
SIMULATION STUDIES AS A GUIDE ON AVOIDING COMMON PITFALLS IN EVALUATING MODEL PERFORMANCE

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

May 8, 2024

ABSTRACT

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

Keywords Model Evaluation · Performance Metrics · Simulation Studies

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

17 **1 Introduction**

18 **1.1 Modeling**

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

33 Here, we review two types of challenges that can be encountered during the model evaluation process: challenges in
34 data structure and challenges in evaluation approach. Data structure challenges include those inherent to the types
35 of data used in a modeling exercise. For continuous data types, challenges include measurement variance, extreme
36 observations, and underlying variation structures like blocks. For categorical data, challenges largely center around the
37 balance or lack thereof between categories. Challenges in the evaluation approach are driven by decision-making about
38 which data are used for model derivation and which are used for evaluation. We will review these challenges in this
39 study.

40 **1.2 Model Evaluation**

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

48 The most common CV method is K-fold CV, which partitions the dataset into K equally sized folds. In each iteration,
49 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

50 set (noted as $\mathcal{D}_{\text{train}}$) to construct the model. Once the model completes training, it is evaluated on the $\mathcal{D}_{\text{test}}$ to obtain a
 51 estimate of the model performance \hat{g} . The process will iterate K times until each fold has been used as the $\mathcal{D}_{\text{test}}$ once.
 52 And the average performance over all K folds is deemed as the expected generalization performance of the model $\mathbb{E}[\hat{g}]$
 53 on new data.

54 However, there is always a evaluation bias between the estimated performance $\mathbb{E}[\hat{g}]$ and the true generalization
 55 performance G , which can only be approximated by evaluating the same model on an infinite number of unseen data. If
 56 RMSE is used as the performance metric, which is lower the better for the model performance, a positive evaluation bias
 57 $\mathbb{E}[\hat{g}] - G$ suggests that the model evaluation procedure concludes a pessimistic estimation of the model performance,
 58 since the true performance is expected to be lower than the estimated performance. Another aspect of model evaluation
 59 error is the variance of each estimated performance \hat{g} across the K folds. For example, there are five estimates in a
 60 5-fold cross-validation. The variance among these five estimates is defined as the evaluation variance. A high evaluation
 61 variance suggests that the performance is sensitive to the choice of the test set $\mathcal{D}_{\text{test}}$, which may be caused by a small
 62 sample size or an over-complex model.

63 There is a trade-off relationship between the evaluation bias and variance from a squared evaluation bias, the derivation
 64 of the relationship is shown in the Eq. 23 in the Appendix. When performing K-fold CV with a fixed sample size
 65 and model complexity, the choice of K is the pivotal element shaping the model evaluation. When the K is set to a
 66 larger value; each training set $\mathcal{D}_{\text{train}}$ is larger in size, resulting in a model trained on a more representative subset of the
 67 population of interest, leading to lower bias. However, a large K comes with a trade-off: the corresponding test subset
 68 $\mathcal{D}_{\text{test}}$ is compressed in size, making the tested model more sensitive to the specific data points, and thus inflating the
 69 validation variance. Conversely, a smaller K, along with a minor training set $\mathcal{D}_{\text{train}}$, reduces their representativeness and
 70 increases bias. Nevertheless, a larger size of the test set $\mathcal{D}_{\text{test}}$ leads to more consistent estimations across the folds and,
 71 consequently, reduces the validation variance.

72 Leave-one-out cross-validation (LOOCV) is a variant of K-fold CV where K equals the sample size of the complete
 73 dataset \mathcal{D} . It provides an unbiased estimation of model performance because the training set $\mathcal{D}_{\text{train}}$ closely resembles the
 74 unseen population of interest, given its size of $N - 1$, where N is the sample size. However, as the trade-off discussion
 75 suggested, this method can lead to high validation variance due to the model is evaluated on one sample at a time. The
 76 true unbiased nature of LOOCV is fully realized only when all K folds are utilized. Performing an incomplete LOOCV
 77 can introduce significant bias because of the inherent high validation variance, which often occurs when training each
 78 model iteration is prohibitively time-consuming or computationally demanding. In specific contexts, such as genomic
 79 prediction, strategies like the one described by Cheng et al. leverage the matrix inverse lemma, which allows for
 80 computational savings by avoiding the inversion of large matrices in each fold. This technique significantly reduces
 81 the dependency of computational resources on the sample size [1]. Van Dixhoorn et al. exemplify the use of LOOCV
 82 with a small dataset, aiming to predict cow resilience with limited data resources [2]. Nevertheless, for large datasets,
 83 LOOCV is generally not recommended due to computational inefficiency. Further details of bias-variance trade-off
 84 have been extensively explored in the statistical literature [3, 4].

85 **1.3 Model Selection**

86 Model selection becomes necessary when models are not entirely determined by the data alone. For example, in a
87 regularized linear regression model such as a ridge regression [5] or the least absolute shrinkage and selection operator
88 (LASSO) [6], it is essential to define a regularization parameter, λ , before fitting the model to the data. A larger λ value
89 yields a more regularized model, which tends to reduce smaller coefficients to negligible values or zero. This approach
90 helps in preventing overfitting noise in the training data. The definition of loss functions for the regularized models
91 were described in 25 and 26 of the Appendix.

92 These pre-defined parameters, which influence model fitting and remain constant during the training process, are known
93 as hyperparameters. Beyond regularized models, hyperparameters are crucial in other predictive models, enhancing
94 flexibility and robustness. For example, in the Support Vector Regression (SVR) [7], the regressors X are projected
95 onto a linear subspace to approximate the target variable Y. By choosing a suitable kernel function, which transforms
96 the regressors into a non-linear space, as a hyperparameter, SVR can more effectively capture non-linear relationships,
97 thus significantly improving model performance. Another hyperparameter example is the number of latent variables in
98 the Partial Least Square (PLS) Regression [8], which condenses the original regressors into a more manageable set of
99 latent variables, reducing multicollinearity issues. Fewer latent variables might lose significant information from the
100 original regressors, while too many can lead to overfitting. Similarly, in Random Forest [9], hyperparameters such as
101 tree depth and the number of trees dictate model complexity. The same applies to the number of hidden layers and
102 the size of filters in convolutional neural networks [10]. All these examples highlight the fact that selecting the most
103 suitable hyperparameters, which is known as hyperparameter tuning, is crucial for optimizing model performance.
104 Feature selection is another crucial aspect of model selection. This process involves fitting the model to a selected
105 subset of the original features, particularly essential in high-dimensional data scenarios where the number of features
106 exceeds the number of observations, leading to poor model generalization. For instance, Ghaffari et al. sought to predict
107 health traits in 38 multiparous Holstein cows using a metabolite profiling strategy. Out of 170 metabolites, only 12
108 were identified as effective discriminators between healthy and over-conditioned cows and were thus selected for the
109 predictive model [11]. Therefore, optimizing feature subsets is a vital model selection strategy that significantly affects
110 model performance. Including the model selection process within the cross-validation is essential to avoid common
111 pitfalls. The risk of inflated model performance arises when model selection is guided by results on the test dataset.
112 Even if the chosen model is subjected to k-fold cross-validation afterward, its selection bias toward the test set can
113 lead to overestimating its efficacy. This issue has been highlighted in statistical literature [3]. A practical solution is to
114 divide the dataset into training, validation, and test sets. The validation set is then used for model selection, ensuring the
115 test set remains completely unused during the training phase, thereby providing a more accurate measure of model
116 performance. For instance, the study by Rovere et al. exemplifies best practices in hyperparameter tuning and feature
117 selection by employing an independent cross-validation step prior to assessing model performance. This approach
118 enabled the precise selection of relevant spectral bands from the mid-infrared spectrum and the optimal number of
119 latent dimensions in PLS with Bayesian regression for predicting the fatty acid profile in milk [12]. Similarly, Becker et

120 al. demonstrated a robust evaluation by using nested cross-validation loops; the inner loop conducted a grid search
121 for the best hyperparameters in logistic regression, while the outer loop was designed to evaluate the performance
122 of the resulting optimized model [13]. Both examples underscore the importance of separating model selection from
123 performance evaluation to ensure the validity and reliability of the results.

