
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

Ryan Wright

School of Animal Sciences
Virginia Tech
Blacksburg, VA 24061
ryanw22@vt.edu

January 8, 2025

ABSTRACT

Predictive modeling is a cornerstone of data-driven research and decision-making in precision agriculture, yet achieving robust, interpretable, and reproducible model evaluations remains challenging. This study addresses two central issues in model evaluation — methodological pitfalls in cross-validation (CV) and data-structure effects on performance metrics — across five simulation experiments supplemented by real-world data. First, we show how the choice of estimator (e.g., 2-fold, 5-fold, or leave-one-out CV) and sample size affects the reliability of performance estimates: although leave-one-out CV can be unbiased for error-based metrics, it systematically underestimates correlation-based metrics. Second, we demonstrate that reusing the test data during model selection (e.g., feature selection, hyperparameter tuning) inflates performance estimates, reinforcing the need for proper separation of training, validation, and test sets. Third, we reveal how ignoring experimental block effects, such as seasonal or herd variations, introduces an upward bias in performance measures highlighting the importance of block CV when predictions are intended for new, previously unseen environment. Fourth, we highlight that different regression metrics — ranging from correlation-based to error-based (e.g., root mean squared error) — capture distinct aspects of predictive performance an under varying error biases and variances. Finally, for classification tasks, class imbalance and threshold settings significantly alter performance metrics, illustrating why a single metric rarely suffices to characterize model performance comprehensively. Collectively, these findings emphasize the

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

18 need for careful alignment between modeling objectives, CV strategies, and metric selection, thereby
19 ensuring trustworthy and generalizable model assessments in precision agriculture and beyond.

20 **Keywords** Model Evaluation · Performance Metrics · Simulation Studies

21 **1 Introduction**

22 **1.1 Modeling**

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

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

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

51 **1.2 Study Objectives**

52 This simulation study aims to highlight how biased or over-optimistic estimations of model performance usually come
53 from inappropriately conducting CV, a technique crucial for characterizing expected model performance on "new"

54 data. We demonstrate how common pitfalls, including using the exact data for both training and model assessment,
 55 excluding the model selection process from CV, and neglecting experimental block effects, contribute to challenges
 56 in model evaluation. Further, we scrutinize common metrics used in evaluating prediction models, including those
 57 used for regression and classification tasks. Because no single metric provides a comprehensive perspective of model
 58 performance, we seek, through this work, to highlight the importance of understanding the underlying theory of each
 59 metric to avoid misleading conclusions.

60 1.3 Cross Validation

61 The most common CV method is K-fold CV, which partitions the dataset into K equally sized folds. In each iteration,
 62 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
 63 (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
 64 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
 65 performance over all K folds is deemed as the expected generalized performance of the model $\mathbb{E}[\hat{g}]$ on new data.

66 However, there is always an evaluation bias between the estimated performance $\mathbb{E}[\hat{g}]$ and the true generalization
 67 performance G , which can only be approximated by evaluating the same model on an infinite number of unseen data.
 68 For example, when the evaluation metric is root mean squared error (RMSE), which decreases as the model's accuracy
 69 improves, a positive evaluation bias ($\mathbb{E}[\hat{g}] - G$) typically implies a pessimistic assessment of the model's performance.
 70 This is because the true error is likely lower than the estimated performance. Conversely, a negative evaluation bias
 71 indicates an optimistic assessment, suggesting that the model may produce larger errors on new data than estimated.

72 Another aspect of model evaluation error is the variance of each estimated performance \hat{g} across the K folds. For
 73 example, there are five estimates in a 5-fold cross-validation. The variance among these five estimates is defined as the
 74 evaluation variance. A high evaluation variance suggests that the performance is sensitive to the choice of data folds,
 75 and a small size or an over-complex model can lead to a high evaluation variance.

76 There is a trade-off relationship between evaluation bias and variance, which can be understood through the framework
 77 of the squared evaluation bias (see Appendix for a detailed derivation). When performing K-fold CV with a fixed
 78 sample size and model complexity, the choice of K is the pivotal element shaping the model evaluation. When the K is
 79 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
 80 of the population of interest, leading to lower bias. However, a large K comes with a trade-off: the corresponding test
 81 subset $\mathcal{D}_{\text{test}}$ is compressed in size, making the tested model more sensitive to the specific data points, and thus inflating
 82 the validation variance. Conversely, a smaller K, along with a minor training set $\mathcal{D}_{\text{train}}$, reduces the representativeness
 83 of each fold and increases bias. Nevertheless, a larger size of the test set $\mathcal{D}_{\text{test}}$ leads to more consistent estimations
 84 across the folds and, consequently, reduces the validation variance.

85 Leave-one-out cross-validation (LOOCV) is a variant of K-fold CV where K equals the sample size of the complete
 86 dataset \mathcal{D} . It provides an unbiased estimation of model performance because the training set $\mathcal{D}_{\text{train}}$ closely resembles the

87 unseen population of interest, given its size of $N - 1$, where N is the sample size. However, as the trade-off discussion
88 suggested, this method can lead to high validation variance due to the model being evaluated on one sample at a time.
89 The true unbiased nature of LOOCV is fully realized only when each individual data point is used sequentially for
90 evaluation. Performing an incomplete LOOCV can introduce significant bias because of the inherent high validation
91 variance, which often occurs when training each model iteration is prohibitively time-consuming or computationally
92 demanding. In specific contexts, such as genomic prediction, strategies like the one described by Cheng et al. leverage
93 the matrix inverse lemma, which allows for computational savings by avoiding the inversion of large matrices in each
94 fold. This technique significantly reduces the dependency of computational resources on the sample size [1]. Van
95 Dixhoorn et al. exemplify the use of LOOCV with a small dataset, aiming to predict cow resilience with limited data
96 resources [2]. Nevertheless, for large datasets, LOOCV is generally not recommended due to computational inefficiency.
97 Further details of bias-variance trade-off have been extensively explored in the statistical literature [3, 4].

98 **1.4 Model Selection**

99 Model selection becomes necessary when models are not entirely determined by the data alone. For example, in a
100 regularized linear regression model such as a ridge regression [5] or the least absolute shrinkage and selection operator
101 (LASSO) [6], it is essential to define a regularization parameter, λ , before fitting the model to the data. A larger λ value
102 yields a more regularized model, which tends to reduce smaller coefficients to negligible values or zero. This approach
103 helps in preventing overfitting noise in the training data. The definition of loss functions for the regularized models
104 were described in S.9 and S.10 of the Appendix.

105 These pre-defined parameters, like λ , influence model fitting and remain constant during the training process. Such
106 parameters are referred to as hyperparameters. Beyond regularized models, hyperparameters are crucial in other
107 predictive models, enhancing flexibility and robustness. For example, in the Support Vector Regression (SVR) [7],
108 the regressors X are projected onto a linear subspace to approximate the target variable y . By choosing a suitable
109 kernel function, which transforms the regressors into a non-linear space, as a hyperparameter, SVR can more effectively
110 capture non-linear relationships, thus significantly improving model performance. Another hyperparameter example is
111 the number of latent variables in the Partial Least Square (PLS) Regression [8], which condenses the original regressors
112 into a more manageable set of latent variables, reducing multicollinearity issues. Fewer latent variables might lose
113 significant information from the original regressors, while too many can lead to overfitting. Similarly, in Random
114 Forest [9], hyperparameters such as tree depth and the number of trees influence model complexity by dictating how
115 many feature splits are possible and how many weak learners comprise the ensemble. The same principle applies to
116 convolutional neural networks, where increasing the number of hidden layers or filter sizes can capture more complex
117 patterns in the data but also heightens the risk of overfitting [10]. All these examples highlight the fact that selecting the
118 most suitable hyperparameters, which is known as hyperparameter tuning, is crucial for optimizing model performance.
119 Feature selection is another crucial aspect of model selection. This process involves fitting the model to a selected
120 subset of the original features, particularly essential in high-dimensional data scenarios where the number of features

121 exceeds the number of observations, leading to poor model generalization. For instance, Zhang et al. used a feature
122 selection strategy to identify six of the most informative spectral bands for detecting weed species in hyperspectral
123 images containing 470 bands [11]. The reduction in the number of features effectively reduce the risk of collinearity
124 and overfitting, thereby improving model performance.

125 Optimizing feature subsets is a vital model selection strategy that significantly affects model performance. Therefore,
126 including the model selection process within the cross-validationz is essential to avoid common pitfalls. The risk of
127 inflated model performance arises when model selection is guided by results on the test dataset. Even if the chosen
128 model is subjected to k-fold cross-validation afterward, its selection bias toward the test set can lead to overestimating
129 its efficacy. This issue has been highlighted in statistical literature [3]. A practical solution is to divide the dataset into
130 training, validation, and test sets. The validation set is then used for model selection, ensuring the test set remains
131 completely unused during the training phase, thereby providing a more accurate measure of model performance.
132 For instance, the study by Rovere et al. exemplifies best practices in hyperparameter tuning and feature selection
133 by employing an independent cross-validation step prior to assessing model performance. This approach enabled
134 the precise selection of relevant spectral bands from the mid-infrared spectrum and the optimal number of latent
135 dimensions in PLS with Bayesian regression for predicting the fatty acid profile in milk [12]. Similarly, Becker et
136 al. demonstrated a robust evaluation by using nested cross-validation loops; the inner loop conducted a grid search
137 for the best hyperparameters in logistic regression, while the outer loop was designed to evaluate the performance
138 of the resulting optimized model [13]. Both examples underscore the importance of separating model selection from
139 performance evaluation to ensure the validity and reliability of the results.

140 **1.5 Cross Validation Design with Block Effects**

141 Blocking is an essential approach in experimental design to control for variations that can confound the variable of
142 interest. For instance, Lahart et al. investigated the dry matter intake of grazing cows using mid-infrared (MIR)
143 spectroscopy technology across multiple herds under varying experimental conditions [14]. Given the significant
144 variation between herds, which may contribute to individual differences in both dry matter intake (i.e., response variable)
145 and MIR spectra (i.e., independent variables), it is crucial to consider the herd as a blocking factor before evaluating the
146 predictability of dry matter intake using MIR spectra. This consideration should also extend to model evaluation. In the
147 cited study, variations in dry matter intake, the primary focus of the prediction model, were observed to exceed one
148 standard deviation among some herds. In cross-validation, if samples from the same herd are assigned to different folds,
149 with one fold used as the test set, the model is likely to achieve high accuracy. This accuracy may largely result from
150 explaining the inter-herd variation rather than individual variations in dry matter intake, leading to an overestimation of
151 model performance. To avoid this pitfall, block cross-validation, where each block (i.e., herd in this example) is used as
152 a fold, is recommended for unbiased model evaluation. Literature reviews have indicated that block cross-validation
153 effectively evaluates model performance on external or unseen datasets [15]. In the same study by Lahart et al., three
154 cross-validation strategies were compared: random cross-validation (Random CV), which randomly assigns samples

155 to folds; within-herd validation, training and testing the model within each herd; and across-herd validation (Block
 156 CV), where each herd is used as a fold and tested in turn. The results showed that performance estimates in block CV
 157 were noticeably lower than the other two strategies, supporting the hypothesis that ignoring block effects inflates model
 158 performance. Other studies considering block effects, including diet [16], herd [12], and farm location [17, 18], have
 159 shown similar results in cross-validation, demonstrating block CV's effectiveness in evaluating model performance on
 160 external datasets.

161 1.6 Model Performance Metrics

162 Model performance metrics serve as quantitative indicators for evaluating model performance. They are critical for
 163 benchmarking various modeling approaches and for evaluating hypotheses underpinning these different approaches.
 164 Choosing appropriate metrics to support hypothesis testing is crucial, as in-ideal selection may lead to overly optimistic
 165 conclusions. Due to the different goals of regression and classification tasks, it is critical to ensure that these different
 166 model types are evaluated using different metrics. As such, metrics for regression and classification are discussed
 167 individually.

168 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	$[0, \infty]$
Mean absolute error (MAE)	Error-based	No	$[0, \infty]$
Root mean squared percentage error (RMSPE)	Error-based	Yes	$[0, \infty]$
Root mean standard deviation ratio (RSR)	Error-based	Yes	$[0, \infty]$
Pearson's correlation coefficient (r)	Linearity-based	Yes	$[-1, 1]$
Coefficient of determination (R^2)	Linearity-based	Yes	$[-\infty, 1]$
Lin's concordance correlation coefficient (CCC)	Linearity-based	Yes	$[-1, 1]$

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

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

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1.1)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1.2)$$

177 where y_i and \hat{y}_i are the true and predicted values, respectively, and n is the sample size. Both metrics preserve the
 178 scale of the original data, making them easy to interpret in real-world units. Additionally, compared to MAE, RMSE
 179 penalizes large errors more due to the squared term, making it more sensitive to outliers. Monitoring animal body
 180 weight is a common practice to aid in the management of dairy cows. Studies by Song et al. and Xavier et al. have
 181 utilized RMSE to assess the effectiveness of three-dimensional cameras in estimating dairy cow body weight, yielding
 182 RMSE values of 41.2 kg and 12.1 kg, respectively [21, 22]. These figures provide a straightforward value for farmers to
 183 gauge whether the prediction error is tolerable, considering their specific operational costs and management thresholds.
 184 In essence, RMSE translates complex model accuracy into practical insights for productive agricultural units. When
 185 evaluating the same model across different traits, which may have different scales, a common practice is to express
 186 error metrics in a scale-free manner. This can be achieved by expressing RMSE as a percent of the deviation from the
 187 observed value, such as root mean squared percentage error (RMSPE), or as a Root Mean Standard Deviation Ratio
 188 (RSR) that normalizes the RMSE by the standard deviation of the observed values:

$$\text{RMSPE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2} \quad (1.3)$$

$$\text{RSR} = \frac{\text{RMSE}}{\sigma_y} \quad (1.4)$$

189 where σ_y is the standard deviation of the observed values. When expressed as a percent, RMSPE typically ranges from
 190 0 and above, with values closer to 0 indicating perfect prediction. Much like expressing RMSE as a percent, RSR is
 191 valuable to interpret RMSE in terms of the context of the variance in the observations. Values below 1 suggest that the
 192 model yields predictions less variable than the standard deviation, while values above 1 suggest that the prediction is
 193 imprecise.

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

$$\begin{aligned} r &= \frac{\text{cov}(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}} \\ &= \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \end{aligned} \quad (1.5)$$

$$\begin{aligned} R^2 &= 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}} \\ &= 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \end{aligned} \quad (1.6)$$

196 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 i th elements
 197 of the actual response vector y and the predicted response vector \hat{y} , respectively. \bar{y} and $\bar{\hat{y}}$ are their respective means.
 198 Both r^2 and R^2 are scale invariant, meaning their values are unaffected by the scale of the observed data because they
 199 are normalized by the variation in the denominator.

200 The correlation coefficient r measures the strength of the linear relationship between two continuous variables, y and \hat{y} ,
 201 and ranges from -1 to 1. A value of 0 indicates no prediction accuracy in the evaluated model. One special characteristic
 202 of correlation r is that it is unaffected by the scale of the predictions or biases; it focuses on the relative changes
 203 in the predicted values compared to the true values. Thus, even if the prediction biases are scaled up or down, the
 204 correlation r between \hat{y} and y remains the same. This property is particularly useful when the focus is more on ranking
 205 predictions rather than their absolute values. For example, this metric has been used to evaluate models that identify
 206 high-performing production individuals, demonstrating the ability to predict nutrient digestibility in dairy cows [26] and
 207 to select models based on their ability to rank traits such as feed intake and milk composition in dairy cows [27, 12].

