COMMON PITFALLS IN EVALUATING MODEL PERFORMANCE AND STRATEGIES FOR AVOIDANCE IN AGRICULTURAL STUDIES

A PREPRINT

C. P. James Chen*
School of Animal Sciences
Virginia Tech
Blacksburg, VA 24061
niche@vt.edu

Robin R. White
School of Animal Sciences
Virginia Tech
Blacksburg, VA 24061
rrwhite@vt.edu

January 2, 2025

ABSTRACT

This study critically examines the methodologies and metrics used for evaluating prediction models in regression and classification tasks, making a case for the application of rigorous and standardized approaches in model performance assessment. Within the context of this work, we define modeling as a structured framework for hypothesis formulation and decision-making, which relies on the analysis and extrapolation of empirical data. The advancement of modeling is contingent on the accumulation of prior knowledge within the scientific community. The study conducted a series of simulations to delve into common pitfalls in cross-validation (CV), a technique crucial for characterizing expected model performance on "new" data. Issues such as using the same data for both training and assessment, excluding model selection from CV, and overlooking experimental block effects were explored through simulation examples. Moreover, the simulations in this study highlight that no single model performance metric suffices to represent model performance adequately and conservatively, emphasizing the need for understanding the underlying theory of each metric to avoid misleading conclusions. In conclusion, this simulation study aims to guide researchers in accurately and consistently reporting model performance, thereby supporting integrity and scientific rigor in prediction modeling research.

Keywords Model Evaluation · Performance Metrics · Simulation Studies

*Corresponding author: James Chen <niche@vt.edu>

1 Introduction

1.1 Review on ultrasound imaging technology

1.2 Modeling

Modeling is an essential tool for hypothesis formulation and decision-making. It functions as a structured investigatory framework that allows researchers to explore system understanding through the summary and analysis of empirical data. Carefully constructed and evaluated models offer the potential to extend this understanding by enabling the extrapolation of results to novel trials and conditions. Although only one focus of the science of modeling, the predictive role is often explicitly or implicitly the ultimate goal of models derived within the precision agriculture context. Through this lens, modeling provides opportunity to standardize and formalize research advancement, through developing quantitative constructs that accumulate prior knowledge derived by the broader the scientific community. Evaluating model performance becomes particularly critical when considering this role within the knowledge generation enterprise, necessitating a rigorous and standardized approach that allows for both reproducibility and comparability. As more and more model-based exercises are developed using slightly different methods, or slightly different datasets, it becomes increasingly challenging to evaluate, characterize, compare, and balance information generated by the resulting modeling tools, particularly when results are conflicting. Specifically, reporting model performance through poorly-defined metrics or incomplete procedures can create opportunity for confusion, misinterpretation, and miscommunication, and can ultimately result in distrust in model-based tools and impede scientific progress.

This study examines two primary challenges that arise during model evaluation: those associated with the evaluation methodology and those stemming from the data structure. The former emphasizes the reliability of estimated performance and essential measures to avoid overestimating a model's capabilities. The latter depends on the nature of the modeling exercise: for regression tasks, variance and bias are particularly important for assessing performance, whereas for classification tasks, class imbalance poses a critical concern. Employing multiple performance metrics can help prevent misinterpretation due to these factors. To illustrate the significance of these challenges and effective strategies to address them, we conduct a series of simulations complemented by real-world data examples.

1.3 Model Evaluation

Model evaluation in the context of predictive analytics seeks to explore how well a model can generalize to new prediction contexts not seen during model training. Although commonly referred to as "model validation" in the literature, this term implies a false degree of confidence given that the word “validation” means to prove something true. There is no single test, or recognized suite of tests, to prove a model valid. Instead, the term "evaluation," which involves assessing the value, nature, character, or quality of something, is more fitting. It is essential to evaluate model performance on unseen data to ensure the approach is applicable to new experiments. To this end, cross-validation (CV) is widely recognized as a standard method for model evaluation.

The most common CV method is K-fold CV, which partitions the dataset into K equally sized folds. In each iteration, one fold is reserved as the test set (i.e., new data, noted as $\mathcal{D}_{\text{test}}$), while the remaining folds are used as the training set (noted as $\mathcal{D}_{\text{train}}$) to construct the model. Once the model is trained, it is evaluated on the $\mathcal{D}_{\text{test}}$ to obtain an estimate of the model performance $\hat{g}$. The process will iterate K times until each fold has been used as the $\mathcal{D}_{\text{test}}$ once. The average performance over all K folds is deemed as the expected generalization performance of the model $\mathbb{E}[\hat{g}]$ on new data.

However, there is always an evaluation bias between the estimated performance $\mathbb{E}[\hat{g}]$ and the true generalization performance G, which can only be approximated by evaluating the same model on an infinite number of unseen data. Depending on the performance metric used in evaluation, a positive evaluation bias ($\mathbb{E}[\hat{g}] - G$) typically suggests that the model evaluation procedure concludes a pessimistic estimation of the model performance, since the true performance is expected to be lower than the estimated performance. Another aspect of model evaluation error is the variance of each estimated performance $\hat{g}$ across the K folds. For example, there are five estimates in a 5-fold cross-validation. The variance among these five estimates is defined as the evaluation variance. A high evaluation variance suggests that the performance is sensitive to the choice of data folds, and a small size or an over-complex model can lead to a high evaluation variance.

There is a trade-off relationship between the evaluation bias and variance from a squared evaluation bias, the derivation of the relatinoship is shown in the Eq. 24 in the Appendix. When performing K-fold CV with a fixed sample size and model complexity, the choice of K is the pivotal element shaping the model evaluation. When the K is set to a larger value; each training set $\mathcal{D}_{\text{train}}$ is larger in size, resulting in a model trained on a more representative subset of the population of interest, leading to lower bias. However, a large K comes with a trade-off: the corresponding test subset $\mathcal{D}_{\text{test}}$ is compressed in size, making the tested model more sensitive to the specific data points, and thus inflating the validation variance. Conversely, a smaller K, along with a minor training set $\mathcal{D}_{\text{train}}$, reduces their representativeness and increases bias. Nevertheless, a larger size of the test set $\mathcal{D}_{\text{test}}$ leads to more consistent estimations across the folds and, consequently, reduces the validation variance.

Leave-one-out cross-validation (LOOCV) is a variant of K-fold CV where K equals the sample size of the complete dataset $\mathcal{D}$. It provides an unbiased estimation of model performance because the training set $\mathcal{D}_{\text{train}}$ closely resembles the unseen population of interest, given its size of $N - 1$, where N is the sample size. However, as the trade-off discussion suggested, this method can lead to high validation variance due to the model being evaluated on one sample at a time. The true unbiased nature of LOOCV is fully realized only when all K folds are utilized. Performing an incomplete LOOCV can introduce significant bias because of the inherent high validation variance, which often occurs when training each model iteration is prohibitively time-consuming or computationally demanding. In specific contexts, such as genomic prediction, strategies like the one described by Cheng et al. leverage the matrix inverse lemma, which allows for computational savings by avoiding the inversion of large matrices in each fold. This technique significantly reduces the dependency of computational resources on the sample size [1]. Van Dixhoorn et al. exemplify the use of LOOCV with a small dataset, aiming to predict cow resilience with limited data resources [2]. Nevertheless, for large

datasets, LOOCV is generally not recommended due to computational inefficiency. Further details of bias-variance trade-off have been extensively explored in the statistical literature [3, 4].

1.4 Model Selection

Model selection becomes necessary when models are not entirely determined by the data alone. For example, in a regularized linear regression model such as a ridge regression [5] or the least absolute shrinkage and selection operator (LASSO) [6], it is essential to define a regularization parameter, λ, before fitting the model to the data. A larger λ value yields a more regularized model, which tends to reduce smaller coefficients to negligible values or zero. This approach helps in preventing overfitting noise in the training data. The definition of loss functions for the regularized models were described in 26 and 27 of the Appendix.

These pre-defined parameters, like λ, influence model fitting and remain constant during the training process. Such parameters are referred to as hyperparameters. Beyond regularized models, hyperparameters are crucial in other predictive models, enhancing flexibility and robustness. For example, in the Support Vector Regression (SVR) [7], the regressors X are projected onto a linear subspace to approximate the target variable Y. By choosing a suitable kernel function, which transforms the regressors into a non-linear space, as a hyperparameter, SVR can more effectively capture non-linear relationships, thus significantly improving model performance. Another hyperparameter example is the number of latent variables in the Partial Least Square (PLS) Regression [8], which condenses the original regressors into a more manageable set of latent variables, reducing multicollinearity issues. Fewer latent variables might lose significant information from the original regressors, while too many can lead to overfitting. Similarly, in Random Forest [9], hyperparameters such as tree depth and the number of trees influence model complexity by dictating how many feature splits are possible and how many weak learners comprise the ensemble. The same principle applies to convolutional neural networks, where increasing the number of hidden layers or filter sizes can capture more complex patterns in the data but also heightens the risk of overfitting [10]. All these examples highlight the fact that selecting the most suitable hyperparameters, which is known as hyperparameter tuning, is crucial for optimizing model performance. Feature selection is another crucial aspect of model selection. This process involves fitting the model to a selected subset of the original features, particularly essential in high-dimensional data scenarios where the number of features exceeds the number of observations, leading to poor model generalization. For instance, Ghaffari et al. sought to predict health traits in 38 multiparous Holstein cows using a metabolite profiling strategy. Out of 170 metabolites, only 12 were identified as effective discriminators between healthy and over-conditioned cows and were thus selected for the predictive model [11]. Therefore, optimizing feature subsets is a vital model selection strategy that significantly affects model performance. Including the model selection process within the cross-validation is essential to avoid common pitfalls. The risk of inflated model performance arises when model selection is guided by results on the test dataset. Even if the chosen model is subjected to k-fold cross-validation afterward, its selection bias toward the test set can lead to overestimating its efficacy. This issue has been highlighted in statistical literature [3]. A practical solution is to divide the dataset into training, validation, and test sets. The validation set is then used for model selection, ensuring the

test set remains completely unused during the training phase, thereby providing a more accurate measure of model performance. For instance, the study by Rovere et al. exemplifies best practices in hyperparameter tuning and feature selection by employing an independent cross-validation step prior to assessing model performance. This approach enabled the precise selection of relevant spectral bands from the mid-infrared spectrum and the optimal number of latent dimensions in PLS with Bayesian regression for predicting the fatty acid profile in milk [12]. Similarly, Becker et al. demonstrated a robust evaluation by using nested cross-validation loops; the inner loop conducted a grid search for the best hyperparameters in logistic regression, while the outer loop was designed to evaluate the performance of the resulting optimized model [13]. Both examples underscore the importance of separating model selection from performance evaluation to ensure the validity and reliability of the results.

1.5 Cross Validation Design with Block Effects

Blocking is an essential approach in experimental design to control for variations that can confound the variable of interest. For instance, Lahart et al. investigated the dry matter intake of grazing cows using mid-infrared (MIR) spectroscopy technology across multiple herds under varying experimental conditions [14]. Given the significant variation between herds, which may contribute to individual differences in both dry matter intake (i.e., response variable) and MIR spectra (i.e., independent variables), it is crucial to consider the herd as a blocking factor before evaluating the predictability of dry matter intake using MIR spectra. This consideration should also extend to model evaluation. In the cited study, variations in dry matter intake, the primary focus of the prediction model, were observed to exceed one standard deviation among some herds. In cross-validation, if samples from the same herd are assigned to different folds, with one fold used as the test set, the model is likely to achieve high accuracy. This accuracy may largely result from explaining the inter-herd variation rather than individual variations in dry matter intake, leading to an overestimation of model performance. To avoid this pitfall, block cross-validation, where each block (i.e., herd in this example) is used as a fold, is recommended for unbiased model evaluation. Literature reviews have indicated that block cross-validation effectively evaluates model performance on external or unseen datasets [15]. In the same study by Lahart et al., three cross-validation strategies were compared: random cross-validation (Random CV), which randomly assigns samples to folds; within-herd validation, training and testing the model within each herd; and across-herd validation (Block CV), where each herd is used as a fold and tested in turn. The results showed that performance estimates in block CV were noticeably lower than the other two strategies, supporting the hypothesis that ignoring block effects inflates model performance. Other studies considering block effects, including diet [16], herd [12], and farm location [17, 18], have shown similar results in cross-validation, demonstrating block CV's effectiveness in evaluating model performance on external datasets.

1.6 Model Performance Metrics

Model performance metrics serve as quantitative indicators for evaluating model performance. They are critical for benchmarking various modeling approaches and for evaluating hypotheses underpinning these different approaches.

Choosing appropriate metrics to support hypothesis testing is crucial, as in-ideal selection may lead to overly optimistic conclusions. Due to the different goals of regression and classification tasks, it is critical to ensure that these different model types are evaluated using different metrics. As such, metrics for regression and classification are discussed individually.

1.6.1 Metrics in Regression Tasks

Table 1: Summary of model performance metrics for regression tasks.