124 **1.4 Cross Validation Design with Block Effects**

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

144 **1.5 Model Performance Metrics**

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

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	$[0, 1]$
Root mean standard deviation ratio (RSR)	Error-Based	Yes	$[0, 1]$
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	$[-1, 1]$

151 1.5.1 Metrics in Regression Tasks

152 Regression models aim to predict continuous variables and are commonly employed in diverse applications, such as
 153 estimating body condition scores [19, 20], body weight [21, 22], milk composition [12, 18, 23, 24], efficiency of feed
 154 resource usage [16, 25, 26], and early-lactation behavior [2]. The metrics in regression tasks evaluate the agreement
 155 between the predicted value \hat{y} and the true values y . The agreement can be generally quantified in two ways: error-based
 156 metrics and linearity-based metrics. The metrics are summarized in Table 1. Error-based metrics focus on the deviation
 157 of each pair of predicted and true values, while linearity-based metrics consider overall linear relationships between the
 158 predictions and the truths. The root mean square error (RMSE) and the mean absolute error (MAE) are two common
 159 error-based metrics:

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

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

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

$$\text{RMSPE} = \frac{\text{RMSE}}{\bar{y}} \quad (3)$$

$$\text{RSR} = \frac{\text{RMSE}}{\sigma_y} \quad (4)$$

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

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

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

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

179 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
 180 of the actual response vector y and the predicted response vector \hat{y} , respectively. \bar{y} and $\bar{\hat{y}}$ are their respective means.
 181 Both r^2 and R^2 are scale invariant, meaning their values are unaffected by the scale of the observed data because they
 182 are normalized by the variation in the denominator.

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

191 The coefficient of determination R^2 quantifies model performance from the proportion of variance in the dependent
 192 variable that is predictable from the independent variables. It ranges from negative infinity to 1, where 1 indicates
 193 that the model explains all the variance in the dependent variable, and 0 indicates that the model performs no better

Table 2: Summary of model performance metrics for classification tasks.

Metric	Label-invariant	Threshold-independent
Accuracy	No	No
Precision	No	No
Recall	No	No
F1 score	No	No
Area under the precision-recall curve (AUC-PR)	No	Yes
Area under the receiver operating characteristic curve (AUC-ROC)	Yes	Yes
Matthews correlation coefficient (MCC)	Yes	Yes

194 than predicting all samples as the mean of the observed values. R^2 is useful in comparing multiple regression models,
 195 as demonstrated in studies that regress body weight of dairy cows on a set of morphological traits [22], examine
 196 the relationship between milk spectral profiles and nitrogen utilization efficiency [16], and evaluate the predictive
 197 performance of milk fatty acid composition [23].

198 It worth noting that many literatures have misinterpreted the relationship between r and R^2 . The coefficient of
 199 determination R^2 is not always equivalent to the square of the correlation coefficient r^2 . The equivalence only holds
 200 when the same dataset is used for both model fitting and evaluation in a least squares regression model. The model
 201 assumes a zero covariance between the fitted residual and the predicted values \hat{y} , and it also assumes that the residuals
 202 (i.e., prediction biases) are centered on zero. In practice when predictions are made on new data, those assumptions
 203 are often violated, leading to discrepancies between r^2 and R^2 . A details derivation of the equivalence is provided in
 204 Equation 27 28 in the Appendix.

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

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

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

213 1.5.2 Metrics in Classification Tasks

214 Classification models aim to predict categorical outcomes such as 'healthy' or 'sick,' 'susceptible' or 'resistant,' and
 215 'high yield' or 'low yield.' To evaluate classification performance, one must first establish a confidence threshold to

216 dichotomize the prediction probabilities. For instance, if a prediction probability exceeds the threshold, the sample is
 217 predicted as a positive sample. It is worth mentioning that this threshold is adjustable to fine-tune model performance
 218 for particular uses. The discussed metrics in classification tasks are summarized in Table 2.

219 Accuracy is the most straightforward metric for evaluating classification models:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Total Correct Predictions}}{\text{Total Predictions}} \\ &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \end{aligned} \quad (8)$$

220 where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives,
 221 respectively. It summarizes an overall model performance by calculating the proportion of correctly classified samples
 222 among all samples. Nonetheless, accuracy can be misleading when the classes are imbalanced. For example, if a study
 223 predicting the presence of a specific event, of which the prevalence was only 10%. In this case, a model that predicts all
 224 samples as negative would achieve an accuracy of 90%, which is misleadingly high. To address this issue, precision and
 225 recall are introduced:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{Total Predicted Positives}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FP}} \end{aligned} \quad (9)$$

$$\begin{aligned} \text{Recall} &= \frac{\text{TP}}{\text{Total Actual Positives}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FN}} \end{aligned} \quad (10)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

226 Precision and recall refine the assessment of a classification model by offering insights that accuracy alone may
 227 overlook. Precision calculates the fraction of true positives among all positive predictions, essentially measuring
 228 the trustworthiness of positive predictions made by the model (Eq. 9). High precision is crucial in scenarios where
 229 false positives incur significant costs, and false negatives are more tolerable. For instance, in contexts where clinical
 230 treatments and culling are expensive, such as detecting bovine tuberculosis [30] or mastitis [31] using non-invasive
 231 methods, a high-precision model is crucial to minimize unnecessary costs and interventions from false positives. On the
 232 other hand, recall, also known as sensitivity, quantifies the ratio of true positives to all actual positives, assessing the
 233 model's ability to identify positive cases (Eq. 10). High recall is essential where missing a positive case has serious
 234 consequences, or where false positives are easily rectifiable. For instance, detecting lameness or abnormal gait is crucial,

as these can indicate underlying pathologies [32] and impact welfare-related transport decisions [33]. An automated detection system [32, 34, 35] with high recall can mitigate economic losses by flagging at-risk cows. The benefit here lies in the feasibility of re-examining false positives, thus preventing more severe outcomes from undetected cases. Lastly, the F1 score, which is the harmonic mean of precision and recall, provides a balanced measure of model performance (Eq. 11). It is usually used as an overall performance metric when precision and recall are equally important.

However, it is worth emphasizing that precision and recall focus predominantly on positive samples. Inappropriately assigning a predominant background event as the positive class can lead to skewed interpretations. Hence, the Receiver Operating Characteristic (ROC) curve provides an another crucial tool for assessing a model's performance in a label-agnostic manner, meaning it is not biased by the class distribution as precision and recall are. An ROC curve plots one minus specificity against sensitivity. The equations for specificity and sensitivity are as follows:

$$\begin{aligned} \text{Specificity} &= \frac{\text{TN}}{\text{Total Actual Negatives}} \\ &= \frac{\text{TN}}{\text{FP} + \text{TN}} \end{aligned} \quad (12)$$

$$\begin{aligned} \text{Sensitivity} &= \text{Recall} \\ &= \frac{\text{TP}}{\text{Total Actual Positives}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FN}} \end{aligned} \quad (13)$$

A model's effectiveness, as depicted on the ROC curve, is gauged by how closely a point on the curve approaches the top-left corner. A steep ascent from the left side of the curve signifies the model's ability to correctly identify most true positives while incurring a low rate of false positives. A random guess, with a 50% chance of correct prediction, corresponds to a diagonal line on the ROC curve. In dairy science, the ROC curve has been extensively utilized, for example, in predicting mastitis from milk composition [36] and diagnosing pregnancy using spectroscopy technology [37]. In this hypothetical example, the ROC curve also demonstrates robustness and label-invariance with a consistent AUC of 0.875, regardless of whether the original or inverted labels are used.

In addition to the metrics, the Matthews Correlation Coefficient (MCC) provides a symmetrical measure of the quality of binary classifications. The MCC considers both positive and negative samples in the dataset, providing a balanced measure of a model's performance [38]. It is defined as:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (14)$$

256 The equation 14 symmetrically incorporates all four components of TP, TN, FP, and FN). This symmetry makes MCC
257 invariant to class distribution changes. The coefficient ranges from -1 to 1, where 1 indicates perfect classification,
258 0 indicates no better performance than random guessing, and -1 signifies total disagreement between prediction and
259 observation. In a case study that used feeding and daily activity behaviors to diagnose Bovine Respiratory Disease
260 in dairy calves, MCC proved particularly insightful [39]. The models in this study exhibited strong performance on
261 negative samples (i.e., healthy calves), which were more prevalent, resulting in high specificity. However, sensitivity
262 was relatively low at 0.54. In this context, MCC, with a value of 0.36, provided a more nuanced and representative
263 measure of model performance, especially given the skew towards negative samples.

264 **1.6 Study Objectives**

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

273 There are five simulation studies being conducted to address these challenges. The first simulation study will focus
274 on the bias-variance trade-off in CV, demonstrating how the choice of K in K-fold CV affects the evaluation bias and
275 variance. The second simulation study will investigate the impact of mistakenly using the same data for model selection
276 and evaluation, highlighting the inflated model performance. The third simulation study will explore the effect of
277 excluding block effects in CV, demonstrating how ignoring block effects can lead to over-optimistic model performance.
278 The fourth simulation study will present four hypothetical predictions made in the same regression tasks, leading to
279 different interpretations with different metrics. The fifth simulation study will demonstrate the impact of imbalanced
280 data on classification model evaluation, showing how the choice of metrics can lead to misleading conclusions. Overall,
281 this series of simulation studies aims to guide researchers in accurately and consistently reporting model performance,
282 thereby supporting integrity and scientific rigor in prediction modeling research.

