Interpretable Machine Learning for Robotic Choreography Evaluation:
Predictive Modeling and SHAP-Based Insights

Ehsan Ramezani
ehsan.ramezani@studio.unibo.it
Student ID: 0001109969

Razieh Soleimanbeigi
razieh.soleimanbeigi@studio.unibo.it
Student ID: 0001119907

Mohammad Pourtaheri
mohammad.pourtaheri@studio.unibo.it
Student ID: 0001121324

July 8, 2025

1 Introduction

Robot choreography has progressed from handcrafted, expert-designed routines—traditionally evaluated by small expert panels—to automated, AI-driven generation and scalable, data-informed evaluation methods.

Despite these advances, the systematic prediction and interpretable explanation of audience judgments remains a largely underexplored area. In this study, we apply machine learning to a dataset comprising 8,563 unique humanoid-robot performances [1][2], annotated with 19 motion features and 7 audience-rated Likert-scale targets. Our goals are twofold:

1. **Prediction:** Develop robust classifiers and regressors capable of accurately forecasting audience scores from low-level motion descriptors, thereby enabling efficient offline screening of new routines.
2. **Explanation:** Uncover the underlying rationale behind model predictions to inform choreographers and HRI engineers, facilitating an iterative and interpretable design process—rather than relying on black-box models.

Our work contributes both a high-performance predictive framework and actionable design insights. Together, these advances support the scalable and data-driven development of expressive robotic performances.

2 Exploratory Data Analysis

The exploratory data analysis was conducted on the robotic choreography dataset to understand the underlying patterns, distributions, and relationships within the data prior to model development. This comprehensive analysis provides critical insights into the dataset characteristics and informs subsequent modeling decisions.

2.1 Data Cleaning and Preparation

The initial dataset comprised 8,624 observations with 26 variables As shown in Table 1, representing various features of robotic choreographies and their corresponding evaluation scores. Following data cleaning procedures, including standardization of text formats in the musicGenre variable and removal of duplicate entries, the final dataset contained 8,563 unique observations. The preprocessing phase also involved one-hot encoding of the categorical musicGenre variable, expanding the feature space to 32 dimensions.

Table 1: Dataset Features and Target Variables Description

Variable Name	**Type**	**Description**
	Choreographic Features	
timeDuration	Continuous	Duration of choreography in seconds
nMovements	Continuous	Total number of different movements in choreography
movementsDifficulty	Ordinal (1-3)	Degree of movement difficulty (1=low, 2=medium, 3=high)
robotSpeech	Binary (0/1)	Whether robot speech is included in performance
acrobaticMovements	Ordinal (1-3)	Level of acrobatic movements (1=low, 2=medium, 3=high)
movementsRepetition	Ordinal (1-3)	Level of movement repetitions (1=low, 2=medium, 3=high)
musicGenre	Categorical	Music genre (folk, electronic, rock, pop, classical, latin, indie, rap)
movementsTransitionsDuration	Ordinal (1-3)	Duration level of transitions between movements
humanMovements	Ordinal (1-3)	Level of human-like movement presence
balance	Ordinal (1-3)	Level of balance movements incorporated
speed	Ordinal (1-3)	Degree of movement speed (1=slow, 2=medium, 3=fast)
bodyPartsCombination	Ordinal (1-3)	Level of combinations involving different body parts
musicBPM	Continuous	Beats per minute of accompanying music
sameStartEndPositionPlace	Binary (0/1)	Whether start and end positions are identical
headMovement	Ordinal (1-3)	Level of head movement combinations
armsMovement	Ordinal (1-3)	Level of arm movement combinations
handsMovement	Ordinal (1-3)	Level of hand movement presence
legsMovement	Ordinal (1-3)	Level of leg movement presence
feetMovement	Ordinal (1-3)	Level of feet movement presence
	Evaluation Metrics (Target Variables)	
StoryTelling	Ordinal (1-5)	Audience evaluation of choreography's narrative quality
Rhythm	Ordinal (1-5)	Audience evaluation of rhythmic coherence with music
MovementTechnique	Ordinal (1-5)	Audience evaluation of movement technique and fluidity
PublicInvolvement	Ordinal (1-5)	Audience evaluation of public engagement level
SpaceUse	Ordinal (1-5)	Audience evaluation of spatial utilization effectiveness
HumanCharacterization	Ordinal (1-5)	Audience evaluation of human-like characteristics
HumanReproducibility	Ordinal (1-5)	Audience evaluation of human reproducibility potential

2.2 Feature Distributions

An analysis of feature distributions revealed key characteristics of the dataset.

- Continuous variables such as `nMovements` and `musicBPM` follow approximately normal distributions.
- The skewness analysis identified `timeDuration` as the only highly skewed variable (skewness = 14.046, kurtosis = 207.23), suggesting the presence of outliers or extreme values in choreography duration. All other numerical features demonstrated moderate skewness ($|\text{skewness}| < 1$), indicating relatively normal distributions suitable for parametric analyses.
- The majority of movement-related features follow a relatively balanced distribution across their 1-3 scale. For instance, `movementsDifficulty` shows counts of 2,880 for level 1, 3,120 for level 2, and 2,563 for level 3, indicating a slight preference for medium-difficulty movements. Similar patterns emerge across other ordinal features, with most showing relatively even distributions across the three levels.
- A review of the `musicGenre` feature, based on the generated bar plot of the data, highlighted a notable class imbalance. Genres like 'folk' and 'electronic' are far more represented than 'indie' or 'rap'.

2.3 Correlation Analysis

To investigate the relationships between variables, a Spearman's rank correlation matrix was generated from the data. This non-parametric method was chosen for its robustness to outliers and its suitability for the dataset's numerous ordinal features. The analysis of this correlation matrix reveals two primary insights.

First, there are strong, positive correlations among the six **main artistic evaluation targets**: Storytelling, Rhythm, Movement Technique, Public Involvement, Space Use, and Human Characterization. This indicates that when the audience rated a choreography highly on one of these artistic metrics, they were likely to rate it highly on the others as well, suggesting a consistent perception of overall performance quality.

Second, the correlations between the input features and the evaluation targets are generally weak. The most discernible relationship is a slight negative correlation between nMovements and movements TransitionsDuration and most of the evaluation targets. This suggests that choreographies with a higher number of movements did not necessarily receive higher scores, a finding that aligns with the analysis of AI-created choreographies in prior research [1].

2.4 Target Variable Analysis

The seven target variables, representing audience scores on a 1-to-5 Likert scale, were analyzed to understand their distributions before modeling (4). The generated histograms of the original scores show that most targets have multi-modal distributions with peaks at integer values, which is characteristic of Likert data. The distribution for EvaluationChoreographyHumanReproducibility is notably different, exhibiting a strong left skew with the majority of ratings being 4 or 5. This confirms that audiences consistently found the choreographies easy to reproduce.