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

215 It worth noting that much of the existing literature has misinterpreted the relationship between r and R^2 . The coefficient
 216 of determination R^2 is not always equivalent to the square of the correlation coefficient r^2 . The equivalence only holds
 217 when the same dataset is used for both model fitting and evaluation in a least squares regression model. The model
 218 assumes a zero covariance between the fitted residual and the predicted values \hat{y} , and it also assumes that the residuals
 219 (i.e., prediction biases) are centered on zero. In practice when predictions are made on new data, those assumptions
 220 are often violated, leading to discrepancies between r^2 and R^2 . A details derivation of the equivalence is provided in
 221 Equation S.11–S.12 in the Appendix.

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

$$\begin{aligned} \text{CCC} &= \frac{2r\sigma_y\sigma_{\hat{y}}}{\sigma_y^2 + \sigma_{\hat{y}}^2 + (\bar{y} - \bar{\hat{y}})^2} \\ &= \frac{2\text{cov}(y, \hat{y})}{\sigma_y^2 + \sigma_{\hat{y}}^2 + (\bar{y} - \bar{\hat{y}})^2} \end{aligned} \tag{1.7}$$

223 where r is the Pearson correlation coefficient. The CCC is a comprehensive metric because it considers both the
 224 correlation and the scale bias between the predicted and true values. It fills the gap left by r^2 where the scale bias is
 225 ignored. Geometrically, CCC measures how well the predicted values \hat{y} fall on the 45-degree line in a scatter plot of

226 the predicted (x-axis) and true values (y-axis). It is advantageous over R^2 because it consistently ranges from -1 to 1,
 227 making it easier to interpret and compare across different studies. The CCC is crucial when accurate predictions are
 228 required for both the scale and the rank of the trait of interest, such as in studies predicting cotton crop yields based on
 229 soil and terrain profiles [29].

230 **1.6.2 Metrics in Classification Tasks**

231 Classification models aim to predict categorical outcomes such as 'healthy' or 'sick,' 'susceptible' or 'resistant,' and
 232 'high yield' or 'low yield.' To evaluate classification performance, one must first establish a confidence threshold to
 233 dichotomize the prediction probabilities. For instance, if a classification model predict a sample as 'sick' with a 0.7
 234 probability, and the threshold is set at 0.5. Since the 0.7 prediction probability exceeds the threshold, the sample is
 235 predicted as a positive sample. It is worth mentioning that this threshold is adjustable to fine-tune model performance
 236 for particular focus, such as minimizing false positives or false negatives. All classification metrics are derived from
 237 the confusion matrix, which summarizes the model's performance in a 2x2 table, where the rows represent the actual
 238 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)

239 The confusion matrix (Table 2) consists of four components: true positives (TP), true negatives (TN), false positives
 240 (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

241 The metrics can be characterized by two key factors: their denominator and their focus on either correctness or error.
 242 Understanding the denominator of a metric helps clarify its scope of interest. For instance, if one wants to evaluate

243 how well the model correctly predicts positive samples, metrics that use actual positives as the denominator should be
 244 prioritized. It is noted that in Table 3, the metrics are organized in four subsections. The metrics in the first subsection
 245 have self-explanatory names, each emphasizing a specific aspect of the model’s performance:

$$\text{True positive rate (TPR)} = \text{Sensitivity}$$

$$= \text{Recall} \quad (1.8)$$

$$= \frac{\text{TP}}{\text{Total Actual Positives}}$$

$$\text{True negative rate (TNR)} = \text{Specificity}$$

$$= \frac{\text{TN}}{\text{Total Actual Negatives}} \quad (1.9)$$

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

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

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

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

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

256 The third section includes precision and recall, which focus on different aspects of positive cases. Machine learning
 257 community used to report precision and recall together, as the community focus more on the positive samples than the
 258 negative samples. For example, in computer vision applications, how well a model can correctly classify and localize
 259 the object of interest (positives) from an image is more important than how well the model can correctly know what area
 260 is irreleavnt background (negatives). Precision evaluates the correctness of the predicted positive cases, ensuring that
 261 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. 1.10). 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.

$$\text{Precision} = \frac{\text{TP}}{\text{Total Predicted Positives}} \quad (1.10)$$

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.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Total Correct Predictions}}{\text{Total Predictions}} \\ &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \end{aligned} \quad (1.11)$$

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:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1.12)$$

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 :

$$\text{F-beta} = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \quad (1.13)$$

283 A common variant is the F2 score, which places more emphasis on false negatives (i.e., recall) than false positives, by
 284 setting $\beta = 2$:

$$F2 = 5 \times \frac{\text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}} \quad (1.14)$$

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

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (1.15)$$

287 The equation 1.15 symmetrically incorporates all four components of TP, TN, FP, and FN). This symmetry makes
 288 MCC invariant to class distribution changes. The coefficient ranges from -1 to 1, where 1 indicates perfect classification,
 289 0 indicates no better performance than random guessing, and -1 signifies total disagreement between prediction and
 290 observation. In a case study that used feeding and daily activity behaviors to diagnose Bovine Respiratory Disease
 291 in dairy calves, MCC proved particularly insightful [37]. The models in this study exhibited strong performance on
 292 negative samples (i.e., healthy calves), which were more prevalent, resulting in high specificity. However, sensitivity
 293 was relatively low at 0.54. In this context, MCC, with a value of 0.36, provided a more nuanced and representative
 294 measure of model performance, especially given the skew towards negative samples

295 1.7 Simulation experiments to Understand Model Evaluation Pitfalls

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

308 **2 Materials and Methods**

309 **2.1 Study datasets**

310 This study utilized three datasets to demonstrate the common challenges in model evaluation: A null dataset, a simulated
 311 spectral dataset, and a real-world spectral dataset.

312 **2.1.1 Null dataset**

313 The null dataset serves as a baseline for the null hypothesis, designed to evaluate the risk of introducing bias in the
 314 estimation of model performance. In this dataset, the predictors X and the target variable y are independently drawn
 315 from the same normal distribution, ensuring no linear or nonlinear relationship between the input features and the target
 316 variable:

$$\begin{cases} X \sim \mathcal{N}(0, 1) \\ y \sim \mathcal{N}(0, 1) \end{cases} \quad (2.1)$$

317 If any model evaluation exercise applied to this dataset produces a significant performance metric, it would indicate a
 318 potential bias in the evaluation process. This serves as a critical check to ensure that the evaluation methodology does
 319 not artificially inflate the perceived performance of the model.

320 **2.1.2 Simulated spectral dataset**

321 To further investigate how these identified challenges impact data with complex structures, both simulated and real
 322 spectral datasets were utilized. Spectral data is commonly encountered in agricultural studies, where the target variable
 323 is predicted using a series of spectral measurements. This type of data serves as an excellent example for this study
 324 because it often presents a significant challenge due to the strong collinearity among predictors. Effectively selecting
 325 predictors with reduced correlation is essential to mitigate overfitting and improve model robustness. The simulated
 326 spectral dataset was generated following the procedure outlined in [38], which characterizes spectral signals X as the
 327 outcome of a linear combination of a score matrix T and a loading matrix P :

$$X_{n \times m} = T_{n \times k} P_{m \times k}^\top + E_{n \times m} \quad (2.2)$$

328 where X is the spectral data matrix with n samples and m spectral variables, T is the score matrix with n samples
 329 and k latent variables, P is the loading matrix with m variables and k latent variables, and E is the residual matrix to
 330 simulate noise (Figure 1). The k latent variables represent the underlying structure of the spectral data, such as the
 331 peaks and valleys in the spectrum. In this study, the spectral data consists of 300 spectral bands ($m = 300$) and four
 332 latent variables ($k = 4$). Among these latent variables, only the first two are assumed to contribute meaningfully to the
 333 target variable, while the other two are treated as noise.

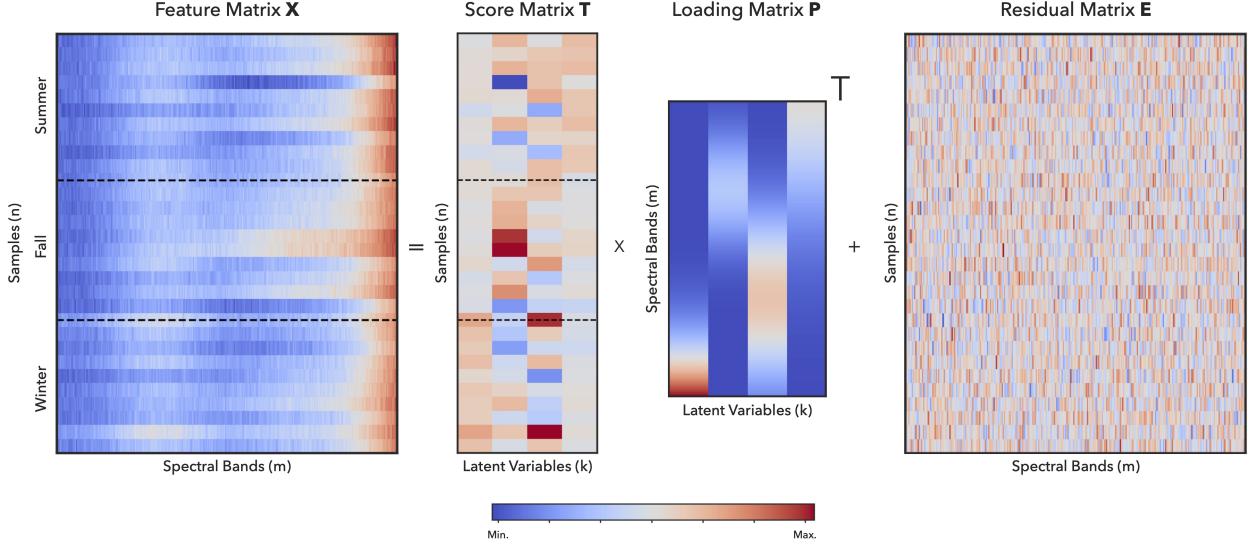


Figure 1: Matrix decomposition of the simulated spectral data. The spectral data matrix X is generated as a linear combination of the score matrix T and the loading matrix P , with added noise E . The color scale is independently normalized for each matrix.

- 334 The score matrix T defines how each sample contributes to each latent variable. For example, if the third sample
 335 exhibits higher spectral measurements around the first peak (as defined by the first latent variable), the value in the
 336 third row and first column of the score matrix will be higher relative to other rows. In this study, T was sampled from
 337 a multivariate normal distribution with a mean vector of $[1, 1, 1, 1]$ and standard deviations of $[0.02, 0.10, 0.10, 0.02]$.
 338 This setup reflects a scenario where the second and third latent variables (corresponding to specific peaks) are more
 339 pronounced compared to the first and fourth latent variables. It is inspired by the spectrum measured in the past work
 340 [39], which used 150 hyperspectral bands ranging from 1,000 nm to 1600 nm to evaluate the wheat kernel quality trait.
 341 The loading matrix P defines how each spectral variable contributes to each latent variable. Each latent variable in P
 342 was simulated using a Gaussian probability function with peaks at the -30th, 90th, 200th, and 345th spectral positions
 343 and standard deviations of $[100, 40, 60, 60]$ to simulate the width of the peaks. Negative peak positions simulate signals
 344 outside the measured spectral range. The residual matrix E is sampled from a normal distribution $\mathcal{N}(0, 0.01)$ to
 345 simulate the noise in the spectral data.
- 346 Seasonal variation is an important factor in agricultural studies and is often overlooked in model evaluation. To
 347 incorporate this effect, the spectral measurements were simulated across three seasons, with random effects applied
 348 to the latent variables. These seasonal effects were modeled by multiplying different scalars with the latent variables
 349 in the score matrix T . The scalars for the first two latent variables were $[1.00, 1.10, 1.07]$, and for the latter two were
 350 $[1.07, 1.00, 1.00]$. This setup reflects a scenario where the first two latent variables are more pronounced in the second
 351 and third seasons, while the latter two latent variables dominate in the first season.
- 352 The response variable y was generated as a nonlinear function of selected spectral variables. Specifically, four spectral
 353 bands (B) were selected at indices $[50, 100, 180, 230]$ from the 300 spectral bands. Nonlinear effects were introduced by

354 applying a sinusoidal transformation to the selected spectral variables, raised to the power of three (cubic nonlinearity).
 355 To ensure that each effect has a unique sinusoidal component, a phase shift was added to each effect. The phase shift is
 356 defined as $\frac{i\pi}{m}$, where i is the index of the selected spectral band, and m is the total number of selected spectral bands.
 357 This approach introduces variation in the sinusoidal behavior of each effect, ensuring they are distinct:

$$y = \sum_{i=1}^m \sin(b_i^3 + \frac{i\pi}{m}), \quad b_i \in B$$

358 where b_i represents the i -th selected spectral band from the four bands B . Finally, Gaussian noise was added to the
 359 response variable y to simulate measurement or modeling errors. The noise was generated with a standard deviation
 360 equal to that of the response variable y , simulating a scenario where only 50% of the variance in y can be explained by
 361 the spectral data. This approach introduces realistic variability, reflecting the inherent uncertainties and complexities
 362 often encountered in real-world prediction tasks.

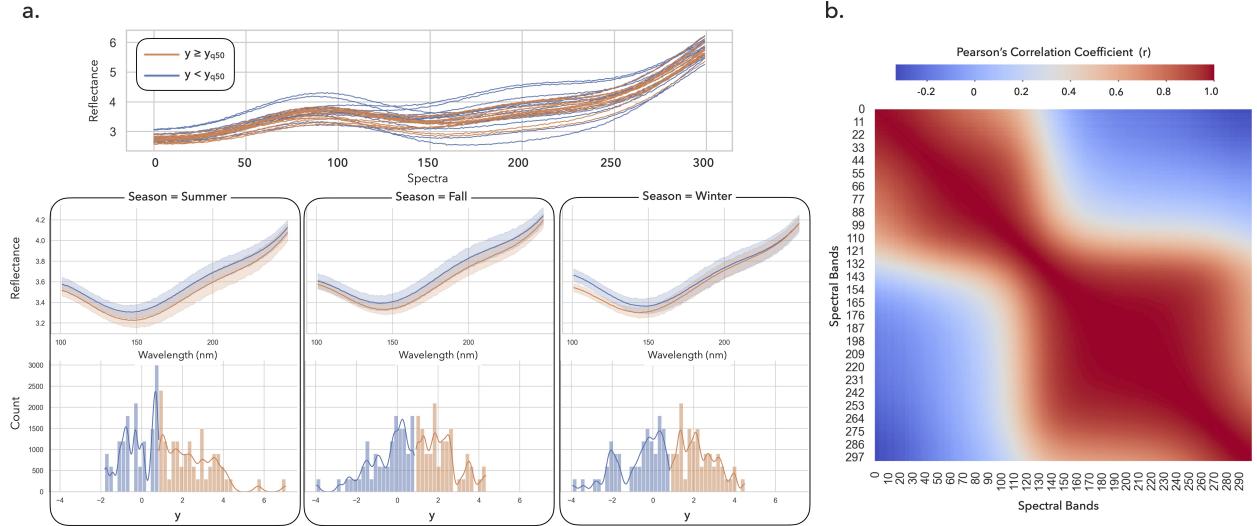


Figure 2: Overview of the simulated spectral dataset. (a) The spectral data matrix X is visualized with the target variable y categorized by its median value. (b) The autocorrelation plot of the spectral data matrix X shows a bi-modular structure with the least pair-wise correlation around the 100th band.

363 The simulated spectral data exhibit a bi-modular autocorrelation structure, with the least pair-wise correlation ($r=0.4$)
 364 observed near the 100th band, which serves as the cutoff between the two modules (Figure 2b). The non-linear
 365 relationship between the spectral data and the target variable y is evident when y is categorized by its median value
 366 (y_{q50}) and visualized in two color groups within the spectral space (Figure 2a). The resulting plot reveals that the
 367 data are not linearly separable in the spectral space, confirming the expected complexity and presenting a challenging
 368 task for model evaluation. Additionally, seasonal effects simultaneously influence both the spectral data and the target
 369 variable. For instance, the spectral reflection is less pronounced around the 150th band during the first season, while
 370 the separability of the two categorical groups decreases around the 250th band in the third season. Furthermore, the
 371 distribution of the target variable varies across seasons: the first season displays a right-skewed distribution, whereas

372 the other two seasons have more symmetric distributions. These seasonal variations introduce additional layers of
 373 complexity, further highlighting the importance of robust evaluation methods for classification models.