Metric	Type	Scale-invariant	Range
Root mean square error (RMSE)	Error-based	No	[0, ∞]
Mean absolute error (MAE)	Error-based	No	[0, ∞]
Root mean squared percentage error (RMSPE)	Error-based	Yes	[0, ∞]
Root mean standard deviation ratio (RSR)	Error-based	Yes	[0, ∞]
Pearson's correlation coefficient (r)	Linearity-based	Yes	[-1, 1]
Coefficient of determination (R^2)	Linearity-based	Yes	[-∞, 1]
Lin's concordance correlation coefficient (CCC)	Linearity-based	Yes	[-1, 1]

Regression models aim to predict continuous variables and are commonly employed in diverse applications, such as estimating body condition scores [19, 20], body weight [21, 22], milk composition [12, 18, 23, 24], efficiency of feed resource usage [16, 25, 26], and early-lactation behavior [2]. The metrics in regression tasks evaluate the agreement between the predicted value $\hat{y}$ and the true values y. The agreement can be generally quantified in two ways: error-based metrics and linearity-based metrics. The metrics are summarized in Table 1.

Error-based metrics focus on the deviation of each pair of predicted and true values, while linearity-based metrics consider overall linear relationships between the predictions and the truths. The root mean square error (RMSE) and the mean absolute error (MAE) are two common error-based metrics:

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

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

where y_i and $\hat{y}_i$ are the true and predicted values, respectively, and n is the sample size. Both metrics preserve the scale of the original data, making them easy to interpret in real-world units. Additionally, compared to MAE, RMSE penalizes large errors more due to the squared term, making it more sensitive to outliers. In the cow production, monitoring animal body weight is a common practice to aid in the management of dairy cows. Studies by Song et al. and Xavier et al. have utilized RMSE to assess the effectiveness of three-dimensional cameras in estimating dairy cow body weight, yielding RMSE values of 41.2 kg and 12.1 kg, respectively [21, 22]. These figures provide a straightforward value for farmers to gauge whether the prediction error is tolerable, considering their specific operational costs and management thresholds. In essence, RMSE translates complex model accuracy into practical insights for productive agricultural

units. When evaluating the same model across different traits, which may have different scales, a common practice is to express error metrics in a scale-free manner. This can be achieved by expressing RMSE as a percent of the deviation from the observed value, such as root mean squared percentage error (RMSPE), or as a Root Mean Standard Deviation Ratio (RSR) that normalizes the RMSE by the standard deviation of the observed values:

$$\text{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2} \tag{3}$$

$$\text{RSR} = \frac{\text{RMSE}}{\sigma_y} \tag{4}$$

where σ_y is the standard deviation of the observed values. When expressed as a percent, RMSPE typically ranges from 0 and above, with values closer to 0 indicating perfect prediction. Much like expressing RMSE as a percent, RSR is valuable to interpret RMSE in terms of the context of the variance in the observations. Values below 1 suggest that the model yields predictions less variable than the standard deviation, while values above 1 suggest that the prediction is imprecise.

On the other hand, Pearson's correlation coefficients (r) and the coefficient of determination (R^2) are two common linearity-based metrics:

$$\begin{aligned} r &= \frac{\text{cov}(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}} \\ &= \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \end{aligned} \tag{5}$$

$$\begin{aligned} R^2 &= 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}} \\ &= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \end{aligned} \tag{6}$$

where SS_{residual} is the residual sum of squares and SS_{total} is the total sum of squares. Each y_i and $\hat{y}_i$ are the ith elements of the actual response vector y and the predicted response vector $\hat{y}$, respectively. $\bar{y}$ and $\bar{\hat{y}}$ are their respective means. Both r^2 and R^2 are scale invariant, meaning their values are unaffected by the scale of the observed data because they are normalized by the variation in the denominator.

The correlation coefficient r measures the strength of the linear relationship between two continuous variables, y and $\hat{y}$, and ranges from -1 to 1. A value of 0 indicates no prediction accuracy in the evaluated model. One special characteristic of correlation r is that it is unaffected by the scale of the predictions or biases; it focuses on the relative changes in the predicted values compared to the true values. Thus, even if the prediction biases are scaled up or down, the correlation r between $\hat{y}$ and y remains the same. This property is particularly useful when the focus is more on ranking predictions rather than their absolute values. For example, this metric has been used to evaluate models that identify

high-performing production individuals, demonstrating the ability to predict nutrient digestibility in dairy cows [26] and to select models based on their ability to rank traits such as feed intake and milk composition in dairy cows [27, 12].

The coefficient of determination R^2 quantifies model performance from the proportion of variance in the dependent variable that is predictable from the independent variables. It ranges from negative infinity to 1, where 1 indicates that the model explains all the variance in the dependent variable, and 0 indicates that the model performs no better than predicting all samples as the mean of the observed values. R^2 is useful in comparing multiple regression models, as demonstrated in studies that regress body weight of dairy cows on a set of morphological traits [22], examine the relationship between milk spectral profiles and nitrogen utilization efficiency [16], and evaluate the predictive performance of milk fatty acid composition [23].

It worth noting that many literatures have misinterpreted the relationship between r and R^2. The coefficient of determination R^2 is not always equivalent to the square of the correlation coefficient r^2. The equivalence only holds when the same dataset is used for both model fitting and evaluation in a least squares regression model. The model assumes a zero covariance between the fitted residual and the predicted values $\hat{y}$, and it also assumes that the residuals (i.e., prediction biases) are centered on zero. In practice when predictions are made on new data, those assumptions are often violated, leading to discrepancies between r^2 and R^2. A details derivation of the equivalence is provided in Equation 28 29 in the Appendix.

In addition to r^2 and R^2, another linearity-based metric is Lin's concordance correlation coefficient (CCC) [28]:

$$\begin{aligned} \text{CCC} &= \frac{2r\sigma_y\sigma_{\hat{y}}}{\sigma_y^2 + \sigma_{\hat{y}}^2 + (\bar{y} - \bar{\hat{y}})^2} \\ &= \frac{2\text{cov}(y, \hat{y})}{\sigma_y^2 + \sigma_{\hat{y}}^2 + (\bar{y} - \bar{\hat{y}})^2} \end{aligned} \tag{7}$$

where r is the Pearson correlation coefficient. The CCC is a comprehensive metric because it considers both the correlation and the scale bias between the predicted and true values. It fills the gap left by r^2 where the scale bias is ignored. Geometrically, CCC measures how well the predicted values $\hat{y}$ fall on the 45-degree line in a scatter plot of the predicted (x-axis) and true values (y-axis). It is advantageous over R^2 because it consistently ranges from -1 to 1, making it easier to interpret and compare across different studies. The CCC is crucial when precise predictions are required for both the scale and the rank of the trait of interest, such as in studies predicting cotton crop yields based on soil and terrain profiles [29].

1.6.2 Metrics in Classification Tasks

Classification models aim to predict categorical outcomes such as 'healthy' or 'sick,' 'susceptible' or 'resistant,' and 'high yield' or 'low yield.' To evaluate classification performance, one must first establish a confidence threshold to dichotomize the prediction probabilities. For instance, if a classification model predict a sample as 'sick' with a 0.7 probability, and the threshold is set at 0.5. Since the 0.7 prediction probability exceeds the threshold, the sample is

predicted as a positive sample. It is worth mentioning that this threshold is adjustable to fine-tune model performance for particular focus, such as minimizing false positives or false negatives. All classification metrics are derived from the confusion matrix, which summarizes the model's performance in a 2x2 table, where the rows represent the actual classes and the columns represent the predicted classes.

Table 2: Confusion matrix for binary classification.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The confusion matrix (Table 2) consists of four components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Most common metrics used in classification tasks are summarized in Table 3.

Table 3: Summary of model performance metrics for classification tasks.

Metric	Denominator	Focus
True positive rate (TPR)	Actual positives	Correctness
True negative rate (TNR)	Actual negatives	Correctness
False negative rate (FNR)	Actual positives	Error
False positive rate (FPR)	Actual negatives	Error
Sensitivity	Actual positives	Correctness
Specificity	Actual negatives	Correctness
Precision	Predicted positives	Correctness
Recall	Actual positives	Correctness
Accuracy	All samples	Balance
F1 score	All samples	Balance
F-beta score	All samples	Balance
MCC	All samples	Balance

The metrics can be characterized by two key factors: their denominator and their focus on either correctness or error. Understanding the denominator of a metric helps clarify its scope of interest. For instance, if one wants to evaluate how well the model correctly predicts positive samples, metrics that use actual positives as the denominator should be prioritized. It is noted that in Table 3, the metrics are organized in four subsections. The metrics in the first subsection have self-explanatory names, each emphasizing a specific aspect of the model's performance:

$$\begin{aligned} \text{True positive rate (TPR)} &= \text{Sensitivity} \\ &= \text{Recall} \\ &= \frac{\text{TP}}{\text{Total Actual Positives}} \end{aligned} \tag{8}$$

$$\begin{aligned}\text{True negative rate (TNR)} &= \text{Specificity} \\ &= \frac{\text{TN}}{\text{Total Actual Negatives}}\end{aligned} \tag{9}$$

Both TPR and TNR focus on the correctness of the model's predictions, but TPR is concerned with positive samples, while TNR is concerned with negative samples. High TPR is essential where missing a positive case has serious consequences, or where false positives are easily rectifiable. For instance, detecting lameness or abnormal gait is crucial, as these can indicate underlying pathologies [30] and impact welfare-related transport decisions [31]. An automated detection system [30, 32, 33] with high TPR can mitigate economic losses by flagging at-risk cows. The benefit here lies in the feasibility of re-examining false positives, thus preventing more severe outcomes from undetected cases.

In contrast, the false negative rate (FNR) and false positive rate (FPR) focus on the model's errors:

$$\text{False negative rate (FNR)} = \frac{\text{FN}}{\text{Total Actual Positives}}$$

$$\text{False positive rate (FPR)} = \frac{\text{FP}}{\text{Total Actual Negatives}}$$

The second section of Table 3 includes sensitivity and specificity, which are equivalent to TPR and TNR, respectively. These terms are widely used in medical diagnostics due to their emphasis on accurately identifying true positive and true negative cases, which are critical requirement for tests and screenings for disease detection.

The third section includes precision and recall, which focus on different aspects of positive cases. Precision evaluates the correctness of the predicted positive cases, ensuring that the predictions are accurate, while recall measures the completeness of identifying all actual positive cases, emphasizing the model's ability to capture true positives. Precision measure the trustworthiness of positive predictions made by the model (Eq. 10). High precision is crucial in scenarios where false positives incur significant costs. For instance, in contexts where clinical treatments and culling are expensive, such as detecting bovine tuberculosis [34] or mastitis [35] using non-invasive methods, a high-precision model is crucial to minimize unnecessary costs and interventions from false positives. Precision and recall are a pair of metrics commonly used in machine learning applications, particularly in multi-class classification or detection scenarios. In these contexts, the evaluation of negative samples (i.e., non-positive samples) is often replaced by examining the precision and recall for each individual class. This approach allows for a more granular assessment of the model's performance across all classes, ensuring that both the quality of predictions and the ability to identify all relevant samples are accounted for.

$$\text{Precision} = \frac{\text{TP}}{\text{Total Predicted Positives}} \tag{10}$$

The last section of Table 3 includes accuracy, F1 score, F-beta score, and Matthews Correlation Coefficient (MCC). These metrics offer a balanced evaluation of the model's performance by taking into account both correctness and error rates, as well as both positive and negative samples. Among them, accuracy is the most straightforward metric for evaluating classification models, as it measures the proportion of correctly classified samples out of the total samples.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Total Correct Predictions}}{\text{Total Predictions}} \\ &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \end{aligned} \tag{11}$$

It summarizes an overall model performance by calculating the proportion of correctly classified samples among all samples. Nonetheless, accuracy can be misleading when the classes are imbalanced. For example, if a study predicting the presence of a specific event, of which the prevalence was only 10%. In this case, a model that predicts all samples as negative would achieve an accuracy of 90%, which is misleadingly high. The F1 score, which is the harmonic mean of precision and recall (i.e., TPR), provides a balanced measure of model performance:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

Unlike accuracy, the F1 score considers both false positives and false negatives by balancing precision and recall, making it a more reliable metric for imbalanced datasets. A variant of the F1 score is the F-beta score, which allows for the adjustment of the balance between precision and recall by introducing a weight parameter β:

$$\text{F-beta} = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{13}$$

A common variant is the F2 score, which places more emphasis on recall, which is false negatives than false positives, by setting $\beta = 2$:

$$\text{F2} = 5 \times \frac{\text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}} \tag{14}$$

Lastly, the Matthews correlation coefficient (MCC) considers both positive and negative samples in the dataset, providing a balanced measure of a model's performance [36]. It is defined as:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{15}$$

The equation 15 symmetrically incorporates all four components of TP, TN, FP, and FN). This symmetry makes MCC invariant to class distribution changes. The coefficient ranges from -1 to 1, where 1 indicates perfect classification, 0 indicates no better performance than random guessing, and -1 signifies total disagreement between prediction and

observation. In a case study that used feeding and daily activity behaviors to diagnose Bovine Respiratory Disease in dairy calves, MCC proved particularly insightful [37]. The models in this study exhibited strong performance on negative samples (i.e., healthy calves), which were more prevalent, resulting in high specificity. However, sensitivity was relatively low at 0.54. In this context, MCC, with a value of 0.36, provided a more nuanced and representative measure of model performance, especially given the skew towards negative samples

1.7 Study Objectives

This simulation study aims to highlight how biased or over-optimistic estimations of model performance usually come from inappropriately conducting CV, a technique crucial for characterizing expected model performance on "new" data. We demonstrate how common pitfalls, including using the exact data for both training and model assessment, excluding the model selection process from CV, and neglecting experimental block effects, contribute to challenges in model evaluation. Further, we scrutinize common metrics used in evaluating prediction models, including those used for regression and classification tasks. Because no single metric provides a comprehensive perspective of model performance, we seek, through this work, to highlight the importance of understanding the underlying theory of each metric to avoid misleading conclusions.