283 **2 Materials and Methods**

284 **2.1 Study 1: Evaluateion bias and variance of cross-validation**

285 This study investigated the interplay between sample size and various performance estimators and their collective
286 impact on bias and variance during model validation. It is hypothesized that increasing the sample size will reduce
287 both bias and variance. Additionally, it is expected that the validation variance will increase with the number of folds

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

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

$$\begin{aligned} \text{Training: } Y_{-k} &= f_{\mathcal{D}_{-k}}(X_{-k}) + \epsilon \\ &= X_{-k}\beta + \epsilon \\ \text{Testing: } \hat{Y}_k &= f_{\mathcal{D}_{-k}}(X_k) \\ &= X_k\beta \quad k = 1, 2, \dots, K \end{aligned} \tag{15}$$

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

$$\begin{aligned} \text{Training: } Y &= f_{\mathcal{D}}(X) \\ &= X\beta + \epsilon \\ \text{Testing: } \hat{Y} &= f_{\mathcal{D}}(X) \\ &= X\beta \end{aligned} \tag{16}$$

Where:

- X denotes the input regressors sampled from a standard normal distribution $\mathcal{N}(0, 1)$ with dimensions $N \times 10$.
- Y denotes the target variable sampled from a standard normal distribution $\mathcal{N}(0, 1)$ with dimensions $N \times 1$.
- X_{-k} and Y_{-k} are the input regressors and target variable in the training set \mathcal{D}_{-k} .
- X_k denotes the input regressors in the test set \mathcal{D}_k .
- \hat{Y}_k denotes the predicted target variable in the test set \mathcal{D}_k .

- 309 • β denotes the estimated regression coefficient with dimensions 10×1 .
 310 • ϵ denotes the error term assumed to be normally distributed.

311 Estimated performance $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$ was derived by averaging the performance metrics across all K folds as per Eq. 20.
 312 The bias and variance of the evaluation were calculated using Eqs. 21 and 22, respectively. To approximate true
 313 model performance $G(f_{\mathcal{D}})$, a hundred unseen datasets \mathcal{D}^* were generated identically to \mathcal{D} , and the performance $G(f_{\mathcal{D}})$
 314 was estimated by averaging the performance metrics across all \mathcal{D}^* . The detailed steps to compute evaluation bias and
 315 variance are provided in the supplementary materials.

316 **2.2 Study 2: Model Selection in Cross-Validation**

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

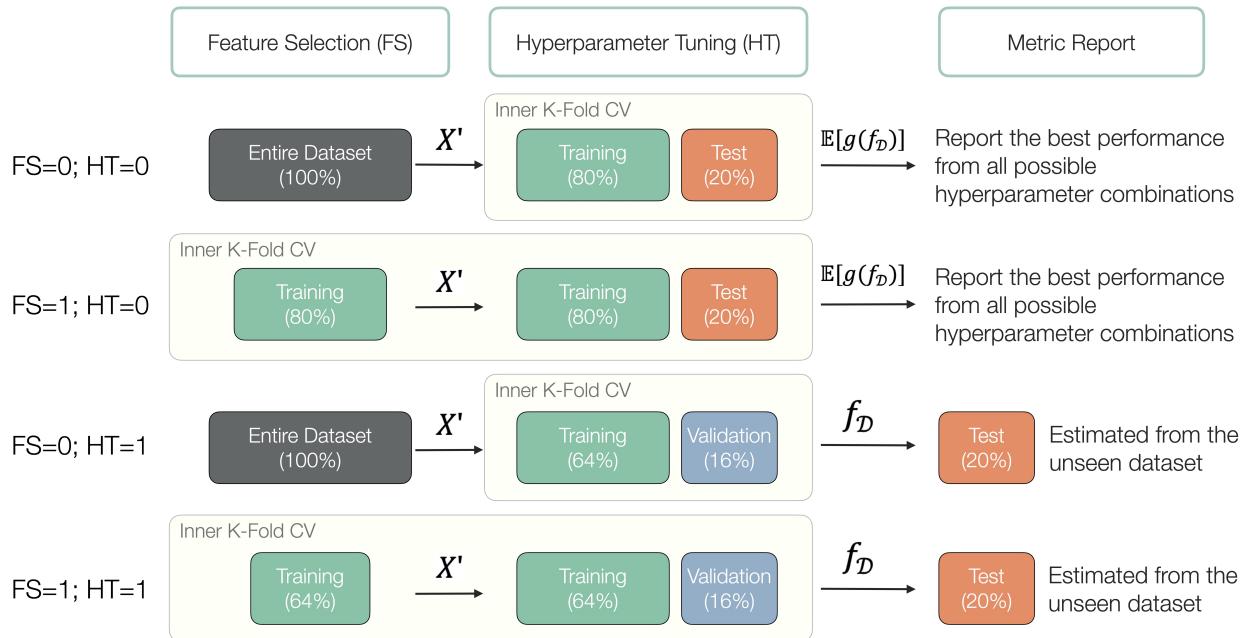


Figure 1: Workflow diagram illustrating four cross-validation strategies of feature selection (FS) and hyperparameter tuning (HT), where 0 denotes incorrect implementation and 1 indicates correct practice. X' is the selected feature subset, $\mathbb{E}[\hat{g}(f_{\mathcal{D}})]$ is the expected generalization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

325 This study introduces notations FS for feature selection and HT for hyperparameter tuning, assigning a binary indicator
326 (0 or 1) to denote incorrect (0) or correct (1) implementation of model selection. This yields four possible combinations
327 of model selection strategies: “FS=0; HT=0”, “FS=0; HT=1”, “FS=1; HT=0”, “FS=1; HT=1” (Figure 1). When
328 FS=0, feature selection precedes cross-validation splitting. If FS=1, feature selection occurs within each fold of the
329 training set during cross-validation. With hyperparameter tuning, a correct implementation (HT=1) involves splitting
330 the dataset into training (64%), validation (16%), and test (20%) sets. The model is trained and tuned using the training
331 and validation sets, respectively, while the test set is reserved for a single evaluation of model performance. Conversely,
332 with HT=0, only training (80%) and test (20%) sets are used, risking validation bias as the test set informs both
333 training and performance reporting. A 5-fold cross-validation approach was deployed for all strategies. Validation
334 bias is measured as the discrepancy between the model selection-influenced performance estimate and the expected
335 generalization performance ($r=0$), using the Pearson correlation coefficient between predicted and observed values.
336 Over 1000 sampling iterations, the study assesses the distribution of validation bias. A t-test will determine whether the
337 validation bias significantly deviates from zero.

338 **2.3 Study 3: Block Effects in Cross-Validation**

339 The objective of the study is to demonstrate how a Random CV, which randomly assigns the samples to folds without
340 considering the block effects, could overestimate the model performance. This study also conducts a block CV, where
341 each block is used as a fold in the cross-validation, as the benchmark. The hypothesis is that the model performance
342 estimated by Random CV is significantly higher than the estimation by block CV. This study simulated a regression
343 task with 100 instances across ten features, denoted as X, and one single response variable, Y. Both X and Y are
344 derived from a standard normal distribution. To introduce a block factor, the study groups every 20 observations into a
345 block, with each block incrementally increasing by b units from zero, where b was simulated from 0.5 to 3.0 with an
346 increment of 0.5. Within these ten features, one is substituted as the block level, represented by an integer from 0 to 4,
347 augmented with random noise drawn from a standard normal distribution. This setup aims to simulate a scenario where
348 the predictors primarily capture block variation, given the null expectation in predictability when using ten random
349 variables X to forecast another random variable Y. The study investigates two model validation strategies: Block CV
350 and Random CV, both utilizing a 5-fold cross-validation method. In block CV, each block serves as a separate fold,
351 while in Random CV, samples are randomly allocated to each fold (Figure 2). The predictive model is linear regression,
352 and the performance is evaluated using Pearson’s correlation coefficient. This simulation runs for 1000 iterations, with
353 X and Y being resampled in each cycle. A one-tailed t-test assesses if the mean estimated performance significantly
354 exceeds zero. Additionally, an Analysis of Variance (ANOVA) table is calculated when b is 0.5 to ascertain if the
355 simulated block variation notably exceeds the assumed individual variation, representing the primary interest.