374 2.1.3 Real-world spectral dataset

375 This dataset contains spectral data collected across 18 bands ranging from 410 nm to 940 nm, aimed at assessing forage
 376 quality. The spectral data were captured using a SparkFun ESP32 Thing Plus microprocessor paired with a SparkFun
 377 Triad Spectroscopy Sensor (SparkFun Electronics, Niwot, CO). The sensor suite was programmed using Arduino IDE
 378 v2.0.4 (Arduino Core Team, 2024) to export measurements.

379 Forage quality was quantified based on neutral detergent fiber (NDF) content, a critical parameter for evaluating
 380 livestock nutrition. Ground truth NDF values were determined using traditional bench chemistry methods with the
 381 ANKOM 200 fiber analyzer system (ANKOM Technology, Macedon, NY). The dataset comprises 599 samples collected
 382 over three distinct time periods reflecting the seasonal effects on both the spectral data and the NDF response: 189
 383 samples were collected from May to June, 198 from July to August, and 212 from September to October. Sampling
 384 took place weekly between May 1 and October 30, 2023, with two samples collected each week from each of 12 fields.
 385 Within each field, sampling locations were chosen at random and varied from week to week.

386 The spectral data were collected in the field using a handheld sensor, while NDF content was measured in the lab. The
 387 fields were primarily grazed by cattle, with some fields grazed by other species, including sheep and horses. This
 388 dataset provides a comprehensive view of how seasonal variation influences forage quality and spectral characteristics,
 389 offering valuable insights into the dynamics of pasture composition and livestock nutrition.

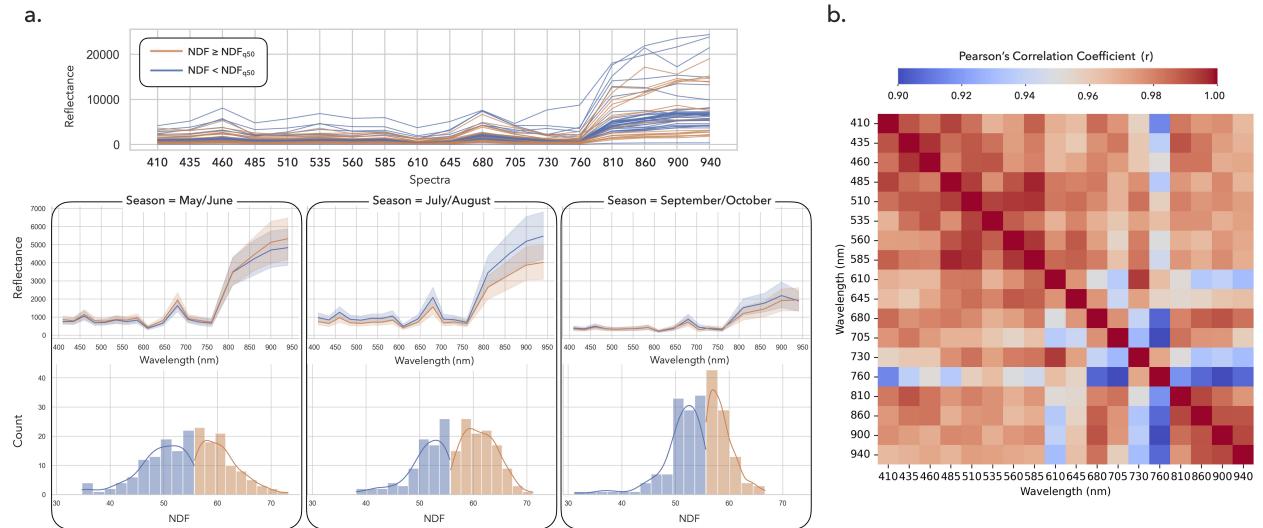


Figure 3: Overview of the real spectral dataset. (a) The spectral data matrix X is visualized with the neutral detergent fiber (NDF) y categorized by its median value. (b) The autocorrelation plot of the spectral data matrix X .

390 A similar examination of the data structure was conducted for both the spectral measurements and the target variable
 391 (Figure 3). The autocorrelation in the spectral measurements is notably stronger than in the simulated dataset, with at

392 least 0.90 for pairwise Pearson correlation coefficients. The seasonal interactions among the spectral measurements are
393 also more pronounced compared to the simulated data. For instance, the spectral reflectance measured in September
394 and October is roughly halved compared to the other seasons. Additionally, a distinct seasonal pattern was observed in
395 the reflectance data beyond 800 nm. In July and August, samples with lower NDF value tend to exhibit higher spectral
396 reflectance, whereas this trend is not evident in May–June or September–October. Moreover, the NDF distribution
397 shows greater variability in July and August, with a higher standard deviation (7.15) compared to May–June (6.19)
398 and September–October (5.23). These observations highlight the stronger seasonal effects and variability in the real
399 dataset compared to the simulated data, providing another example of the challenges in evaluating model performance
400 in real-world agricultural studies.

401 **2.2 Experiment 1: Evaluation bias and variance of cross-validation**

402 This experiment examined the reliability of CV in estimating model performance, with a focus on different performance
403 estimators and their interaction with sample size. It is hypothesized that increasing the number of folds in CV will
404 generally provide a more accurate estimate of model performance but will also lead to increased variance in each
405 estimate, as suggested by the bias-variance trade-off theory. Additionally, sample size is considered a critical factor in
406 reducing the bias difference between estimators, with larger sample sizes expected to mitigate the impact of estimator
407 bias and improve the reliability of performance evaluation.

408 Since K-fold CV employs a fraction (i.e., $K - 1$ folds) of the data for training, it may provide a pessimistic estimate
409 of model performance. Such underestimation is explored in this experiment by comparing the performance metrics
410 of 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"
411 evaluation, which assesses model performance on the same dataset used for training, potentially leading to an overly
412 optimistic bias. To gauge model performance, four metrics are employed: RMSE (Eq. 1.1), MAE (Eq. 1.2), r (Eq.
413 1.5), and R^2 (Eq. 1.6). The evaluation model is a linear regression with ten input features and one output target, all
414 drawn from the null dataset. The sample sizes N are varied among 50, 250, and 500 to explore the dynamics between
415 sample size and performance estimators. Each configuration is repeated across 500 iterations to assess the distribution
416 of evaluation bias and variance.

417 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
418 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
419 does not occur. The linear model f is trained on the training set \mathcal{D}_{-k} (denoted as $f_{\mathcal{D}_{-k}}$) to estimate regression coefficients
420 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:

$$\begin{aligned}
\text{Training: } & y_{-k} = f_{\mathcal{D}_{-k}}(X_{-k}) + \epsilon \\
& = X_{-k}\beta + \epsilon \\
\text{Testing: } & \hat{y}_k = f_{\mathcal{D}_{-k}}(X_k) \\
& = X_k\beta \quad k = 1, 2, \dots, K
\end{aligned} \tag{2.3}$$

421 For the “In-Sample” performance estimator, predictions were made without splitting, as:

$$\begin{aligned}
\text{Training: } & y = f_{\mathcal{D}}(X) \\
& = X\beta + \epsilon \\
\text{Testing: } & \hat{y} = f_{\mathcal{D}}(X) \\
& = X\beta
\end{aligned} \tag{2.4}$$

422 Where:

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

430 Estimated performance $\mathbb{E}[\hat{y}(f_{\mathcal{D}})]$ was derived by averaging the performance metrics across all K folds as per Eq. S.4.
431 The bias and variance of the evaluation were calculated using Eqs. S.5 and S.6, respectively. To approximate true
432 model performance $G(f_{\mathcal{D}})$, a hundred unseen datasets \mathcal{D}^* were generated identically to \mathcal{D} , and the performance $G(f_{\mathcal{D}})$
433 was estimated by averaging the performance metrics across all \mathcal{D}^* . The detailed steps to compute evaluation bias and
434 variance are provided in the supplementary materials.

435 2.3 Experiment 2: Model Selection in Cross-Validation

436 The objective of this simulation experiment is to investigate the impact of improper model selection practices on
437 evaluation bias. Two critical steps in the model selection process are considered: feature selection and hyperparameter
438 tuning. The experiment hypothesizes that improper model selection — particularly the leakage of test set information
439 during feature selection or hyperparameter tuning — will result in a significant overestimation of model performance.

440 To evaluate this hypothesis, three datasets are utilized: a null dataset with a baseline performance of $r = 0$, a simulated
 441 spectral dataset, and a real spectral dataset. Feature selection is conducted by selecting the top 10 features most strongly
 442 correlated with the target variable, y . The original number of feature candidates varies across datasets, with 1000 for
 443 the null dataset, 300 for the simulated spectral dataset, and 18 for the real spectral dataset.

444 For hyperparameter tuning, the experiment employs a Support Vector Regression (SVR) model with two hyperparam-
 445 eters: the kernel function and the regularization parameter (c). The kernel functions considered are linear, sigmoid,
 446 and radial basis function (RBF), while the regularization parameter is set to two values: $c = 1.0$ and $c = 0.01$. The
 447 kernel function determines how the selected features are transformed — either linearly or nonlinearly — to predict the
 448 target variable, y . The regularization parameter c controls the trade-off between minimizing prediction error and model
 449 complexity; a larger c allows for more error but reduces the likelihood of overfitting. In total, six SVR model variants
 450 (i.e., three kernel functions combined with two regularization parameter values) are available for selection during the
 451 evaluation process.

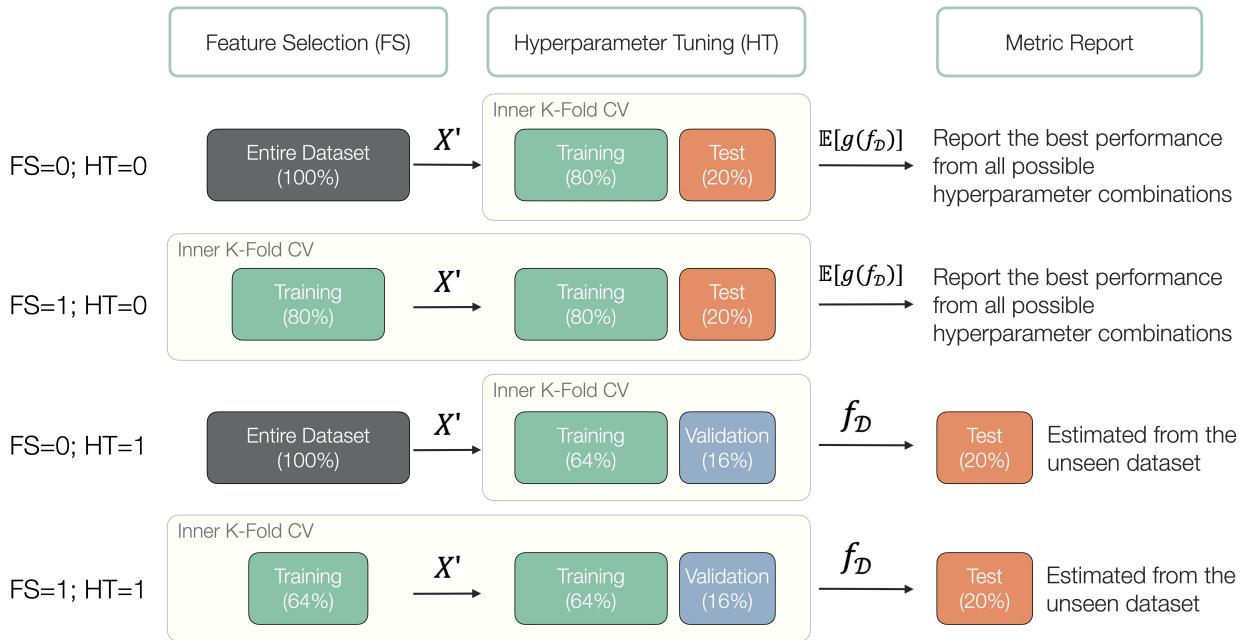


Figure 4: Workflow diagram illustrating four cross-validation strategies of feature selection (FS) and hyperparameter tuning (HT), where 0 denotes incorrect implementation and 1 indicates correct practice. X' is the selected feature subset, $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$ is the expected generalization performance, $f_{\mathcal{D}}$ is the model trained on the training set without being revealed to the test set.

452 This experiment introduces notations FS for feature selection and HT for hyperparameter tuning, assigning a binary
 453 indicator (0 or 1) to denote incorrect (0) or correct (1) implementation of model selection. This yields four possible
 454 combinations of model selection strategies: “FS=0; HT=0”, “FS=0; HT=1”, “FS=1; HT=0”, “FS=1; HT=1” (Figure 4).
 455 When FS=0, feature selection precedes cross-validation splitting. If FS=1, feature selection occurs within each fold
 456 of the training set during cross-validation. With hyperparameter tuning, a correct implementation (HT=1) involves
 457 splitting the dataset into training (64%), validation (16%), and test (20%) sets. The model is trained and tuned using the

458 training and validation sets, respectively, while the test set is reserved for a single evaluation of model performance.
 459 Conversely, with HT=0, only training (80%) and test (20%) sets are used, risking evaluation bias as the test set informs
 460 both training and performance reporting. A 5-fold cross-validation approach was deployed for all strategies. evaluation
 461 bias is measured as the discrepancy between the model selection-influenced performance estimate and the expected
 462 generalization performance ($r=0$), using the Pearson correlation coefficient between predicted and observed values. Over
 463 500 sampling iterations, the experiment assesses the distribution of evaluation bias. A two-way analysis of variance
 464 (ANOVA) was conducted to examine the main effects of HT and FS on model performance. The ANOVA model was
 465 specified as $y \sim 1 + HT + FS$. This experimental setup is designed to quantify the extent of performance overestimation
 466 under improper model selection practices and provide insights into its implications for predictive modeling.

467 **2.4 Experiment 3: Block Effects in Cross-Validation**

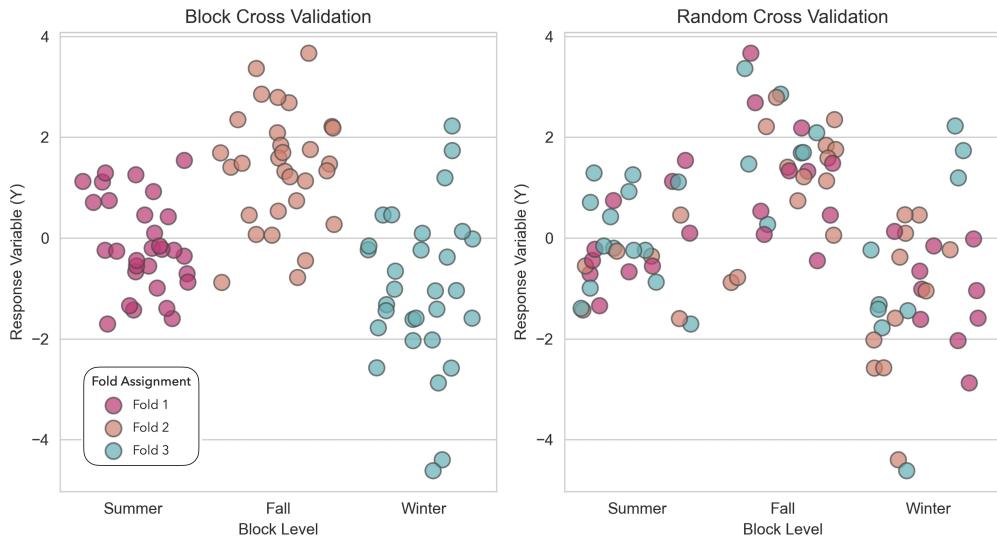


Figure 5: Illustration of fold assignment in block cross validation (left) and random cross validation (right). Folds are color-coded, and the block effect is set to 3 in this example.