There are five simulation studies being conducted to address these challenges. The first simulation study will focus on the bias-variance trade-off in CV, demonstrating how the choice of K in K-fold CV affects the evaluation bias and variance. The second simulation study will investigate the impact of mistakenly using the same data for model selection and evaluation, highlighting the inflated model performance. The third simulation study will explore the effect of excluding block effects in CV, demonstrating how ignoring block effects can lead to over-optimistic model performance. The fourth simulation study will explore how various metrics respond to different combinations of bias and variance in prediction errors, illustrating how these variations can lead to distinct interpretations of model performance. The fifth simulation study will examine the impact of imbalanced data on classification model evaluation, highlighting how the choice of metrics can influence conclusions and potentially lead to misleading interpretations. Together, this series of simulation studies aims to provide guidance for researchers on accurately and consistently reporting model performance, thereby promoting integrity and scientific rigor in prediction modeling research.

2 Materials and Methods

2.1 Study 1: Evaluation bias and variance of cross-validation

This study investigated the interplay between sample size and various performance estimators and their collective impact on bias and variance during model evaluation. It is hypothesized that increasing the sample size will reduce both bias and variance. Additionally, it is expected that the validation variance will increase with the number of folds in the CV, while simultaneously reducing bias. Since K-fold CV employs a fraction (i.e., $K-1$ folds) of the data for training, it may provide a pessimistic estimate of model performance. Hence, this study designed to assess the underestimation from each performance estimators, including K-fold CV with K set to 2, 5, and 10, as well as LOOCV where K equals the sample size N, and the "In-Sample" evaluation, which assesses model performance on the same dataset used for training, potentially leading to an overly optimistic bias. To gauge model performance, three metrics are employed: RMSE (Eq. 1), r (Eq. 5), and R^2 (Eq. 6). The validation model is a multivariate linear regression with ten input features and one output target, all drawn from a standard normal distribution $\mathcal{N}(0,1)$, implying no expected linear relationship between inputs and the target, with an expected correlation r of zero. The sample sizes N are varied among 50, 100, and 500 to explore the dynamics between sample size and performance estimators. Each configuration is repeated across 1000 iterations to assess the distribution of bias and variance.

For each iteration, the dataset $\mathcal{D}=(X,Y)$ was sampled as per the simulation's premise. In the case of K-fold CV, the dataset $\mathcal{D}$ was partitioned into K folds in which each fold is $\mathcal{D}_k=(X_k,Y_k)$. For the "In-Sample" approach, partitioning does not occur. The linear model f is trained on the training set $\mathcal{D}_{\text{-k}}$ (denoted as $f_{\mathcal{D}_{\text{-k}}}$) to estimate regression coefficients β, which then predicts the target variable $\hat{Y}_k$ from the test set $\mathcal{D}_k$. The procedure of K-fold CV can be expressed as:

$$\begin{aligned} \text{Training:} \quad Y_{\text{-k}} &= f_{\mathcal{D}_{\text{-k}}}(X_{\text{-k}}) + \epsilon \\ &= X_{\text{-k}}\beta + \epsilon \\ \text{Testing:} \quad \hat{Y}_k &= f_{\mathcal{D}_{\text{-k}}}(X_k) \\ &= X_k\beta \qquad k=1,2,\ldots,K \end{aligned} \tag{16}$$

For the "In-Sample" performance estimator, predictions were made without splitting, as:

$$\begin{aligned} \text{Training:} \quad Y &= f_{\mathcal{D}}(X) \\ &= X\beta + \epsilon \\ \text{Testing:} \quad \hat{Y} &= f_{\mathcal{D}}(X) \\ &= X\beta \end{aligned} \tag{17}$$

Where:

- X denotes the input regressors sampled from a standard normal distribution $\mathcal{N}(0,1)$ with dimensions $N \times 10$.
- Y denotes the target variable sampled from a standard normal distribution $\mathcal{N}(0,1)$ with dimensions $N \times 1$.
- $X_{\text{-k}}$ and $Y_{\text{-k}}$ are the input regressors and target variable in the training set $\mathcal{D}_{\text{-k}}$.
- X_k denotes the input regressors in the test set $\mathcal{D}_k$.
- $\hat{Y}_k$ denotes the predicted target variable in the test set $\mathcal{D}_k$.
- β denotes the estimated regression coefficient with dimensions 10×1.
- ϵ denotes the error term assumed to be normally distributed.

Estimated performance $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$ was derived by averaging the performance metrics across all K folds as per Eq. 21. The bias and variance of the evaluation were calculated using Eqs. 22 and 23, respectively. To approximate true model performance $G(f_{\mathcal{D}})$, a hundred unseen datasets $\mathcal{D}^*$ were generated identically to $\mathcal{D}$, and the performance $G(f_{\mathcal{D}})$ was estimated by averaging the performance metrics across all $\mathcal{D}^*$. The detailed steps to compute evaluation bias and variance are provided in the supplementary materials.

2.2 Study 2: Model Selection in Cross-Validation

The objective of this simulation study is to examine the effect of improper model selection implementation on validation bias. The focus will be on the model selection procedures of feature selection and hyperparameter tuning. The study hypothesizes that utilizing the test set inappropriately during any model selection stage will lead to a significant overestimation of model performance. This study simulated a regression task using an SVR model, which utilized various kernel functions to project a subset of features, X, to predict a target variable, Y. Both X and Y are drawn from a normal distribution $\mathcal{N}(0,1)$ to establish a baseline null correlation (performance r=0) for assessing validation bias. This study set the sample size and number of features at 100 and 1000, respectively. Feature selection is executed by choosing the top 50 features that correlate most strongly with Y. For hyperparameter tuning, four kernel functions were evaluated: linear, polynomial, radial basis function, and sigmoid.

This study introduces notations FS for feature selection and HT for hyperparameter tuning, assigning a binary indicator (0 or 1) to denote incorrect (0) or correct (1) implementation of model selection. This yields four possible combinations of model selection strategies: "FS=0; HT=0", "FS=0; HT=1", "FS=1; HT=0", "FS=1; HT=1" (Figure 1). When FS=0, feature selection precedes cross-validation splitting. If FS=1, feature selection occurs within each fold of the training set during cross-validation. With hyperparameter tuning, a correct implementation (HT=1) involves splitting the dataset into training (64%), validation (16%), and test (20%) sets. The model is trained and tuned using the training and validation sets, respectively, while the test set is reserved for a single evaluation of model performance. Conversely, with HT=0, only training (80%) and test (20%) sets are used, risking validation bias as the test set informs both training and performance reporting. A 5-fold cross-validation approach was deployed for all strategies. Validation

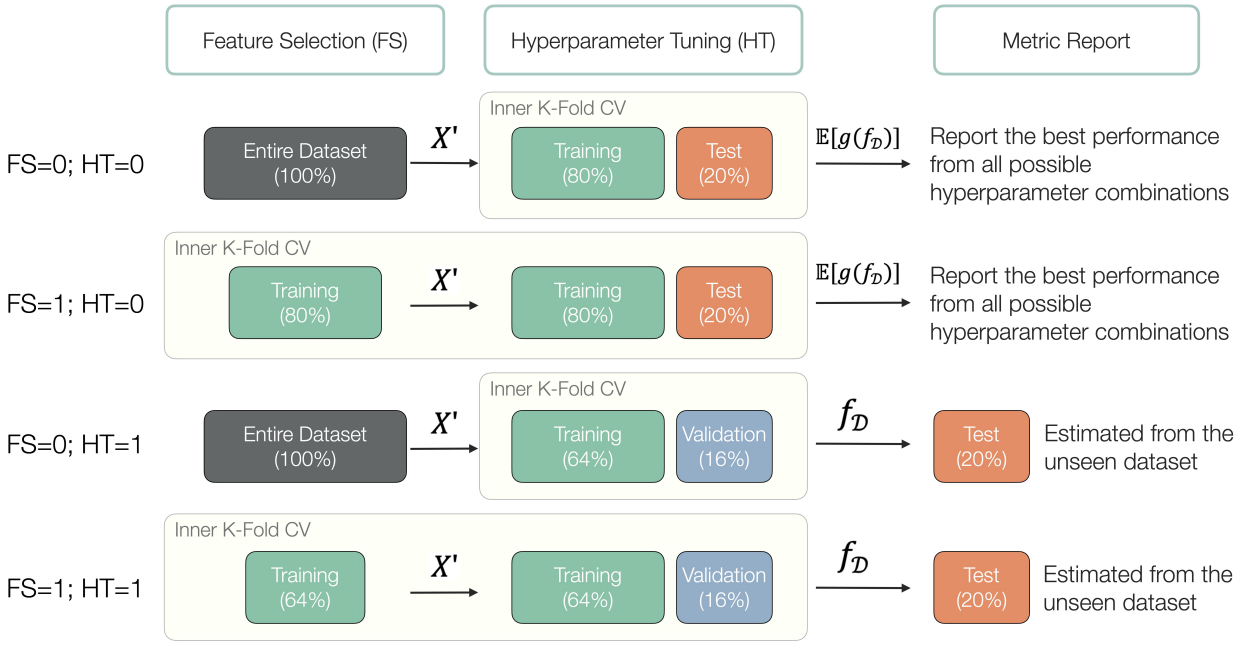

Figure 1: Workflow diagram illustrating four cross-validation strategies of feature selection (FS) and hyperparameter tuning (HT), where 0 denotes incorrect implementation and 1 indicates correct practice. X' is the selected feature subset, $\mathbb{E}[\hat{g}(f_\mathcal{D})]$ is the expected generalization performance, $f_\mathcal{D}$ is the model trained on the training set without being revealed to the test set.

bias is measured as the discrepancy between the model selection-influenced performance estimate and the expected generalization performance (r=0), using the Pearson correlation coefficient between predicted and observed values. Over 1000 sampling iterations, the study assesses the distribution of validation bias. A t-test will determine whether the validation bias significantly deviates from zero.

2.3 Study 3: Block Effects in Cross-Validation

The objective of the study is to demonstrate how a Random CV, which randomly assigns the samples to folds without considering the block effects, could overestimate the model performance. This study also conducts a block CV, where each block is used as a fold in the cross-validation, as the benchmark. The hypothesis is that the model performance estimated by Random CV is significantly higher than the estimation by block CV. This study simulated a regression task with 100 instances across ten features, denoted as X, and one single response variable, Y. Both X and Y are derived from a standard normal distribution. To introduce a block factor, the study groups every 20 observations into a block, with each block incrementally increasing by b units from zero, where b was simulated from 0.5 to 3.0 with an increment of 0.5. Within these ten features, one is substituted as the block level, represented by an integer from 0 to 4, augmented with random noise drawn from a standard normal distribution. This setup aims to simulate a scenario where the predictors primarily capture block variation, given the null expectation in predictability when using ten random variables X to forecast another random variable Y. The study investigates two model evaluation strategies: Block CV and Random CV, both utilizing a 5-fold cross-validation method. In block CV, each block serves as a separate fold,

while in Random CV, samples are randomly allocated to each fold (Figure 2). The predictive model is linear regression, and the performance is evaluated using Pearson's correlation coefficient. This simulation runs for 1000 iterations, with X and Y being resampled in each cycle. A one-tailed t-test assesses if the mean estimated performance significantly exceeds zero. Additionally, an Analysis of Variance (ANOVA) table is calculated when b is 0.5 to ascertain if the simulated block variation notably exceeds the assumed individual variation, representing the primary interest.