For the classification task, these target variables were binarized to represent a positive (1) or negative (0) evaluation (3).

- Scores from 1 to 3 were mapped to Class 0 (low evaluation).
- Scores from 4 to 5 were mapped to Class 1 (high evaluation).

3 Classification Analysis

This section describes the methodology and results of the classification models developed to predict the binarized choreography evaluation scores. The goal is to identify the most effective model for each of the seven evaluation metrics. Four distinct classification algorithms were implemented: Logistic Regression, Random Forest, XGBoost, and CatBoost.

3.1 Methodology and Justification of Model Selection:

We chose tree-based ensemble models, especially gradient boosting implementations like XGBoost and CatBoost, as the main emphasis of this classification task, coupled with Random Forest and Logistic Regression (baseline model). This decision was based on a combination of their inherent strengths with tabular data, empirical evidence from testing automated machine learning (AutoML) exploration, and practical considerations regarding model interpretability and computational efficiency.

Better Performance on Structured Tabular Data: Tree-based models are widely recognized as the state-of-the-art for structured (tabular) data, which characterizes the choreography dataset used in this project. They outperform linear models in automatically capturing complex, non-linear relationships and feature interactions without the need for extensive manual feature engineering. The exploratory data analysis (EDA) showed that the relationships between the choreographic features and the evaluation targets were not strongly linear. Such findings made tree-based models a more suitable alternative. Additionally, they are naturally resistant to outliers and do not require feature scaling, simplifying the preprocessing pipeline and reducing the risk of poor scaling decisions—especially for skewed features such as timeDuration.

Using AutoML Frameworks for Empirical Validation: To confirm the architectural choice, initial tests were performed using an AutoML framework (AutoGluon). This system automatically trained and evaluated a wide range of models, including boosted trees, linear models, and more complex deep neural networks (DNNs) tailored for tabular data. The results consistently demonstrated that gradient-boosted tree models were highly competitive. Since tree-based models provided comparable or superior predictive accuracy without the added complexity and training time of DNNs, focusing on them proved to be a smart and data-driven decision.

Computational Efficiency and Interpretability of SHAP: A key objective of this study is not only to predict outcomes, but also to understand which choreographic elements influence audience perception. The SHAP (SHapley Additive exPlanations) framework is well-suited for this purpose. The computational cost of calculating SHAP values varies significantly across model families. The TreeExplainer algorithm is highly optimized for tree-based models, offering fast and accurate calculation of SHAP values. In contrast, explaining predictions from neural networks or transformer-based models such as TabPFN requires heavier computation (e.g., via GradientExplainer, DeepExplainer, or sampling-based approaches). These methods are typically slower and more memory-intensive, making large-scale interpretability analysis and the generation of comprehensive summary plots across multiple targets impractical. By selecting tree-based models, we ensure efficient and comprehensive interpretability, enabling deeper insights into the "why" behind the predictions.

A systematic approach was adopted for building, tuning, and evaluating the predictive models:

3.1.1 Data Splitting and Preprocessing

The task was framed as a separate binary classification problem for each of the seven evaluation metrics. For each of the seven target variables, the dataset was independently split into a training set (80%) and a holdout test set (20%). A stratified splitting strategy was employed to ensure that the proportion of positive and negative classes was maintained in both the training and test sets, which is crucial for handling imbalanced data.

A scikit-learn `Pipeline` was constructed to streamline the workflow for each model. This automated process included:

- **Imputation:** Missing values in the feature set were handled using K-Nearest Neighbors Imputation (`KNNImputer` with n=5).
- **Scaling:** For the Logistic Regression model, which is sensitive to feature scaling, numerical features were standardized using `RobustScaler`. Tree-based models (Random Forest, XGBoost, CatBoost) do not require feature scaling, so this step was omitted from their pipelines.

3.1.2 Hyperparameter Tuning and Evaluation

To find the optimal configuration for each model and target, `GridSearchCV` was utilized with 3-fold stratified cross-validation on the training data. The search was optimized to maximize the F1-score, chosen as the primary metric (`refit='f1'`) due to its effectiveness in evaluating models on imbalanced datasets.

Model performance was assessed using a comprehensive set of metrics on the unseen holdout test set:

- **F1-Score:** As the primary metric, it provides a harmonic mean of precision and recall.
- **ROC AUC:** To measure the model's ability to distinguish between positive and negative classes.
- **Precision and Recall:** To provide a detailed understanding of the model's performance concerning false positives and false negatives, respectively.
- **Confusion Matrix:** To visualize the counts of true/false positives and negatives.

3.2 Model Performance

The following subsections detail the performance of each of the four models across all seven target variables after hyperparameter tuning.

3.2.1 Logistic Regression

The `Logistic Regression` model served as a linear baseline. Its performance was generally moderate, indicating that linear relationships alone may not be sufficient to capture the full complexity of audience evaluations. The hyperparameter grid included `C` = {0.001, 0.01, 0.1, 1, 10, 100, 1000} and `penalty` = {`'l1'`, `'l2'`}.

Table 2: Logistic Regression Performance on Holdout Set

Target Variable	F1-Score	ROC AUC	Precision	Recall
HumanCharacterization	0.4773	0.5334	0.4223	0.5487
HumanReproducibility	0.6368	0.5321	0.7848	0.5358
MovementTechnique	0.5506	0.6648	0.5028	0.6084
PublicInvolvement	0.5267	0.6658	0.4498	0.6354
Rhythm	0.5973	0.6776	0.6062	0.5887

Table 2 continued

Target Variable	F1-Score	ROC AUC	Precision	Recall
SpaceUse	0.4927	0.6510	0.4074	0.6234
StoryTelling	0.5316	0.6802	0.4985	0.5695

3.2.2 Random Forest

The `Random Forest` classifier, an ensemble of decision trees, demonstrated a notable improvement over Logistic Regression, particularly in its discriminative power (AUC). This suggests that non-linear interactions between features play a significant role in predicting audience scores. The grid search explored the following parameter ranges: `n_estimators` = {100, 200, 300}, `max_depth` = {None, 10, 20}, `min_samples_split` = {2, 5, 10}, `min_samples_leaf` = {1, 2, 4}, and `max_features` = {`'sqrt'`, `'log2'`}.

Table 3: Random Forest Performance on Holdout Set

Target Variable	F1-Score	ROC AUC	Precision	Recall
HumanCharacterization	0.4873	0.5972	0.4792	0.4957
HumanReproducibility	0.8502	0.5631	0.7680	0.9521
MovementTechnique	0.5501	0.6977	0.5425	0.5580
PublicInvolvement	0.5192	0.7014	0.5052	0.5341
Rhythm	0.5966	0.7243	0.6515	0.5502
SpaceUse	0.4915	0.6935	0.4362	0.5628
StoryTelling	0.5489	0.7206	0.5438	0.5540

3.2.3 XGBoost

The `XGBoost` model, a gradient boosting implementation, consistently yielded strong results, often achieving the highest F1-scores and AUC values among the tested models, especially for the `Rhythm` and `MovementTechnique` targets. The hyperparameter search space included: `n_estimators` = {100, 200, 300}, `learning_rate` = {0.01, 0.05, 0.1}, `max_depth` = {3, 5, 7}, `subsample` = {0.7, 0.9, 1.0}, and `colsample_bytree` = {0.7, 0.9, 1.0}.