468 The objective of this experiment is to demonstrate how Random CV, which randomly assigns samples to folds without
 469 accounting for block effects, can lead to an overestimation of model performance. As a benchmark, the experiment
 470 employs Block CV, where each block is treated as a fold in cross-validation. The hypothesis is that the model
 471 performance estimated by Random CV will be significantly higher than that estimated by Block CV.
 472 This experiment utilizes both simulated and real-world datasets, both of which were collected across multiple seasons,
 473 introducing block effects that confound both the predictor features and the response variable. The simulated dataset
 474 includes 200 observations per season, distributed equally across seasons, while the real-world dataset also contains
 475 approximately 200 observations per season. The block effect in both datasets is defined by the seasonal variation.
 476 The experiment evaluates two model validation strategies: Block CV and Random CV, both using a 3-fold cross-
 477 validation approach. Three folds are used to match the number of seasons in the dataset. In Block CV, each block (i.e.,

478 season) is treated as a distinct fold, ensuring that samples from the same block are not split across folds. In Random CV,
479 samples are randomly assigned to folds without consideration of block boundaries (Figure 5). The predictive model
480 used is a random forest regression model, and its performance is assessed using Pearson’s correlation coefficient r and
481 RMSE.

482 The simulation is run for 500 iterations, with X (predictor variables) and Y (response variable) resampled in each
483 iteration for the simulation dataset and also the fold assignment to account for variability. A one-way ANOVA was
484 performed to evaluate whether the choice of CV strategy (i.e., block CV vs. random CV) significantly affects model
485 bias. The model was specified as $y \sim 1 + \text{BlockCV}$,

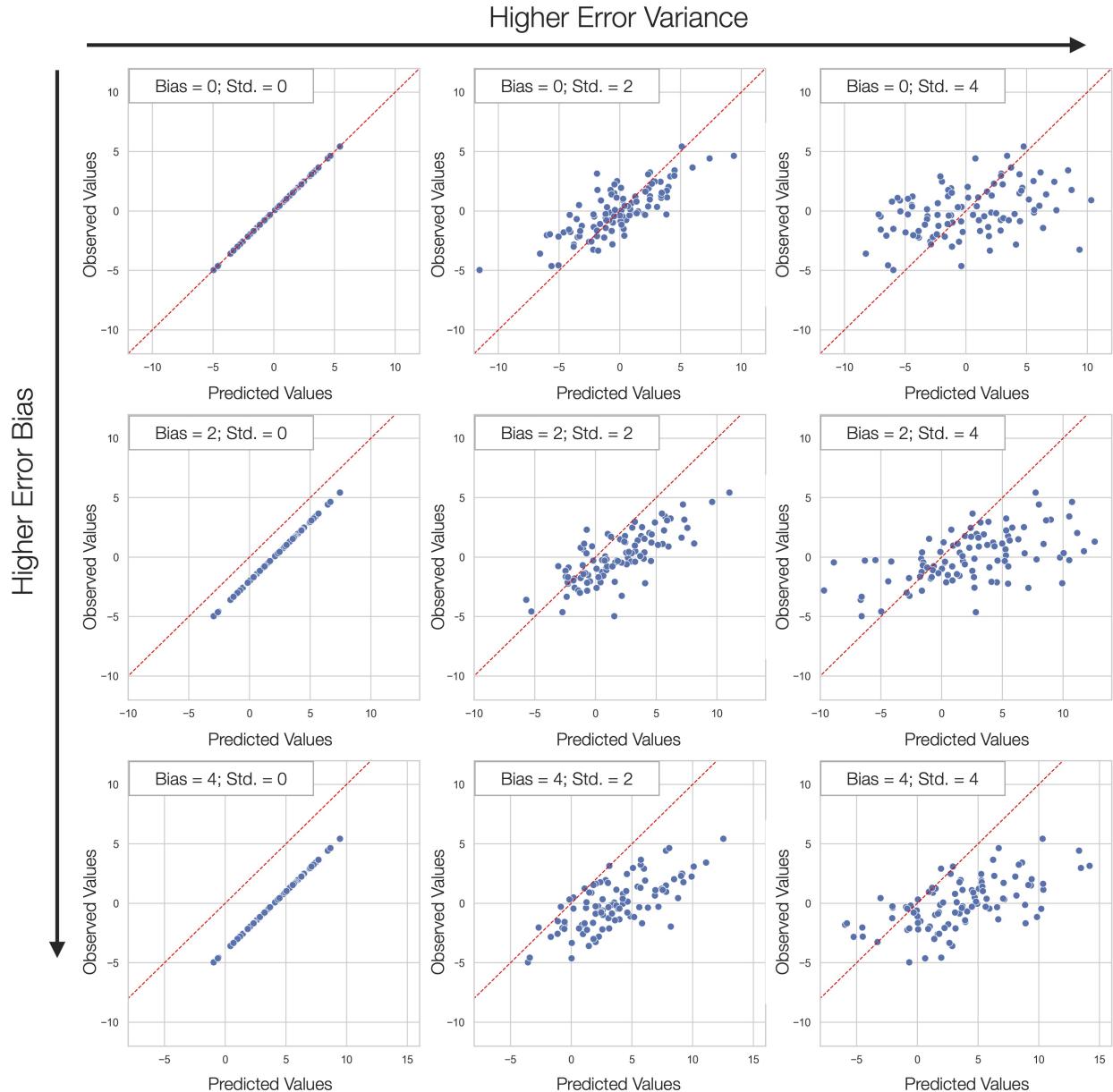
486 **2.5 Experiment 4: Characteristics of Metrics in Regression Tasks**

Figure 6: Scatter plots illustrating the relationship between predicted and actual values for 9 combinations of bias and variance, with each parameter set to one of three levels: 0, 2, or 4. The red diagonal line represents the ideal prediction line.

487 The objective of this experiment is to examine how different performance metrics in regression tasks respond to two
 488 types of prediction errors: bias and variance. The experiment aims to highlight the unique characteristics of each metric,
 489 such as sensitivity to outliers or systematic bias, and provide guidance for selecting appropriate metrics in regression
 490 tasks. Additionally, it seeks to verify the trade-off relationship between bias and variance in prediction errors.

491 To achieve this, six levels of bias and variance are examined: [0, 0.5, 1, 2, 4, 8], forming a total of 36 combinations of
 492 prediction errors. The bias error can be considered as a systematic error that consistently overestimates or underestimates
 493 the ground truth values, while the variance error represents the random fluctuations around the ground truth. Outliers of
 494 prediction errors are considered as extreme cases of variance errors, where the predicted values deviate considerably
 495 from the ground truth. The simulated ground truth values are generated from a normal distribution with a mean of 0 and
 496 a standard deviation of 2, while the predicted values (\hat{y}) are created by adding random errors (ϵ) with specified levels of
 497 bias and variance to the ground truth:

$$\begin{cases} y \sim \mathcal{N}(0, 2) \\ \epsilon \sim \mathcal{N}(b, s) \\ \hat{y} = y + \epsilon \end{cases} \quad (2.5)$$

498 where b represents the bias and s represents the variance of the prediction errors (Figure 6). The choice of a standard
 499 deviation of 2 for the ground truth values is intended to highlight the differences in behavior between the RMSE and
 500 RSR metrics, with RSR being standardized by the standard deviation of the ground truth values while RMSE tracks the
 501 original error scale.

502 The evaluated metrics in this experiment are categorized into two main groups. Error-based metrics include RMSE,
 503 MAE, and RSR. These metrics focus on quantifying the magnitude of errors in the predictions. Linearity-based metrics,
 504 such as r , R^2 , and CCC, assess the linear relationship and agreement between the predicted and actual values. By
 505 systematically exploring how each metric responds to varying levels of bias and variance, this experiment demonstrates
 506 their strengths, limitations, and practical implications for regression analysis. The findings are intended to guide
 507 practitioners in selecting the most appropriate performance metrics based on their specific modeling objectives and the
 508 characteristics of their data.

509 **2.6 Experiment 5: Characteristics of Metrics in Classification Tasks**

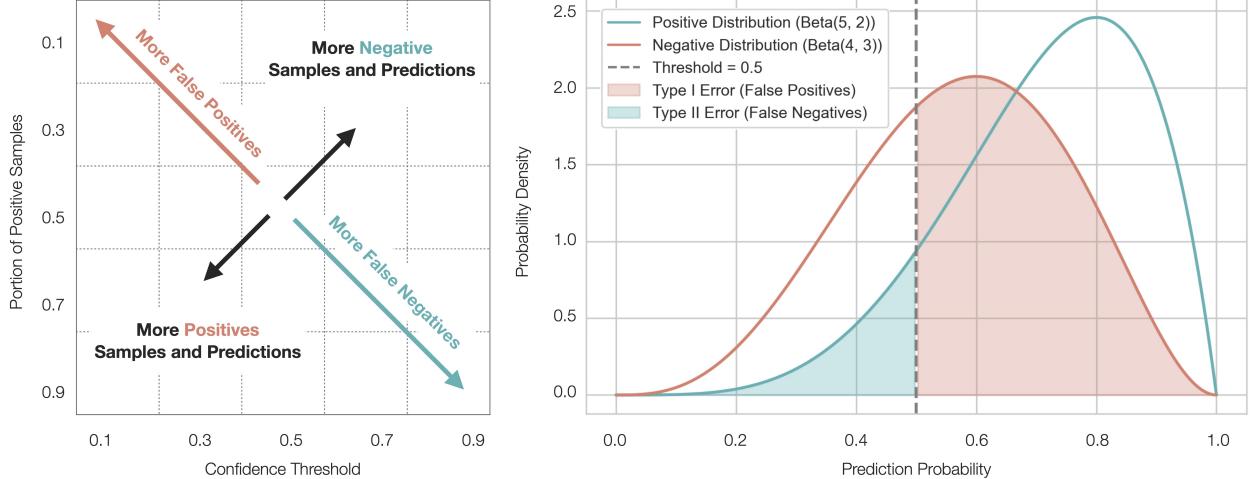


Figure 7: Illustration of the simulation design for evaluating classification metrics under varying balance and confidence thresholds. (a) A 5×5 grid of performance metrics, with each cell representing a unique combination of balance and confidence threshold. (b) The prediction probability distribution for positive and negative samples.

510 The objective of this experiment is to investigate two critical aspects of evaluating binary classification models: the
 511 balance between positive and negative samples and the choice of confidence threshold. The experiment aims to explore
 512 how these factors influence the performance metrics used to evaluate classification models and to provide insights into
 513 the necessity of reporting specific metrics together.

514 To achieve this, five levels of balance and five levels of confidence thresholds are examined, forming a total of 25
 515 combinations. The inspected levels are $[0.1, 0.3, 0.5, 0.7, 0.9]$ for both balance and confidence thresholds. The balance
 516 level is determined by the proportion of positive samples, with a balance level of 0.9 indicating that 90% of the samples
 517 are positive. The confidence threshold is used to dichotomize prediction probabilities, where a higher threshold results
 518 in fewer positive predictions by requiring higher certainty for a positive classification. This simulation produces a 5×5
 519 grid of performance metrics, where each cell represents a unique combination of balance and confidence threshold
 520 (Figure 7a). The top-left corner of the grid corresponds to a scenario where positive samples are rare, and the model
 521 uses a low confidence threshold, resulting in a high false positive rate. In contrast, the bottom-right corner represents a
 522 scenario where positive samples are abundant, and the model applies a high confidence threshold, leading to a high
 523 false negative rate. This design contrasts these two extreme cases and hence provides a comprehensive evaluation of
 524 performance metrics across varying conditions.

525 The prediction probability, which represents the likelihood of a given sample being classified as positive by the model,
 526 is simulated using a beta distribution (Figure 7b). For positive samples, the prediction probability is drawn from a beta
 527 distribution with parameters $\alpha = 5$ and $\beta = 2$, resulting in a peak probability around 0.8.
 528 For negative samples, the beta distribution has parameters $\alpha = 4$ and $\beta = 3$, with a peak probability around 0.6. This
 529 design creates a scenario where more false positives are expected than false negatives, as the negative samples have high

530 prediction probabilities that overlap with the positive samples. The shaded area in Figure 7b represents the overlap
531 between the two distributions, where the confidence threshold is applied. This overlap introduces two types of errors:
532 the region to the right of the threshold intersecting with the negative distribution represents false positives (Type I error),
533 while the region to the left of the threshold intersecting with the positive distribution represents false negatives (Type II
534 error). This setup highlights the trade-off between these error types based on the choice of the confidence threshold.
535 The metrics evaluated in this experiment include TPR, TNR, FPR, FNR, sensitivity, specificity, precision, recall,
536 accuracy, F1 score, F2 score, and MCC. Although some of these metrics are mathematically equivalent but referred to
537 by different names, this experiment also highlights the reasons why certain metrics are commonly reported together and
538 discusses their complementary roles in evaluating model performance.

539 **3 Results and Discussion**

540 **3.1 Experiment 1: The Impact of Estimator Choice and Sample Size on Model Evaluation Reliability**

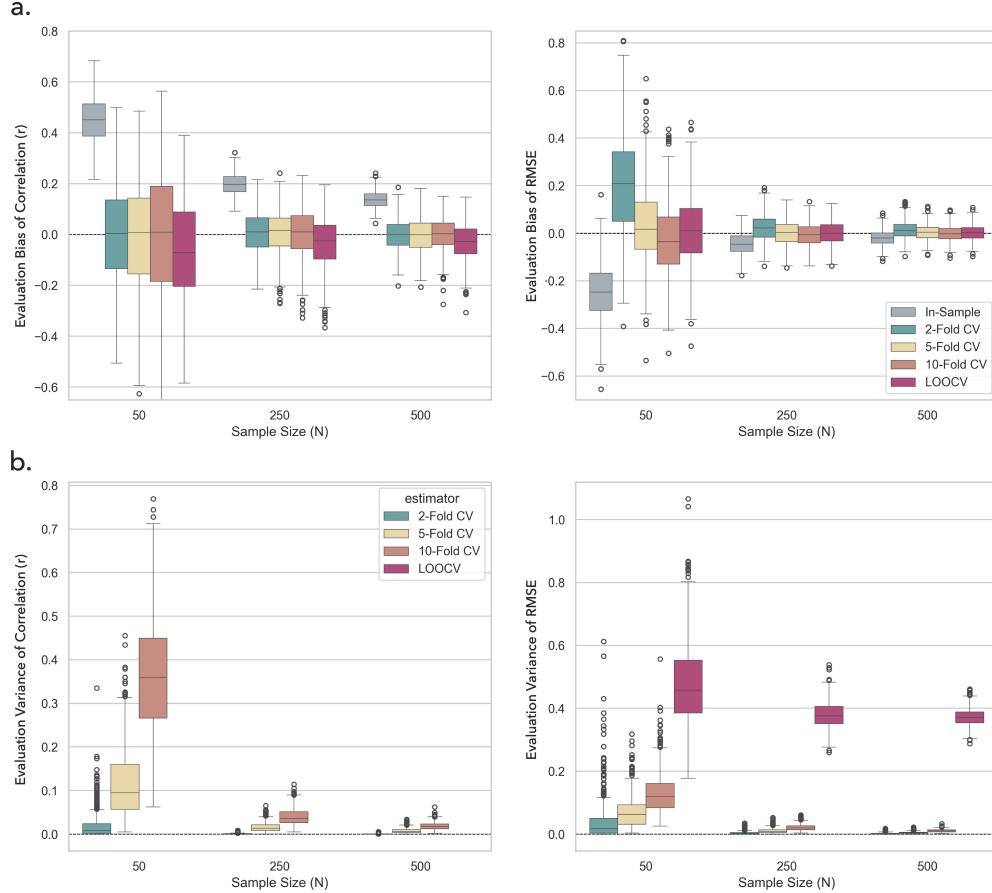


Figure 8: Evaluation bias from 500 sampling iterations on the null dataset with 10 feature variables. Multiple performance estimators across different sample sizes were color-coded. Two metrics: r and RMSE, were displayed in the column facets.

