Industry report

Riccardo Marvasi, Edoardo Saturno, Nicola Palli
Master's Degree in Artificial Intelligence, University of Bologna
{riccardo.marvasi, edoardo.saturno, nicola.palli}@studio.unibo.it

Abstract

In the field of machine learning, the increasing prevalence of intricate models, often deemed as "black boxes", accentuates the growing importance of explainability. As these intricate models play an increasingly pivotal role in decision-making across diverse domains, elucidating the rationale behind their predictions becomes imperative. In this study, we conducted a thorough exploration of various explainability approaches, scrutinizing traditional methods such as LIME, SHAP, and DICE, alongside a novel algorithm. This approach combines ANOVA decomposition and LASSO regularization, applied across a spectrum of machine learning models tailored for a linear regression problem. Our comparative analysis sheds light on the efficacy and interpretability of these methodologies, enhancing our comprehension of model transparency and elucidating the features that influence the decision-making process.

1 Introduction

Artificial intelligence has markedly improved the accuracy of deriving conclusions from complex data. However, algorithms often present a challenge for individuals outside the AI domain, as their operations can be intricate and hard to grasp. This becomes particularly important in practical terms because models driven by observational data can be difficult to correct for biases and other inherent artifacts within the data. In numerous scenarios, there is a necessity for dependable and transparent models, allowing their credibility to be validated against sector-specific expertise. This goes beyond local explanations, such as feature attributions, requiring a broader interpretation. It involves elucidating how the weight of individual input variables influences the model's response consistently across the entire range of inputs.[1]

This paper focuses on implementing, analysing and comparing diverse forms of explainability techniques for a set of machine learning models employed to forecast the score of a Robotic Dance Performance, based on various technical features. The datasets utilized may potentially exhibit noise, and the assessment of dance performance can depend on the independent effects of individual variables or group interactions between them. The models in question employ linear regression to predict evaluations for specific performances, learning from a dataset containing assessments of various robot performances. This paper centers on this specific aspect and benchmarks conventional explainability approaches, such as SHAP, DICE, and LIME, against a custom algorithm founded on the utilization of ANOVA and LASSO.

While rule-based models remain applicable in various scenarios, linear regression is frequently preferred in numerical applications [2]. The rise of powerful AI methods, tho, necessitates the user's ability to comprehend the inner workings of models, emphasizing the importance of model transparency. This transparency refers to a clear and readily understandable flow of information from input to response. In the era of significant AI development, gaining access to methods that enhance the transparency of black-box models becomes crucial. Understanding these complex models is not only beneficial for extracting meaningful insights but also plays a pivotal role in identifying potential problems within the model pipeline. Transparent models facilitate a more effective debugging process, allowing users to pinpoint errors with greater clarity. This could involve the identification of features playing unintended roles or the detection of any anomalies that might affect the model's performance. Overall, the push for transparency aligns with the broader goal of making AI systems more interpretable and user-friendly, enabling users to have a deeper understanding of the decision-making processes within these advanced models. [3-4]

2 System description

Our research on the explainability of machine learning models builds upon an existing script developed for predicting scores of robotic artistic performances based on various input features. The models we employed—Linear Regressor (LR), Decision Tree (DT), Random Forest (RF), and Gradient Boosting Regressor (GBR)—were already implemented, limiting us to their original definitions. Each model incorporates also a mechanism to assign feature importance. For example, decision trees based models in Scickit-Learn employs *Mean Decrease Gini index* to calculate the feature importance values.

Our approach involves using these predefined models as benchmarks to evaluate the performance of other techniques. Specifically, we aim to verify that the features identified as more significative by our explainability techniques align with the 'golden features' identified by each model. In essence, our work ensures that the features highlighted by our explainability methods coincide with the important features identified by the original models. However this is not a shot-out about the correctness of the scikit-learn feature importance attribution mechanism, but rather a way to define some "common ground" between the various approaches in order to be able to identify abnormal behaviour of each specific techniques with respect to the overall trend of the others. In this way we are able to learn where and when to use each technique.

As mentioned earlier, our initial focus in this study was on well-known explainability techniques. Specifically, we developed three separate scripts for each methodology: DICE, LIME, and SHAP. In the following lines, we provide a concise explanation of the functioning of each of these techniques.