Table 4: XGBoost Performance on Holdout Set

Target Variable	F1-Score	ROC AUC	Precision	Recall
HumanCharacterization	0.5437	0.5692	0.4376	0.7178
HumanReproducibility	0.8052	0.5539	0.7829	0.8288
MovementTechnique	0.5708	0.7097	0.5149	0.6403
PublicInvolvement	0.5267	0.6988	0.4703	0.5985
Rhythm	0.6315	0.7345	0.6553	0.6094
SpaceUse	0.5121	0.7002	0.4077	0.6883
StoryTelling	0.5778	0.7246	0.5222	0.6467

3.2.4 CatBoost

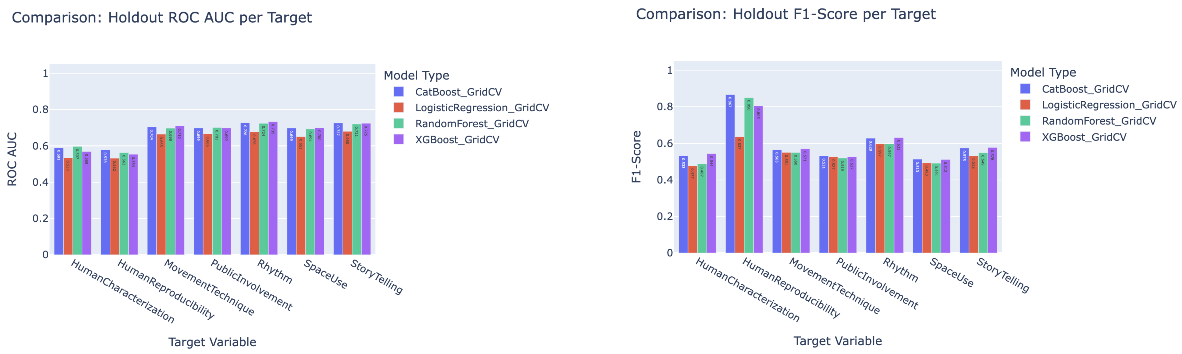
`CatBoost`, another powerful gradient boosting library, delivered competitive performance, excelling particularly on the heavily imbalanced `HumanReproducibility` target, where it achieved the highest F1-score among all models. The hyperparameter search space included: `iterations` = {100, 200, 300}, `learning_rate` = {0.01, 0.05, 0.1}, `depth` = {3, 5, 7}, `l2_leaf_reg` = {1, 3, 5, 7, 9}, and `auto_class_weights` = {`'Balanced'`, `'None'`}.

Table 5: CatBoost Performance on Holdout Set

Target Variable	F1-Score	ROC AUC	Precision	Recall
HumanCharacterization	0.5332	0.5907	0.4574	0.6390
HumanReproducibility	0.8674	0.5779	0.7671	0.9977
MovementTechnique	0.5647	0.7039	0.5269	0.6084
PublicInvolvement	0.5310	0.6989	0.4714	0.6077
Rhythm	0.6281	0.7280	0.6528	0.6052
SpaceUse	0.5134	0.6985	0.4109	0.6840
StoryTelling	0.5747	0.7266	0.5217	0.6398

3.3 Model Comparison and Final Selection

To select the single best model for each target, their performance on the holdout set was compared, prioritizing the F1-Score, followed by the ROC AUC score. Figures 1a and 1b provide a visual summary of this comparison. The tree-based ensemble methods consistently outperformed the linear baseline model, with XGBoost and CatBoost emerging as the top contenders for most targets. An interactive selection process confirmed the final models, which were then saved for interpretability analysis with SHAP.



(a) ROC AUC

(b) F1-score

Figure 1: Model Comparison of classification performance metrics across targets.

Final Model Selection Report

The following models, as shown in table 6 were identified as the best performers for each target and saved for subsequent analysis.

4 Regression Analysis

In this section, we aim to build predictive regression models to estimate audience evaluation scores for seven distinct aspects of robotic choreographies. Specifically, our objective is to train individual regression models for each evaluation metric. Three distinct regression algorithms were implemented: Ridge Regression, XGBoost Regression, and CatBoost Regression.

4.0.1 Data Splitting and Preprocessing

The original dataset contains evaluations for multiple choreography targets, each measured on a 1-to-5 Likert scale. For each evaluation metric, we constructed an individual regression dataset containing: All 19 choreography-related attributes as features, and one evaluation metric at a time, preserving its continuous numeric scale as a target. The dataset was partitioned into: Training Set (80%), Validation Set (10%) and Test Set (10%). This approach provides robust evaluation.

Table 6: Best Performing Models and Hyperparameters per Target

Target Variable	Selected Model	Best Hyperparameters	F1-Score
HumanCharacterization	`XGBoost_GridCV`	`{colsample_bytree: 1.0, learning_rate: 0.01, max_depth: 3, n_estimators: 100, subsample: 0.9}`	0.5437
HumanReproducibility	`CatBoost_GridCV`	`{auto_class_weights: 'None', depth: 3, iterations: 100, l2_leaf_reg: 9, learning_rate: 0.1}`	0.8674
MovementTechnique	`XGBoost_GridCV`	`{colsample_bytree: 0.9, learning_rate: 0.01, max_depth: 3, n_estimators: 100, subsample: 1.0}`	0.5708
PublicInvolvement	`CatBoost_GridCV`	`{auto_class_weights: 'Balanced', depth: 3, iterations: 300, l2_leaf_reg: 9, learning_rate: 0.05}`	0.5310
Rhythm	`XGBoost_GridCV`	`{colsample_bytree: 1.0, learning_rate: 0.05, max_depth: 5, n_estimators: 100, subsample: 1.0}`	0.6315
SpaceUse	`CatBoost_GridCV`	`{auto_class_weights: 'Balanced', depth: 3, iterations: 300, l2_leaf_reg: 1, learning_rate: 0.05}`	0.5134
StoryTelling	`XGBoost_GridCV`	`{colsample_bytree: 0.9, learning_rate: 0.1, max_depth: 3, n_estimators: 200, subsample: 0.9}`	0.5778

4.0.2 Hyper-Parameter Tuning and Evaluation

Hyperparameter tuning was systematically conducted using a 5-fold cross-validation strategy (`GridSearchCV`) to identify the optimal combination Ridge hyperparameters for each regression target. The performance metric used during hyperparameter tuning was `neg_mean_squared_error`, suitable for continuous regression problems. After selecting the best hyperparameters, the final model was evaluated on both the validation and the unseen test sets using three key metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2).