- 541 The results (Figure 8, Table 4, and Table 5) indicate that both the choice of performance estimator and the sample size
 542 notably influence evaluation reliability, which can be decomposed into bias and variance. Although different numbers of
 543 folds in CV and LOOCV show no substantial differences in bias, they do affect variance. Specifically, as the number of
 544 folds increases, the testing sets become smaller, leading to higher variance. Traditionally, LOOCV has been considered
 545 an unbiased estimator for error-based metrics such as R^2 , RMSE, and MAE. In this experiment, LOOCV generally
 546 follows that expectation, except for a few cases at certain sample sizes. Interestingly, when the ratio of sample size
 547 to number of features is sufficiently high (e.g., 25 when $N = 250$), other K -fold estimators can deliver comparable
 548 accuracy and bias, offering a more computationally efficient alternative to LOOCV.
 549 However, a key finding emerges with correlation-based metrics (e.g., r): LOOCV tends to underestimate model
 550 performance and exhibits a pessimistic bias. At $N = 250$ and $N = 500$, LOOCV's bias on r can be 10 to 30 times

551 larger than other K -fold estimators. In contrast, in-sample (or apparent) estimation, while conventionally deemed
552 the most biased due to information leakage from the testing set, can surprisingly achieve comparable reliability at
553 larger sample sizes. For instance, at $N = 250$, the bias of in-sample estimation is only 0.099 for R^2 and -0.044 for
554 RMSE, which is less biased than all K -fold CV estimators for R^2 and less biased than 2-fold CV for RMSE with
555 a smaller sample size of 50. Further examination confirms that a higher number of folds in CV generally reduces
556 bias for error-based metrics (RMSE, MAE) because training sets become more representative of the total data. Yet,
557 correlation-based metrics can display divergent trends under the same conditions. In LOOCV, evaluating a single data
558 point at a time makes its variance particularly evident for RMSE, as single-point predictions inherently fluctuate more.
559 Consequently, across all sample sizes tested, LOOCV consistently exhibits greater variance than lower-fold CV (e.g.,
560 2-fold, 5-fold). Nonetheless, bias and variance across all estimators converge as sample size grows (e.g., $N = 500$).

561 Understanding the potential bias associated with each combination of performance estimator and sample size is critical
562 when comparing similar work across multiple studies. For instance, Haque et al. (2023) reported a classification
563 accuracy of 0.991 using 10-fold cross-validation on a dataset of over 3,000 plant leaf images to train a classifier for
564 identifying potential maize diseases [40]. The study claimed that its methodology outperformed other works in terms of
565 model performance. However, one of the cited studies reported an accuracy of 0.925, which was based on a dataset of
566 only 100 images and evaluated using a 70/30 train-test split (approximately equivalent to 3-fold cross-validation) [41].
567 This comparison is therefore worth questioning, as the performance metrics were derived using different evaluation
568 methods and sample sizes. The apparent performance gap may not reflect differences in the models themselves but
569 rather the evaluation strategies employed.

570 In conclusion, performance estimation reliability depends strongly on the interplay between the estimation method, the
571 metric in use, and the sample size. Larger sample sizes typically reduce both bias and variance, thereby improving the
572 trustworthiness of model evaluations. While LOOCV often provides less biased estimates for error-based metrics, it can
573 severely underestimate correlation-based metrics and suffers from higher variance. K -fold CV methods present a more
574 computationally manageable solution for large datasets and can match LOOCV's performance when the sample size
575 is sufficiently large relative to the number of features. Ultimately, selecting the most appropriate evaluation strategy
576 should be based on practical considerations—such as available sample size, computational resources, and the specific
577 metrics of interest—to ensure robust and reliable model assessments.

Table 4: Evaluation bias (mean \pm std) for the metrics from 500 sampling iterations. The minimum bias given the same sample size is highlighted in bold. N: training sample size; r : Pearson correlation coefficient; R^2 : coefficient of determination; RMSE: root mean squared error; MAE: mean absolute error. CV: cross-validation; LOOCV: leave-one-out cross-validation.

Metric	Estimator	N=50	N=250	N=500
r	In-Sample	0.449 \pm 0.088	0.198 \pm 0.043	0.137 \pm 0.033
	2-Fold CV	0.004\pm0.184	0.009 \pm 0.082	-0.001 \pm 0.061
	5-Fold CV	-0.012 \pm 0.209	0.006 \pm 0.088	-0.001 \pm 0.067
	10-Fold CV	-0.011 \pm 0.254	0.003\pm0.094	0.000\pm0.065
R^2	LOOCV	-0.070 \pm 0.203	-0.035 \pm 0.098	-0.031 \pm 0.071
	In-Sample	0.515 \pm 0.207	0.099 \pm 0.037	0.053 \pm 0.020
	2-Fold CV	-0.694 \pm 0.642	-0.044 \pm 0.071	-0.017 \pm 0.034
	5-Fold CV	-0.401 \pm 0.409	-0.024 \pm 0.049	-0.007\pm0.026
	10-Fold CV	-0.940 \pm 0.857	-0.046 \pm 0.052	-0.014 \pm 0.024
RMSE	LOOCV	-0.013\pm0.256	0.009\pm0.039	0.008 \pm 0.020
	In-Sample	-0.244 \pm 0.116	-0.044 \pm 0.044	-0.020 \pm 0.032
	2-Fold CV	0.215 \pm 0.226	0.022 \pm 0.056	0.013 \pm 0.036
	5-Fold CV	0.035 \pm 0.158	0.002 \pm 0.047	0.004 \pm 0.033
	10-Fold CV	-0.024 \pm 0.149	-0.006 \pm 0.046	-0.001\pm0.033
MAE	LOOCV	0.012\pm0.144	0.001\pm0.046	0.003 \pm 0.033
	In-Sample	-0.195 \pm 0.096	-0.037 \pm 0.037	-0.017 \pm 0.028
	2-Fold CV	0.180 \pm 0.190	0.017 \pm 0.047	0.010 \pm 0.030
	5-Fold CV	0.049 \pm 0.134	0.004 \pm 0.039	0.003 \pm 0.028
	10-Fold CV	0.022 \pm 0.127	0.002 \pm 0.038	0.002 \pm 0.028
	LOOCV	0.011\pm0.119	-0.001\pm0.038	0.001\pm0.028

Table 5: Evaluation variance (mean \pm std) for the metrics from 500 sampling iterations. The minimum variance given the same sample size is highlighted in bold. N: training sample size; r : Pearson correlation coefficient; R^2 : coefficient of determination; RMSE: root mean squared error; MAE: mean absolute error. CV: cross-validation; LOOCV: leave-one-out cross-validation.

Metric	Estimator	N=50	N=250	N=500
r	2-Fold CV	0.019\pm0.030	0.001\pm0.001	0.000\pm0.001
	5-Fold CV	0.117 \pm 0.081	0.016 \pm 0.011	0.008 \pm 0.005
	10-Fold CV	0.362 \pm 0.131	0.040 \pm 0.019	0.019 \pm 0.009
R^2	2-Fold CV	0.859 \pm 2.876	0.003\pm0.006	0.001\pm0.001
	5-Fold CV	0.743\pm1.391	0.008 \pm 0.009	0.002 \pm 0.002
	10-Fold CV	7.164 \pm 37.486	0.018 \pm 0.019	0.003 \pm 0.002
RMSE	2-Fold CV	0.041\pm0.068	0.004\pm0.005	0.002\pm0.003
	5-Fold CV	0.070 \pm 0.050	0.010 \pm 0.008	0.005 \pm 0.004
	10-Fold CV	0.130 \pm 0.067	0.021 \pm 0.010	0.010 \pm 0.005
	LOOCV	0.477 \pm 0.131	0.379 \pm 0.041	0.371 \pm 0.029
MAE	2-Fold CV	0.030\pm0.053	0.003\pm0.003	0.001\pm0.002
	5-Fold CV	0.052 \pm 0.039	0.008 \pm 0.005	0.004 \pm 0.003
	10-Fold CV	0.100 \pm 0.052	0.015 \pm 0.007	0.008 \pm 0.004
	LOOCV	0.477 \pm 0.131	0.379 \pm 0.041	0.371 \pm 0.029

578 **3.2 Experiment 2: Misuse of Model Selection Can Lead to Over-Optimistic Performance Estimates**

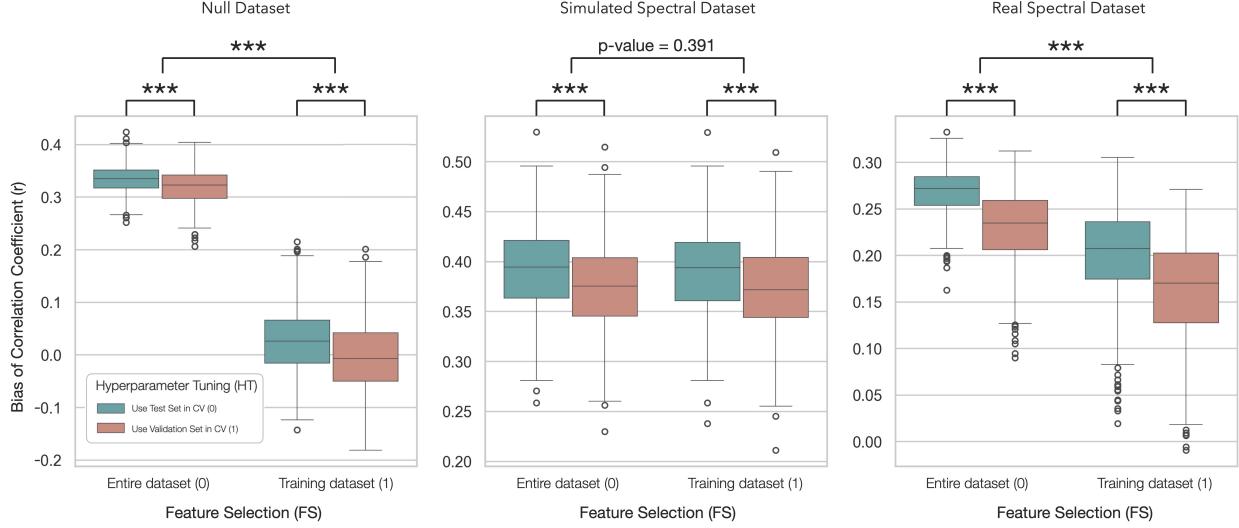


Figure 9: The evaluation bias of the four combinations of model selection strategies: “FS=0; HT=0”, “FS=0; HT=1”, “FS=1; HT=0”, “FS=1; HT=1” across three datasets. The significance level of the difference between the two strategies is noted as *** < 0.00001 .

579 With the exception of the feature selection procedure in the simulated spectral dataset, all other procedures that
 580 erroneously incorporate the testing set into the model selection process substantially inflate evaluation bias (Figure 9 and
 581 Table S.6). However, the magnitude of this inflation varies across datasets. In the null dataset, incorrectly performing
 582 feature selection leads to roughly a 30% performance inflation; on the real dataset, a similar practice results in a 6%
 583 inflation. By contrast, the simulated spectral dataset shows no measurable inflation, with only a negligible 0.02% change
 584 in r . This minimal effect may stem from the degree to which the selected features contribute to predictive accuracy. If a
 585 model’s performance is comparable to that obtained through random feature selection, then leveraging the testing set
 586 for feature selection does not necessarily boost performance. Such a scenario is more plausible in datasets with high
 587 multicollinearity, where multiple features correlate strongly, making any subset of features effectively representative of
 588 the entire feature space.

589 On the other hand, using the entire dataset to perform hyperparameter tuning substantially inflates performance in all
 590 three datasets—by approximately 2.5% in the null dataset, 1.5% in the simulated spectral dataset, and nearly 4% in
 591 the real spectral dataset. It is worth noting that these estimates arise from a relatively small search space of only six
 592 hyperparameter combinations. In contemporary machine learning, hyperparameter spaces can be far larger, especially
 593 for deep learning models in which the architectures themselves are highly configurable, involving potentially millions
 594 of parameters. Even minor alterations (e.g., changing the kernel size in a convolutional layer) may markedly affect
 595 model performance in such complex settings [42].

596 Model selection practices are critical in many precision agricultural applications. It is important to note that feature
 597 selection is not entirely prohibited from using the entire dataset, especially when employing unsupervised feature

selection methods, such as the successive projections algorithm (SPA) [43]. These methods consider feature redundancy and select the most informative features without incorporating information from the target variable in the testing set. For example, Zhang et al. (2019) leveraged ground-based hyperspectral sensors to detect weed species in rice fields [11]. Even after preprocessing and discarding spectral bands with high fluctuations due to hardware limitations, 470 bands remained for consideration. The SPA method was employed to select the most informative bands for the prediction task, ultimately identifying six key bands that also match the past literature on weed detection. This procedure is considered unbiased, as the feature selection was not explicitly performed to maximize model performance.

The same study demonstrated good practices in hyperparameter tuning. Within each training fold, a nested 5-fold cross-validation was implemented to identify the optimal hyperparameter combinations. For the random forest model, the number of trees and the spectral bands considered for splits were tuned during classifier development. Similarly, for the SVM model, the best regularization parameter C and kernel function were identified to avoid overfitting. Other studies have adopted similar practices. For instance, Shahinfar et al. (2019) used a large dataset of sheep growth management records to predict carcass traits, including birth weight, age at slaughter, and breeding values for fat composition from ancestors [44]. Given the large feature space, careful hyperparameter tuning was essential to prevent overfitting. Shahinfar et al. employed nested cross-validation to select the optimal number of decision trees and bootstrap samples for the random forest model. These examples illustrate the importance of rigorous feature selection and hyperparameter tuning to enhance model reliability and generalizability in precision agricultural studies.

Collectively, these findings underscore the importance of rigorous CV practices in model selection, particularly for feature selection and hyperparameter tuning, to achieve accurate performance estimates and robust generalizability in predictive modeling.

618 **3.3 Experiment 3: Overlooking Experimental Block Effects Can Lead to Biased Model Performance Estimates**

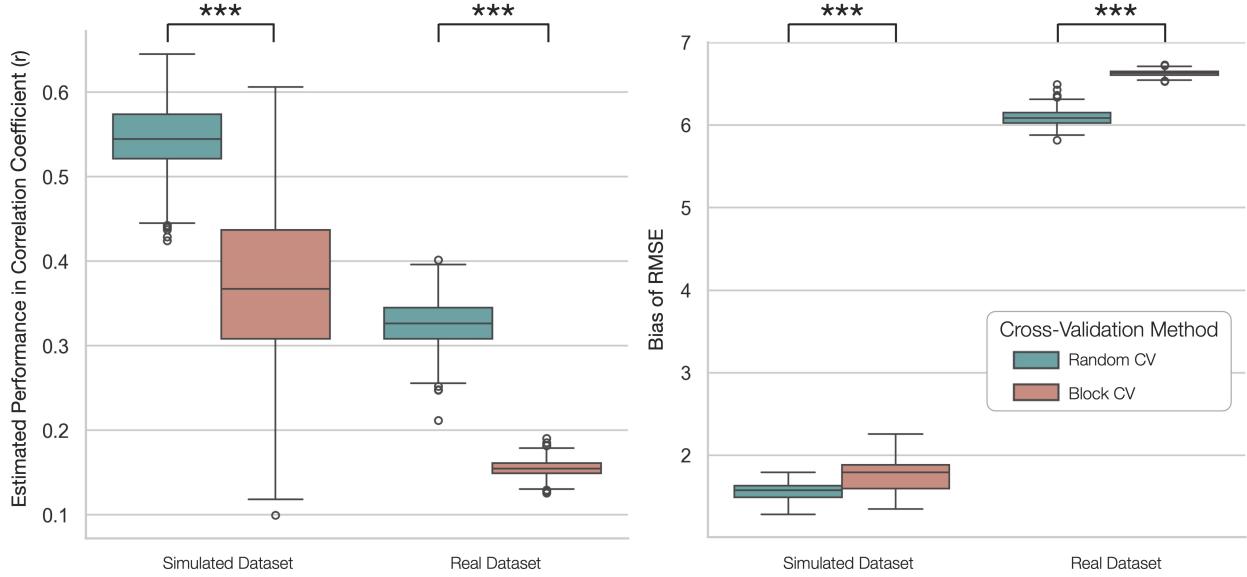


Figure 10: Bias in model performance estimation by Block CV and Random CV across 1000 iterations. The dashed line represents the null hypothesis that the mean performance estimate is zero. The significance level is noted as *** < 0.00001 .