SHAP (SHapley Additive exPlanations): is based on cooperative game theory and on the concept of *shapley values*. It calculates the contribution of each feature to the difference between the actual prediction and the expected prediction. SHAP values consider all possible combinations of features and their effects on predictions, providing a holistic understanding of feature importance. More formally, considering a model prediction $f(x)$, where x denotes the input feature vector, SHAP values are employed to elucidate the deviation between the model's prediction and the average prediction across all subsets of features. Mathematically, the model prediction $f(x)$ is defined as the sum of the average prediction ϕ_0 and the contributions of each individual feature $\phi_i(x)$:

$$f(x) = \phi_0 + \sum_{i=1}^{N} \phi_i(x)$$

Where $\phi_i(x)$ is the *shapely value* and represents the contribution of feature i to the variance between the model's prediction and the average prediction capturing the marginal contribution of the feature by evaluating how much the prediction changes when the feature i is added to different subsets of features. This approach allows the understanding of the impact of each feature on the model's predictions. The resulting *shapely* values represent a balanced distribution of the prediction's deviation from the average across individual features[13].

DICE (Diverse Counterfactual Explanations) is a method used in the field of machine learning interpretability. Its main function is to generate counterfactual explanations, which are possible alternative scenarios that would have led to a different outcome than the one obtained by the model. These explanations help users better understand the model's decisions and identify any biases or weaknesses in its operation. Essentially, DICE provides a clearer perspective on how the model makes its decisions, making the automatic decision-making process more transparent and interpretable. In order to evaluate. In other words, it provides "what-if" explanations for model output and can be a useful complement to other explanation methods, both for end-users and model developers. The importance of a feature is assessed by observing how its variation influences the model's predictions on a series of counterfactual instances generated by DICE. This process provides an estimate of the effect each feature has on the model's output, allowing users to better understand the model's functioning and the impact of individual features on its decisions. In particular, the importance of each feature, in our case, was evaluated by considering the set of counterfactuals that led to a change in the output, thus calculating the absolute (AV) and relative (RV) variation of each feature that resulted in a final change, as follows:

$$AV = \sum_{i=1}^{N} |\delta x_i| * |\delta y| \tag{1}$$

$$RV = AV/N \tag{2}$$

where δx_i represents the feature variation, δy the output variation and N is the number of feature variations[14].

LIME (Local Interpretable Model-agnostic Explanations) approximates the behavior of a model in the proximity of a specific prediction by constructing an interpretable model based on local perturbations of the input data. Formally, let x be the input instance for which interpretation is sought, $f(x)$ be the prediction made by the black-box model, and g be the interpretable model approximating f locally. LIME aims to minimize the following loss function:

$$\underset{g \in G}{\operatorname{argmin}}\ L(f, g, \pi_x) + \Omega(g)$$

where, G is the set of interpretable models, π_x is a sampling distribution centered around x used to generate perturbed instances, $L(f, g, \pi_x)$ is a loss function measuring the dissimilarity between the predictions of f and g on perturbed instances, and $\Omega(g)$ is a regularization term. LIME employs a sampling technique to generate N perturbed instances, $\{x'_i\}_{i=1}^{N}$, around x. The interpretable model g is then trained on these perturbed instances with corresponding labels $f(x'_i)$. The resulting interpretable model elucidates the local decision boundary around x and provides insights into the factors influencing the prediction at that specific point[15].

Subsequently, we introduced a custom algorithm, inspired by the article *Towards interpretable machine learning for clinical decision support* [5]. This algorithm adopts a "remove and retrain" strategy, systematically isolating each feature and using it as input to train the model. The process is then repeated to assess pairs of input features, specifically evaluating their grouped effect on the dependent variable. The entire process first makes predictions using the fitted model and then uses them in correlation with true labels in order to compute p-values for each group of features and capture their statistical relevance. P-values are obtained by comparing the variability in the data that is explained by the model's predictions with the remaining unexplained variability, which is captured by the residuals (the differences between the true and predicted labels). This comparison helps to assess the extent to which the model's predictions significantly differ from what would be expected by random chance alone, providing insight into the statistical relevance of the features being analyzed[8]. The rationale behind this "double" approach lies in the desire of, first of all, understanding the impact of individual features on the model's performance. In fact, by isolating each feature during training, the algorithm gauges its standalone influence. Secondly, the assessment of feature pairs provides insights into potential interactions or dependencies that contribute collectively to the model's predictive capability. This analysis provides the understanding of feature importance and inter-feature relationships for each model. Tri-wise and higher order interactions are not taken into account due to the combinatorial explosion of possible groups as well as the reduced quantity of information they contains.