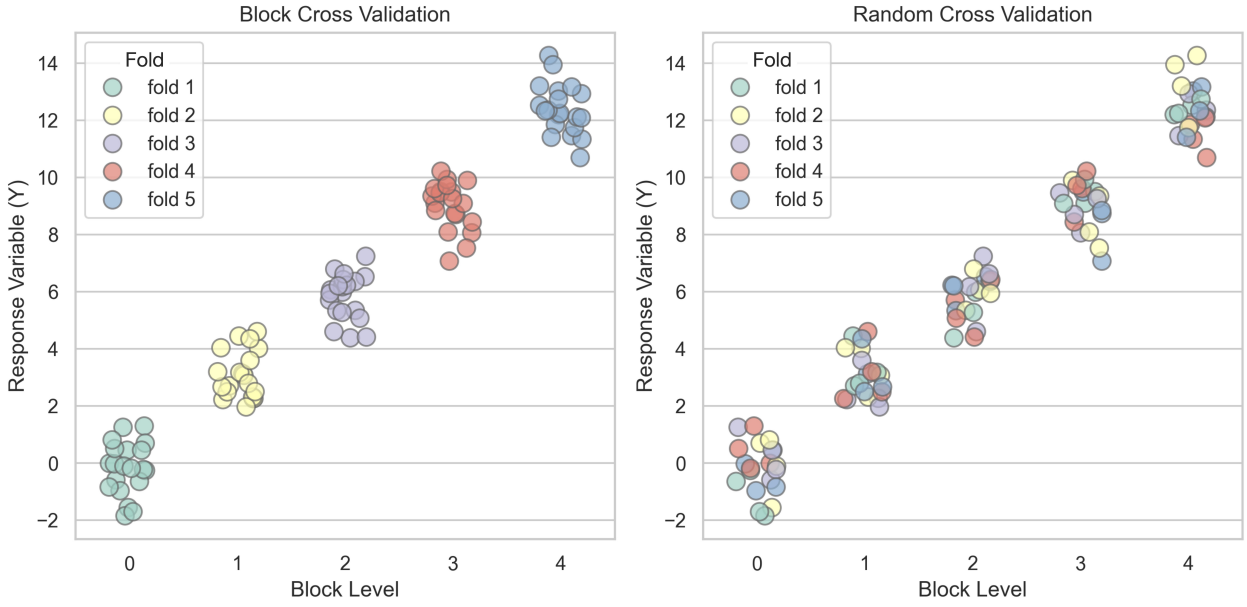

Figure 2: Illustration of fold assignment in block cross validation (left) and random cross validation (right). Folds are color-coded, and the block effect is set to 3 in this example.

2.4 Study 4: Performance Metrics in Regression Tasks

This study explores two error-based metrics, Root Mean Squared Error (RMSE) and Root Mean Squared Percentage Error (RMSPE), and three linearity-based metrics, Pearson Correlation Coefficient (r), the Coefficient of Determination (R^2), and the Concordance Correlation Coefficient (CCC), in a variety of commonly-encountered data challenges. These data challenges are depicted through 4 scenarios, representing data commonly encountered in predictive applications varying in scope (scenarios "Baseline" and "Scaled"), data with outliers disrupting the scale of prediction (scenario "Outlier Focused"), and data with an underlying grouping structure (scenario "Clustered"). The statistical description of the approach to generating each of these scenarios is included below. Practical examples of real-world instances of these types of data challenges are also described.

In the hypothetical example depicted in Figure 7, 100 observations were generated from two separate normal distributions. The first 50 observations were drawn from a normal distribution with a mean of -3 and a standard deviation of 1, denoted as $\mathcal{N}(-3, 1)$. The remaining 50 observations were generated from another normal distribution, $\mathcal{N}(3, 1)$. Utilizing two distinct distributions served to simulate experimental block effects, preset at a magnitude of 6

units for this experiment. Based on the simulated observations, four scenarios of predictions were derived according to the setting below:

- Scenario "Baseline": To establish a correlation relationship, the observations were added another random variable sampled from $\mathcal{N}(0, 1)$ to introduce prediction errors. This scenario represents a "best case" for developing predictive analytics, and could be exemplary of scenarios like predicting a scaled performance response (i.e., milk yield, average daily gain) from measurable input variables like dry matter intake, sensor system data, or past performance data.
- Scenario "Scaled": The prediction outcome from Scenario "Baseline" was multiplied by 5, simulating predictions with a larger variance while maintaining the same relative order as the original predictions. There are some responses that have naturally greater proportional variation compared with others. For example, an animal's body core temperature is unlikely to vary by more than 5%; however, daily variation around measurements like feed intake can range upwards of 30 to 40%. Comparison of scenarios "Baseline" and "Scaled" explore how this natural variation should be included in interpreting predictive analytics.
- Scenario "Outlier Focused": only the top 10% of predictions that deviate the most from zero in Scenario "Baseline" were raised to the power of 3. The rest of the predictions were set to zero. This scenario simulates a prediction that focuses solely on the extreme samples. In disciplines like nutritional exploration, the emphasis of predictive analytics typically focuses on understanding the mean animal or the mean response of an individual animal; however, in predictive analytics focused on health or genetic merit, the emphasis of prediction is often on the extreme observations. Analytics to understand the extreme observations is always complicated by the question of whether extremes are due to true outliers or some sort of measurement error. As precision livestock farming advances, the opportunities for measurement error due to erroneous sensor measurements increases.
- Scenario "Clustered": Values sampled from two normal distributions, $\mathcal{N}(-3, 2)$ and $\mathcal{N}(3, 2)$, were added respectively to the predictions made in Scenario "Baseline" of Block A (cross markers in Figure 7) and Block B (circle markers in Figure 7). In the animal sciences we often rely on blocks as an experimental tool to support analytics given challenging experimental design or constrained animal units. Many times, the difference between blocks dwarfs the differences observed within a block, resulting in a masking of true effects due to the block influence. This scenario amplified the original block effects, simulating a model that effectively distinguished between different blocks (e.g., herd or breed) but was less capable of predicting individual variations within each block. An example of this scenario might be simulating milk production or body weight across species – the magnitude of the difference between sheep and cattle (for example) far outweighs the magnitude of the difference of sheep or cattle over time.

This quartet of predictions serves to simulate potential challenges and complexities encountered in real-world modeling scenarios, thereby providing a foundation for evaluating different performance metrics used in regression problems.

2.5 Study 5: Performance Metrics in Classification Tasks

This study presents a hypothetical example to highlight how the choice of different performance metrics can lead to different interpretations of a model's effectiveness. The example focuses on binary classification, where the outcome is either positive (Y=1) or negative (Y=0). Suppose a binary classification model always outputs a probability between 0 and 1, indicating the likelihood that a sample belongs to the positive class. This example assumes that the model has high confidence in correctly predicting 1 out of 4 positive and 5 out of 6 negative samples. This example intends to illustrate a scenario where the positive outcome is rare, such as predicting the onset of a calving event in dairy cows [38, 39]. The example data is shown in Figure 8. In addition to the original labels, this example also examines a scenario with inverted labels (Figure 8. Upper). Since most classification metrics prioritize positive samples, it is generally advisable to designate the event of interest as the positive class in binary classification problems. Inverting the labels illustrates the potential overestimation of model performance when the more common, but less significant, background event is mistakenly marked as the positive class. It is important to note that inverting the labels in this example only affects the interpretation of model performance, not the model configuration or parameters. To evaluate classification performance, one must first establish a confidence threshold to dichotomize the prediction probabilities. For instance, if a prediction probability exceeds the threshold, the sample is labeled positive. By default, the threshold is set at 0.5 for its simplicity. For example, in the third data row of the example data: With a prediction probability of 0.38 that falls below the threshold, the sample is deemed negative, resulting in a false negative classification since the ground truth is positive. It is worth mentioning that this threshold is adjustable to fine-tune model performance for particular uses. A confusion matrix (Figure 8. Lower), effectively encapsulates prediction outcomes. The rows in this 2x2 matrix correspond to ground truth, while its columns reflect predictions. Correct predictions populate the diagonal cells, and errors fill the off-diagonal ones. This matrix serves as the foundation for computing various metrics to assess model performance, which will be explored in the result sections.

3 Results and Discussion

3.1 Study 1: The Impact of Estimator Choice and Sample Size on Model Evaluation Reliability

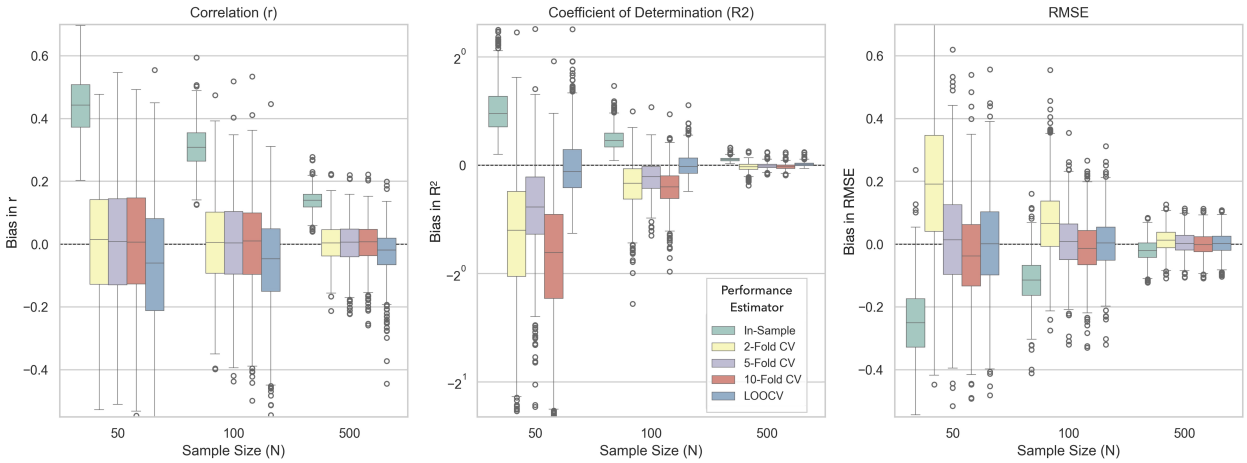

Figure 3: Simulation results of evaluation bias from 1000 sampling iterations. Multiple performance estimators across different sample sizes were color-coded. Three metrics: r, R^2, and RMSE, were displayed in the column facets.

The simulation results, depicted in box plots (Figure 3 and 4), explored the evaluation bias and variance distribution. Figure 3 examines the bias alterations across various estimators and sample sizes. Independent of the estimator and metric, the bias diminishes with increasing sample sizes. The in-sample estimator consistently overestimates across all metrics and sample sizes, underscoring the necessity of CV for unbiased performance evaluation. In CV estimators, although LOOCV is traditionally viewed as unbiased, it shows underestimation in model performance, especially when the metric is correlation coefficient (r). Comparatively, 2-, 5-, and 10-fold CV provide a more unbiased estimation than LOOCV for all sample sizes. However, for metrics like R^2 or RMSE, LOOCV emerges as the least biased estimator. While K-fold CV exhibits higher bias than LOOCV, this difference dwindles when the sample size exceeds 500. Notably, 10-fold CV, contrary to expectations, demonstrates higher bias than 5-fold CV for small sample sizes (50 and 100) in the R^2 metric, though this disparity also becomes insignificant at larger sample sizes.

Considering LOOCV's singular data point testing, its evaluation variance is pertinent only for RMSE, which permits single data point evaluations. Figure 4 illustrates the bias and variance in RMSE across different performance estimators as a function of sample size N. Both bias and variance in RMSE decrease as sample size increases, aligning with the hypothesis. LOOCV provides the least biased estimation, while 2-fold CV exhibits the highest bias without significant reduction at larger sample sizes. However, biases across all estimators converge at a sample size of 500. In terms of evaluation variance, LOOCV consistently shows higher values than other estimators for all sample sizes. Additionally, a lower number of folds K correlates with reduced variance, which is also in line with the hypothesized trend.