619 Performance inflation is evident in both the simulated and real spectral datasets, which inherently exhibit seasonal
620 variation as block effects (Figure 10 and Table S.7). Ignoring this seasonal variation leads to a notable overestimation
621 of model performance, as reflected by a 17.5% bias in r for the simulated dataset and 17.1% for the real dataset. A
622 similar pattern emerges for RMSE, with a 15.5% bias in the simulated dataset and an 11.1% bias in the real dataset.
623 The ANOVA results further support these findings, since all four tests show significant differences (p -value < 0.001)
624 between the two methods on the estimated model performance.

625 St-Pierre (2001) made a similar observation when comparing datasets collected from different studies conducted under
626 distinct time periods or environmental conditions [45]. The distinctness often manifests in differences in variance
627 or value scales, complicating comparisons in the literature. St-Pierre defined this phenomenon as the “study effect,”
628 recommending that it be calibrated as a random variable instead of a fixed variable in a linear mixed model. It is because
629 these study effects are unobserved until the data is collected, fitting these effects as random variables allows for the
630 calibration of a broader set of study effects, including those that are unobserved. The author advocated for calibrating
631 the study variation before making any inferences from the dataset, which parallels the need to calibrate the seasonal
632 variation in prediction modeling to alleviate the overestimation bias observed in random CV.

633 Similar discussion are found in maize breeding. Predicting hybrid yield performance for future years or seasons has
634 been a challenge. De Oliveira et al. (2020) estimated hybrid performance across multiple years and compared two
635 CV systems [46]. The first system used random splits of the available hybrids into validation folds, while the second
636 employed year-based splits, which is an approach akin to block CV as advocated in this study. The results showed

637 substantial performance differences of up to 0.4 in r across years, underscoring the impact of properly accounting for
 638 temporal effects. Since seasons are inherently random effects that cannot be fully observed in historical datasets (as
 639 no two seasons are identical in their environmental responses to yield), a common strategy to address this issue is to
 640 quantify year-to-year variation using quantitative variables rather than directly modeling the seasonal effect. In crop
 641 modeling, such variables are referred to as environmental covariates, which decompose environmental variability into
 642 measurable components like temperature, humidity, and soil moisture. For instance, Cruz et al. (2023) incorporated
 643 183 environmental covariates, including cumulative thermal time, soil water evaporation, leaf area index, and daily
 644 infiltration, into their prediction models [47]. By estimating the effects of these covariates, researchers can address
 645 the missing information from unseen seasonal variations and mitigate the performance drop when transitioning from
 646 random CV to block CV.

647 These observations emphasize the importance of closely examining identifiable sources of variation in experiments
 648 and aligning evaluation strategies with the model's intended real-world application. Variations, such as seasonal
 649 block effects, can simultaneously influence both predictive features and response variables. If a model is intended for
 650 deployment in a new block, such as a future season for which no prior information is available, using block CV is
 651 critical to ensure the evaluation results in generalizability. Conversely, for models designed for a closed environment
 652 where all possible blocks are represented, random CV may offer a more efficient evaluation strategy.

653 3.4 Experiment 4: Characteristics of Metrics in Regression Tasks

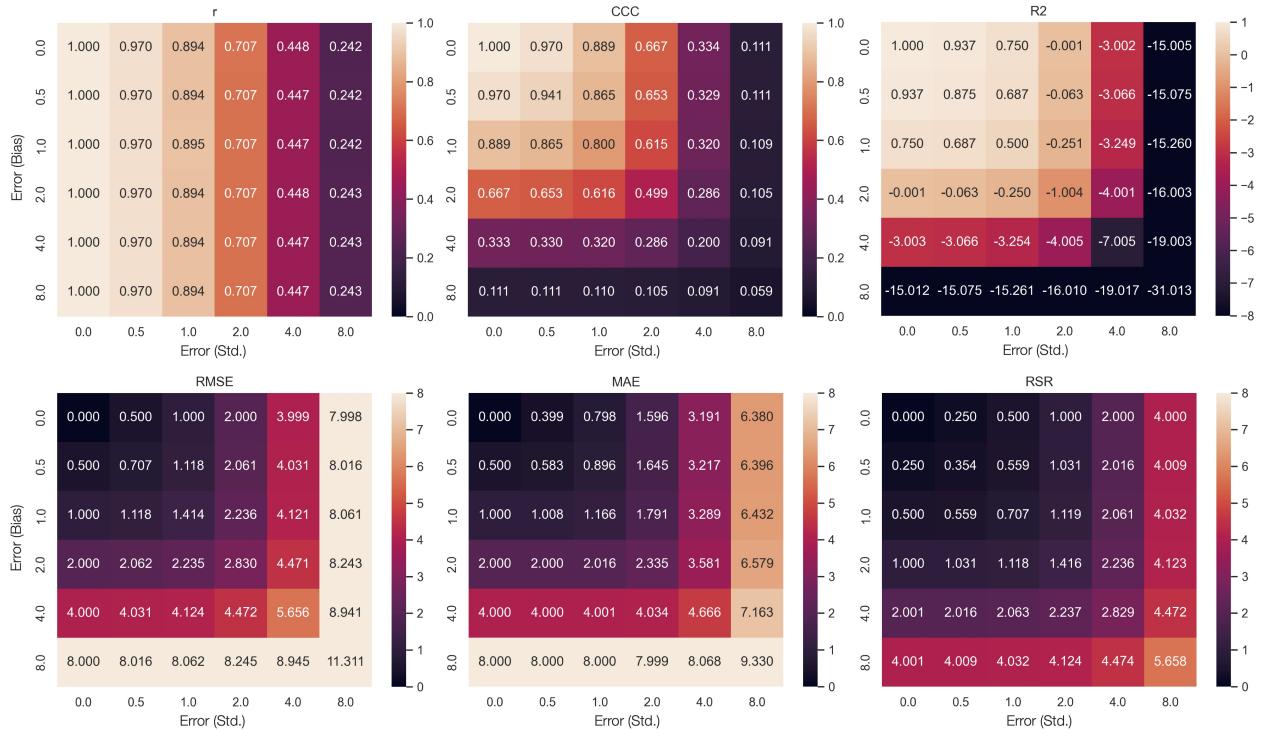


Figure 11: The heatmap of different metrics in response to the bias and variance error in regression tasks where the ground truth variance is 4.

654 Metrics in regression tasks exhibit distinct trends in response to various combinations of error bias and variance (Figure
655 11). Except for r and MAE, all metrics demonstrate symmetric responses in the grid matrix, supporting the trade-off
656 relationship between bias and variance. For instance, R^2 consistently shows a value of 0.75 at the positions (1, 4) and
657 (4, 1), indicating that a model with zero bias and variance of 2 achieves the same R^2 as a model with bias of 2 and zero
658 variance.

659 Both r and CCC are linearity-based metrics that assess the correlation between predicted and actual values, yet they
660 exhibit different trends in the heatmap. Since r is standardized by the standard deviation of both predicted and actual
661 values, it is entirely invariant to bias errors, which reflect the scale of the prediction error. In contrast, CCC is sensitive
662 to both bias and variance, offering a more comprehensive evaluation of correlation. In this experiment, CCC also
663 provides better interpretability compared to r . Specifically, when $CCC > 0.5$, the total squared error (i.e., the sum of
664 squared bias and variance) does not exceed $4 + 4 = 8$, which is twice the ground truth variance ($2^2 = 4$). Furthermore,
665 when $CCC > 0.8$, the error is guaranteed to be less than the ground truth variance. Unlike the ambiguous interpretation
666 of r , these benchmarks provide a straightforward guideline for translating the correlation concept into the scale of
667 prediction errors.

668 R^2 is a widely used metric in the machine learning community and offers good interpretability. When $R^2 = 0$, it
669 indicates that the model performs no better than the mean prediction. In the heatmap, this is evident at positions (4,
670 1) and (1, 4), where the prediction error equals the ground truth variance. Additionally, when $R^2 = -1$, it suggests
671 the prediction error exceeds the mean prediction error by one unit of the ground truth variance, as seen at position (4,
672 4), where both bias and variance errors are 2, resulting in a total squared error of 8 (twice the ground truth variance).
673 Although R^2 shares similar interpretability with CCC, it inflates rapidly in the presence of outliers. For example, when
674 the bias or variance error increases from 4 to 8, R^2 drops dramatically from -3 to -15. In contrast, CCC decreases only
675 by 0.22 under the same conditions, maintaining a range between 0 and 1, which makes it more stable and comparable
676 across different prediction contexts.

677 Error-based metrics like RMSE and MAE are often compared for their robustness to outliers. Due to the squared error
678 term in RMSE, it is more sensitive to large prediction errors. Interestingly, when analyzing their response to increases
679 in bias error, RMSE and MAE exhibit identical trends: a 1-unit increase in bias error results in a 1-unit increase in
680 both metrics. However, differences emerge along the variance error axis, where MAE increases by only 0.8 units
681 per unit increase in variance error, whereas RMSE increases by 1 unit. This indicates that MAE is more robust to
682 variance-related errors than RMSE, making it better suited for handling outlier predictions.

683 RSR, on the other hand, standardizes RMSE by the standard deviation of the ground truth values. In this experiment,
684 where the ground truth values were generated from a normal distribution with a standard deviation of 2, RSR serves
685 as a direct comparison to RMSE. While RMSE reflects the original error scale, RSR normalizes the error scale to the
686 ground truth variance, resulting in values that are always half those of RMSE. This property makes RSR an excellent
687 metric for comparing model performance across multiple datasets with varying variance levels. RSR provides a uniform

standard for performance evaluation while retaining the capability to capture error amplitude, which linearity-based metrics like r and CCC lack.

3.5 Experiment 5: Characteristics of Metrics in Classification Tasks

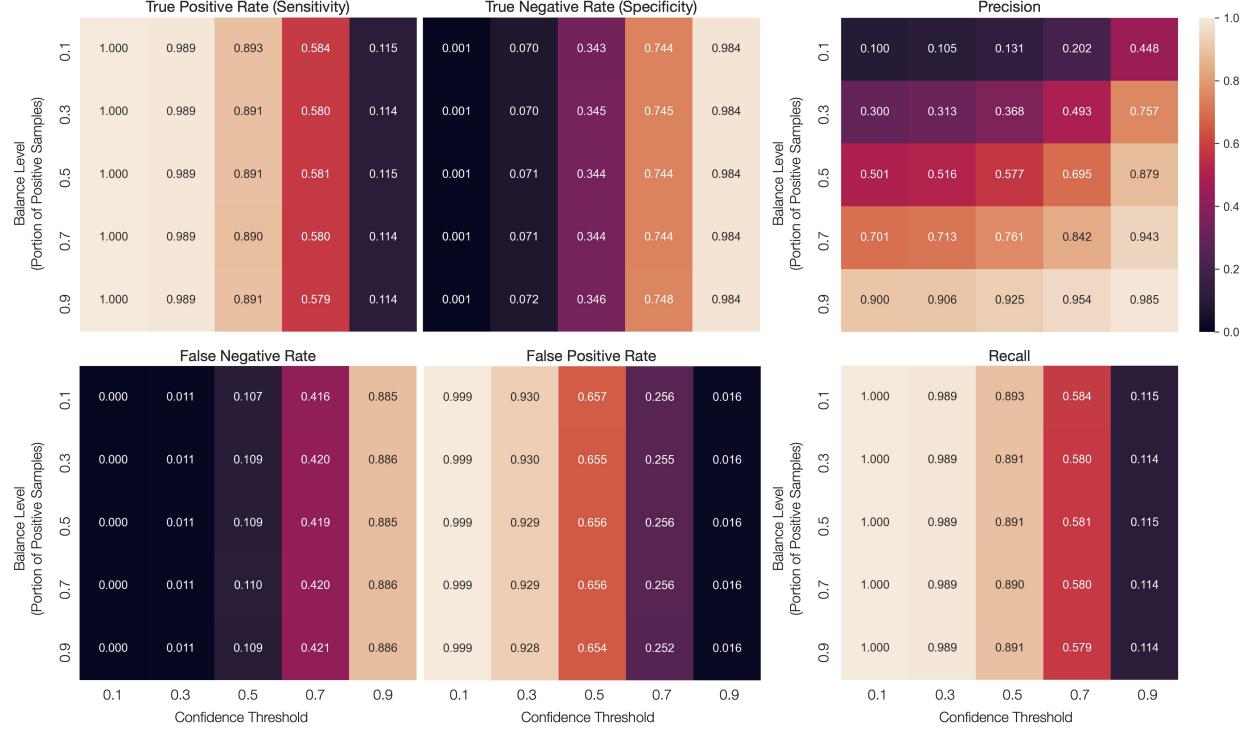


Figure 12: A 5×5 grid of performance metrics, with each cell representing a unique combination of balance and confidence threshold. The inspected metrics focus on one sample distribution at a time. The model is simulated to have higher false positive predictions than false negative predictions with a beta distribution of $\alpha = 4$ and $\beta = 3$ for the negative class.

Except for precision, metrics focused on a single sample distribution (i.e., actual positives or negatives) are unaffected by class imbalance (Figure 12). For instance, the TPR remains stable at 0.989 with a confidence threshold of 0.3, regardless of the proportion of positive samples. This stability arises because such metrics evaluate only one distribution at a time, making them invariant to shifts in class balance.

To capture a more comprehensive view of model performance, multiple metrics are often combined. In this experiment, where false positives are more frequent than false negatives, TPR alone is insufficient, as it focuses only on actual positives. Including TNR accounts for false positives among actual negatives, revealing the model’s overall tendency to produce more false positives than false negatives. Since TNR and FPR (false positive rate) are complementary ($TNR + FPR = 1$), as are TPR and FNR (false negative rate), these pairs offer flexibility in reporting and interpreting classification errors. Researchers can choose to emphasize either the correctness (TPR and TNR) or error (FPR and FNR) aspects of model predictions, depending on the application context.

702 In agricultural disease diagnosis, both TPR (sensitivity) and TNR (specificity) are critical for evaluating performance.
703 These metrics ensure accurate identification of both positive (diseased) and negative (healthy) cases. For example,
704 Buczinski et al. (2018) used biometric indicators to detect bovine respiratory disease [48], while Lu et al. (2017) relied
705 on camera images to identify wheat stripe rust and black chaff [49]. These studies highlight the importance of sensitivity
706 and specificity for balanced evaluations in diagnostic applications.