4.0.3 Background Dataset for SHAP Interpretability

In preparation for subsequent SHAP interpretability analyses, a small representative subset of each training set was saved as a "background dataset." This subset aids in interpreting model predictions by providing a reference distribution for SHAP value calculations.

4.1 Regression Model Implementation and Results

4.1.1 Ridge Regression

In this section, we describe the implementation and training procedure of Ridge regression models to predict audience evaluation scores for each choreography aspect. Ridge regression was selected as a baseline linear regression model due to its effectiveness in handling multicollinearity among features by incorporating an L2 regularization penalty.

Pipeline and Preprocessing

We employed a standardized modeling pipeline to ensure consistent preprocessing and efficient hyperparameter tuning. The pipeline comprises two primary steps:

1. **Feature Scaling**:

   All input features were standardized using `StandardScaler`, ensuring zero mean and unit variance, an essential step for Ridge regression due to its sensitivity to feature scales.

2. **Regression Estimator**:

   Ridge regression was chosen for its robustness in handling correlated input features through L2 regularization.

We tuned `alpha` = [0.001, 0.01, 0.1, 1, 10, 100, 1000] and `solver` = `['auto', 'svd', 'cholesky', 'sag']`.
The trained pipeline (`best_pipe`) was saved for future reproducibility and analysis.

Target	**Best Alpha**	**Val. MAE**	**Val. RMSE**	**Val. R^2**	**Test MAE**	**Test RMSE**	**Test R^2**
Storytelling	100.0000	0.9480	1.1190	0.1010	0.9530	1.1300	0.0840
Rhythm	100.0000	0.9610	1.1450	0.0750	0.9770	1.1510	0.0850
Movement Technique	1000.0000	0.8390	1.0340	0.0400	0.7980	1.0060	0.0270
Public Involvement	100.0000	0.9370	1.1170	0.0410	0.9710	1.1460	0.0550
Space Use	1000.0000	0.8390	1.0140	0.0520	0.7950	0.9880	0.0530
Human Characterization	1000.0000	0.9160	1.1120	0.0020	0.8940	1.0880	−0.0020
Human Reproducibility	1000.0000	0.6950	0.9650	0.0200	0.7160	1.0100	−0.0140

Table 7: Performance Metrics for Regression Models

4.2 XGBoost Regression

Given the limited performance of Ridge regression, we pursued XGBoost (Extreme Gradient Boosting), a powerful gradient boosting algorithm designed for complex datasets. XGBoost is highly effective in modeling nonlinear relationships and capturing intricate feature interactions. It offers built-in regularization parameters (`subsample`, `colsample_bytree`) to manage complexity and reduce overfitting. Additionally, it is optimized for speed, enabling extensive hyperparameter tuning. Hyperparameters and their values for model tuning is as rendering below:
`n_estimators` = [100, 250, 500], `learning_rate` = [0.01, 0.05, 0.1], `max_depth` = [3, 6, 9], `subsample` = [0.6, 0.8, 1.0], and `colsample_bytree` = [0.6, 0.8, 1.0].
Optimized XGBoost models were saved in JSON format, along with small background datasets for subsequent interpretability analyses using SHAP.

Target	**Best `n_est.`**	**Best `max_d.`**	**Val. MAE**	**Val. RMSE**	**Val. R^2**	**T. MAE**	**T. RMSE**	**T. R^2**
Storytelling	250.0000	6.0000	0.9140	1.0850	0.1550	0.9050	1.0880	0.1510
Rhythm	250.0000	6.0000	0.8950	1.0730	0.1880	0.9130	1.0960	0.1700
Movement Technique	250.0000	6.0000	0.8240	1.0160	0.0730	0.7860	0.9880	0.0620
Public Involvement	500.0000	3.0000	0.8930	1.0710	0.1180	0.9310	1.0970	0.1340
Space Use	100.0000	3.0000	0.8160	0.9800	0.1150	0.7840	0.9660	0.0960
Human Characterization	250.0000	6.0000	0.9000	1.0880	0.0450	0.8980	1.0820	0.0100
Human Reproducibility	100.0000	6.0000	0.6900	0.9580	0.0360	0.7040	0.9950	0.0150

Table 8: Performance Metrics After Hyperparameter Tuning

4.3 CatBoost Regression

Building upon the promising results from XGBoost, we further employed CatBoost, another powerful gradient-boosting algorithm particularly noted for handling structured datasets. CatBoost offers distinct advantages including automatic handling of categorical features, built-in overfitting detection mechanisms, and superior performance due to its ordered boosting strategy. CatBoost was chosen due to its robustness to overfitting, enhanced efficiency, and ability in automatic handling of data. The selected hyperparameters for tuning included: `iterations` = [500, 1000], `learning_rate` = [0.01, 0.05, 0.1], `depth` = [4, 6, 8], and `l2_leaf_reg` = [1, 3, 5].

The base CatBoost model was configured with `RMSE` and `early stopping`. Optimized CatBoost models (`.cbm` files) and associated small background datasets were systematically stored, facilitating future interpretability analyses using SHAP.

Target	**Best Iter.**	**Best Depth**	**Val. MAE**	**Val. RMSE**	**Val. R^2**	**T. MAE**	**T. RMSE**	**T. R^2**
Storytelling	500.0000	4.0000	0.9070	1.0800	0.1620	0.9070	1.0940	0.1420
Rhythm	500.0000	4.0000	0.8970	1.0830	0.1730	0.9080	1.0930	0.1750
Movement Technique	500.0000	4.0000	0.8250	1.0130	0.0780	0.7890	0.9860	0.0650
Public Involvement	500.0000	4.0000	0.8940	1.0710	0.1180	0.9340	1.0980	0.1310
Space Use	1000.0000	6.0000	0.8200	0.9830	0.1100	0.7870	0.9650	0.0990
Human Characterization	500.0000	4.0000	0.9020	1.0870	0.0470	0.8910	1.0760	0.0210
Human Reproducibility	1000.0000	4.0000	0.6960	0.9560	0.0390	0.7110	0.9940	0.0160

Table 9: Performance Metrics for CatBoost Models Across All Targets

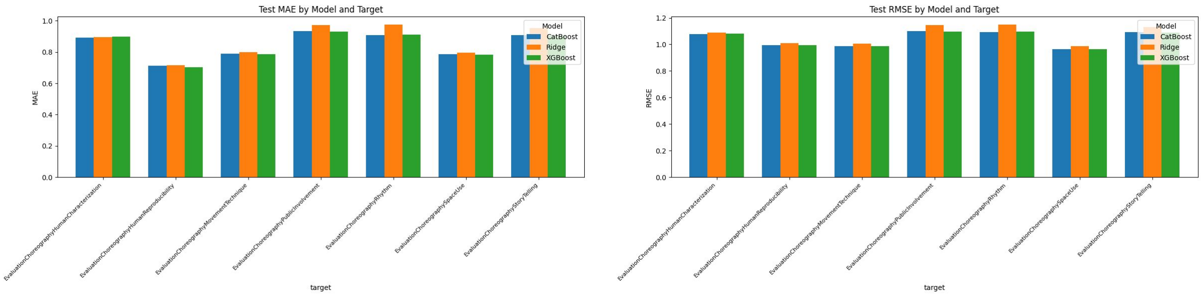
4.4 Model Comparison and Final Selection