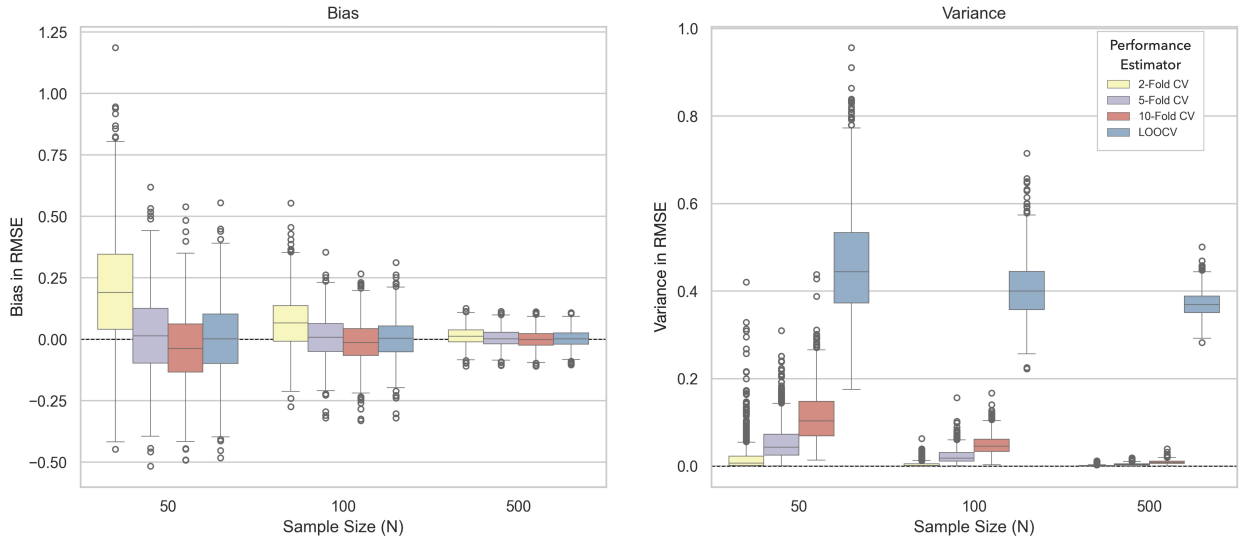

Figure 4: Simulation results of evaluation bias and variance from 1000 sampling iterations. Multiple performance estimators across different sample sizes were color-coded. Only RMSE was displayed. Bias and variance were listed in the left and right facets, respectively.

In conclusion, when conducting model evaluation, it is crucial to consider the estimator and sample size, as they significantly influence evaluation reliability which can be decomposed into bias and variance. Larger sample sizes generally lead to reduced bias and variance, enhancing the reliability of the evaluation process. For unbiased performance estimation, CV methods, such as K-fold CV and LOOCV, are preferable to in-sample estimation. LOOCV often provides less biased estimations for certain metrics but can exhibit higher variance. It is also noteworthy that the number of folds in K-fold CV can affect bias and variance; thus, experimenting with different numbers of folds, especially in smaller sample sizes, can be beneficial. Ultimately, the selection of appropriate evaluation techniques should be tailored to the specific context of the dataset and the objectives of the modeling exercise, ensuring a robust and reliable assessment of model performance.

3.2 Study 2: Misuse of Model Selection Can Lead to Over-Optimistic Performance Estimates

The evaluation bias was visualized using box plots (Figure 5), with the feature selection factor (FS) on the x-axis and hyperparameter tuning (HT) distinguished by color — green for incorrect and yellow for correct implementation. The y-axis represents the evaluation bias as measured by the correlation coefficient. The results indicate a clear overestimation of model performance when feature selection is applied to the entire dataset, regardless of hyperparameter tuning. The median biases were 0.797 for "FS=0; HT=0" and 0.761 for "FS=0; HT=1". Moreover, inappropriate evaluation in hyperparameter tuning resulted in a significant bias (p-value < 0.001) with a median of 0.113 for "FS=1; HT=0". The only scenario without bias significantly occurred when both feature selection and hyperparameter tuning were correctly incorporated within the cross-validation process "FS=1; HT=1", yielding a median bias of -0.008. These findings align

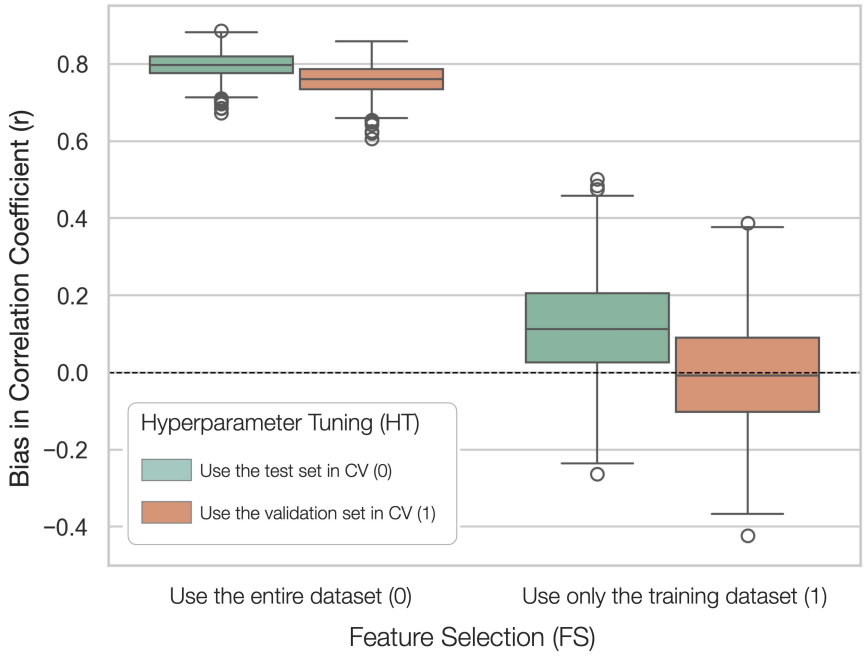

Figure 5: The evaluation bias of the four model selection strategies.

with the initial hypothesis and the prevailing literature, reinforcing that model selection must be integrated into the cross-validation workflow to prevent an overestimation of model performance.

The simulation results robustly confirm the hypothesis that improper implementation of model selection inflates performance estimates. Specifically, the evaluation bias is markedly high when feature selection precedes data splitting, with or without correct hyperparameter tuning. Although integrating feature selection within cross-validation folds mitigates this bias, incorrect hyperparameter tuning still significantly skews performance metrics. Notably, this overestimation from the hyperparameter tuning is even more pronounced in complex models, such as neural network architectures that often entail over a million parameters. These findings underscore the necessity of meticulous cross-validation practices, particularly for feature selection and hyperparameter tuning, to ensure accurate performance estimations and generalizability in predictive modeling.

3.3 Study 3: Overlooking Experimental Block Effects Can Lead to Biased Model Performance Estimates

In this simulation, an ANOVA table (Table 4), calculated from a single iteration for illustrative purposes, demonstrates that the simulated data exhibits block variation significantly greater than the residual variance. The result (Figure 6) shows that regardless of the amplitude of block effects in this simulation study, the Block CV strategy consistently yields a mean performance estimate close to zero, while the Random CV strategy consistently and significantly overestimates the model performance (p-value < 0.001). This finding supports the hypothesis that Random CV tends to overestimate model performance when block variation predominates over residual variation.

Table 4: ANOVA results for a single iteration of the simulated data with b = 0.5. SS: sum of squares; DF: degree of freedom; MS: mean square; F: F-statistic

Source	SS	DF	MS	F	p-value
Between	60.971	4	15.243	20.580	<0.001
Within	70.363	95	0.741		
Total	131.35	99			

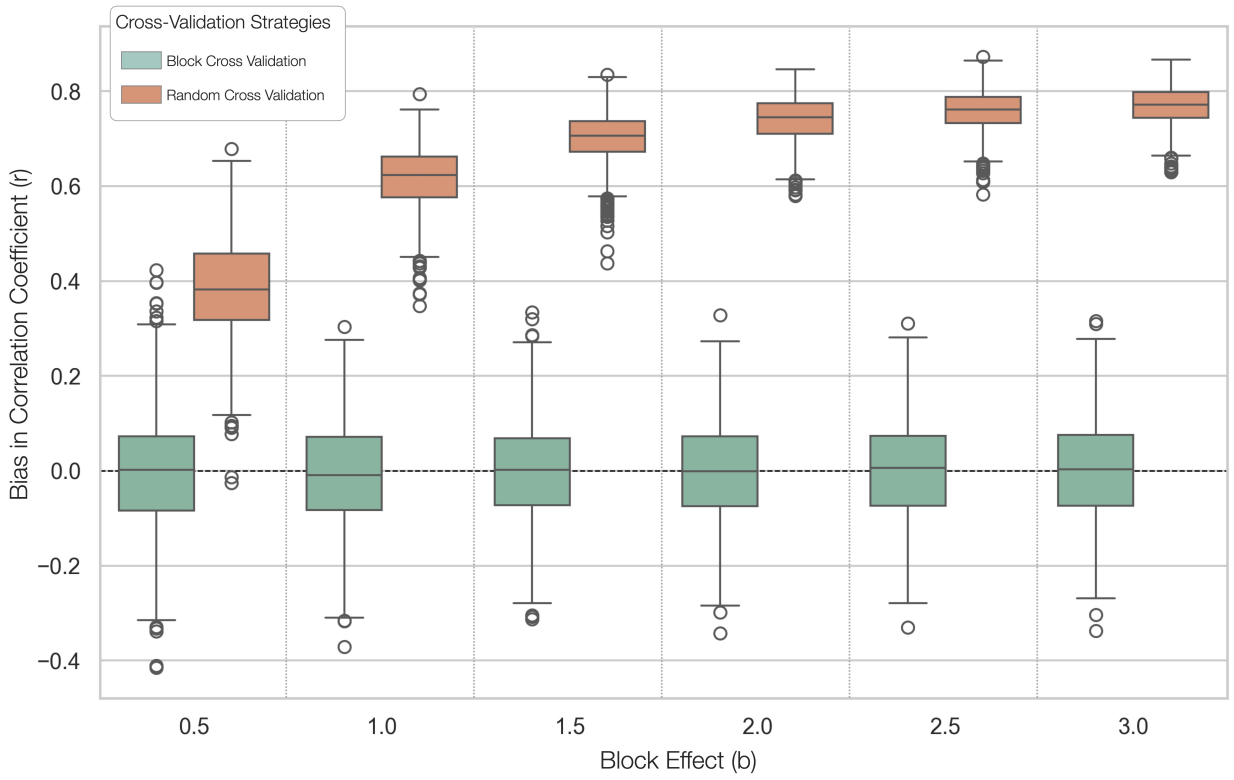

Figure 6: Bias in model performance estimation by Block CV and Random CV across 1000 iterations. The dashed line represents the null hypothesis that the mean performance estimate is zero.

In conclusion, block CV proves to be a vital tool in assessing the generalizability and accuracy of a predictive model, especially in contexts where block effects, such as herd variations, play a significant role in both the predicting features and response variable. The random CV strategy, which randomly assigns samples to folds without considering block effects, tends to overestimate model performance. This study recommends that block CV be used as a benchmark in model evaluation, especially when block effects are present

3.4 Study 4: Different Regression Metrics Illustrate Different Aspects of Model Performance

The simulated hypothetical example in Figure 7 illustrates the performance of four different prediction scenarios. The error-based metrics, RMSE and RMSPE, are sensitive to the magnitude of the error. In Scenario "Scaled", where the errors are five times larger but remain the same in rank order compared to Scenario "Baseline", the RMSE inflates from

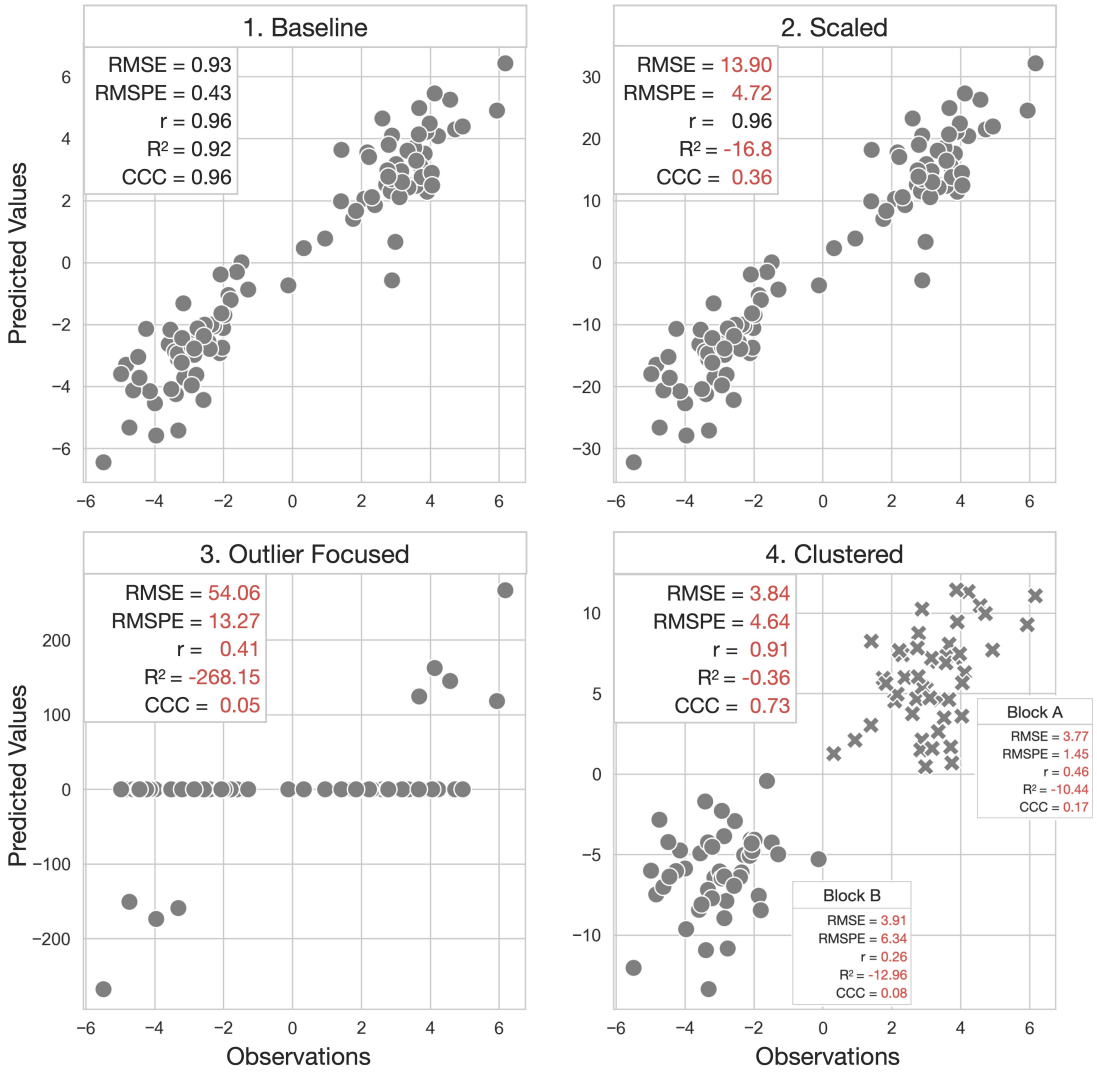

Figure 7: Scatter plots display the same observations against four different prediction scenarios in the given hypothetical example. Scenario "Baseline" serves as a baseline for the metrics, with any metric better than the baseline highlighted in bold and underscored, and any worse metric colored in red.

