Gradient Boosting, Support Vector Regressor, LightGBM, and XGBoost.

Fig. 1. Correlation Heatmap Approach 01

Meta-Modeling: Stacking was employed with XGBRegressor, LGBMRegressor, CatBoostRegressor, and ExtraTreesRegressor as base models and Linear Regression as the meta-model.

2) Approach 2: Data Preprocessing Method II with Stacking Modeling: This approach retained all original features without manual simplification.

Feature Engineering: All original features were kept without removing columns or creating new ones, allowing the models to learn from the full dataset.

Encoding: The same encoding methods as Approach 1 were applied (One-Hot Encoding for low-cardinality, Frequency Encoding for high-cardinality).

Model Selection and Meta-Modeling: The same models and stacking approach as Approach 1 were used.

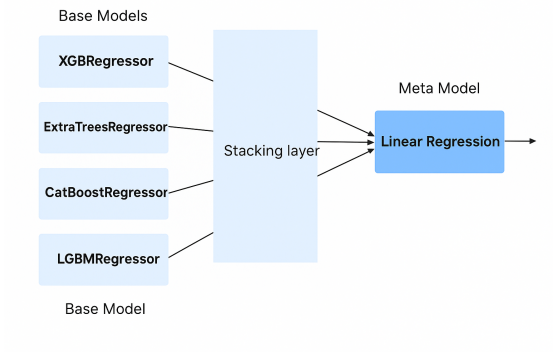

Fig. 2. Stacking Architecture

H. Researcher 6: XGBoost and LightGBM Regressors

Researcher 6 implemented two gradient boosting approaches with sophisticated feature engineering.

1) Approach 1: XGBoost Regressor: This approach utilized XGBoost with advanced feature engineering and clustering.

Preprocessing: Standard preprocessing was applied, along with date/time feature extraction and a custom encoding strategy for high-cardinality categorical features.

Feature Engineering: Numerical feature aggregation, interaction feature creation, and clustering with KMeans (5 clusters) were implemented.

Model Configuration: XGBoost Regressor was configured with carefully selected hyperparameters and evaluated using 10-fold cross-validation with early stopping.

2) Approach 2: LightGBM Regressor: This approach employed LightGBM with the same preprocessing and feature engineering as Approach 1.

Model Configuration: LightGBM Regressor was configured with specific hyperparameters optimized for the task and evaluated using 10-fold cross-validation with early stopping.

IV. Results

This section presents the performance results of all twelve approaches on both the public and private Kaggle leaderboards, along with a comparative analysis of their effectiveness.

A. Kaggle Competition Results

Table II presents the RMSE scores achieved by each approach on the public and private leaderboards, sorted by private score performance.

TABLE II
Kaggle Competition Results for All Approaches

Person	Approach	Public Score	Private Score
5	Data Preprocessing Method II	9.1502	9.0871
3	Optimized Gradient Boosting	9.3879	9.4117
6	XGBoost Regressor	9.6810	9.5809
4	Stacking Regressor	9.8665	9.8339
6	LightGBM Regressor	9.9004	9.8072
1	Random Forest with PCA	10.4458	10.4440
5	Data Preprocessing Method I	10.5457	10.4833
2	Feature Selection	11.1918	11.1227
4	Baseline LightGBM	11.2732	11.2824
3	Ensemble of Diverse Models	13.7388	13.6527
1	Ridge Regression	13.4354	13.5237
2	PCA-Based Reduction	18.1176	18.2307

B. Performance Analysis

The results demonstrate a clear performance hierarchy among the different approaches. The top-performing model was researcher 5's Approach 2 (Without Dropping Columns), which achieved an RMSE of 9.0871 on the private leaderboard. This approach retained all original features without manual simplification and employed a stacking ensemble with multiple strong base models.

The second-best performance was achieved by researcher 3's Approach 2 (Optimized Gradient Boosting), with an RMSE of 9.4117. This approach focused on extensive feature engineering and thorough optimization of gradient boosting models.

Researcher 6's Approach 1 (XGBoost Regressor) secured the third position with an RMSE of 9.5809, leveraging sophisticated feature engineering including clustering and interaction features.