707 In contrast, precision and recall focus on predicted positive samples instead of actual positives, making them essential
708 for tasks prioritizing prediction quality on positive samples over identifying all instances of interest. It is worth noting
709 that precision measures the correctness of positive predictions and is highly sensitive to class imbalance and confidence
710 thresholds. For example, in a dataset with 70% positive samples, a naive model (i.e., when threshold = 0.1) predicting
711 all samples as positive achieves a baseline precision of 0.7. It can only be considered modest improvement if a model
712 reports a high precision of 0.8. However, in a dataset with only 10% positive samples, achieving a precision of 0.448
713 with a high confidence threshold significantly outperforms the baseline of 0.1, demonstrating strong performance in
714 handling imbalance even with a low value of precision in this case.

715 Compared to using the pair of TPR and TNR, precision and recall are particularly useful in computer vision domains
716 like image-based weed detection [11, 50], where the goal is to identify weeds or economic crops, which may constitute
717 rare positive samples, against a background of negative samples (e.g., areas with no objects of interest). In such
718 cases, TNR and FPR are less relevant, as the focus is on the reliability of the model’s predictions (precision) and its
719 ability to capture all positive instances of weeds or crops (recall). Similarly, in natural language processing (NLP)
720 and information retrieval tasks [51, 52], precision and recall play vital roles. For example, in evaluating language
721 models, the focus is often on how reliable the generated responses are (precision) and how many relevant responses are
722 produced (recall). Negative samples, such as irrelevant or nonsensical outputs, are typically less of a concern compared
723 to ensuring the relevance and completeness of positive samples. In information retrieval, where a user might submit a
724 document containing ten key pieces of information (positive samples), the emphasis is on how many of these key pieces
725 the system retrieves (recall) and how accurate those retrieved pieces are (precision), rather than on the proportion of
726 irrelevant information correctly ignored (TNR).

727 In summary, each pair of metrics (TPR and TNR, precision and recall) offers a unique perspective on model performance.
728 TPR (or sensitivity) and TNR (specificity) are essential for evaluating the model’s ability to correctly classify positive
729 and negative samples, respectively, while precision and recall are indispensable metrics for scenarios where identifying
730 and evaluating the quality of positive samples are more critical than focusing on the negative ones. These metrics are
731 not self-explanatory but remain popular in the machine learning community due to their ability to measure the reliability
732 and completeness of positive predictions in tasks where negative samples are of secondary interest.

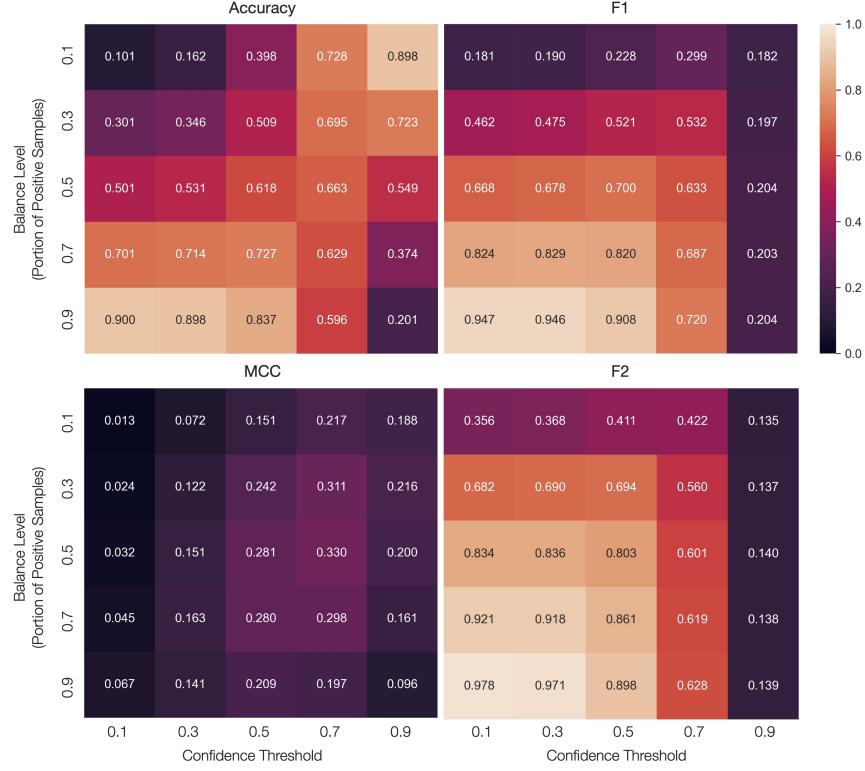


Figure 13: A 5×5 grid of performance metrics, with each cell representing a unique combination of balance and confidence threshold. The inspected metrics focus on multiple sample distributions simultaneously. The prediction model is simulated to have higher false positive rate than false negative rate.

- 733 Metrics that capture multiple aspects of model performance offer more comprehensive insights into a model’s effec-
 734 tiveness (Figure 13). The heatmap in this experiment illustrates extreme scenarios at its corners, highlighting metric
 735 strengths and limitations. For example, the top-left corner features predominantly negative samples with mostly positive
 736 predictions, leading to many false positives, while the bottom-right corner features predominantly positive samples
 737 with mostly negative predictions, resulting in false negatives. The other corners illustrate cases where predictions
 738 align overwhelmingly with the majority class, allowing naive models to achieve high performance purely due to class
 739 imbalance rather than true predictive power. Effective metrics must assess performance beyond such “background
 740 effects.”
- 741 Accuracy, while simple and widely used, is limited in these extreme cases. For instance, in the top-left corner, accuracy
 742 drops due to false positives, whereas in corners dominated by a majority class, accuracy can be misleadingly high. For
 743 example, in the top-right corner, where 90% of samples are negative, a model predicting all negatives achieves 90%
 744 accuracy, despite ignoring true positives. Such scenarios demonstrate accuracy’s inability to capture critical trade-offs
 745 between false positives and false negatives.
- 746 The F1-score addresses accuracy’s limitations by balancing precision and TPR (i.e., recall), making it suitable for
 747 moderately imbalanced datasets. For example, in the same top-right corner of the heatmap, the F1-score of 0.182
 748 reflects sensitivity to both false positives and false negatives, unlike accuracy, which may inflate performance due to

749 imbalance. Real-world studies, such as Haque et al. (2023), underscore this: in a maize leaf blight detection task with a
750 1:2 class imbalance, an accuracy of 99.02% masked the model’s limitations. Instead, an F1-score of 97.49% provided a
751 clearer measure of the model’s ability to balance precision and TPR [40].

752 Variants like the F2-score further refine performance evaluation by prioritizing TPR over precision, making it useful
753 for scenarios where missing positive cases is costly. For instance, Minni (2024) applied the F2-score in breast cancer
754 diagnosis to emphasize capturing positive cases, addressing the severe consequences of false negatives [53].

755 The MCC provides a robust alternative by considering all confusion matrix components and mitigating sensitivity to
756 class imbalance. Becker et al. (2021) demonstrated MCC’s stability across datasets with varying imbalances, ranging
757 from mildly (119 vs. 153 samples) to severely imbalanced (265 non-stress vs. 50 stress samples), when predicting
758 heat stress in dairy cattle. Unlike F1-scores, which can reflect class distribution more than true classifier quality,
759 MCC delivered a more comprehensive and reliable evaluation [13]. These examples highlight MCC’s effectiveness in
760 assessing model performance across diverse contexts.

761 4 Conclusion

762 In conclusion, this study presents a comprehensive evaluation of five simulation experiments, uncovering critical
763 insights into the interplay of performance estimators, metrics, and contextual factors that influence model evaluation
764 reliability. The findings highlight the nuanced impact of estimator choices and sample sizes on bias and variance,
765 emphasizing that while traditional estimators such as LOOCV can be reliable for error-based metrics, they may
766 severely underestimate correlation-based metrics under certain conditions. The misuse of model selection processes
767 was shown to substantially inflate performance estimates, reinforcing the importance of adhering to rigorous cross-
768 validation practices. Additionally, the role of experimental block effects in biasing performance estimates underscores
769 the necessity of aligning evaluation strategies with real-world applications. In both regression and classification
770 tasks, metric characteristics vary significantly, with different metrics offering complementary perspectives on model
771 performance. For regression tasks, CCC and R^2 provided interpretable benchmarks for understanding prediction errors,
772 while MAE demonstrated greater robustness to variance compared to RMSE. In classification tasks, metrics such as
773 precision, F1 score, and MCC were shown to capture different facets of performance, with MCC offering a balanced,
774 stringent evaluation across multiple dimensions. Collectively, these findings stress the importance of tailoring model
775 evaluation strategies to specific research and application contexts to ensure robust, reliable, and interpretable results.

776 5 Code Availability

777 All the implementation codes for the simulation experiments and data analysis are available at <https://github.com/>
778 Niche-Lab/modeling-guide/.

779 6 Acknowledgement

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

782 Declaration of generative AI and AI-assisted technologies in the writing process

783 During the preparation of this work the author(s) used ChatGPT in order to correct grammar and improve the readability
784 of the manuscript. The author(s) reviewed and edited the content as needed after using this tool/service and take(s) full
785 responsibility for the content of the publication.

786 References

- 787 [1] Hao Cheng, Dorian J. Garrick, and Rohan L. Fernando. Efficient strategies for leave-one-out cross validation for
788 genomic best linear unbiased prediction. *Journal of Animal Science and Biotechnology*, 8(1):38, May 2017.
- 789 [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
790 resilience during the transition period in dairy cows: A case study. *Journal of Dairy Science*, 101(11):10271–10282,
791 November 2018.
- 792 [3] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and
793 Prediction*. Springer series in statistics. Springer, 2009.
- 794 [4] Gavin C. Cawley and Nicola L.C. Talbot. On Over-fitting in Model Selection and Subsequent Selection Bias in
795 Performance Evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, August 2010.
- 796 [5] Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems.
797 *Technometrics*, 12(1):55–67, 1970. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American
798 Society for Quality].
- 799 [6] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society:
800 Series B (Methodological)*, 58(1):267–288, January 1996.
- 801 [7] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression
802 machines. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS’96,
803 pages 155–161, Cambridge, MA, USA, December 1996. MIT Press.
- 804 [8] Hervé Abdi. Partial Least Square Regression PLS-Regression. *Encyclopedia of social sciences research methods*,
805 pages 792–795, 2003.
- 806 [9] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.
- 807 [10] Yann LeCun. Generalization and Network Design strategies. 1989.
- 808 [11] Yanchao Zhang, Junfeng Gao, Haiyan Cen, Yongliang Lu, Xiaoyue Yu, Yong He, and Jan G. Pieters. Automated
809 spectral feature extraction from hyperspectral images to differentiate weedy rice and barnyard grass from a rice
810 crop. *Computers and Electronics in Agriculture*, 159:42–49, April 2019.
- 811 [12] G. Rovere, G. de los Campos, A. L. Lock, L. Worden, A. I. Vazquez, K. Lee, and R. J. Tempelman. Prediction of
812 fatty acid composition using milk spectral data and its associations with various mid-infrared spectral regions in
813 Michigan Holsteins. *Journal of Dairy Science*, 104(10):11242–11258, October 2021.
- 814 [13] C. A. Becker, A. Aghalari, M. Marufuzzaman, and A. E. Stone. Predicting dairy cattle heat stress using machine
815 learning techniques. *Journal of Dairy Science*, 104(1):501–524, January 2021.
- 816 [14] B. Lahart, S. McParland, E. Kennedy, T.M. Boland, T. Condon, M. Williams, N. Galvin, B. McCarthy, and
817 F. Buckley. Predicting the dry matter intake of grazing dairy cows using infrared reflectance spectroscopy analysis.
818 *Journal of Dairy Science*, 102(10):8907–8918, October 2019.

- 819 [15] Tiago Bresolin and João R. R. Dórea. Infrared Spectrometry as a High-Throughput Phenotyping Technology to
820 Predict Complex Traits in Livestock Systems. *Frontiers in Genetics*, 11, 2020.
- 821 [16] C. Grelet, E. Froidmont, L. Foldager, M. Salavati, M. Hostens, C. P. Ferris, K. L. Ingvartsen, M. A. Crowe, M. T.
822 Sorensen, J. A. Fernandez Pierna, A. Vanlierde, N. Gengler, and F. Dehareng. Potential of milk mid-infrared
823 spectra to predict nitrogen use efficiency of individual dairy cows in early lactation. *Journal of Dairy Science*,
824 103(5):4435–4445, May 2020.
- 825 [17] I. Adriaens, N. C. Friggens, W. Ouweltjes, H. Scott, B. Aernouts, and J. Statham. Productive life span and
826 resilience rank can be predicted from on-farm first-parity sensor time series but not using a common equation
827 across farms. *Journal of Dairy Science*, 103(8):7155–7171, August 2020.
- 828 [18] Lucio F. M. Mota, Diana Giannuzzi, Vittoria Bisutti, Sara Pegolo, Erminio Trevisi, Stefano Schiavon, Luigi Gallo,
829 David Fineboym, Gil Katz, and Alessio Cecchinato. Real-time milk analysis integrated with stacking ensemble
830 learning as a tool for the daily prediction of cheese-making traits in Holstein cattle. *Journal of Dairy Science*,
831 105(5):4237–4255, May 2022.
- 832 [19] Roii Spoliansky, Yael Edan, Yisrael Parmet, and Ilan Halachmi. Development of automatic body condition scoring
833 using a low-cost 3-dimensional Kinect camera. *Journal of Dairy Science*, 99(9):7714–7725, September 2016.
- 834 [20] Sun Yukun, Huo Pengju, Wang Yujie, Cui Ziqi, Li Yang, Dai Baisheng, Li Runze, and Zhang Yonggen. Automatic
835 monitoring system for individual dairy cows based on a deep learning framework that provides identification via
836 body parts and estimation of body condition score. *Journal of Dairy Science*, 102(11):10140–10151, November
837 2019.
- 838 [21] X. Song, E.A.M. Bokkers, P.P.J. Van Der Tol, P.W.G. Groot Koerkamp, and S. Van Mourik. Automated body
839 weight prediction of dairy cows using 3-dimensional vision. *Journal of Dairy Science*, 101(5):4448–4459, May
840 2018.
- 841 [22] C. Xavier, Y. Le Cozler, L. Depuille, A. Caillot, A. Lebreton, C. Allain, J. M. Delouard, L. Delattre, T. Luginbuhl,
842 P. Faverdin, and A. Fischer. The use of 3-dimensional imaging of Holstein cows to estimate body weight and
843 monitor the composition of body weight change throughout lactation. *Journal of Dairy Science*, 105(5):4508–4519,
844 May 2022.
- 845 [23] P. Mäntysaari, E.A. Mäntysaari, T. Kokkonen, T. Mehtio, S. Kajava, C. Grelet, P. Lidauer, and M.H. Lidauer.
846 Body and milk traits as indicators of dairy cow energy status in early lactation. *Journal of Dairy Science*,
847 102(9):7904–7916, September 2019.
- 848 [24] M. Frizzarin, I. C. Gormley, D. P. Berry, T. B. Murphy, A. Casa, A. Lynch, and S. McParland. Predicting cow
849 milk quality traits from routinely available milk spectra using statistical machine learning methods. *Journal of
850 Dairy Science*, 104(7):7438–7447, July 2021.
- 851 [25] J. A. D. R. N. Appuhamy, J. V. Judy, E. Kebreab, and P. J. Kononoff. Prediction of drinking water intake by dairy
852 cows. *Journal of Dairy Science*, 99(9):7191–7205, September 2016.