Upon acquiring p-values for each individual feature and feature pair, our methodology engages in hypothesis testing to confirm their statistical significance. The null hypothesis (H_0) assumes that there is no significant relationship between the feature (or feature pair) and the dependent variable, while the alternative hypothesis (H_1) suggests the presence of a significant relationship [9]. Afterwards, a crucial step involves comparing the calculated p-values with a predetermined threshold α, which provides *Significance Level* value. In line with the principles of hypothesis testing, we considered a p-value falling below α as a sufficient evidence to reject the null hypothesis. This implies that the feature (or feature pair) is considered statistically significant and retains information contributing to the model's predictive performance. On the other hand, features with p-values exceeding α have insufficient evidence to reject the null hypothesis, leading to the conclusion they are not statistically significant[6].

Moreover, our approach includes also a recall mechanism crafted to grasp the relational importance of features. This mechanism functions on the principle that, if numerous pairs of relevant features groups share a specific common feature, then its impact on the dependent variable is deemed significant, even if originally dismissed by the initial threshold mechanism. This approach aligns with the theoretical framework of feature interdependence and synergistic contributions within a dataset. In machine learning and statistical analysis, the concept of feature interaction acknowledges that the impact of one feature on the dependent variable may be contingent on the presence or absence of other features. By recognizing features that co-

occur frequently with important features, our recall mechanism aims to identify additional dimensions of relevance[10].

Following the initial feature selection based on p-values, a secondary feature selection process is introduced, leveraging the LASSO (Least Absolute Shrinkage and Selection Operator) method on the refined dataset. The rationale behind this approach is deeply rooted in regularization theory and variable selection principles. LASSO introduces a penalty term, incorporating the sum of the absolute values of the coefficients into the standard regression loss function. Mathematically, the objective function for LASSO is expressed as trying to minimize the term L:

$$L = \text{Loss} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Where Loss represents the standard regression loss, λ is the regularization parameter controlling the strength of the penalty, and β_j denotes the coefficients associated with each feature. The key characteristic of LASSO lies in its ability to shrink certain coefficients to exactly zero. This facilitates automatic feature selection, as features with zero coefficients are considered insignificant and are consequently excluded from the model[11]. Therefore, the application of LASSO is a valuable tool for refining the feature set, retaining only those features that contribute significantly to the predictive power of the model. In our methodology, LASSO is employed on the p-filtered dataset, and the resulting set of shrunk coefficients provides a clear indication of feature significance. Features with non-zero coefficients survive the regularization process, signifying their importance, while features with coefficients reduced to zero are considered negligible and are consequently excluded. This two-step feature selection process, integrating statistical significance and LASSO regularization, has the intent of striking a balance between interpretability and predictive accuracy in the final model [7].

In the final step of our algorithm, we aggregate comprehensive information by constructing a dictionary that encapsulates the status of each feature. This dictionary serves to categorize features based on their relevance during distinct stages of our interpretability framework. Features are classified into three key categories within the dictionary:

1. **P-Irrelevant (P-value Irrelevant):** Features discarded during the initial feature selection process based on p-values. This categorization aligns with the statistical significance assessment.

2. **L-Irrelevant (LASSO Irrelevant):** Features eliminated during the secondary feature selection through LASSO application.

3. **Non-zero Coefficient Value (LASSO):** Features retaining non-zero coefficient values calculated by LASSO. These features are considered important and contribute meaningfully to the model, as determined by the regularization process.

This interpretability pipeline facilitates transparency and traceability, helping users in comprehending the rationale behind feature selection decisions. The compiled dictionary is then stored within the same folder, alongside other pertinent outputs, such as the original feature importance coefficients, tables, images generated by interpretability techniques (DICE, LIME, SHAP). This all-inclusive documentation contributes to the reproducibility and interpretability of the entire model interpretation process.

3 Data

We utilized two distinct datasets, both obtained from the University of Bologna, focusing specifically on statistical information related to a robotic dance performance and its reception by the audience. Notably, the divergence between the two versions of the dataset stems from evaluations conducted by audiences with contrasting backgrounds—one with an artistic orientation and the other with a scientific background. These datasets comprise a diverse set of features, encompassing categorical aspects such as the genre of the music, and numerical values like the beats per minute (bpm) of the music. Additionally, the datasets include the seven target variables, reflecting evaluations of rhythm, storytelling, public involvement, and other parameters. All the models are employed to generate predictions for each of the seven target variables across both datasets.

Our preprocessing strategy involves transforming the initial input features to obtain 27 distinct features. These features encompass the original numerical values, standardized by removing the mean and scaling to unit variance, alongside

discrete categorical levels derived from the categorical values. In particular, the categorical input feature "AI_technique" is divided in three boolean variables AI_planning, AI_searchstrategy and AI_constraint (the three possible AI techniques) and the musicGenre feature is divided in mPop, mRock, mElectric, mFolk, mIndie.