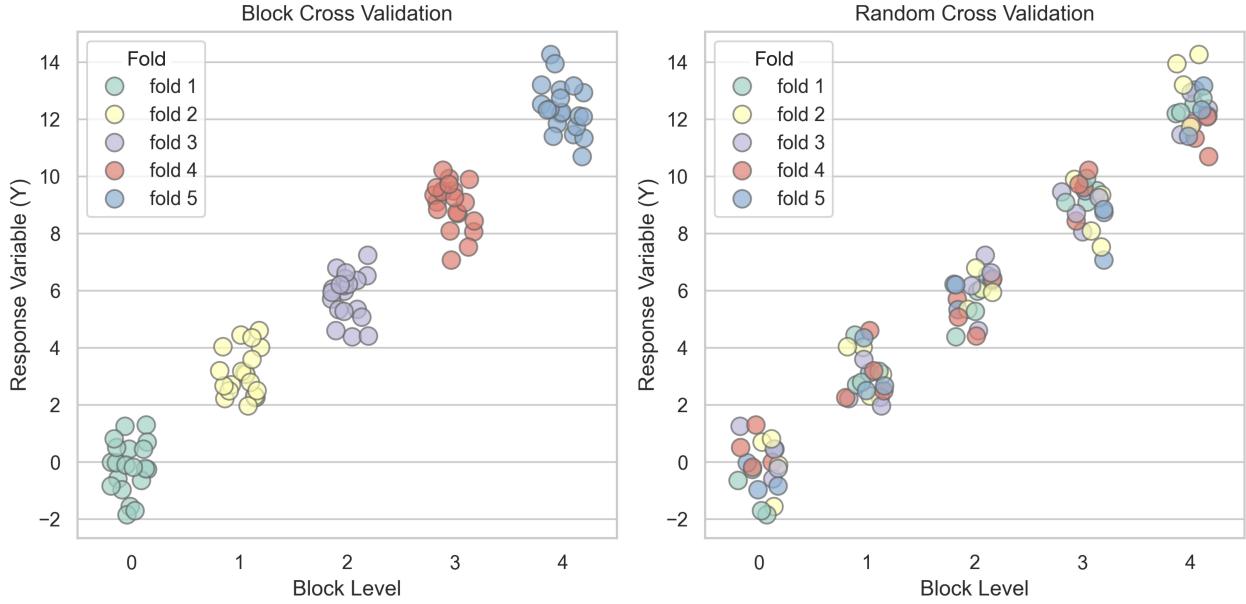


Figure 2: Illustration of fold assignment in block cross validation (left) and random cross validation (right). Folds are color-coded, and the block effect is set to 3 in this example.

356 2.4 Study 4: Performance Metrics in Regression Tasks

357 This study explores two error-based metrics, Root Mean Squared Error (RMSE) and Root Mean Squared Percentage
 358 Error (RMSPE), and three linearity-based metrics, Pearson Correlation Coefficient (r), the Coefficient of Determination
 359 (R^2), and the Concordance Correlation Coefficient (CCC), in a variety of commonly-encountered data challenges.
 360 These data challenges are depicted through 4 scenarios, representing data commonly encountered in predictive
 361 applications varying in scope (scenarios 1 and 2), data with outliers disrupting the scale of prediction (scenario 3), and
 362 data with an underlying grouping structure (scenario 4). The statistical description of the approach to generating each
 363 of these scenarios is included below. Practical examples of real-world instances of these types of data challenges are
 364 also described.

365 In the hypothetical example depicted in Figure 7, 100 observations were generated from two separate normal
 366 distributions. The first 50 observations were drawn from a normal distribution with a mean of -3 and a standard
 367 deviation of 1, denoted as $\mathcal{N}(-3, 1)$. The remaining 50 observations were generated from another normal distribution,
 368 $\mathcal{N}(3, 1)$. Utilizing two distinct distributions served to simulate experimental block effects, preset at a magnitude of 6
 369 units for this experiment. Based on the simulated observations, four scenarios of predictions were derived according to
 370 the setting below:

- 371 • Scenario 1: To establish a correlation relationship, the observations were multiplied by 0.3, followed by the
 372 addition of random noise $\mathcal{N}(0, 0.7)$ to introduce prediction errors. This scenario represents a “best case” for
 373 developing predictive analytics, and could be exemplary of scenarios like predicting a scaled performance

374 response (i.e., milk yield, average daily gain) from measurable input variables like dry matter intake, sensor
375 system data, or past performance data.

- 376 • Scenario 2: The prediction outcome from Scenario 1 was multiplied by 5, simulating predictions with a larger
377 variance while maintaining the same relative order as the original predictions. There are some responses
378 that have naturally greater proportional variation compared with others. For example, an animal's body core
379 temperature is unlikely to vary by more than 5%; however, daily variation around measurements like feed
380 intake can range upwards of 30 to 40%. Comparison of scenarios 1 and 2 explore how this natural variation
381 should be included in interpreting predictive analytics.
- 382 • Scenario 3: only the top 10% of predictions that deviate the most from zero in Scenario 1 were raised to the
383 power of 5. The rest of the predictions were set to zero. This scenario simulates a prediction that focuses
384 solely on the extreme samples. In disciplines like nutritional exploration, the emphasis of predictive analytics
385 typically focuses on understanding the mean animal or the mean response of an individual animal; however,
386 in predictive analytics focused on health or genetic merit, the emphasis of prediction is often on the extreme
387 observations. Analytics to understand the extreme observations is always complicated by the question of
388 whether extremes are due to true outliers or some sort of measurement error. As precision livestock farming
389 advances, the opportunities for measurement error due to erroneous sensor measurements increases.
- 390 • Scenario 4: Values sampled from two normal distributions, $\mathcal{N}(-3, 1)$ and $\mathcal{N}(3, 1)$, were added respectively
391 to the predictions made in Scenario 1 of Block A (colored orange in Figure 1) and Block B (colored green in
392 Figure 1). In the animal sciences we often rely on blocks as an experimental tool to support analytics given
393 challenging experimental design or constrained animal units. Many times, the difference between blocks
394 dwarfs the differences observed within a block, resulting in a masking of true effects due to the block influence.
395 This scenario amplified the original block effects, simulating a model that effectively distinguished between
396 different blocks (e.g., herd or breed) but was less capable of predicting individual variations within each
397 block. An example of this scenario might be simulating milk production or body weight across species –
398 the magnitude of the difference between sheep and cattle (for example) far outweighs the magnitude of the
399 difference of sheep or cattle over time.

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

402 **2.5 Study 5: Performance Metrics in Classification Tasks**

403 This study presents a hypothetical example to highlight how the choice of different performance metrics can lead to
404 different interpretations of a model's effectiveness. The example focuses on binary classification, where the outcome is
405 either positive ($Y=1$) or negative ($Y=0$). Suppose a binary classification model always outputs a probability between 0
406 and 1, indicating the likelihood that a sample belongs to the positive class. This example assumes that the model has

407 high confidence in correctly predicting 1 out of 4 positive and 5 out of 6 negative samples. This example intends to
408 illustrate a scenario where the positive outcome is rare, such as predicting the onset of a calving event in dairy cows
409 [40, 41]. The example data is shown in Figure 8. In addition to the original labels, this example also examines a
410 scenario with inverted labels (Figure 8. Upper). Since most classification metrics prioritize positive samples, it is
411 generally advisable to designate the event of interest as the positive class in binary classification problems. Inverting
412 the labels illustrates the potential overestimation of model performance when the more common, but less significant,
413 background event is mistakenly marked as the positive class. It is important to note that inverting the labels in this
414 example only affects the interpretation of model performance, not the model configuration or parameters. To evaluate
415 classification performance, one must first establish a confidence threshold to dichotomize the prediction probabilities.
416 For instance, if a prediction probability exceeds the threshold, the sample is labeled positive. By default, the threshold
417 is set at 0.5 for its simplicity. For example, in the third data row of the example data: With a prediction probability
418 of 0.38 that falls below the threshold, the sample is deemed negative, resulting in a false negative classification since
419 the ground truth is positive. It is worth mentioning that this threshold is adjustable to fine-tune model performance for
420 particular uses. A confusion matrix (Figure 8. Lower), effectively encapsulates prediction outcomes. The rows in this
421 2x2 matrix correspond to ground truth, while its columns reflect predictions. Correct predictions populate the diagonal
422 cells, and errors fill the off-diagonal ones. This matrix serves as the foundation for computing various metrics to assess
423 model performance, which will be explored in the result sections.

424 3 Results and Discussion

425 3.1 Study 1: The Impact of Estimator Choice and Sample Size on Model Evaluation Reliability

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

436 Considering LOOCV's singular data point testing, its evaluation variance is pertinent only for RMSE, which permits
437 single data point evaluations. Figure 4 illustrates the bias and variance in RMSE across different performance estimators
438 as a function of sample size N. Both bias and variance in RMSE decrease as sample size increases, aligning with the
439 hypothesis. LOOCV provides the least biased estimation, while 2-fold CV exhibits the highest bias without significant

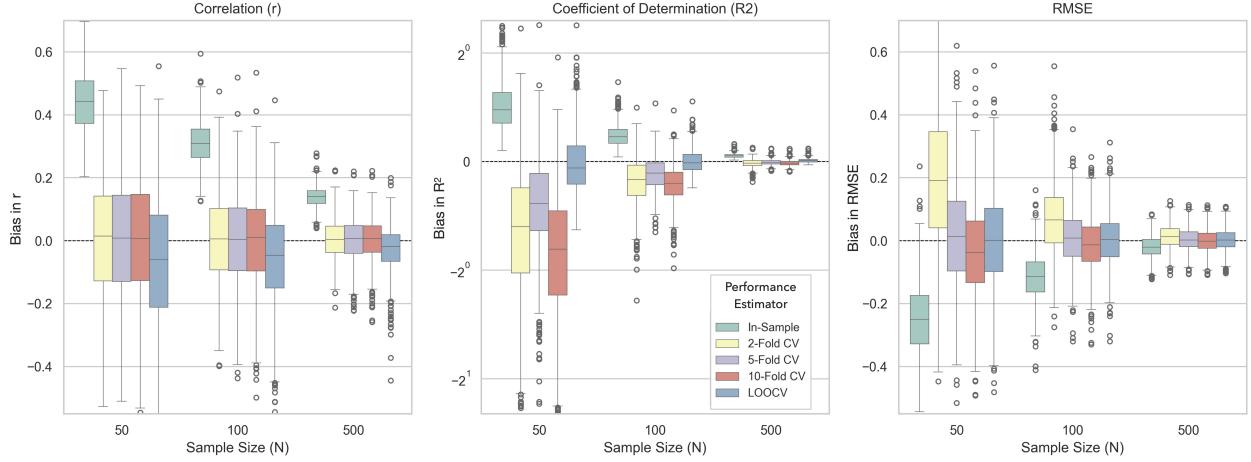


Figure 3: Simulation results of evaluation bias from 1000 sampling iterations. Multiple performance estimators across different sample sizes were color-coded. Three metrics: r , R^2 , and RMSE, were displayed in the column facets.