- 853 [26] R. A. de Souza, R. J. Tempelman, M. S. Allen, W. P. Weiss, J. K. Bernard, and M. J. VandeHaar. Predicting
854 nutrient digestibility in high-producing dairy cows. *Journal of Dairy Science*, 101(2):1123–1135, February 2018.
- 855 [27] J. R. R. Dórea, G. J. M. Rosa, K. A. Weld, and L. E. Armentano. Mining data from milk infrared spectroscopy to
856 improve feed intake predictions in lactating dairy cows. *Journal of Dairy Science*, 101(7):5878–5889, July 2018.
- 857 [28] L. I. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268, March
858 1989.
- 859 [29] Edward J. Jones, Thomas F. A. Bishop, Brendan P. Malone, Patrick J. Hulme, Brett M. Whelan, and Patrick
860 Filippi. Identifying causes of crop yield variability with interpretive machine learning. *Computers and Electronics*
861 in Agriculture, 192:106632, January 2022.
- 862 [30] N. W. O’Leary, D. T. Byrne, A. H. O’Connor, and L. Shalloo. Invited review: Cattle lameness detection with
863 accelerometers. *Journal of Dairy Science*, 103(5):3895–3911, May 2020.
- 864 [31] J. Stojkov, G. Bowers, M. Draper, T. Duffield, P. Duivenvoorden, M. Groleau, D. Haupstein, R. Peters,
865 J. Pritchard, C. Radom, N. Sillett, W. Skippon, H. Trépanier, and D. Fraser. Hot topic: Management of cull
866 dairy cows—Consensus of an expert consultation in Canada. *Journal of Dairy Science*, 101(12):11170–11174,
867 December 2018.
- 868 [32] Maher Alsaad, Mahmoud Fadul, and Adrian Steiner. Automatic lameness detection in cattle. *The Veterinary
869 Journal*, 246:35–44, April 2019.
- 870 [33] X. Kang, X. D. Zhang, and G. Liu. Accurate detection of lameness in dairy cattle with computer vision: A new
871 and individualized detection strategy based on the analysis of the supporting phase. *Journal of Dairy Science*,
872 103(11):10628–10638, November 2020.
- 873 [34] S. J. Denholm, W. Brand, A. P. Mitchell, A. T. Wells, T. Krzyzelewski, S. L. Smith, E. Wall, and M. P. Coffey.
874 Predicting bovine tuberculosis status of dairy cows from mid-infrared spectral data of milk using deep learning.
875 *Journal of Dairy Science*, 103(10):9355–9367, October 2020.
- 876 [35] S.A. Kandeel, A.A. Megahed, M.H. Ebeid, and P.D. Constable. Ability of milk pH to predict subclinical mastitis
877 and intramammary infection in quarters from lactating dairy cattle. *Journal of Dairy Science*, 102(2):1417–1427,
878 February 2019.
- 879 [36] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1
880 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, January 2020.
- 881 [37] J. M. Bowen, M. J. Haskell, G. A. Miller, C. S. Mason, D. J. Bell, and C-A. Duthie. Early prediction of
882 respiratory disease in preweaning dairy calves using feeding and activity behaviors. *Journal of Dairy Science*,
883 104(11):12009–12018, November 2021.
- 884 [38] Maxime Metz, Alessandra Biancolillo, Matthieu Lesnoff, and Jean-Michel Roger. A note on spectral data
885 simulation. *Chemometrics and Intelligent Laboratory Systems*, 200:103979, May 2020.

- 886 [39] Chun-Peng James Chen, Yang Hu, Xianran Li, Craig F. Morris, Stephen Delwiche, Arron H. Carter,
887 Camille Steber, and Zhiwu Zhang. An independent validation reveals the potential to predict Hag-
888 berg–Perten falling number using spectrometers. *The Plant Phenome Journal*, 6(1):e20070, 2023. _eprint:
889 <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ppj2.20070>.
- 890 [40] Md. Ashraful Haque, Sudeep Marwaha, Chandan Kumar Deb, Sapna Nigam, and Alka Arora. Recognition of
891 diseases of maize crop using deep learning models. *Neural Computing and Applications*, 35(10):7407–7421,
892 April 2023.
- 893 [41] Malusi Sibya and Mbuyu Sumbwanyambe. A Computational Procedure for the Recognition and Classification of
894 Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks. *AgriEngineering*, 1(1):119–
895 131, March 2019. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
- 896 [42] Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning, February 2017.
897 arXiv:1611.01578 [cs].
- 898 [43] Sófacles Figueiredo Carreiro Soares, Adriano A. Gomes, Mario Cesar Ugulino Araujo, Arlindo Rodrigues Galvão
899 Filho, and Roberto Kawakami Harrop Galvão. The successive projections algorithm. *TrAC Trends in Analytical
900 Chemistry*, 42:84–98, January 2013.
- 901 [44] Saleh Shahinfar, Khama Kelman, and Lewis Kahn. Prediction of sheep carcass traits from early-life records using
902 machine learning. *Computers and Electronics in Agriculture*, 156:159–177, January 2019.
- 903 [45] N. R. St-Pierre. *Invited Review: Integrating Quantitative Findings from Multiple Studies Using Mixed Model
904 Methodology1*. *Journal of Dairy Science*, 84(4):741–755, April 2001.
- 905 [46] Amanda Avelar de Oliveira, Marcio F. R. Resende, Luís Felipe Ventorim Ferrão, Rodrigo Rampazo Amadeu,
906 Lauro José Moreira Guimarães, Claudia Teixeira Guimarães, Maria Marta Pastina, and Gabriel Rodrigues Alves
907 Margarido. Genomic prediction applied to multiple traits and environments in second season maize hybrids.
908 *Heredity*, 125(1):60–72, August 2020. Publisher: Nature Publishing Group.
- 909 [47] Marco Lopez-Cruz, Fernando M. Aguate, Jacob D. Washburn, Natalia de Leon, Shawn M. Kaepller,
910 Dayane Cristina Lima, Ruijuan Tan, Addie Thompson, Laurence Willard De La Bretonne, and Gustavo de los
911 Campos. Leveraging data from the Genomes-to-Fields Initiative to investigate genotype-by-environment inter-
912 actions in maize in North America. *Nature Communications*, 14(1):6904, October 2023. Publisher: Nature
913 Publishing Group.
- 914 [48] S. Buczinski, G. Fecteau, J. Dubuc, and D. Francoz. Validation of a clinical scoring system for bovine respiratory
915 disease complex diagnosis in preweaned dairy calves using a Bayesian framework. *Preventive Veterinary Medicine*,
916 156:102–112, August 2018.
- 917 [49] Jiang Lu, Jie Hu, Guannan Zhao, Fenghua Mei, and Changshui Zhang. An in-field automatic wheat disease
918 diagnosis system. *Computers and Electronics in Agriculture*, 142:369–379, November 2017.

- 919 [50] Wen-Hao Su. Advanced Machine Learning in Point Spectroscopy, RGB- and Hyperspectral-Imaging for Automatic
920 Discriminations of Crops and Weeds: A Review. *Smart Cities*, 3(3):767–792, September 2020. Number: 3
921 Publisher: Multidisciplinary Digital Publishing Institute.
- 922 [51] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang,
923 and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024.
924 arXiv:2312.10997.
- 925 [52] Alireza Salemi and Hamed Zamani. Evaluating Retrieval Quality in Retrieval-Augmented Generation, April 2024.
926 arXiv:2404.13781.
- 927 [53] N. Minni, N. Rehma, and P. M. Harikrishnaa. Exploring Machine Learning Algorithms for Accurate Breast
928 Cancer Classification: A Comparative Analysis Using F2 Metric. In B. Rama Devi, Kishore Kumar, M. Raju,
929 K. Srujan Raju, and Mathini Sellathurai, editors, *Proceedings of Fifth International Conference on Computer and*
930 *Communication Technologies*, pages 153–162, Singapore, 2024. Springer Nature.

931 **Appendix**

932 **Cross Validation**

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

$$\begin{aligned}\mathcal{D} &= \{(X, Y)\} \\ &= \{(X_1, Y_1), (X_2, Y_2), \dots, (X_K, Y_K)\}\end{aligned}\tag{S.1}$$

937 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
 938 variable. The value of n corresponds to the total number of samples, while p represents the number of features. In
 939 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
 940 remaining folds make up the training set $\mathcal{D}_{\text{train}}$ (or \mathcal{D}_{-k}):

$$\begin{aligned}\mathcal{D}_{\text{train}} &= \mathcal{D}_{-k} \\ &= \{(X_1, Y_1), (X_2, Y_2), \dots, (X_{k-1}, Y_{k-1}), (X_{k+1}, Y_{k+1}), \dots, (X_K, Y_K)\} \\ \mathcal{D}_{\text{test}} &= \mathcal{D}_k \\ &= \{(X_k, Y_k)\}\end{aligned}\tag{S.2}$$

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

$$\begin{aligned}\hat{g}(f_{\mathcal{D}_{-k}}) &= \mathcal{L}(Y_k, \hat{Y}_k) \\ &= \mathcal{L}(Y_k, f_{\mathcal{D}_{-k}}(X_k))\end{aligned}\tag{S.3}$$

944 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
 945 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
 946 performance over all K folds. The model's generalization performance can be expressed as:

$$\begin{aligned}\mathbb{E}[\hat{g}(f_{\mathcal{D}})] &= \mathbb{E}[\hat{g}(f_{\mathcal{D}_k})] \\ &= \frac{1}{K} \sum_{k=1}^K \hat{g}(f_{\mathcal{D}_k})\end{aligned}\tag{S.4}$$

947 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
 948 all $\hat{g}(f_{\mathcal{D}_k})$ over K folds, which is also the definition of $\mathbb{E}[\hat{g}(f_{\mathcal{D}_k})]$.

949 Cross Validation Bias and Variance

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

$$\text{Bias} = \mathbb{E}[\hat{g}(f_{\mathcal{D}})] - G(f_{\mathcal{D}})\tag{S.5}$$

953 For example, if RMSE is used as the performance metric, a positive validation bias suggests that the model validation
 954 procedure concludes a pessimistic estimation of the model performance, since the true performance is expected to be
 955 lower than the estimated performance. Another aspect of model validation is the variance of the estimated performance.
 956 For example, in a 5-fold cross-validation, there are five estimates of the model performance. The variance among these
 957 five estimates is known as validation variance. A high validation variance suggests that the performance is sensitive to
 958 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
 959 variance can be defined as:

$$\begin{aligned}\text{Variance} &= \mathbb{E}[(\hat{g}(f_{\mathcal{D}_k}) - \mathbb{E}[\hat{g}(f_{\mathcal{D}})])^2] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k}) - 2\hat{g}(f_{\mathcal{D}_k})\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - 2\mathbb{E}[\hat{g}(f_{\mathcal{D}_k})]\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]\end{aligned}\tag{S.6}$$

960 Combining the Equations S.5 and S.6, the mean squared error (MSE) of the model validation can be decomposed as:

$$\begin{aligned}
\text{MSE} &= \mathbb{E}[(\hat{g}(f_{D_k}) - G(f_D))^2] \\
&= \mathbb{E}[\hat{g}^2(f_{D_k})] - 2\mathbb{E}[\hat{g}(f_{D_k})]G(f_D) + G^2(f_D) + \\
&\quad \mathbb{E}^2[\hat{g}(f_{D_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})] \\
&= (\mathbb{E}^2[\hat{g}(f_{D_k})] - 2\mathbb{E}[\hat{g}(f_{D_k})]G(f_D) + G^2(f_D)) + \\
&\quad (\mathbb{E}[\hat{g}^2(f_{D_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})]) \\
&= (\mathbb{E}[\hat{g}(f_{D_k})] - G(f_D))^2 + (\mathbb{E}[\hat{g}^2(f_{D_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})]) \\
&= (\mathbb{E}[\hat{g}(f_D)] - G(f_D))^2 + (\mathbb{E}[\hat{g}^2(f_D)] - \mathbb{E}^2[\hat{g}(f_D)]) \\
&= \text{Bias}^2 + \text{Variance}
\end{aligned} \tag{S.7}$$

961 **Hyperparameter**

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

$$\mathcal{L}_{\text{OLS}}(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 \tag{S.8}$$

$$\mathcal{L}_{\text{ridge}}(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda \sum_{j=1}^p \beta_j^2 \tag{S.9}$$

$$\mathcal{L}_{\text{LASSO}}(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \tag{S.10}$$

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

967 **Squared Correlation Coefficient r^2 and Determination Coefficient R^2**

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

$$\begin{aligned}
\text{cov}(Y, \hat{Y}) &= \text{cov}(\hat{Y} + \epsilon, \hat{Y}) \\
&= \text{cov}(\hat{Y}, \hat{Y}) + \text{cov}(\hat{Y}, \epsilon) \\
&= \text{var}(\hat{Y}) + \text{cov}(\hat{Y}, \epsilon) \\
&= \text{var}(\hat{Y})
\end{aligned} \tag{S.11}$$

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

$$\begin{aligned}
r^2 &= \frac{\text{cov}^2(Y, \hat{Y})}{\text{var}(Y)\text{var}(\hat{Y})} \\
&= \frac{\text{var}(\hat{Y})^2}{\text{var}(Y)\text{var}(\hat{Y})} \\
&= \frac{\text{var}(\hat{Y})}{\text{var}(Y)} \\
&= \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{\hat{Y}})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \\
&= \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{\hat{Y}})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \\
&= \frac{SS_{\text{regression}}}{SS_{\text{total}}} \\
&= R^2
\end{aligned} \tag{S.12}$$

975 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
976 i th elements of the actual response vector Y and the predicted response vector \hat{Y} , while \bar{Y} and $\bar{\hat{Y}}$ are their respective
977 means. This proof highlights that under certain assumptions, r^2 and R^2 can indeed be equivalent, but such conditions
978 are specific to least squares regression where errors are normally distributed and predictions are unbiased estimates of
979 the actual values.

980 Original results of the simulation experiments

Table S.6: Experiment 2: ANOVA results of how each CV procedure affects the evaluation bias measured in the correlation coefficient (r). FS: Feature Selection, HT: Hyperparameter Tuning. DF: Degree of Freedom, SS: Sum of Squares, MS: Mean Squares. Significant p-values (< 0.05) are highlighted in bold.

(a) Dataset: Null dataset, Metric: r

Factor	DF	SS	MS	F-value	p-value
FS	1	49.72	49.72	20073.41	< 1e-6
HT	1	0.24	0.24	97.83	< 1e-6
FS:HT	1	0.03	0.03	14.33	< 1e-6
Residual	1996	4.94	0.00	—	—

(b) Dataset: Simulated spectral dataset, Metric: r

Factor	DF	SS	MS	F-value	p-value
FS	1	1.87e-03	1.87e-03	1.03	0.391
HT	1	1.64e-01	1.64e-01	91.10	< 1e-6
FS:HT	1	2.85e-08	2.85e-08	0.00	0.997
Residual	1996	3.60e+00	1.80e-03	—	—

(c) Dataset: Real spectral dataset, Metric: r

Factor	DF	SS	MS	F-value	p-value
FS	1	2.31	2.31	1198.87	< 1e-6
HT	1	0.73	0.73	382.76	< 1e-6
FS:HT	1	0.00	0.00	0.20	0.648
Residual	1996	3.85	0.00	—	—

Table S.7: Experiment 3: ANOVA results for the effect of deploying block CV and random CV. DF: Degree of Freedom, SS: Sum of Squares, MS: Mean Squares. Significant p-values (< 0.05) are highlighted in bold.

(a) Dataset: Simulated spectral dataset, Metric: r

Factor	DF	SS	MS	F-value	p-value
method	1	9.61	9.61	2122.69	< 1e-6
Residual	998	4.52	0.00	—	—

(b) Dataset: Real spectral dataset, Metric: r

Factor	DF	SS	MS	F-value	p-value
method	1	8.64	8.64	29744.48	< 1e-6
Residual	998	0.29	0.00	—	—

(c) Dataset: Simulated spectral dataset, Metric: $RMSE$

Factor	DF	SS	MS	F-value	p-value
method	1	11.57	11.57	559.59	< 1e-6
Residual	998	20.64	0.02	—	—

(d) Dataset: Real spectral dataset, Metric: $RMSE$

Factor	DF	SS	MS	F-value	p-value
method	1	88.40	88.40	26768.87	< 1e-6
Residual	998	3.29	0.00	—	—