A level refers to a distinct value or category that a discrete variable can take. This process involves the creation of two dedicated pipelines, each tailored to handle numerical and categorical features separately. For numerical features, our pipeline executes a dual-phase transformation. Firstly it employs a median imputation strategy addressing missing values within numerical features by replacing them with the corresponding median values, in order to ensure robust handling of missing data. Following imputation, the pipeline proceeds to scale the numerical features using an appropriate scaler. On the categorical side the process starts with the separation of distinct values for each categorical feature. Subsequently, one-hot encoding is applied to each isolated categorical value.

With the application of this pre-processing step, our final dataset exhibits 27 variables, including the original 18 numerical features, complemented by an additional 9 that encapsulate the diverse values derived from the categorical ones.

4 Experimental setup and results

In this section, we will begin by detailing the experimental setup that underpins our results. Subsequently, we will present the outcomes coming from an exhaustive exploration, employing the three classical explainability methodologies, together with the insights gained from the ANOVA-based algorithm. This examination seeks to offer a lucid comprehension of the factors influencing our results and the interpretability of the employed methodologies. Additionally, it serves as a means to assess the reliability and trustworthiness of the employed methodologies.

Our evaluation process diverges from traditional train/test assessments of model performance. Rather than focusing on conventional metrics, our emphasis is on assessing how effectively the models assign importance values to relevant features. In order to reason on the validity of the obtained results, as mentioned before, to establish benchmarks, we defined "golden standards" based on the importance coefficients directly assigned by the scikit-learn models (in figure 1 we can see an example of which features are relevant for the linear regression model). These benchmarks serve as a reference point for comparison against the alternative methodologies implemented. The golden standard for each model are obtained in the following ways:

1. **Decision Trees and Random Forests** : The default method for assessing feature importance in decision trees and random forests in scikit-learn is the "mean decrease in impurity" (MDI). It measures the total decrease in node impurity weighted by the probability of reaching that node.

2. **Gradient Boosting Regressor (GBR)**: The gradient boosting regressor model in scikit-learn uses the default method of "gain-based importance" for assessing feature importance. It measures the improvement in the loss function brought by a feature across all splits it is used in.

3. **Linear Regression**: The linear regression model assesses feature importance basing on the magnitude of the coefficients. The absolute value of the coefficients indicates the importance of each feature in the linear regression model, with higher magnitude coefficients suggesting higher importance.

The coefficients derived from the mentioned methods serve as a benchmark for comparison with values obtained through various implemented strategies. Specifically, for SHAP, DICE, and LIME, we leverage their built-in methods designed to elucidate feature importance. In contrast, for our custom model, we employ a distinct approach. We first obtain feature importance coefficients using LASSO, and subsequently, with the predefined dictionary, we provide a comparative analysis against the established golden standard.

For the sake of simplicity, in this report we will only focus on the explainability results obtained on the *Linear Regressor* model, applied on the *Scientific Background* dataset, utilizing as target variable *EvaluationPublicInvolvement*, however this can be repeated for any model and for any target feature. The complete list of results is attached on the results folder [citazione].

In Figures 2 and 3 two of the SHAP results are presented. SHAP is particularly well suited for

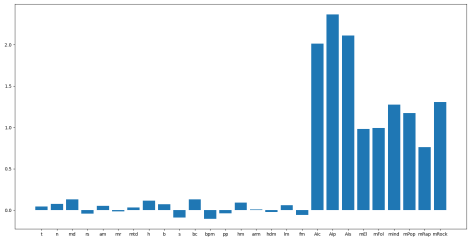

Figure 1: Golden features for the linear regression model (LR).