440 reduction at larger sample sizes. However, biases across all estimators converge at a sample size of 500. In terms of
 441 evaluation variance, LOOCV consistently shows higher values than other estimators for all sample sizes. Additionally,
 442 a lower number of folds K correlates with reduced variance, which is also in line with the hypothesized trend.

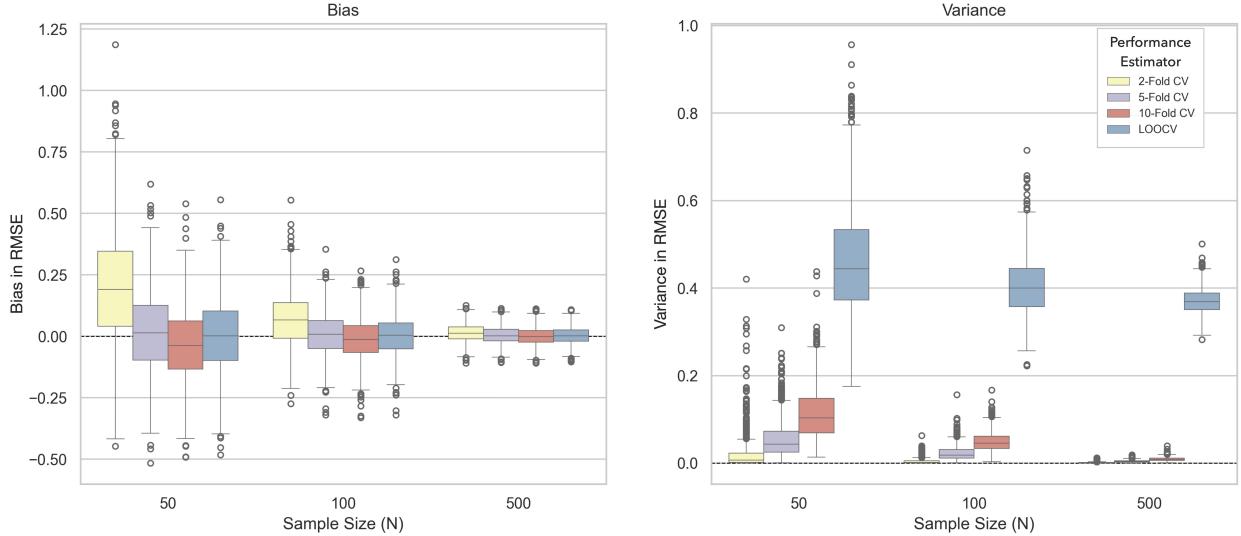


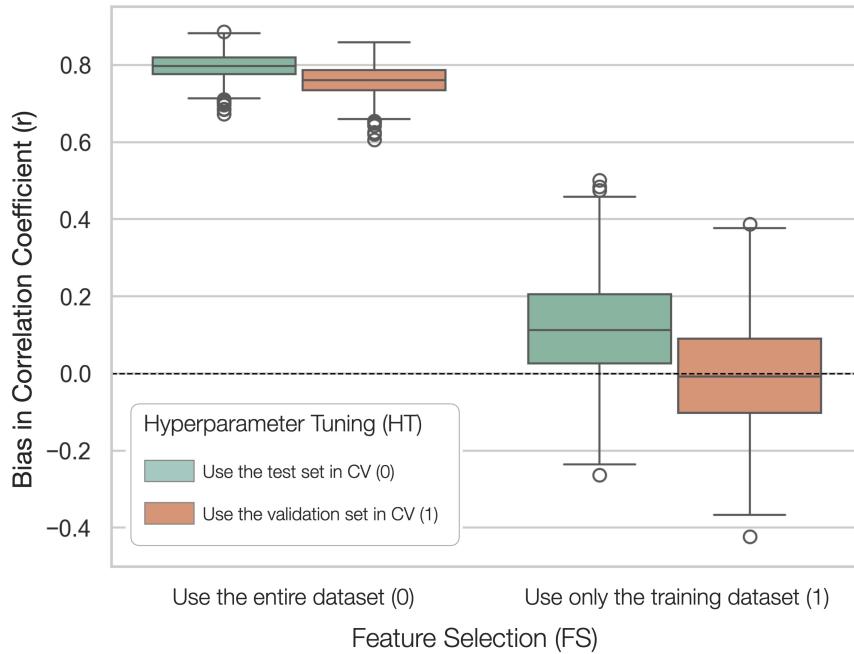
Figure 4: Simulation results of evaluation bias and variance from 1000 sampling iterations. Multiple performance estimators across different sample sizes were color-coded. Only RMSE was displayed. Bias and variance were listed in the left and right facets, respectively.

443 In conclusion, when conducting model evaluation, it is crucial to consider the estimator and sample size, as they
 444 significantly influence evaluation reliability which can be decomposed into bias and variance. Larger sample sizes
 445 generally lead to reduced bias and variance, enhancing the reliability of the evaluation process. For unbiased performance
 446 estimation, CV methods, such as K-fold CV and LOOCV, are preferable to in-sample estimation. LOOCV often
 447 provides less biased estimations for certain metrics but can exhibit higher variance. It is also noteworthy that the number
 448 of folds in K-fold CV can affect bias and variance; thus, experimenting with different numbers of folds, especially

449 in smaller sample sizes, can be beneficial. Ultimately, the selection of appropriate evaluation techniques should be
 450 tailored to the specific context of the dataset and the objectives of the modeling exercise, ensuring a robust and reliable
 451 assessment of model performance.

452 **3.2 Study 2: Misuse of Model Selection Can Lead to Over-Optimistic Performance Estimates**

453 The evaluation bias was visualized using box plots (Figure 5), with the feature selection factor (FS) on the x-axis and
 454 hyperparameter tuning (HT) distinguished by color — green for incorrect and yellow for correct implementation. The y-
 455 axis represents the evaluation bias as measured by the correlation coefficient. The results indicate a clear overestimation
 456 of model performance when feature selection is applied to the entire dataset, regardless of hyperparameter tuning. The
 457 median biases were 0.797 for “FS=0; HT=0” and 0.761 for “FS=0; HT=1”. Moreover, inappropriate evaluation in
 458 hyperparameter tuning resulted in a significant bias (p -value < 0.001) with a median of 0.113 for “FS=1; HT=0”. The
 459 only scenario without bias significantly occurred when both feature selection and hyperparameter tuning were correctly
 460 incorporated within the cross-validation process “FS=1; HT=1”, yielding a median bias of -0.008. These findings align
 461 with the initial hypothesis and the prevailing literature, reinforcing that model selection must be integrated into the
 462 cross-validation workflow to prevent an overestimation of model performance.



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

464 The simulation results robustly confirm the hypothesis that improper implementation of model selection inflates
 465 performance estimates. Specifically, the evaluation bias is markedly high when feature selection precedes data splitting,
 466 with or without correct hyperparameter tuning. Although integrating feature selection within cross-validation folds
 mitigates this bias, incorrect hyperparameter tuning still significantly skews performance metrics. Notably, this

Table 3: ANOVA results for a single iteration of the simulated data with $b = 0.5$. SS: sum of squares; DF: degree of freedom; MS: mean square; F: F-statistic

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

467 overestimation from the hyperparameter tuning is even more pronounced in complex models, such as neural network
 468 architectures that often entail over a million parameters. These findings underscore the necessity of meticulous cross-
 469 validation practices, particularly for feature selection and hyperparameter tuning, to ensure accurate performance
 470 estimations and generalizability in predictive modeling.