Comparing Ridge, XGBoost, and CatBoost regression models, we observed **Ridge Regression** provided a linear baseline with limited predictive power. In the other hand, **XGBoost** and **CatBoost** Substantially improved predictive accuracy, capturing complex feature interactions and nonlinear relationships. Both models exhibited similar performance, with minor differences across evaluation targets. Given their comparable performance and considering ease of interpretation and robustness to overfitting, both XGBoost and CatBoost emerge as strong candidates. For interpretability analyses and practical deployment, we selected models individually per target based on their test-set RMSE/MAE and R^2 performance, leveraging CatBoost's or XGBoost's strengths as appropriate.

In this section, we perform a comprehensive comparison of Ridge, XGBoost, and CatBoost regression models trained on each choreography evaluation target. Our aim is to determine the most appropriate evaluation metric tailored to the characteristics of each target, subsequently selecting the optimal regression model accordingly.

We aggregated the performance of each model across all seven evaluation targets, producing a unified summary table and visualizations to facilitate direct comparisons.

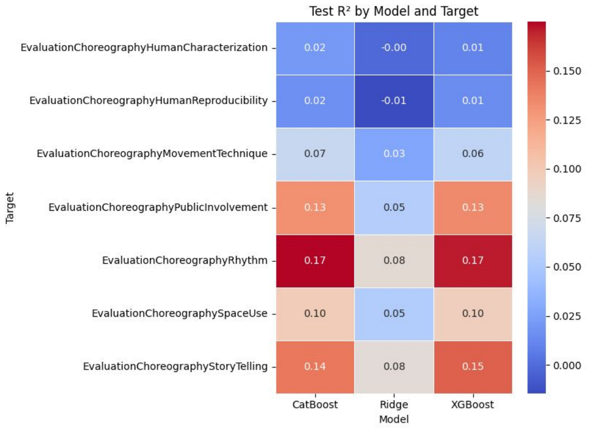
To visually interpret model performance, the following plots were generated:



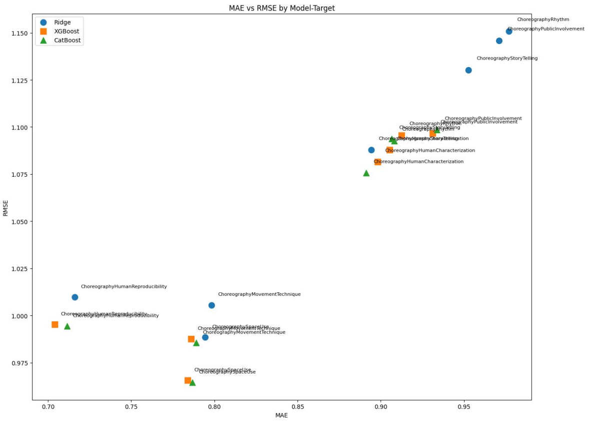
(a) Test MAE by Model and Target

(b) Test RMSE by Model and Target

Figure 2: Comparison of Test Performance Metrics Across Models and Targets.



(a) Heatmap of Test R^2 by Model and Target.



(b) MAE vs. RMSE Scatter Plot

These visualizations provided intuitive guidance in selecting models based on the error distribution characteristics per target.

4.4.1 Selection of Primary Evaluation Metrics

To align model evaluation precisely with the data's underlying nature, we selected either **Mean Absolute Error (MAE)** or **Root Mean Squared Error (RMSE)** as our primary evaluation metric per target. This decision considered each target's inherent variability (standard deviation, σ).

4.4.2 Final Model Selection per Target

Applying the selected primary metrics, we identified the best-performing regression model per target:

Target	Selected Metric	Ridge	XGBoost	CatBoost	Best Model
Storytelling	RMSE	1.130	**1.088**	1.094	**XGBoost**
Rhythm	RMSE	1.151	1.096	**1.093**	**CatBoost**
Public Involvement	RMSE	1.146	**1.097**	1.098	**XGBoost**
Movement Technique	MAE	0.798	**0.786**	0.789	**XGBoost**
Space Use	MAE	0.795	**0.784**	0.787	**XGBoost**
Human Characterization	MAE	0.894	0.898	**0.891**	**CatBoost**
Human Reproducibility	MAE	0.716	**0.704**	0.711	**XGBoost**

Table 10: Final Model Selection Per Target Based on Chosen Evaluation Metric

This table provides the final, clear-cut recommendations for model selection based upon data-informed metrics, leveraging each model's strengths across different evaluation targets.

5 SHAP Analysis

This section details the SHAP analysis conducted to interpret our machine learning models and support our goal of improving human–robot interaction through choreographic evaluation.

5.1 Methodology

Our SHAP analysis is structured into four distinct phases, each building upon the last to ensure a robust, interpretable, and actionable understanding of model behavior.

5.1.1 Phase 1: Establish a Robust SHAP Foundation

Phase 1 produces the background sample, raw SHAP values, and interaction matrices, and stores all metadata for reproducibility.

Outputs:

- **Background dataset** (`background.csv`)
- **Raw SHAP values** (`raw_<target>.npy`) (see Figure 4)
- **Interaction values** (tree models only)
- Configuration file (feature names, file paths, explainer settings, random seeds)

Raw SHAP Values (`raw_<target>.npy`) **What it is:** A NumPy array of shape $(N_{\text{instances}}, N_{\text{features}})$, where each row is a choreography from the hold-out set and each column is a model feature (e.g. `timeDuration`, `nMovements`). Each cell contains the SHAP value quantifying that feature's contribution to the model's prediction for that instance. **Goal:** To quantify the exact contribution of every feature to each individual prediction, forming the raw evidence for all downstream explanations. **How to interpret:**

- **Positive value:** the feature pushed the prediction higher (e.g. raised the "Storytelling" score).
- **Negative value:** the feature pushed the prediction lower.
- **Near zero:** the feature had little to no impact on that specific prediction.

Connection to goal: This raw matrix is the foundation from which all higher-level insights (PDP, dependence plots, decision plots, etc.) are derived.

Having built the foundation, Phase 2 shows how features act in aggregate and individually.