0.93 to 13.90, and RMSPE also increases from 0.43 to 4.72. Another notable characteristic of RMSE and RMSPE is that they weigh more on large errors, which is essential when making a large error is costly and should be prioritized for avoidance. In Scenario "Outliers", where certain predictions deviate substantially from the majority, the squaring operation in Equation 1 accentuates these outliers, culminating in an RMSE of 54.06 and RMSPE of 13.27. However, when investigating into the intra-block performance in the scenario "Clustered", the RMSE failed to detect the inflated performance due to the strong block effects. It resulted in a similar RMSE of 3.84 from the entire prediction set and 3.77 and 3.91 within each block. This phenomenon emphasizes again that RMSE is affected solely by the magnitude of the error, which neglects the ability of the model to capture relative trends in intra-block or inter-block predictions. On the other hand, when the goal is to rank observations of interest rather than predict the absolute magnitude of the error, linearity-based metrics can provide more insights. The correlation r is an example showing its consistency across

the Scenario "Baseline" and "Scaled", despite the latter having five times larger errors. This metric is particularly useful when the relative order of predictions is more important than the absolute error magnitude. However, it is worth noting that the correlation r can be misleading in certain scenarios, such as the Scenario "Outlier Focused", where 90% of the predictions are zero. In this case, the correlation r show a moderate performance of 0.41, which is mainly contributed by the 10% of the outlier predictions that are "ranked" correctly but with a large error magnitude. This example highlights the importance of visually inspecting the regression results through scatter plots to avoid misleading conclusions. Moreover, one common pitfall of the correlation r is that block effects can influence it, leading to an inflated performance estimate if individual variation is of greater interest than inter-block variation. This was demonstrated in Scenario "Clustered", where the overall coefficient r was 0.91, but the metric within each block was only 0.46 and 0.26, respectively. Therefore, it is essential to examine regression results within identifiable blocks. Besides the correlation r, R^2 provides a more comprehensive insight, as it focuses both the linear trend from the variance composition and the error magnitude from the residual sum of squares. From the "Scaled" scenario, R^2 successfully detected the inflated error magnitude, resulting in a negative value of -16.8. It also captured the outlier-induced variance in the "Outliers" scenario, with a negative value of -268.15. Lastly, in Scenario "Clustered", the value of R^2 indicated a weak performance by the model with a score of -0.36. This score is statistically reasonable, as the predictions have larger variance than the observations. The metric also successfully detected the model failure in capturing intra-block variation, as the R^2 values within each block were -10.44 and -12.96, respectively. However, an obvious limitation of R^2 is that it has no standard scale. Considering this, a more nuanced evaluation metric, CCC, showcases a more balanced performance evaluation. It always range from -1 to 1, and it successfully captures all the characteristics of the four scenarios. In Scenario "Scaled", the CCC value dropped from 0.96 to 0.36. In Scenario "Outliers", the CCC value plummeted to 0.05, showcasing the model's failure to "align" the predictions with the observations with 90% of the predictions being zero. In the Scenario "Clustered", although the CCC value was 0.73, the metric also showed the model's weakness in each block, with the CCC values of 0.17 and 0.08, respectively. This study demonstrates that CCC is a more balanced metric that considers both the linear trend and the error magnitude, making it a more comprehensive evaluation metric for regression models.

3.5 Study 5: Label-Invariant Metrics Provide Balanced Assessment in Binary Classification

Different metrics in binary classification were evaluated in a simulated example (Figure 8). The original labels were inverted to examine the robustness of the metrics against label choices. The accuracy metric, with a 0.5 threshold in this example, stands at 0.60. This figure might suggest modest efficacy, marginally surpassing random chance, with an accuracy of 0.50. Nonetheless, the same accuracy level could be achieved by classifying every sample as negative in an imbalanced dataset where negatives are predominant. In contrast, precision and recall provide a more nuanced evaluation of model performance by separately assessing the correctness of positive predictions and the ability to detect actual positives. With a threshold of 0.5, the example dataset yields precision and recall values of 0.5 and 0.25, respectively. These metrics deliver more interpretable information that only half of the positive predictions are correct, and just a

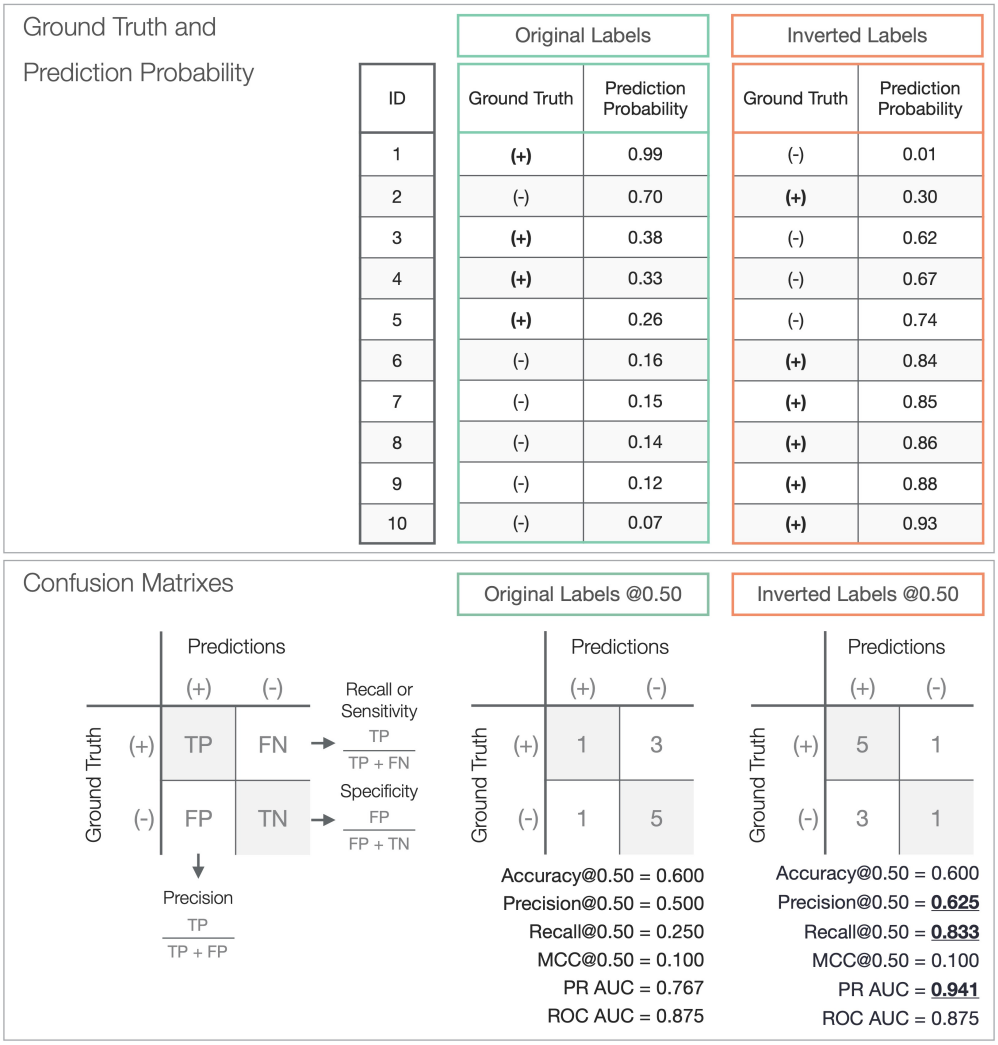

Figure 8: Simulated hypothetical example of binary classification. TP: true positive; FN: false negative; FP: false positive; TN: true negative; **Upper**: The ground truth and prediction probability. **Lower**: The confusion matrix of the prediction at a threshold of 0.5, followed by classification metrics of accuracy, precision, recall, MCC, PR curve AUC, and ROC curve AUC. The performance of the original labels serves as a baseline for comparison. Any better performance metrics from the inverted labels are highlighted in bold and underscored

quarter of the actual positives are detected. This contrasts with an accuracy of 0.6, which may appear misleadingly high due to the abundance of negative samples. Additionally, it is noted that the chosen confidence threshold significantly impacts precision and recall. While the trade-off between these two metrics is not always linear, it is generally observed that a higher threshold increases precision but decreases recall, and vice versa. A high threshold indicates a conservative approach in predicting positives, reducing false positives, and thus enhancing precision. However, this often leads to missing actual positive cases, lowering recall. Hence, the Precision-Recall (PR) curve is an essential tool for evaluating model performance across various thresholds. Plotted with recall on the x-axis and precision on the y-axis, this curve is derived by computing these metrics at different thresholds (Figure 9, Left). The Area Under the Curve (AUC) provides a summary measure of the PR curve's overall performance. A model's effectiveness is generally indicated by how close

a point on the PR curve is to the top-right corner. For example, at a threshold of 0.25, which is positioned near the top-right of the PR curve, the model demonstrates impressive performance with an accuracy of 0.90, precision of 0.80, and recall at 1.00.

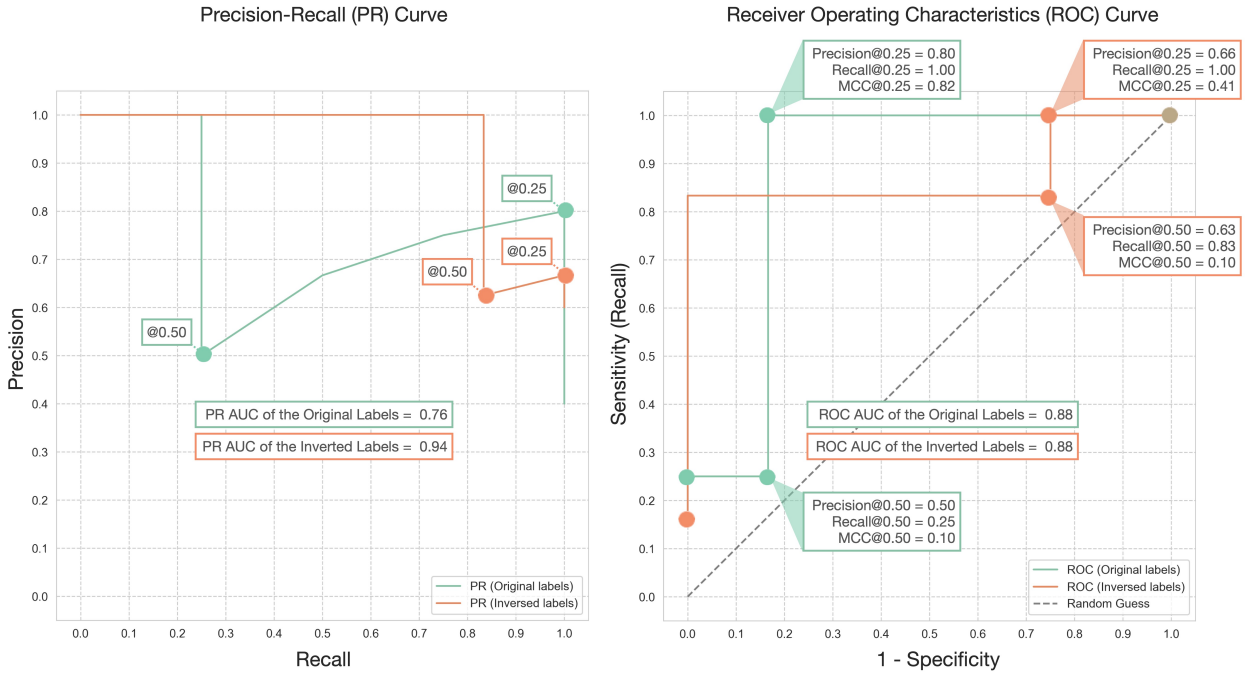

Figure 9: **(Left)** Precision-recall (PR) curve and **(Right)** Receiver operating characteristic (ROC) curve for the hypothetical example are displayed. The performance at confidence thresholds of 0.25 and 0.50 is highlighted. Original labels are marked in green, while inverted labels appear in orange. The Area Under the Curve (AUC) is depicted at the center of each curve.