471 3.3 Study 3: Overlooking Experimental Block Effects Can Lead to Biased Model Performance Estimates

472 In this simulation, an ANOVA table (Table 3), calculated from a single iteration for illustrative purposes, demonstrates
 473 that the simulated data exhibits block variation significantly greater than the residual variance. The result (Figure 6)
 474 shows that regardless of the amplitude of block effects in this simulation study, the Block CV strategy consistently yields
 475 a mean performance estimate close to zero, while the Random CV strategy consistently and significantly overestimates
 476 the model performance ($p\text{-value} < 0.001$). This finding supports the hypothesis that Random CV tends to overestimate
 477 model performance when block variation predominates over residual variation.

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

483 3.4 Study 4: Different Regression Metrics Illustrate Different Aspects of Model Performance

484 The simulated hypothetical example in Figure 7 illustrates the performance of four different prediction scenarios.
 485 When the goal is to rank observations of interest rather than predict the absolute magnitude of the error, this metric is
 486 appropriate. The property of this metric was demonstrated in both Scenario 1 and 2, where the coefficient remained
 487 consistent despite Scenario 2 having errors five times greater than in Scenario 1. If the absolute error is of interest,
 488 this metric should be used in conjunction with other metrics, such as RMSE or R^2 . It is also worth noting that this
 489 metric can provide a value of 0.27 in Scenario 3, where 90% of the predictions failed to capture the trend and resulted
 490 in zero-value predictions. The positive performance of the metric came from the predictions ranking the remaining 10%
 491 of the observations in a fairly accurate order, regardless of the large error magnitude. Moreover, one common pitfall
 492 of this metric is that block effects can influence it, leading to an inflated performance estimate if individual variation
 493 is of greater interest than inter-block variation. This was demonstrated in Scenario 4, where the overall coefficient r

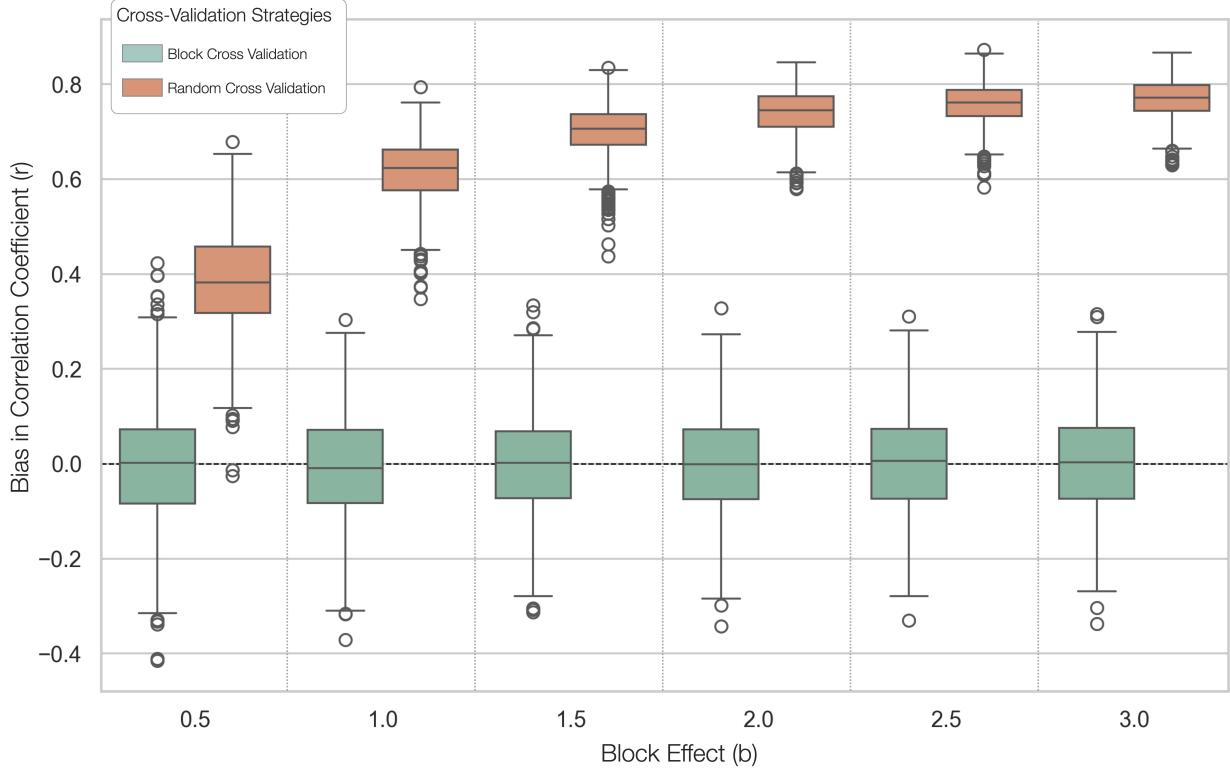


Figure 6: Bias in model performance estimation by Block CV and Random CV across 1000 iterations. The dashed line represents the null hypothesis that the mean performance estimate is zero.

494 was 0.94, but the metric within each block was only 0.33 and 0.25, respectively. Therefore, it is essential to visually
 495 inspect regression results through scatter plots or examine them within individual blocks. Distinct from the correlation
 496 coefficient r , RMSE is sensitive to scale, implying that achieving predictions with a variance akin to the observed values
 497 takes precedence over maintaining their order or trend. This property is evident in Scenario 2, where the RMSE inflates
 498 from 2.41 to 3.63, despite the fact that the predictions in both scenarios rank the observations identically. Another
 499 notable characteristic of RMSE is it weighs more on large errors, which is essential when making a large error is
 500 costly and should be prioritized for avoidance. In Scenario 3, where certain predictions deviate substantially from the
 501 majority, the squaring operation in Equation 1 accentuates these outliers, culminating in an RMSE value of 25.49.
 502 It is also worth mentioning that RMSE is impervious to block effects, which was illustrated in Scenario 4. In this
 503 scenario, both the complete set of predictions and the intra-block predictions yield similar RMSE values—1.49, 1.46,
 504 and 1.52, respectively. This phenomenon emphasizes again that RMSE is affected solely by the magnitude of the error,
 505 which neglects the ability of the model to capture relative trends in intra-block or inter-block predictions. In Scenario
 506 1, a moderate R^2 of 0.47 indicates that the model explains nearly half of the observed variation. With a five times
 507 larger variance compared to Scenario 1, Scenario 2 yields a negative R^2 of -0.21 despite retaining the same prediction
 508 order. Similarly to RMSE, where errors are squared, the presence of outliers leads to a dramatic R^2 drop to -58.83,
 509 showcasing the sensitivity of R^2 to outlier-induced variance. Lastly, in Scenario 4, the value of R^2 indicated a strong
 510 performance by the model with a score of 0.80. This score is statistically reasonable, as the model explained 80% of the

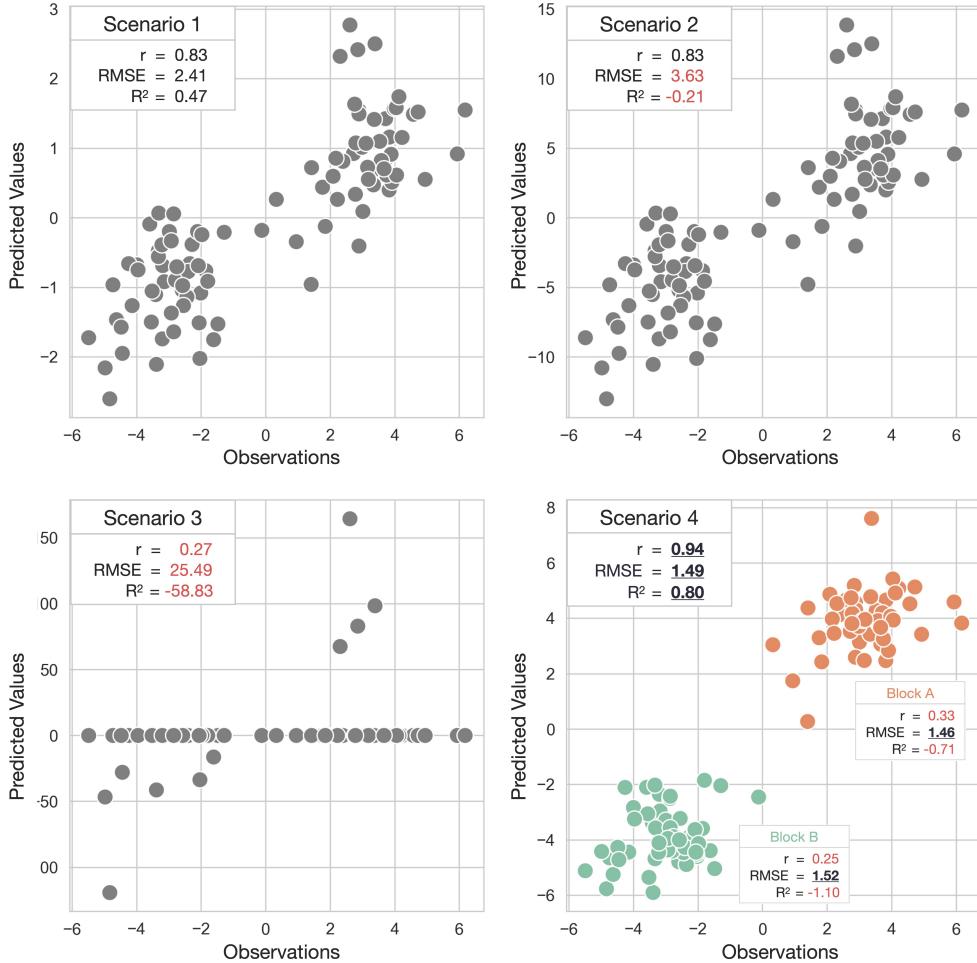


Figure 7: Scatter plots display the same observations against four different prediction scenarios in the given hypothetical example. Scenario 1 serves as a baseline for the metrics, with any metric better than the baseline highlighted in bold and underscored, and any worse metric colored in red.

