

A Machine Learning Approach to Predict Music Popularity Scores

Diwakar J.S.P. – 220144P

1. Introduction

This report describes two different approaches used to predict music track popularity scores in a Kaggle regression competition. The goal of the competition was to predict a continuous popularity score using a dataset of music-related features. The performance of each model was evaluated using Root Mean Squared Error (RMSE), the main metric used by the competition.

The two approaches are:

- **Random Forest Regressor with Principal Component Analysis (PCA)**
- **Ridge Regression with Feature Engineering**

Both methods aimed to reduce overfitting and improve prediction accuracy. Mutual Information (MI) scores were used for feature selection in both approaches. This report explains how each model was built, compares their results, and discusses their pros and cons. The final Kaggle score and leaderboard ranking are also reported.

2. Dataset Description

The dataset includes a mix of:

- Audio features (e.g., `organic_immersion_[0-2]`, `emotional_charge_[0-2]`)
- Contextual features (e.g., `album_component_count`, `artist_count`)
- Temporal features (e.g., `release_month`, `release_year`)

The target variable is a popularity score, which is a continuous number.

Preprocessing Steps:

Missing values were filled using mean, median and mode according to the distribution of each column for numerical features. For categorical columns mode is used. Since the missing amount is considerable dropping was not much a good option. One-hot encoding was applied to categorical features with a few numbers of unique values such as 'season_of_release' and target encoding for other categorical features. 'publication_timestamp' feature is splitted into 'release_year' and 'release_month' for better usage.

Top features were identified using Mutual Information (MI) scores to keep only the most predictive ones. Both approaches used the same processed data set.

3. Methodology

3.1 Random Forest Regressor with Principal Component Analysis

This approach aimed to reduce the high dimensionality of the dataset and address

multicollinearity, while still building a reasonably accurate prediction model.

The original dataset contained a large number of features. High-dimensional data can make machine learning models more complex, slower, and prone to overfitting. To address this:

The top 40 most informative features by MI scores were standardized, because it was essential since PCA is sensitive to the scale of data, and unscaled features could dominate the principal components.

Principal Component Analysis (PCA) was applied to the standardized features to reduce dimensionality. After testing different numbers of components, 34 components were selected, as they explained around 95% of the variance in the original dataset and produced the lowest RMSE on the validation set compared to other component counts.

After generating the PCA components, several regression models were tested, such as Linear Regression, Gradient Boosting Regressor, CatBoost Regressor, Random Forest Regressor, etc. Among these, the Random Forest Regressor gave the best validation performance in terms of prediction accuracy (lowest RMSE and MAE), and was chosen as the final model for this approach.

Advantages and disadvantages of this approach:

PCA helped reduce the number of input variables while preserving most of the original information. This made the model simpler and faster to train. Also PCA converts correlated features into a set of uncorrelated components, solving the issue of multicollinearity. Furthermore, Random Forest is robust to noise and overfitting, especially when combined with dimensionality reduction and it gave better prediction results than simpler models.

It's difficult to explain which exact feature affects the prediction which leads to loss of interpretability. Also, even though the PCA components covered most of the variance, some useful information may still be lost from the discarded components. Although Random Forest improves accuracy, it is more computationally expensive and harder to interpret compared to linear models as well.

3.2 Ridge Regression with Feature Engineering

This approach focused on improving prediction accuracy by creating new meaningful features from the existing data and using regularized regression to prevent overfitting and manage multicollinearity. The original dataset provided several audio and metadata features. While informative, these alone might not capture all the underlying patterns influencing a track's popularity. Therefore, feature engineering was done to better represent the data.

These 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 Mutual Information (MI) scores were calculated to assess their importance with respect to the popularity score. From this, the top 60 features were selected based on MI scores.

For this approach, Ridge Regression was used. This is an extension of linear regression that includes L2 regularization. The regularization term helps in:

- Reducing overfitting by penalizing large coefficients.
- Handling multicollinearity by shrinking the influence of correlated predictors.
- Improving generalization to unseen data.

The model was evaluated by testing different counts of top features (e.g., top 20,30,40, 50, 55, etc.). The best validation performance was achieved using 55 features. To optimize the performance of Ridge Regression, Grid Search was applied to find the best alpha value (regularization strength). This ensures the balance between underfitting and overfitting is well-tuned.

Advantages and disadvantages of this approach:

Feature engineering is useful to find out domain-specific information that the model could not learn from raw data alone. And Ridge Regression's penalty term helped reduce overfitting and improve the model's performance on unseen data while making it robust to multicollinearity, even when using many related features. Also, Grid search ensured the model used an optimized regularization strength for the best possible validation accuracy.

The combination of engineered features, MI filtering, and hyperparameter tuning made the pipeline more complex than simpler methods like PCA. Also more computation is required and while Ridge can handle complex datasets better than plain linear regression, it still assumes a linear relationship between features and target, which may limit performance in capturing non-linear trends.