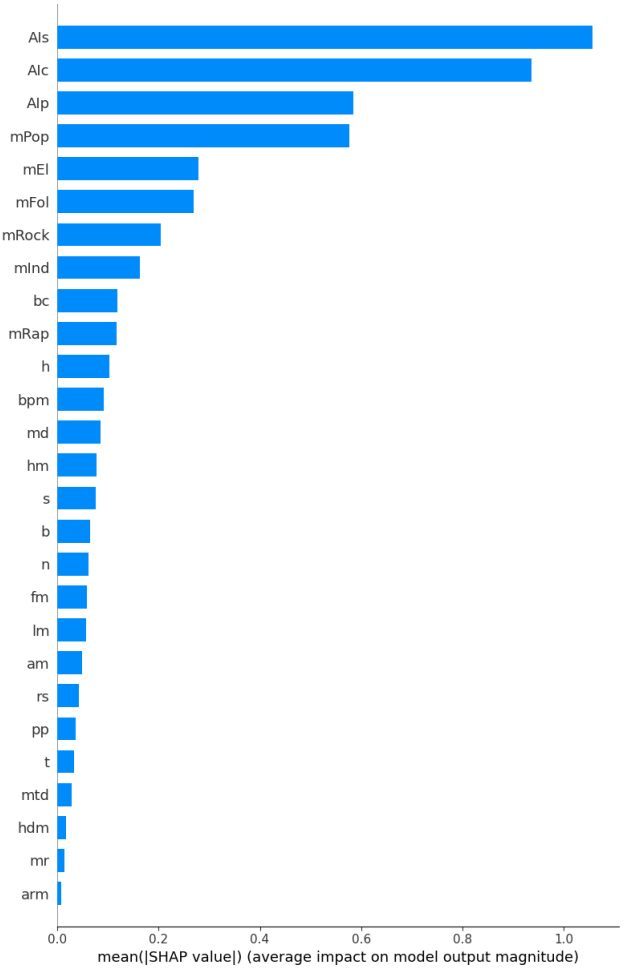

Figure 2: SHAP values for the linear regression model

global analysis of examples because it applies an analysis on all the input data.

In Figure 2, the mean absolute impact of each feature across all examples is portrayed, offering a comprehensive perspective on the average contribution of each feature to the model's predictions across the entire dataset. This representation allows us to discern the overall significance of individual features, providing valuable insights into the consistent patterns of influence exhibited by different input variables. The sorting of features based on their absolute magnitude of impact facilitates a clear identification of the most influential factors driving the model's outputs. Figure 3 provides a deeper exploration through the distribution of each feature's influence on the dependent variable. This visualization captures the diversity in the impact of individual features across the dataset, revealing how their contributions vary. The sorting mechanism based on absolute magnitude not only highlights the most impactful features but also facilitates a dynamic understanding of their collective effect on the model's predictions.

As already mentioned SHAP is a powerful tool to obtain a comprehensive and lucid overview of the model's global behavior. It excels at highlighting features that, on average, exert the highest impact on the target variable, providing also insightful details on how each variable contributes to the model's predictions.

LIME and DICE, on the other hand, are designed to provide local, instance-specific explanations for model predictions, so they are more suited for the understanding of the decision-making process for a single data point rather than the overall behavior of the model. For this reason, our analysis for what concerns DICE and LIME focuses on the results provided for the single instances.

In Figures 4 and 5, two singular LIME explanations are presented, offering insights into model predictions by highlighting the top 5 most important features. Alongside this, LIME provides a threshold or range for each of these features. This threshold indicates whether a feature's value, when surpassing or falling within the specified limits, has a positive or negative impact on the predicted output.

As mentioned earlier, feature importance with DICE is measured here through the absolute and relative variation of the feature that leads to a change in the output. Specifically, the absolute variation allows us to observe the frequency with which the feature has been chosen, while the relative variation deals with the magnitude of the change: the lower it is, the more relevant the feature might be within the model. The DICE results are collected in Table 1.

The results of the ANOVA-based algorithm are reported in Table 2. The table provides a comparison between the golden values visualized in Figure

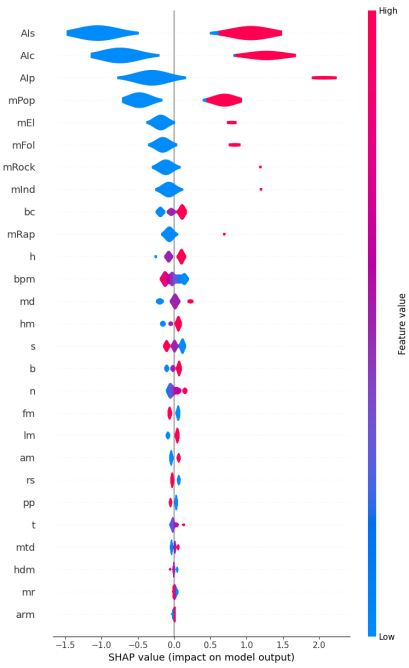

Figure 3: Distribution of feature impact

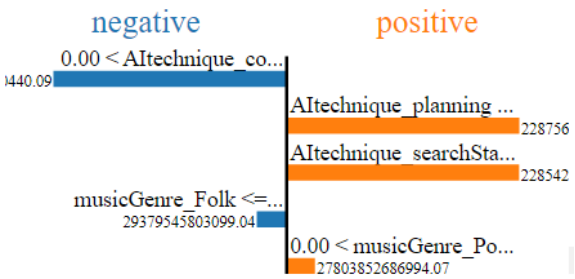

Figure 4: Lime explanation for an instance

1 and the responses obtained from the algorithm for each feature. Specifically, the terms "P-irrelevant" and "L-irrelevant" indicate that a particular feature has been deemed not relevant by the ANOVA or LASSO selection, respectively. For features identified as relevant by both ANOVA and LASSO, the numeric value represents the LASSO coefficient: if the value is different from 0 the feature is deemed important. The table is a way to provide a complete overview of the algorithm's assessments for each feature with respect to the golden features. Different parameter values have been tested for both ANOVA and Lasso:

- **alpha** represents the threshold used to assess feature importance through ANOVA decomposition. A feature with a p-value lower than alpha is deemed relevant. The conventional value for alpha is 0.10, but adjusting this parameter could impact the stringency of feature selection based on importance.