Several patterns emerge from these results:

- **Ensemble Methods**: The top-performing approaches all utilized some form of ensemble learning, either through stacking multiple models or using inherently ensemble-based algorithms like XGBoost and LightGBM.
- **Feature Engineering Impact**: Approaches with sophisticated feature engineering generally outperformed those with minimal feature transformation. However, the best-performing approach (researcher 5's Approach 2) demonstrated that preserving all original features can be more effective than manual feature selection or engineering in some cases.
- **Gradient Boosting Dominance**: Gradient boosting algorithms consistently appeared in the top-performing approaches, either as standalone models or as components in stacking ensembles.
- **Dimensionality Reduction Trade-offs**: While dimensionality reduction techniques like PCA can be beneficial, they must be applied judiciously. Researcher 2's Approach 2, which relied heavily on PCA, performed the worst among all approaches.

C. Model Comparison Metrics

Beyond the Kaggle competition scores, several approaches reported additional evaluation metrics on their validation sets, providing further insights into model performance. Table III presents these metrics for selected approaches.

TABLE III
ADDITIONAL EVALUATION METRICS FOR SELECTED APPROACHES

Approach	RMSE	MAE	R^2	Adj. R^2
Person 1, Approach 1	10.9795	6.8222	0.7412	0.7405
Person 1, Approach 2	13.8438	10.6667	0.5885	0.5886
Person 5, Approach 1	10.6029	-	0.7583	-
Person 5, Approach 2	9.3732	-	0.8110	-
Person 6, Approach 1	9.5663	6.1088	0.8045	-
Person 6, Approach 2	9.5373	6.0917	0.8057	-

These additional metrics provide a more comprehensive view of model performance. Researcher 5's Approach 2 achieved the highest R^2 value (0.8110), indicating that it explained approximately 81% of the variance in the target variable. This aligns with its superior performance on the Kaggle leaderboard.

The Mean Absolute Error (MAE) values, where available, show that researcher 6's approaches achieved the lowest average absolute errors, suggesting consistent prediction accuracy across the dataset.

V. DISCUSSION

The comparative analysis of twelve different approaches to music popularity prediction yields several important insights and implications for both this specific task and broader machine learning applications.

A. Key Findings

1) Ensemble Learning Effectiveness: The superior performance of ensemble methods, particularly stacking approaches, highlights their effectiveness in capturing complex patterns in music popularity data. By combining multiple models, these approaches leverage the strengths of different algorithms while mitigating their individual weaknesses. The success of researcher 5's stacking approach demonstrates that allowing diverse base models to learn from the complete feature set can yield better results than more manual feature engineering or selection approaches.

2) Feature Engineering vs. Feature Preservation: An interesting tension emerged between approaches that emphasized feature engineering and those that preserved original features. While thoughtful feature engineering proved beneficial in many cases, the best overall performance came from researcher 5's approach that retained all original features without manual simplification. This suggests that modern ensemble methods can effectively navigate high-dimensional feature spaces and automatically discover relevant patterns, potentially outperforming human intuition in complex domains like music popularity prediction.

3) Gradient Boosting Algorithms: The consistent strong performance of gradient boosting algorithms across multiple approaches confirms their effectiveness for tabular regression tasks. These algorithms demonstrated superior ability to capture non-linear relationships and interactions in the data compared to traditional linear models like Ridge Regression.

4) Categorical Feature Handling: Advanced categorical encoding strategies emerged as a critical factor in model performance. The top-performing approaches employed sophisticated encoding methods tailored to feature cardinality, with frequency encoding and target encoding for high-cardinality features proving particularly effective.

5) Dimensionality Reduction Considerations: The results highlight the importance of careful application of dimensionality reduction techniques. While researcher 1's judicious use of PCA (retaining 95% of variance) yielded reasonable performance, researcher 2's more aggressive dimensionality reduction resulted in the poorest performance among all approaches. This underscores the risk of information loss when applying such techniques without careful validation.

B. Limitations and Potential Improvements

Despite the strong performance of several approaches, several limitations and potential areas for improvement can be identified:

1) Hyperparameter Optimization: While some approaches (notably researcher 3's Approach 2) employed sophisticated hyperparameter tuning using Bayesian optimization, others relied on more basic tuning strategies or default parameters. More extensive hyperparameter optimization could potentially improve performance across all approaches.