5.1.2 Phase 2: Visualize "How" and "When" Features Matter

Phase 2 generates PDP & ICE and SHAP dependence plots to reveal global and local feature effects.
Outputs:

- PDP & ICE plots (for the top-N features)
- SHAP dependence plots

Partial Dependence & ICE Plots (`plots_pdp_ice_<feature>.png`) **What it is:** A combined visualization where the *partial dependence* curve (thick line) depicts the average effect of a single feature on predictions, and the *individual conditional expectation* curves (thin lines) trace each choreography's response. **Goal:** To quantify how variations in one feature influence the predicted score both on average and for individual instances. **How to interpret:**

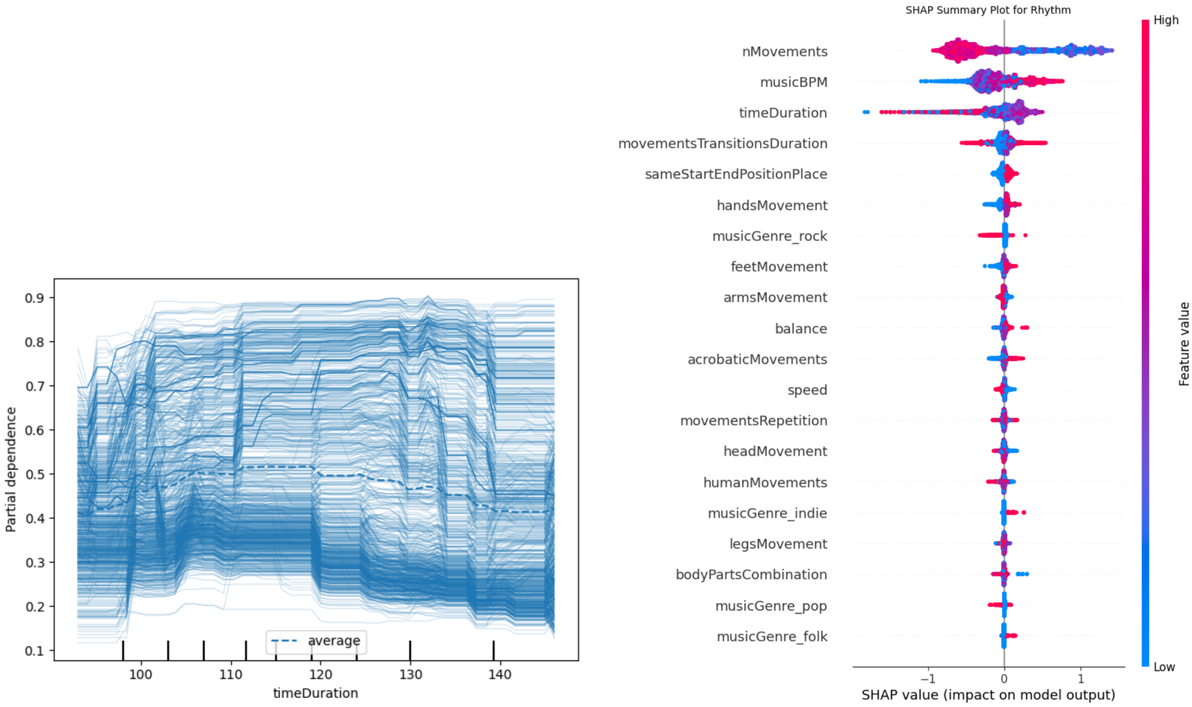
- The PDP curve answers questions like "As `timeDuration` increases, how does the 'Rhythm' score change on average?"
- Parallel ICE curves indicate consistent feature effects across instances; diverging lines reveal interactions.

Connection to goal: These plots provide the foundational "how much" and "under what conditions" insights that guide subsequent analyses. (see Figure 4)

SHAP Dependence Plot (`dep_int_feature>.png`) **What it is:** A scatter plot where the x-axis shows a feature's values, the y-axis its SHAP contributions, and each point is colored by its most strongly interacting feature. **Goal:** To reveal a feature's main effect and how it is modulated by another feature, uncovering interaction dynamics. **How to interpret:**

- Points above zero indicate positive impact; points below zero indicate negative impact.
- Vertical color separation shows that the feature's influence varies with the interacting feature's value.

Connection to goal: Answers "How does the effect of Feature A change depending on Feature B?", providing deeper insight into conditional relationships. (see Figure 5)