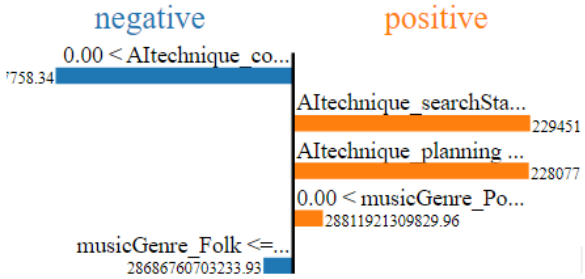

Figure 5: Enter Caption

Feature	Relative var	Absolute var
t	2.141	404.820
n	1.671	357.708
md	1.386	275.990
rs	1.401	299.949
am	1.414	288.614
mr	1.804	330.278
mtd	1.386	242.709
h	1.964	412.562
b	1.403	249.823
s	1.419	288.149
bc	1.396	301.554
bpm	1.660	395.249
pp	1.329	293.768
hm	1.617	292.722
arm	1.820	356.828
hdm	1.733	417.676
lm	1.480	288.686
fm	1.439	295.199
AIc	2.916	70.000
AIp	2.200	33.000
AIs	2.161	67.000
mEl	1.520	73.000
mFol	1.615	84.000
mInd	1.767	99.000
mPop	1.870	116.000
mRap	2.022	91.000
mRock	1.589	89.000

Table 1: DICE Counterfactual relative and absolute variations for each feature.

- **lambda** is a regularization parameter that controls the strength of regularization applied to the model. It plays a crucial role in balancing between minimizing the sum of squared residuals and the penalty term, which is the sum of the absolute values of the coefficients multiplied by lambda. It's conventionally set

on 0.01, but increasing it increases the amount of regularization, leading to more shrinkage of coefficient values and potentially sparse solutions.

- It is possible to choose the maximum number **r** of dependent variables to be used in ANOVA calculation. As illustrated in the paper, this parameter is commonly set to 2. Higher values help capture the importance of correlations among multiple features, increasing the Lasso's effort to reduce feature coefficients.

Feature	Golden	ANOVA r=1	ANOVA r=2
t	0,044	P-irrelevant	L-irrelevant
n	0,074	P-irrelevant	L-irrelevant
md	0,129	P-irrelevant	0.042
rs	-0,045	0.052	0.025
am	0,050	P-irrelevant	0.034
mr	-0,016	P-irrelevant	L-irrelevant
mtd	0,033	L-irrelevant	L-irrelevant
h	0,113	P-irrelevant	0.018
b	0,071	P-irrelevant	0.023
s	-0,898	P-irrelevant	-0.033
bc	0,130	0.015	0.016
bpm	-0,104	P-irrelevant	-0.05
pp	-0,038	-0.07	-0.06
hm	0,089	0.06	0.09
arm	0,010	0.065	0.06
hdm	-0,024	-0.07	-0.07
lm	0,060	0.031	0.01
fm	-0,059	P-irrelevant	0.08
AIc	2,013	0,035	L-irrelevant
AIp	2,369	-0.30	-0.291
AIs	2,113	P-irrelevant	L-irrelevant
mEl	0,982	0,029	L-irrelevant
mFol	0,994	-0,023	0.211
mInd	1,276	0,107	L-irrelevant
mPop	1,173	0,172	L-irrelevant
mRap	0,761	-0,361	L-irrelevant
mRock	1,309	-0,07	-0.034

Table 2: Feature importance results provided by the ANOVA-based algorithm for the linear regression model

5 Discussion

In this section we will provide a discussion on the results presented in section 4.