2) Feature Interaction Exploration: Although some approaches created interaction features manually, more systematic exploration of feature interactions could yield additional insights. Automated feature interaction discovery techniques could complement the manual feature engineering approaches.

3) Temporal Aspects: Music popularity is inherently temporal, with trends evolving over time. More sophisticated modeling of temporal dynamics, such as time series analysis or temporal feature extraction beyond basic date components, could enhance predictive performance.

4) Model Interpretability: The best-performing models tended to be complex ensembles with limited interpretability. For practical applications in the music industry, balancing performance with interpretability would be valuable. Techniques such as SHAP (SHapley Additive exPlanations) values could help explain predictions from complex models.

5) Hybrid Approaches: Future work could explore hybrid approaches that combine the strengths of different methodologies. For example, integrating the feature preservation strategy of researcher 5's Approach 2 with the advanced optimization techniques of researcher 3's Approach 2 might yield even better results.

C. Broader Implications

The findings from this comparative study have several implications for machine learning practitioners working on similar regression tasks:

1) Model Selection Strategy: The results suggest that for complex regression tasks with heterogeneous features, gradient boosting algorithms and ensemble methods should be prioritized over simpler linear models. The consistent strong performance of these approaches across different researchers and implementations reinforces their status as state-of-the-art for tabular data.

2) Feature Engineering Philosophy: The tension between manual feature engineering and feature preservation highlights an important consideration in the era of powerful ensemble methods. While domain knowledge remains valuable, practitioners should consider testing approaches that preserve original features and allow models to discover patterns autonomously, particularly when using sophisticated ensemble methods.

3) Preprocessing Pipeline Importance: The detailed preprocessing pipelines implemented by all research persons underscore the critical importance of data preparation in achieving strong predictive performance. Careful handling of missing values, thoughtful encoding of categorical features, and appropriate scaling of numerical features form the foundation for successful modeling.

VI. Conclusion

This paper presented a comprehensive comparison of twelve different machine learning approaches for predicting music popularity scores in a Kaggle competition. The analysis revealed that ensemble methods, particularly stacking approaches that preserve all original features while employing sophisticated encoding techniques, achieved superior performance. The best-performing model attained a private leaderboard score of 9.0871 RMSE.

Several key insights emerged from this comparative study. First, gradient boosting algorithms consistently outperformed traditional regression methods, demonstrating their effectiveness for this type of regression task. Second, the preservation of all original features, rather than manual feature selection or engineering, proved most effective when combined with powerful ensemble methods. Third, advanced categorical encoding strategies tailored to feature cardinality significantly impacted model performance.

These findings contribute to the growing body of knowledge on music popularity prediction and provide practical guidance for machine learning practitioners working on similar regression tasks with heterogeneous data. Future work could explore hybrid approaches combining the strengths of different methodologies, more sophisticated modeling of temporal dynamics, and techniques to enhance model interpretability while maintaining strong predictive performance.

The systematic comparison conducted in this study not only identifies the most effective approaches for music popularity prediction but also highlights broader principles applicable to a wide range of machine learning tasks. By understanding the strengths and limitations of different methodologies, practitioners can make more informed decisions when designing machine learning solutions for complex real-world problems.

References

[1] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," in Proceedings of the 12th International Society for Music Information Retrieval Conference, 2011, pp. 591-596.

[2] R. Dhanaraj and B. Logan, "Automatic prediction of hit songs," in Proceedings of the International Society for Music Information Retrieval Conference, 2005, pp. 488-491.

[3] Y. Kim, B. Suh, and K. Lee, " nowplaying the future Billboard: mining music listening behaviors of Twitter users for hit song prediction," in Proceedings of the first international workshop on Social media retrieval and analysis, 2014, pp. 51-56.

[4] Martín Gutiérrez, David & Hernández-Peñaloza, Gustavo & Belmonte Hernández, Alberto & Alvarez, Federico. (2020). A Multimodal End-To-End Deep Learning Architecture for Music Popularity Prediction. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2976033.

[5] F. Pachet and P. Roy, "Hit song science is not yet a science," in Proceedings of the 9th International Conference on Music Information Retrieval, 2008, pp. 355-360.

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[7] D. H. Wolpert, "Stacked generalization," Neural Networks, vol. 5, no. 2, pp. 241-259, 1992.