4. Evaluation Metrics

The following metrics were used to evaluate both approaches on the validation set. These metrics provide insights into both the accuracy of the predictions and the efficiency of the models in capturing patterns from the data.

RMSE (Root Mean Squared Error) : RMSE measures the square root of the average of the squared differences between predicted and actual values. It gives more weight to larger errors, making it sensitive to outliers.

MAE (Mean Absolute Error) : MAE measures the average of the absolute differences between predicted and actual values. It's less sensitive to outliers than RMSE and provides a straightforward interpretation of prediction error.

R^2 : This shows the proportion of variance in the target variable that is explained by the model. A value of 1 indicates perfect prediction, and a value of 0 means the model performs no better than the mean.

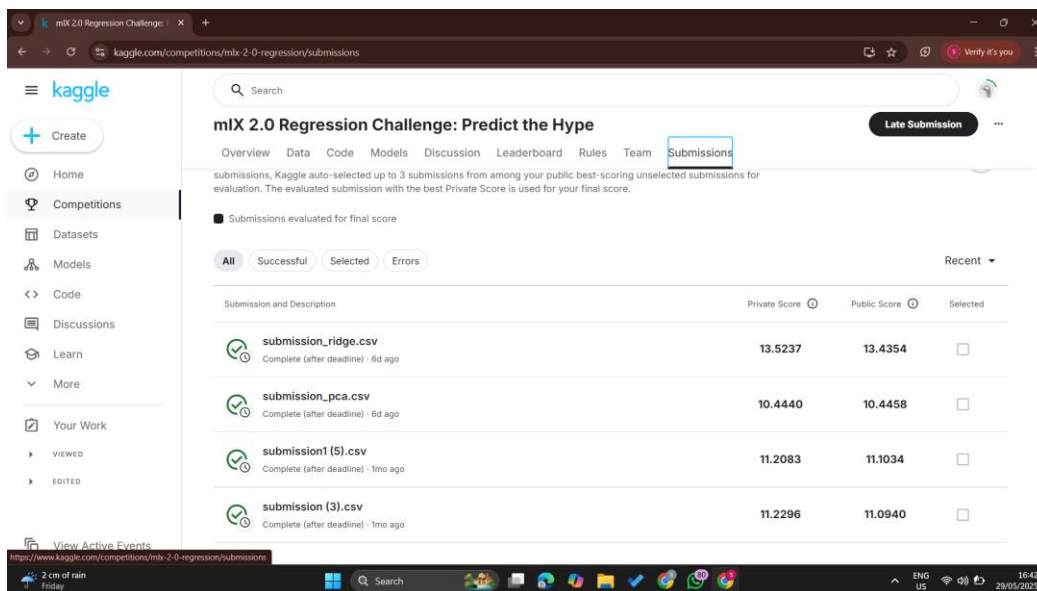
Adjusted R^2 : This is a modification of R^2 that penalizes the use of too many predictors. It adjusts for the number of features used and is useful when comparing models with different feature counts. A higher value indicates a better-fitting model without overfitting.

Metric	Approach 1 (PCA + Random Forest)	Approach 2 (Ridge + Feature Engineering)
RMSE	10.9795	13.8438
MAE	6.8222	10.6667
R ²	0.7412	0.5885
Adjusted R ²	0.7405	0.5886

Approach 1 (PCA + Random Forest) showed better performance across all metrics. It had a significantly lower RMSE and MAE, indicating more accurate and consistent predictions. Its R² and Adjusted R² values were also higher, meaning it explained more variance in the target variable, even with fewer features. However, Ridge Regression still offered good generalization due to regularization and showed decent explanatory power.

5. Results and Discussion

Two different regression approaches were evaluated to predict music popularity scores:



Submission and Description	Private Score	Public Score	Selected
submission_ridge.csv Complete (after deadline) · 6d ago	13.5237	13.4354	<input type="checkbox"/>
submission_pca.csv Complete (after deadline) · 6d ago	10.4440	10.4458	<input type="checkbox"/>
submission1 (5).csv Complete (after deadline) · 1mo ago	11.2083	11.1034	<input type="checkbox"/>
submission (3).csv Complete (after deadline) · 1mo ago	11.2296	11.0940	<input type="checkbox"/>

Evaluation metrics and Kaggle scores showed that the Random Forest with PCA model outperformed the Ridge Regression model. This confirmed that the PCA-based model generalized better to unseen test data.

The higher performance of the Random Forest model can be because of PCA effectively handled the high dimensionality of the dataset and reduced multicollinearity, which is crucial for models sensitive to feature redundancy. Moreover, Random Forests are well-suited for capturing nonlinear relationships in data.

While the Ridge Regression approach was enhanced through manual feature engineering and L2 regularization, it was ultimately limited by its linear assumptions. Although, it showed reasonable generalization and served as a useful baseline.

In conclusion, the PCA + Random Forest Regressor approach proved to be more robust and accurate for predicting music popularity, both in validation and competition testing scenarios.