511 total variation, which was mainly contributed by block effects. However, when each block was analyzed individually,
 512 the R^2 values decreased to -0.71 and -1.10, respectively, because the model failed to account for intra-block variation.
 513 In summary, while both RMSE and R^2 aim to measure prediction errors, R^2 offers additional statistical insights that
 514 facilitate a more nuanced evaluation of model performance. All three metrics discussed in this section are effective in
 515 examining the performance of regression models. Correlation Coefficient r evaluates the ability of the model to rank the
 516 observations. RMSE is a metric that is easy to interpret and is suitable to evaluate the absolute error. The coefficient of
 517 determination R^2 measures the proportion of variation that the model can explain. Choosing the appropriate metric
 518 hinges on the specific goals of model evaluation. For a thorough analysis, employing multiple metrics is advisable to
 519 gain a multifaceted view of the model's performance.

Ground Truth and Prediction Probability		Original Labels		Inverted Labels	
ID		Ground Truth	Prediction Probability	Ground Truth	Prediction Probability
1		(+)	0.99	(-)	0.01
2		(-)	0.70	(+)	0.30
3		(+)	0.38	(-)	0.62
4		(+)	0.33	(-)	0.67
5		(+)	0.26	(-)	0.74
6		(-)	0.16	(+)	0.84
7		(-)	0.15	(+)	0.85
8		(-)	0.14	(+)	0.86
9		(-)	0.12	(+)	0.88
10		(-)	0.07	(+)	0.93

Confusion Matrixes		Original Labels @0.50	Inverted Labels @0.50																																													
<table border="1"> <thead> <tr> <th colspan="2"></th> <th colspan="2">Predictions</th> </tr> <tr> <th colspan="2"></th> <th>(+)</th> <th>(-)</th> </tr> <tr> <th rowspan="2">Ground Truth</th> <th>(+)</th> <td>TP</td> <td>FN</td> </tr> </thead> <tbody> <tr> <th>(-)</th> <td>FP</td> <td>TN</td> </tr> </tbody> </table> <p style="text-align: center;">↓</p> $\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$			Predictions				(+)	(-)	Ground Truth	(+)	TP	FN	(-)	FP	TN	$\text{Recall or Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$ $\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}$	<table border="1"> <thead> <tr> <th colspan="2"></th> <th colspan="2">Predictions</th> </tr> <tr> <th colspan="2"></th> <th>(+)</th> <th>(-)</th> </tr> <tr> <th rowspan="2">Ground Truth</th> <th>(+)</th> <td>1</td> <td>3</td> </tr> </thead> <tbody> <tr> <th>(-)</th> <td>1</td> <td>5</td> </tr> </tbody> </table> <p style="text-align: center;">Accuracy@0.50 = 0.600 Precision@0.50 = 0.500 Recall@0.50 = 0.250 MCC@0.50 = 0.100 PR AUC = 0.767 ROC AUC = 0.875</p>			Predictions				(+)	(-)	Ground Truth	(+)	1	3	(-)	1	5	<table border="1"> <thead> <tr> <th colspan="2"></th> <th colspan="2">Predictions</th> </tr> <tr> <th colspan="2"></th> <th>(+)</th> <th>(-)</th> </tr> <tr> <th rowspan="2">Ground Truth</th> <th>(+)</th> <td>5</td> <td>1</td> </tr> </thead> <tbody> <tr> <th>(-)</th> <td>3</td> <td>1</td> </tr> </tbody> </table> <p style="text-align: center;">Accuracy@0.50 = 0.600 Precision@0.50 = 0.625 Recall@0.50 = 0.833 MCC@0.50 = 0.100 PR AUC = 0.941 ROC AUC = 0.875</p>			Predictions				(+)	(-)	Ground Truth	(+)	5	1	(-)	3	1
		Predictions																																														
		(+)	(-)																																													
Ground Truth	(+)	TP	FN																																													
	(-)	FP	TN																																													
		Predictions																																														
		(+)	(-)																																													
Ground Truth	(+)	1	3																																													
	(-)	1	5																																													
		Predictions																																														
		(+)	(-)																																													
Ground Truth	(+)	5	1																																													
	(-)	3	1																																													

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

520 3.5 Study 5: Label-Invariant Metrics Provide Balanced Assessment in Binary Classification

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

529 quarter of the actual positives are detected. This contrasts with an accuracy of 0.6, which may appear misleadingly high
 530 due to the abundance of negative samples. Additionally, it is noted that the chosen confidence threshold significantly
 531 impacts precision and recall. While the trade-off between these two metrics is not always linear, it is generally observed
 532 that a higher threshold increases precision but decreases recall, and vice versa. A high threshold indicates a conservative
 533 approach in predicting positives, reducing false positives, and thus enhancing precision. However, this often leads to
 534 missing actual positive cases, lowering recall. Hence, the Precision-Recall (PR) curve is an essential tool for evaluating
 535 model performance across various thresholds. Plotted with recall on the x-axis and precision on the y-axis, this curve is
 536 derived by computing these metrics at different thresholds (Figure 9, Left). The Area Under the Curve (AUC) provides
 537 a summary measure of the PR curve's overall performance. A model's effectiveness is generally indicated by how close
 538 a point on the PR curve is to the top-right corner. For example, at a threshold of 0.25, which is positioned near the
 539 top-right of the PR curve, the model demonstrates impressive performance with an accuracy of 0.90, precision of 0.80,
 540 and recall at 1.00.

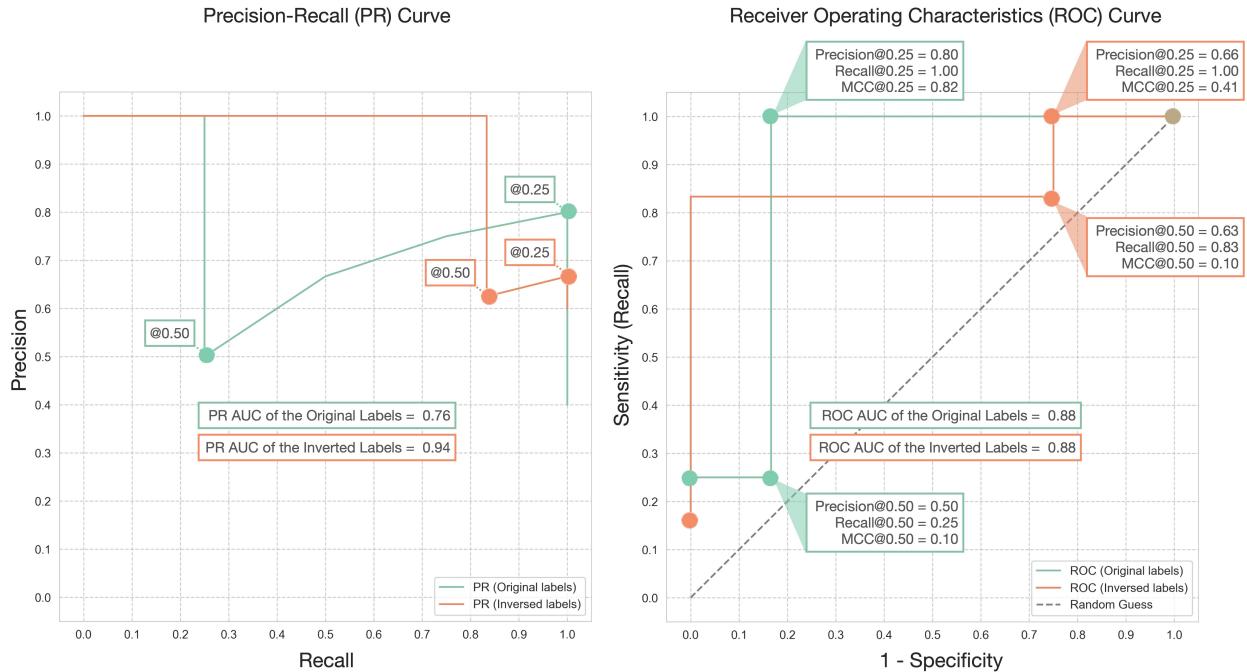


Figure 9: **(Left)** Precision-recall (PR) curve and **(Right)** Receiver operating characteristic (ROC) curve for the hypothetical example are displayed. The performance at confidence thresholds of 0.25 and 0.50 is highlighted. Original labels are marked in green, while inverted labels appear in orange.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