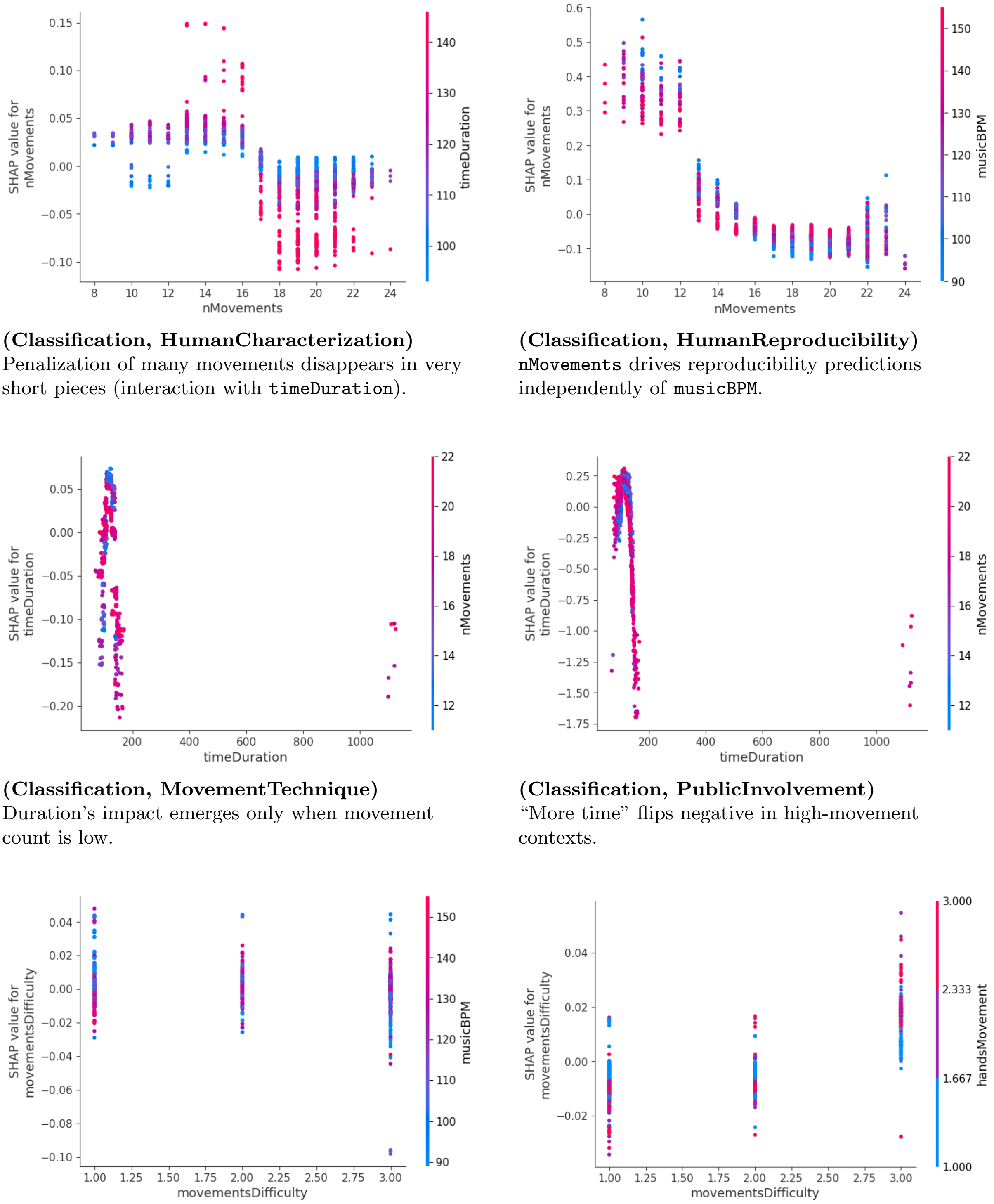


(a) Partial Dependence & ICE for `timeDuration`.

(b) SHAP Summary for `Rhythm`.

Figure 4: Contrasting model insights: average trends (left) and individual contributions (right).

Figure 5: SHAP dependence plots revealing feature interactions.



(Classification, HumanCharacterization)
Penalization of many movements disappears in very short pieces (interaction with `timeDuration`).

(Classification, HumanReproducibility)
`nMovements` drives reproducibility predictions independently of `musicBPM`.

(Classification, MovementTechnique)
Duration's impact emerges only when movement count is low.

(Classification, PublicInvolvement)
"More time" flips negative in high-movement contexts.

(Classification, Rhythm)
Higher difficulty uplifts score and reverses with tempo.

(Regression, SpaceUse)
Movement difficulty flips from harmful to helpful depending on hand activity.

Phase 3 evaluates stability and subgroup-specific effects.

5.1.3 Phase 3: Prove Robustness & Contextualize via Bootstrap & Subgroups

Phase 3 assesses feature stability and subgroup dependencies via bootstrap replicates and subgroup analysis.

Outputs:

- **Feature Stability Matrix** (`bootstrap_stability_features_<target>.csv`)
- **Mean Stability Score** (`mean_bootstrap_rho_<target>.csv`)
- **Subgroup Analysis** (`subgroup_mean_abs_shap_<feature>.csv`)

Feature Stability Matrix (`bootstrap_stability_features_<target>.csv`) **What it is:** A Spearman's correlation matrix quantifying consistency of feature rankings across bootstrap replicates. **Goal:** To verify that top features remain stable under resampling. **How to interpret:** Values near 1.0 indicate reliable rankings; lower values flag instability. **Connection to goal:** Confirms which features are truly influential rather than dataset-specific artifacts.

Mean Stability Score (`mean_bootstrap_rho_<target>.csv`) **What it is:** A CSV listing each feature's average Spearman's ϱ across bootstrap samples. **Goal:** To summarize stability into a single score per feature. **How to interpret:** Scores above 0.8 denote highly consistent importance. **Connection to goal:** Highlights the most dependable predictors for reporting.

Table 11: Mean Bootstrap Spearman's ρ (Feature Stability Scores)

Feature	**Class.Rhythm**	**Class.HumanRep.**	**Reg.HumanRep.**	**Reg.Rhythm**
`timeDuration`	0.9600	0.9900	0.9300	1.0000
`nMovements`	1.0000	1.0000	1.0000	1.0000
`movementsDifficulty`	1.0000	0.9400	0.9100	1.0000
`robotSpeech`	0.9400	1.0000	0.9300	0.9400
`acrobaticMovements`	0.8700	1.0000	0.9300	0.9400
`movementsRepetition`	0.8900	1.0000	0.9400	0.9800
`movementsTransitionsDuration`	1.0000	0.9900	0.9300	1.0000
`humanMovements`	1.0000	0.9600	1.0000	0.9100
`balance`	0.9400	0.9600	0.9300	1.0000
`speed`	0.9500	0.9600	0.9700	0.9600
`bodyPartsCombination`	0.9900	1.0000	0.9200	0.9200
`musicBPM`	0.9600	1.0000	1.0000	1.0000
`sameStartEndPositionPlace`	0.9300	0.9500	0.9500	0.9200
`headMovement`	0.9500	0.9400	0.9500	0.9500
`armsMovement`	0.8900	0.9600	0.9700	0.9300
`handsMovement`	0.9300	1.0000	0.9600	0.9800
`legsMovement`	0.9900	0.9600	1.0000	1.0000
`feetMovement`	1.0000	0.9400	0.9500	0.9800
`musicGenre_electronic`	1.0000	0.9100	1.0000	0.9200
`musicGenre_folk`	0.9400	1.0000	1.0000	0.9800
`musicGenre_indie`	0.9100	0.9300	0.9300	0.9300
`musicGenre_latin`	0.9300	1.0000	1.0000	0.8600
`musicGenre_pop`	0.9200	0.9500	0.9400	1.0000
`musicGenre_rap`	0.9300	1.0000	0.9300	1.0000
`musicGenre_rock`	1.0000	1.0000	0.9300	0.9500

Subgroup Analysis (`subgroup_mean_abs_shap_<feature>.csv`) **What it is:** A table of mean absolute SHAP values computed for each category of predefined grouping variables (e.g. `musicGenre`, `robotType`). **Goal:** To detect context-dependent variations in feature importance. **How to interpret:** Compare subgroup columns to reveal differing model logic. **Connection to goal:** Ensures explanations account for subgroup-specific behavior.

Finally, Phase 4 synthesizes these insights into summaries and narratives.

5.2 SHAP Value Analysis

This section presents aggregated results and visualizations from the SHAP analysis for feature contributions and model behavior.

5.2.1 Phase 4: Synthesize Insights & Drive Decisions

Phase 4 merges all metrics into summary tables, generates decision plots, and produces Report Snippets (report snippets.md) for actionable narratives.