However, it is worth re-emphasizing that precision and recall focus predominantly on positive samples. Inappropriately assigning a predominant background event as the positive class can lead to skewed interpretations. This pitfall is demonstrated in this example by inverting the labels. At a threshold of 0.50, precision increases from 0.50 to 0.63, and recall jumps from 0.25 to 0.83. With the threshold set at 0.25, precision drops to 0.66 from 0.80, while recall remains unchanged. The PR AUC also rises from 0.76 to 0.94. Such shifts in metrics, driven merely by label rearrangement unrelated to the data or model characteristics, underscore the importance of label-invariant metrics that remain unaffected by label assignments. Unlike metrics focusing solely on positive samples, the ROC curve accounts for both positive and negative samples, making it a label-invariant metric. Specificity is plotted on the x-axis and sensitivity on the y-axis, calculated at different thresholds (Figure 9, Right). In this hypothetical example, the ROC curve demonstrates robustness and label-invariance with a consistent AUC of 0.875, regardless of whether the original or inverted labels are used. Lastly, another label-invariant metric is MCC which provides a balanced assessment of both positive and negative samples. Considering MCC's balanced approach to evaluating model performance, this study introduces the concept of an MCC curve. This curve, which plots the MCC value against various threshold levels (Figure 10), serves as a powerful tool for identifying the optimal confidence thresholds for model predictions. By examining this curve, one

can determine the specific threshold at which the MCC value peaks, thereby optimizing the model's performance. For example, when applied to the hypothetical example, the optimum MCC value of 0.82 was attained at a threshold of 0.25. This particular threshold corresponded to accuracy, precision, and recall values of 0.90, 0.75, and 1.00, respectively. Notably, the MCC curve retains its symmetry even when labels are reversed, affirming its status as a label-invariant measure. In scenarios with inverted labels, the maximum MCC value observed was 0.83, achieved at a threshold of 0.75, leading to accuracy, precision, and recall values of 0.90, 1.00, and 0.83, respectively. Such findings underscore the MCC's ability to provide a balanced and comprehensive assessment of both positive and negative samples, thereby reinforcing its utility as a versatile and effective metric for thorough model evaluation.

In conclusion, binary classification models are often evaluated using metrics focusing on positive samples, such as precision and recall. It is generally advisable to designate the event of interest as the positive class. Otherwise, these metrics can be misleading when the more common but less significant background event is mistakenly marked as the positive class. To circumvent this potential bias, adopting label-invariant metrics is recommended. These metrics offer a more balanced and reliable assessment of model performance. Notable examples of such metrics include the ROC curve and the proposed MCC curve by this review, both of which are unaffected by the choice of positive and negative class labels and are thus robust for a thorough model evaluation.

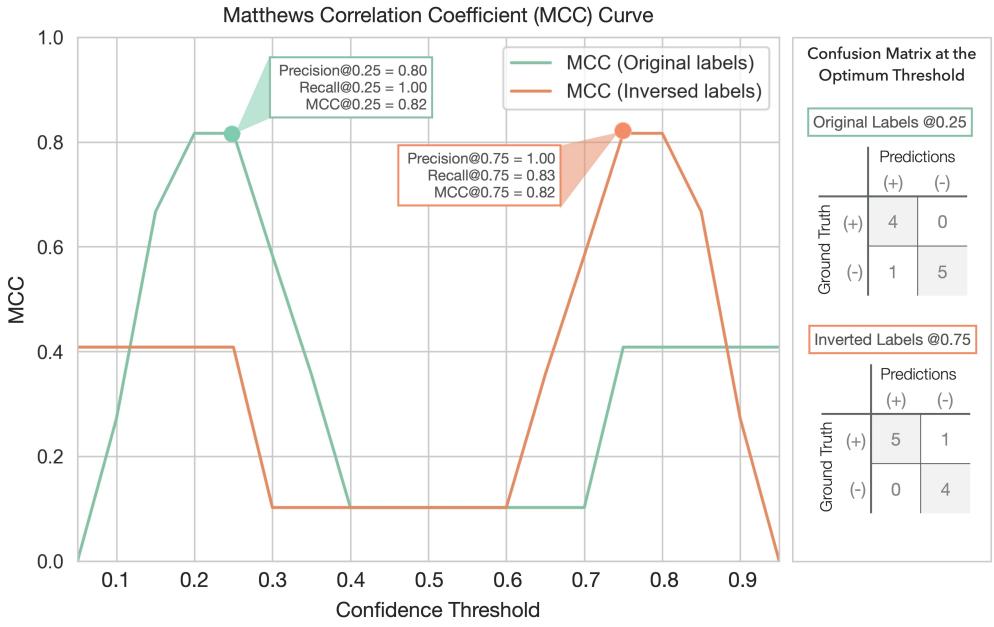

Figure 10: Matthews Correlation Coefficient (MCC) curve. A line chart plotting MCC at different thresholds for the hypothetical example. The optimal threshold is highlighted by the dot marks in green and orange for the original and inverted labels, respectively. The confusion matrix at the optimal threshold is displayed in the right panel.

4 Conclusion

In summary, the review highlights several key considerations for performance assessment in predictive modeling. When evaluating regression models, the choice of metrics like Correlation Coefficient r, RMSE, and R^2 depends on the specific goals of the model. A comprehensive evaluation should include multiple metrics to understand different aspects of model performance. In binary classification models, precision and recall are crucial, but it is essential to correctly designate the positive class to avoid bias. Label-invariant metrics, such as the ROC curve and the proposed MCC curve, provide a balanced assessment, unaffected by class label choices. Additionally, the reliability of model evaluation is significantly influenced by estimator choice and sample size. Larger sample sizes tend to reduce bias and variance, increasing evaluation reliability. Cross-validation methods, such as K-fold CV and LOOCV, are preferable for unbiased performance estimation, with the number of folds in K-fold CV being particularly influential in smaller datasets. Moreover, the review underscores the importance of correct implementation in model selection processes, as improper techniques can inflate performance estimates. This is especially true in complex models where feature selection and hyperparameter tuning need meticulous cross-validation to avoid overestimation of performance. Finally, the utility of Block CV is emphasized in contexts where block effects are significant. It provides a more realistic assessment of model generalizability and accuracy compared to a Random CV, which tends to overestimate performance in such scenarios. Overall, the review recommends a thoughtful selection of metrics and evaluation techniques, tailored to the specific dataset and modeling objectives, to ensure accurate and reliable performance assessments in predictive modeling.

5 Acknowledgement

The author James Chen expresses his gratitude to Drs. Zhiwu Zhang, Hao Cheng, Gota Morota, and Gonzalo Ferreira for their insightful discussions that partially contributed to this study. The authors declare no conflicts of interest.

References

[1] Hao Cheng, Dorian J. Garrick, and Rohan L. Fernando. Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction. *Journal of Animal Science and Biotechnology*, 8(1):38, May 2017.

[2] I. D. E. van Dixhoorn, R. M. de Mol, J. T. N. van der Werf, S. van Mourik, and C. G. van Reenen. Indicators of resilience during the transition period in dairy cows: A case study. *Journal of Dairy Science*, 101(11):10271–10282, November 2018.

[3] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009.

[4] Gavin C. Cawley and Nicola L.C. Talbot. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, August 2010.

[5] Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality].

[6] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996.

[7] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, pages 155–161, Cambridge, MA, USA, December 1996. MIT Press.

[8] Hervé Abdi. Partial Least Square Regression PLS-Regression. *Encyclopedia of social sciences research methods*, pages 792–795, 2003.

[9] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[10] Yann LeCun. Generalization and Network Design strategies. 1989.

[11] Morteza H. Ghaffari, Amirhossein Jahanbekam, Hassan Sadri, Katharina Schuh, Georg Dusel, Cornelia Prehn, Jerzy Adamski, Christian Koch, and Helga Sauerwein. Metabolomics meets machine learning: Longitudinal metabolite profiling in serum of normal versus overconditioned cows and pathway analysis. *Journal of Dairy Science*, 102(12):11561–11585, December 2019.

[12] G. Rovere, G. de los Campos, A. L. Lock, L. Worden, A. I. Vazquez, K. Lee, and R. J. Tempelman. Prediction of fatty acid composition using milk spectral data and its associations with various mid-infrared spectral regions in Michigan Holsteins. *Journal of Dairy Science*, 104(10):11242–11258, October 2021.

[13] C. A. Becker, A. Aghalari, M. Marufuzzaman, and A. E. Stone. Predicting dairy cattle heat stress using machine learning techniques. *Journal of Dairy Science*, 104(1):501–524, January 2021.

[14] B. Lahart, S. McParland, E. Kennedy, T.M. Boland, T. Condon, M. Williams, N. Galvin, B. McCarthy, and F. Buckley. Predicting the dry matter intake of grazing dairy cows using infrared reflectance spectroscopy analysis. *Journal of Dairy Science*, 102(10):8907–8918, October 2019.

[15] Tiago Bresolin and João R. R. Dórea. Infrared Spectrometry as a High-Throughput Phenotyping Technology to Predict Complex Traits in Livestock Systems. *Frontiers in Genetics*, 11, 2020.

[16] C. Grelet, E. Froidmont, L. Foldager, M. Salavati, M. Hostens, C. P. Ferris, K. L. Ingvartsen, M. A. Crowe, M. T. Sorensen, J. A. Fernandez Pierna, A. Vanlierde, N. Gengler, and F. Dehareng. Potential of milk mid-infrared spectra to predict nitrogen use efficiency of individual dairy cows in early lactation. *Journal of Dairy Science*, 103(5):4435–4445, May 2020.

[17] I. Adriaens, N. C. Friggens, W. Ouweltjes, H. Scott, B. Aernouts, and J. Statham. Productive life span and resilience rank can be predicted from on-farm first-parity sensor time series but not using a common equation across farms. *Journal of Dairy Science*, 103(8):7155–7171, August 2020.

[18] Lucio F. M. Mota, Diana Giannuzzi, Vittoria Bisutti, Sara Pegolo, Erminio Trevisi, Stefano Schiavon, Luigi Gallo, David Fineboym, Gil Katz, and Alessio Cecchinato. Real-time milk analysis integrated with stacking ensemble learning as a tool for the daily prediction of cheese-making traits in Holstein cattle. *Journal of Dairy Science*, 105(5):4237–4255, May 2022.

[19] Roii Spoliansky, Yael Edan, Yisrael Parmet, and Ilan Halachmi. Development of automatic body condition scoring using a low-cost 3-dimensional Kinect camera. *Journal of Dairy Science*, 99(9):7714–7725, September 2016.

[20] Sun Yukun, Huo Pengju, Wang Yujie, Cui Ziqi, Li Yang, Dai Baisheng, Li Runze, and Zhang Yonggen. Automatic monitoring system for individual dairy cows based on a deep learning framework that provides identification via body parts and estimation of body condition score. *Journal of Dairy Science*, 102(11):10140–10151, November 2019.

[21] X. Song, E.A.M. Bokkers, P.P.J. Van Der Tol, P.W.G. Groot Koerkamp, and S. Van Mourik. Automated body weight prediction of dairy cows using 3-dimensional vision. *Journal of Dairy Science*, 101(5):4448–4459, May 2018.

[22] C. Xavier, Y. Le Cozler, L. Depuille, A. Caillot, A. Lebreton, C. Allain, J. M. Delouard, L. Delattre, T. Luginbuhl, P. Faverdin, and A. Fischer. The use of 3-dimensional imaging of Holstein cows to estimate body weight and monitor the composition of body weight change throughout lactation. *Journal of Dairy Science*, 105(5):4508–4519, May 2022.

[23] P. Mäntysaari, E.A. Mäntysaari, T. Kokkonen, T. Mehtiö, S. Kajava, C. Grelet, P. Lidauer, and M.H. Lidauer. Body and milk traits as indicators of dairy cow energy status in early lactation. *Journal of Dairy Science*, 102(9):7904–7916, September 2019.

[24] M. Frizzarin, I. C. Gormley, D. P. Berry, T. B. Murphy, A. Casa, A. Lynch, and S. McParland. Predicting cow milk quality traits from routinely available milk spectra using statistical machine learning methods. *Journal of Dairy Science*, 104(7):7438–7447, July 2021.