First of all, regarding SHAP explanations (figure 2 and 3), it is appreciable how the most influencing features are aligned with the golden standard values represented in figure 1. Other than being similar to the golden features, the explanations also provide much more insights on the input features. For instance, the absence of a typically well-received genre of music in favor of another, can detrimentally affect audience reception, illustrating that preferences may differ across audiences; for example, a scientific audience may favor pop music over rock music. To be more specific in figure 3 we can see a huge negative impact for the majority of samples in the *Rock Music* feature. It is obvious that either this means that Rock genre absence is badly received from audience and this negatively affects the output score or that its mere presence is enough to lower that same score. The discrimination between these two scenarios can be concluded by looking and the distribution of Rock Music labels for all examples. Even the LIME explanation instances (figure 4 and 5) confirm that the features which, when perturbed, result in the most significant changes in the model's outputs align with those highlighted by the golden values. This alignment suggests that LIME's local interpretable model successfully captures the influence of important features, providing an reasonable representation of the decision-making process for that particular instance. In particular, in the two instances reported, LIME states that the features with the highest impact on the final prediction are AI_constraint, AI_planning and AI_searchstrategy. LIME, more on detail, provides intuitions on whether there is a positive or negative correlation: for examples, in both instances, values of AI_constraint falling below the value of 0 have a big negative impact on the output. In this specific example, those three features are possible values of the categorical variable "AI_technique", so the meaning of the threshold is relative, but it provides anyway an useful suggestion on the fact that the AI_technique variable (and, in particular, the "constraint" technique) are decisive in the final output prediction of the linear regression model.

Moving on the analysis of the DICE results (1), it is appreciable that the input features that show the lowest absolute variation are again AI planning, AI searchstrategy and AI technique. As a remind, the absolute variation represents the change in the model's prediction when the corresponding feature is perturbed or altered, so when this variation is low, it suggests that small changes in the input feature result in minimal fluctuations in the model's output. In the case of high correlation, the model is

particularly sensitive to variations in the specified feature, indicating that the feature has a substantial impact on the predictions. Therefore, a low absolute variation signifies that the model's output is relatively stable and consistent with small changes in the input feature, highlighting a strong correlation between the two. So, in this case, the fact that AI planning, AI searchstrategy and AI constraint are the input features with the lowest absolute variation suggest that the AI technique is the most highly-correlated features, in total accordance with the LIME and SHAP explanations and with the golden values.

Lastly, analysing the ANOVA-based algorithm results (table 2), some discrepancies with the other techniques emerged. For example, by looking at the results of table 2, it can be shown how an universally-considered highly-correlated feature like *AI-planning* is considered irrelevant by the algorithm. This trend is repeated all over the various models. More examples of this behaviour, associated with SHAP explanation, can be found in the following [folder].

As already pointed, the explanations provided by SHAP, DICE and LIME are coherent with each other and do not show major discrepancies with the golden feature importance values. Instead, it is evident that, despite an overall tendency to approximate the golden standard values, there are some important discrepancies in the ANOVA-based algorithm results: either the p-value feature selection or LASSO-shrinking rendered irrelevant some important features.

Reasons behind this behaviour, which is repeated all over the trials, can be grasped pointing out the properties that are necessary for a reliable application of each one of these methodologies. Regarding ANOVA, the assumptions that must stand are[16]:

1. **Independence:** Observations should be independent of each other, ensuring that observed variability accurately reflects true differences between them.

2. **Normality:** Residuals, which represents the distribution of the differences between the observed values (true labels) and the predicted values, should be approximately normally distributed within the dataset, ensuring the validity of inferential statistics used in ANOVA.

3. **Homogeneity of Variance (*homoscedasticity*):** refers to the assumption that the variance of the residuals is constant across all levels of the independent variable(s). This will ensure the validity of assumptions underlying the F-test and, subsequently, the P-test.

By trying to demonstrate the validity of the above properties, it emerged that *homoscedasticity* and *normality distributions of residuals* do not hold. This was concluded by performing a normality test on the array of residuals obtained by the difference between the true labels and the model-predicted labels. Furthermore, we performed a *Levene* test to check the variance between each group of residuals and if they were homogeneous[12].

A further analysis also highlighted that the assumption that the Total Sum of Squares (TSS) equals the sum of Residual Sum of Squares (RSS) and Regression sum of squares (SSR) was violated too in some models, further highlighting that the algorithm is not capturing all the relevant information or relationships in the data, impacting the reliability of the results.

There are several consequences on the model results. First of all, the non-validity of the two properties violates the assumptions behind the inferential statistics used by ANOVA, leading to inaccurate p-values. This, of course, brings to possible misinterpretations of relationships between feature. If residuals exhibit increasing variance within levels of the independent variable, it might wrongly suggest that the variability in the dependent variable is primarily due to the independent variable, when in reality, it is due to *heteroscedasticity* (with this term we refer to the absence of *homoscedasticity*) itself.