Outputs:

- **Prototypical Decision Plots** (`decision_<target>.png`)
- **Master Summary Table** (`final_summary.csv`)
- **Report Snippets** (`report_snippets.md`)

Prototypical Decision Plots (`decision_<target>.png`) **What it is:** A detailed waterfall plot for a single prediction, showing baseline output and each feature's contribution toward the final score for a chosen choreography. **Goal:** To illustrate the model's decision process with concrete examples. **How to interpret:** Read from the base value upward; arrows indicate how features shift the prediction. **Connection to goal:** Grounds the narrative in real instances, explaining why a choreography scored high or low. (see Figure 6)

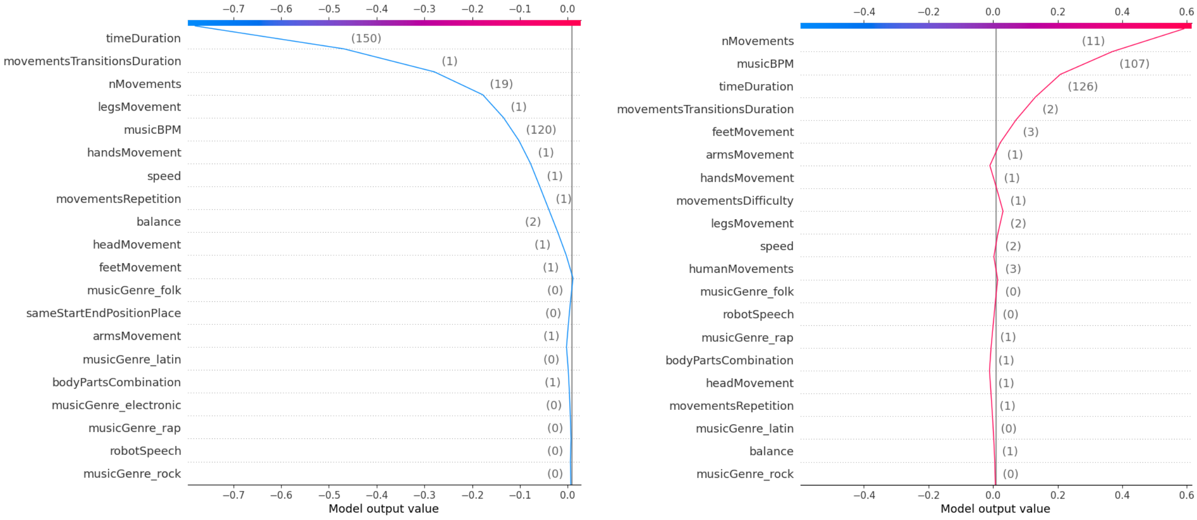


Figure 6: SHAP decision plots for CatBoost on HumanCharacterization: (left) low-score instance, (right) high-score instance.

Master Summary Table (`final_summary.csv`) **What it is:** A CSV aggregating key metrics for each feature—mean SHAP, strongest interaction partner, interaction strength, stability score, and subgroup SHAP means. **Goal:** To provide a comprehensive data source for final reporting. **How to interpret:** Sort or filter by any column to identify influential and robust features. **Connection to goal:** Supplies the quantitative basis for which features drive and stabilize predictions.

Report Snippets (`report_snippets.md`) What it is: A Markdown file auto-generated from the master summary table, translating findings into plain-English bullets (e.g. "The top 3 stable features are..."). **Goal:** To draft the initial narrative without manually parsing tables. **How to interpret:** Read the text directly; it highlights the most important results. **Connection to goal:** Bridges quantitative outputs to a clear narrative for stakeholders. (see Figures 12, 13)

Table 12: Summary of Top Features (Classification) based on SHAP Analysis

Model	**Target**	**Feature 1**	**SHAP**	**Feature 2**	**SHAP**	**Feature 3**	**SHAP**
CatBoost	HumanReproducibility	`nMovements`	0.0989	`musicBPM`	0.0911	`movementsT.D.`	0.0630
CatBoost	PublicInvolvement	`nMovements`	0.4032	`timeDuration`	0.2244	`musicBPM`	0.1890
CatBoost	SpaceUse	`nMovements`	0.5215	`timeDuration`	0.2329	`musicBPM`	0.1935
XGBoost	HumanCharacterization	`timeDuration`	0.0770	`musicBPM`	0.0438	`nMovements`	0.0280
XGBoost	MovementTechnique	`nMovements`	0.3359	`musicBPM`	0.0569	`timeDuration`	0.0456
XGBoost	Rhythm	`nMovements`	0.5917	`musicBPM`	0.2006	`timeDuration`	0.1945
XGBoost	Storytelling	`nMovements`	0.5338	`timeDuration`	0.3190	`musicBPM`	0.1882

Table 13: Summary of Top Features (Regression) based on SHAP Analysis

Model	**Target**	**Feature 1**	**SHAP**	**Feature 2**	**SHAP**	**Feature 3**	**SHAP**
CatBoost	HumanCharacterization	`nMovements`	0.0545	`movementsT.D.`	0.0544	`timeDuration`	0.0531
CatBoost	Rhythm	`nMovements`	0.3386	`musicBPM`	0.0955	`timeDuration`	0.0867
XGBoost	HumanReproducibility	`musicBPM`	0.0511	`nMovements`	0.0283	`timeDuration`	0.0147
XGBoost	MovementTechnique	`nMovements`	0.1991	`musicBPM`	0.0672	`timeDuration`	0.0537
XGBoost	PublicInvolvement	`nMovements`	0.2447	`musicBPM`	0.0931	`timeDuration`	0.0889
XGBoost	SpaceUse	`nMovements`	0.2164	`musicBPM`	0.0905	`timeDuration`	0.0508
XGBoost	Storytelling	`nMovements`	0.2624	`timeDuration`	0.1075	`musicBPM`	0.0746

5.3 Insights from Model Interpretability

Our four-phase SHAP pipeline demonstrates that `nMovements`, `musicBPM`, and `timeDuration` are consistently the most powerful drivers of choreographic scores, individually as well as together. Specifically, our analysis shows that when different features work together—like a fast tempo making the number of movements more important or a longer time duration changing how tempo affects the score—their combined impact is often greater than what each feature contributes on its own. Tables and Report Snippets clearly show these findings, while PDP/ICE and dependence plots illustrate the complex ways in which different feature combinations have the strongest effects. Mean Bootstrap Spearman's and Subgroup analyses show that these core drivers stay consistent across samples and also across features such as music genres, allowing for confident, context-aware explanations of model behavior.

6 Conclusion

Gradient-boosted tree models, notably XGBoost and CatBoost, consistently outperformed linear baselines in predicting audience evaluations, while our SHAP interpretability pipeline identified movement count, duration, and music tempo as the most influential drivers of perceived choreography quality. Interaction and stability analyses confirmed that these findings hold across resampled data and distinct subgroups, and decision plots provided concrete examples of model reasoning. By combining strong predictive performance with detailed, reproducible explanations, this work establishes a robust framework for understanding and enhancing robot choreography design through data-driven insights.

References

[1] Allegra De Filippo, Luca Giuliani, Eleonora Mancini, Andrea Borghesi, Paola Mello, Michela Milano, et al. Towards symbiotic creativity: A methodological approach to compare human and ai robotic dance creations. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 5806–5814. ijcai, 2023.

[2] Allegra De Filippo, Michela Milano, et al. Large language models for human-ai co-creation of robotic dance performances. In *IJCAI*, pages 7627–7635. International Joint Conferences on Artificial Intelligence, 2024.