[25] J. A. D. R. N. Appuhamy, J. V. Judy, E. Kebreab, and P. J. Kononoff. Prediction of drinking water intake by dairy cows. *Journal of Dairy Science*, 99(9):7191–7205, September 2016.

[26] R. A. de Souza, R. J. Tempelman, M. S. Allen, W. P. Weiss, J. K. Bernard, and M. J. VandeHaar. Predicting nutrient digestibility in high-producing dairy cows. *Journal of Dairy Science*, 101(2):1123–1135, February 2018.

[27] J. R. R. Dórea, G. J. M. Rosa, K. A. Weld, and L. E. Armentano. Mining data from milk infrared spectroscopy to improve feed intake predictions in lactating dairy cows. *Journal of Dairy Science*, 101(7):5878–5889, July 2018.

[28] L. I. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268, March 1989.

[29] Edward J. Jones, Thomas F. A. Bishop, Brendan P. Malone, Patrick J. Hulme, Brett M. Whelan, and Patrick Filippi. Identifying causes of crop yield variability with interpretive machine learning. *Computers and Electronics in Agriculture*, 192:106632, January 2022.

[30] N. W. O'Leary, D. T. Byrne, A. H. O'Connor, and L. Shalloo. Invited review: Cattle lameness detection with accelerometers. *Journal of Dairy Science*, 103(5):3895–3911, May 2020.

[31] J. Stojkov, G. Bowers, M. Draper, T. Duffield, P. Duivenvoorden, M. Groleau, D. Haupstein, R. Peters, J. Pritchard, C. Radom, N. Sillett, W. Skippon, H. Trépanier, and D. Fraser. Hot topic: Management of cull dairy cows—Consensus of an expert consultation in Canada. *Journal of Dairy Science*, 101(12):11170–11174, December 2018.

[32] Maher Alsaaod, Mahmoud Fadul, and Adrian Steiner. Automatic lameness detection in cattle. *The Veterinary Journal*, 246:35–44, April 2019.

[33] X. Kang, X. D. Zhang, and G. Liu. Accurate detection of lameness in dairy cattle with computer vision: A new and individualized detection strategy based on the analysis of the supporting phase. *Journal of Dairy Science*, 103(11):10628–10638, November 2020.

[34] S. J. Denholm, W. Brand, A. P. Mitchell, A. T. Wells, T. Krzyzelewski, S. L. Smith, E. Wall, and M. P. Coffey. Predicting bovine tuberculosis status of dairy cows from mid-infrared spectral data of milk using deep learning. *Journal of Dairy Science*, 103(10):9355–9367, October 2020.

[35] S.A. Kandeel, A.A. Megahed, M.H. Ebeid, and P.D. Constable. Ability of milk pH to predict subclinical mastitis and intramammary infection in quarters from lactating dairy cattle. *Journal of Dairy Science*, 102(2):1417–1427, February 2019.

[36] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, January 2020.

[37] J. M. Bowen, M. J. Haskell, G. A. Miller, C. S. Mason, D. J. Bell, and C-A. Duthie. Early prediction of respiratory disease in preweaning dairy calves using feeding and activity behaviors. *Journal of Dairy Science*, 104(11):12009–12018, November 2021.

[38] V. Ouellet, E. Vasseur, W. Heuwieser, O. Burfeind, X. Maldague, and É. Charbonneau. Evaluation of calving indicators measured by automated monitoring devices to predict the onset of calving in Holstein dairy cows. *Journal of Dairy Science*, 99(2):1539–1548, February 2016.

[39] M.R. Borchers, Y.M. Chang, K.L. Proudfoot, B.A. Wadsworth, A.E. Stone, and J.M. Bewley. Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle. *Journal of Dairy Science*, 100(7):5664–5674, July 2017.

Appendix

Cross Validation

Model cross validation aims to evaluate how well a given model generalizes to an independent dataset that it has not seen during the training process. The most common method is K-fold cross-validation (**K-fold CV**). To implement the K-fold CV, the available dataset, denoted as $\mathcal{D}$, is partitioned into K equally sized folds. We can express the dataset as below:

$$\begin{aligned}\mathcal{D} &= \{(X, Y)\} \\ &= \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_K, Y_K)\}\end{aligned} \tag{18}$$

where $X \in \mathbb{R}^{n \times p}$ represents the input features, and $Y \in \mathbb{R}^{n \times 1}$ symbolizes the ground truth labels for a single target variable. The value of n corresponds to the total number of samples, while p represents the number of features. In each iteration of the K-fold CV, a single fold is reserved as the test set, $\mathcal{D}_{\text{test}}$ (or $\mathcal{D}_k$), to act as unseen data, while the remaining folds make up the training set $\mathcal{D}_{\text{train}}$ (or $\mathcal{D}_{\text{-k}}$):

$$\begin{aligned}\mathcal{D}_{\text{train}} &= \mathcal{D}_{\text{-k}} \\ &= \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_{k-1}, Y_{k-1}), (X_{k+1}, Y_{k+1}), \ldots, (X_K, Y_K)\} \\ \mathcal{D}_{\text{test}} &= \mathcal{D}_k \\ &= \{(X_k, Y_k)\}\end{aligned} \tag{19}$$

After splitting the dataset into $\mathcal{D}_{\text{-k}}$ and $\mathcal{D}_k$, the examined model f is trained on the training set $\mathcal{D}_{\text{-k}}$ and denoted as $f_{\mathcal{D}_{\text{-k}}}$. The hold-out test set $\mathcal{D}_k$ is then used to evaluate the model performance $\hat{g}\left(f_{\mathcal{D}_{\text{-k}}}\right)$, which is defined by comparing the predicted labels $\hat{Y}_k = f_{\mathcal{D}_{\text{-k}}}(X_k)$ with the true labels Y_k using a performance metric $\mathcal{L}$ (e.g., RMSE or R^2):

$$\begin{aligned}\hat{g}(f_{\mathcal{D}_{\text{-k}}}) &= \mathcal{L}(Y_k, \hat{Y}_k) \\ &= \mathcal{L}(Y_k, f_{\mathcal{D}_{\text{-k}}}(X_k))\end{aligned} \tag{20}$$

To estimate the generalization performance of a model $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$, the K-fold CV procedure is repeated K times until each fold has been used as the test set $\mathcal{D}_k$ once. The entire dataset $\mathcal{D}$ is leveraged to calculate the average prediction performance over all K folds. The model's generalization performance can be expressed as:

$$\begin{aligned}\mathbb{E}[\hat{g}(f_{\mathcal{D}})] &= \mathbb{E}[\hat{g}(f_{\mathcal{D}_{-k}})] \\ &= \frac{1}{K}\sum_{k=1}^{K}\hat{g}(f_{\mathcal{D}_{-k}})\end{aligned} \tag{21}$$

It is noted that $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$ is equivalent to $\mathbb{E}[\hat{g}(f_{\mathcal{D}_{-k}})]$ in K-fold CV. It is because the $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$ is estimated by averaging all $\hat{g}(f_{\mathcal{D}_{-k}})$ over K folds, which is also the definition of $\mathbb{E}[\hat{g}(f_{\mathcal{D}_{-k}})]$.

Cross Validation Bias and Variance

The true generalization performance of the model $G(f_{\mathcal{D}})$ can only be approximated by averaging the performance metrics over infinite unseen datasets. However, in practice, the dataset $\mathcal{D}$ is finite and therefore, there is always a bias when using a finite dataset to estimate $G(f_{\mathcal{D}})$. The bias is known as validation bias:

$$\text{Bias} = \mathbb{E}[\hat{g}\ (f_{\mathcal{D}})] - G(f_D) \tag{22}$$

For example, if RMSE is used as the performance metric, a positive validation bias suggests that the model validation procedure concludes a pessimistic estimation of the model performance, since the true performance is expected to be lower than the estimated performance. Another aspect of model validation is the variance of the estimated performance. For example, in a 5-fold cross-validation, there are five estimates of the model performance. The variance among these five estimates is known as validation variance. A high validation variance suggests that the performance is sensitive to the choice of the test set $\mathcal{D}_k$, which may be caused by a small sample size or an over-complex model. The validation variance can be defined as:

$$\begin{aligned}\text{Variance} &= \mathbb{E}[(\hat{g}(f_{\mathcal{D}_{-k}}) - \mathbb{E}[\hat{g}(f_{\mathcal{D}})])^2] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_{-k}}) - 2\hat{g}(f_{\mathcal{D}_{-k}})\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_{-k}})] - 2\mathbb{E}[\hat{g}(f_{\mathcal{D}_{-k}})]\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_{-k}})] - \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]\end{aligned} \tag{23}$$

Combining the Equations 22 and 23, the mean squared error (MSE) of the model validation can be decomposed as:

$$
\begin{aligned}
\text{MSE} &= \mathbb{E}[(\hat{g}(f_{\mathcal{D}_k}) - G(f_{\mathcal{D}}))^2] \\
&= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - 2\mathbb{E}[g(f_{D_k})]G(f_{\mathcal{D}}) + G^2(f_{\mathcal{D}}) + \\
&\quad \mathbb{E}^2[\hat{g}(f_{\mathcal{D}_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})] \\
&= (\mathbb{E}^2[\hat{g}(f_{\mathcal{D}_k})] - 2\mathbb{E}[\hat{g}(f_{D_k})]G(f_{\mathcal{D}}) + G^2(f_{\mathcal{D}})) + \\
&\quad (\mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - \mathbb{E}^2[\hat{g}(f_{\mathcal{D}_k})]) \\
&= (\mathbb{E}[\hat{g}(f_{\mathcal{D}_k})] - G(f_{\mathcal{D}}))2 + (\mathbb{E}[g2(f_{\mathcal{D}_k})] - E^2[\hat{g}(f_{\mathcal{D}_k})]) \\
&= (\mathbb{E}[\hat{g}(f_{\mathcal{D}})] - G(f_{\mathcal{D}}))^2 + (\mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]) \\
&= \text{Bias}^2 + \text{Variance}
\end{aligned}
\tag{24}
$$

Hyperparameter

Here are the loss functions for ordinary least squares (OLS), ridge regression, and LASSO regression, respectively:

$$\mathcal{L}_{\text{OLS}}(\beta) = \sum_{i=1}^{n}(y_i - x_i\beta)^2 \tag{25}$$

$$\mathcal{L}_{\text{ridge}}(\beta) = \sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{26}$$

$$\mathcal{L}_{\text{LASSO}}(\beta) = \sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{27}$$

Where x_i and y_i represent the ith row of the design matrix X and the response vector Y, respectively. The term n denotes the sample size, and β is the coefficient vector. All three models aim to find the optimal β that minimizes their respective loss function, $\mathcal{L}$. In the regularized models (i.e., ridge and LASSO regression), the vector length of β is penalized in the loss function.

Sqaured Correlation Coefficient r^2 and Determination Coefficient R^2

The squared Pearson correlation coefficient, r^2, is not necessarily equivalent to the coefficient of determination, R^2. This equivalence holds true specifically in the context of least squares regression when the same model and data are used for both fitting and evaluation. However, this may not be the case when the model is assessed using new data. To demonstrate the equivalence between r^2 and R^2 under these specific conditions, we begin by assuming that the covariance between the predicted values $\hat{Y}$ and the residuals ϵ is zero:

$$\begin{aligned}\text{cov}(Y, \hat{Y}) &= \text{cov}(\hat{Y} + \epsilon, \hat{Y}) \\ &= \text{cov}(\hat{Y}, \hat{Y}) + \text{cov}(\hat{Y}, \epsilon) \\ &= \text{var}(\hat{Y}) + \text{cov}(\hat{Y}, \epsilon) \\ &= \text{var}(\hat{Y})\end{aligned} \tag{28}$$

With the assumption that $\bar{\hat{Y}} = \bar{Y}$, which typically holds when $\mathbb{E}[\epsilon] = 0$, the squared correlation coefficient r^2 is expressed as follows:

$$\begin{aligned}r^2 &= \frac{\text{cov}^2(Y, \hat{Y})}{\text{var}(Y)\text{var}(\hat{Y})} \\ &= \frac{\text{var}(\hat{Y})^2}{\text{var}(Y)\text{var}(\hat{Y})} \\ &= \frac{\text{var}(\hat{Y})}{\text{var}(Y)} \\ &= \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{\hat{Y}})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \\ &= \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \\ &= \frac{SS_{\text{regression}}}{SS_{\text{total}}} \\ &= R^2\end{aligned} \tag{29}$$

where $SS_{\text{regression}}$ is the variation explained by the model and SS_{total} is the total sum of squares. Each Y_i and $\hat{Y}_i$ are the ith elements of the actual response vector Y and the predicted response vector $\hat{Y}$, while $\bar{Y}$ and $\bar{\hat{Y}}$ are their respective means. This proof highlights that under certain assumptions, r^2 and R^2 can indeed be equivalent, but such conditions are specific to least squares regression where errors are normally distributed and predictions are unbiased estimates of the actual values.