While ANOVA is generally considered to be robust to violation of assumption number 2 when sample sizes are large, severe *heteroscedasticity* can still affect the validity of results. Due to these issues, ANOVA filtering and LASSO feature selection are unreliable, as it can be seen looking at table 2.

This situation does not affect the SHAP, DICE and LIME explanations due to the fact that the requirements of these techniques are less stringent compared to those of ANOVA. They typically assume that the relationships between input features and predictions are locally consistent, which is the case in our experiment. In fact, small changes in the input data around a particular feature, produce a noticeable change in the model's predictions. More than that, while not strictly necessary, these

methods often work better when features are relatively independent or have limited interactions, which is the case of our dataset, since each feature refers to an individual parameter of the robot performance and correlations between features are limited. More on detail LIME and DICE only require a sufficiently dense distribution of data points around the instance of interest to accurately approximate the local behavior of the model, since they rely on perturbing input data to generate explanations.

6 Conclusion

In conclusion, our findings suggest that the custom methodology relying on ANOVA is not well-suited for addressing the challenges posed by the structure of the dataset in this particular case. To overcome this limitation, it becomes imperative to explore alternative methods that can better accommodate the dataset's characteristics without necessitating extensive preprocessing and normalization efforts, which may not be universally applicable to all features.

Subsequently, we have identified DICE, LIME, and SHAP as promising alternatives, as they exhibit greater adaptability to datasets with specific constraints compared to the custom ANOVA-based approach. Notably, the distinct computational approaches employed by DICE, LIME, and SHAP allow them to handle issues such as the absence of homoscedasticity and non-normally distributed residuals more effectively.

Conversely, the custom methodology's performance is notably hampered by these properties, primarily due to its reliance on different computational mechanisms. Therefore, in scenarios where datasets exhibit deviations from homoscedasticity and normal distribution of residuals, opting for DICE, LIME, or SHAP would likely yield more robust and reliable results compared to the ANOVA-based approach

In order to enhance the applicability of ANOVA, various techniques can be explored. Firstly, Data Transformation can be employed whenever feasible. Techniques such as log transformation, square root transformation, or Box-Cox transformation can be applied to the data. These transformations aim to stabilize variance and improve the normality of residuals, thereby bolstering the robustness of ANOVA assumptions.

Secondly, Feature Engineering can be investigated. This involves exploring the possibility of creating new features or modifying existing ones to better align with ANOVA assumptions. For instance, generating interaction terms between variables can capture relationships that individual features may not adequately represent.

Thirdly, Outlier Detection and Handling is imperative. Identifying and managing outliers in the dataset is crucial as they can significantly impact ANOVA results. Outliers can be addressed through removal, transformation, or the utilization of robust statistical methods less influenced by their presence.

Lastly, Data Normalization should be considered if applicable. Normalizing the data ensures that variables are on similar scales, mitigating issues related to heteroscedasticity and enhancing the performance of ANOVA.

References

[1] *C. Rudin*, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (2019) 206–215.

[2] *E. Christodoulou, J. Ma, G.S. Collins, E.W. Steyerberg, J.Y. Verbakel, B.Van Calster*, A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models

[3] https://www.algolia.com/blog/ai/what-is-explainable-ai-and-why-is-transparency-so-important-for-machine-learning-solutions/

[4] https://www.forbes.com/sites/forbestechcouncil/2023/01/13/e
ai-the-importance-of-adding-interpretability-into-machine-learning/?sh=39f75afea9e4

[5] Towards interpretable machine learning for clinical decision support, *Bradley Walters*, *Sandra Ortega-Martorell*, *Paulo J. G. Lisboa*, *Ivan Olier*

[6] https://link.springer.com/chapter/10.1007/978-3-030-90639-9$_3$0

[7] https://www.blog.trainindata.com/lasso-feature-selection-with-python/

[8] https://www.analyticsvidhya.com/blog/2021/06/feature-selection-using-statistical-tests/

[9] https://courses.lumenlearning.com/introstats1/chapter/null-and-alternative-hypotheses/

[10] https://www.researchgate.net/publication/350971982$_B ig_d at$

[11] https://medium.com/@agrawalsam1997/feature-selection-using-lasso-regression-10f49c973f08

[12] https://datatab.net/tutorial/levene-test

[13] "A Guide for Making Black Box Models Explainable", chapter 9.6, *Christoph Molnar*

[14] https://interpret.ml/DiCE/

[15] "A Guide for Making Black Box Models Explainable", chapter 9.2, *Christoph Molnar*

[16] https://online.stat.psu.edu/stat500/book/export/html/607: :text=There%20are%20three%20primary%20assumptions