541 However, it is worth re-emphasizing that precision and recall focus predominantly on positive samples. Inappropriately
 542 assigning a predominant background event as the positive class can lead to skewed interpretations. This pitfall is
 543 demonstrated in this example by inverting the labels. At a threshold of 0.50, precision increases from 0.50 to 0.63, and
 544 recall jumps from 0.25 to 0.83. With the threshold set at 0.25, precision drops to 0.66 from 0.80, while recall remains
 545 unchanged. The PR AUC also rises from 0.76 to 0.94. Such shifts in metrics, driven merely by label rearrangement
 546 unrelated to the data or model characteristics, underscore the importance of label-invariant metrics that remain unaffected

547 by label assignments. Unlike metrics focusing solely on positive samples, the ROC curve accounts for both positive
548 and negative samples, making it a label-invariant metric. Specificity is plotted on the x-axis and sensitivity on the
549 y-axis, calculated at different thresholds (Figure 9, Right). In this hypothetical example, the ROC curve demonstrates
550 robustness and label-invariance with a consistent AUC of 0.875, regardless of whether the original or inverted labels are
551 used. Lastly, another label-invariant metric is MCC which provides a balanced assessment of both positive and negative
552 samples. Considering MCC's balanced approach to evaluating model performance, this study introduces the concept
553 of an MCC curve. This curve, which plots the MCC value against various threshold levels (Figure 10), serves as a
554 powerful tool for identifying the optimal confidence thresholds for model predictions. By examining this curve, one
555 can determine the specific threshold at which the MCC value peaks, thereby optimizing the model's performance. For
556 example, when applied to the hypothetical example, the optimum MCC value of 0.82 was attained at a threshold of 0.25.
557 This particular threshold corresponded to accuracy, precision, and recall values of 0.90, 0.75, and 1.00, respectively.
558 Notably, the MCC curve retains its symmetry even when labels are reversed, affirming its status as a label-invariant
559 measure. In scenarios with inverted labels, the maximum MCC value observed was 0.83, achieved at a threshold of
560 0.75, leading to accuracy, precision, and recall values of 0.90, 1.00, and 0.83, respectively. Such findings underscore the
561 MCC's ability to provide a balanced and comprehensive assessment of both positive and negative samples, thereby
562 reinforcing its utility as a versatile and effective metric for thorough model evaluation.

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

570 4 Conclusion

571 In summary, the review highlights several key considerations for performance assessment and validation in predictive
572 modeling. When evaluating regression models, the choice of metrics like Correlation Coefficient r , RMSE, and R^2
573 depends on the specific goals of the model. A comprehensive evaluation should include multiple metrics to understand
574 different aspects of model performance. In binary classification models, precision and recall are crucial, but it is
575 essential to correctly designate the positive class to avoid bias. Label-invariant metrics, such as the ROC curve and the
576 proposed MCC curve, provide a balanced assessment, unaffected by class label choices. Additionally, the reliability of
577 model validation is significantly influenced by estimator choice and sample size. Larger sample sizes tend to reduce
578 bias and variance, increasing validation reliability. Cross-validation methods, such as K-fold CV and LOOCV, are
579 preferable for unbiased performance estimation, with the number of folds in K-fold CV being particularly influential
580 in smaller datasets. Moreover, the review underscores the importance of correct implementation in model selection

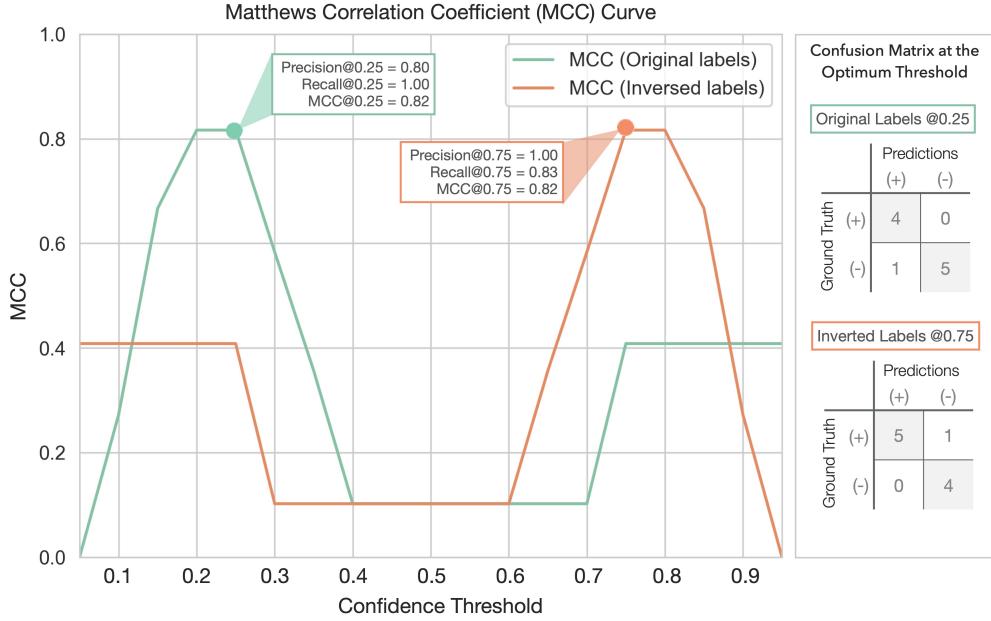


Figure 10: Matthews Correlation Coefficient (MCC) curve. A line chart plotting MCC at different thresholds for the hypothetical example. The optimal threshold is highlighted by the dot marks in green and orange for the original and inverted labels, respectively. The confusion matrix at the optimal threshold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

581 processes, as improper techniques can inflate performance estimates. This is especially true in complex models where
 582 feature selection and hyperparameter tuning need meticulous cross-validation to avoid overestimation of performance.
 583 Finally, the utility of Block CV is emphasized in contexts where block effects are significant. It provides a more realistic
 584 assessment of model generalizability and accuracy compared to a Random CV, which tends to overestimate performance
 585 in such scenarios. Overall, the review recommends a thoughtful selection of metrics and validation techniques, tailored
 586 to the specific dataset and modeling objectives, to ensure accurate and reliable performance assessments in predictive
 587 modeling.

588 5 Acknowledgement

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

591 References

- 592 [1] Hao Cheng, Dorian J. Garrick, and Rohan L. Fernando. Efficient strategies for leave-one-out cross validation for
593 genomic best linear unbiased prediction. *Journal of Animal Science and Biotechnology*, 8(1):38, May 2017.
- 594 [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
595 resilience during the transition period in dairy cows: A case study. *Journal of Dairy Science*, 101(11):10271–10282,
596 November 2018.
- 597 [3] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and
598 Prediction*. Springer series in statistics. Springer, 2009.
- 599 [4] Gavin C. Cawley and Nicola L.C. Talbot. On Over-fitting in Model Selection and Subsequent Selection Bias in
600 Performance Evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, August 2010.
- 601 [5] Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems.
602 *Technometrics*, 12(1):55–67, 1970. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American
603 Society for Quality].
- 604 [6] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society:
605 Series B (Methodological)*, 58(1):267–288, January 1996.
- 606 [7] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression
607 machines. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS’96,
608 pages 155–161, Cambridge, MA, USA, December 1996. MIT Press.
- 609 [8] Hervé Abdi. Partial Least Square Regression PLS-Regression. *Encyclopedia of social sciences research methods*,
610 pages 792–795, 2003.
- 611 [9] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.
- 612 [10] Yann LeCun. Generalization and Network Design strategies. 1989.
- 613 [11] Morteza H. Ghaffari, Amirhossein Jahanbekam, Hassan Sadri, Katharina Schuh, Georg Dusel, Cornelia Prehn,
614 Jerzy Adamski, Christian Koch, and Helga Sauerwein. Metabolomics meets machine learning: Longitudinal
615 metabolite profiling in serum of normal versus overconditioned cows and pathway analysis. *Journal of Dairy
616 Science*, 102(12):11561–11585, December 2019.
- 617 [12] G. Rovere, G. de los Campos, A. L. Lock, L. Worden, A. I. Vazquez, K. Lee, and R. J. Tempelman. Prediction of
618 fatty acid composition using milk spectral data and its associations with various mid-infrared spectral regions in
619 Michigan Holsteins. *Journal of Dairy Science*, 104(10):11242–11258, October 2021.
- 620 [13] C. A. Becker, A. Aghalari, M. Marufuzzaman, and A. E. Stone. Predicting dairy cattle heat stress using machine
621 learning techniques. *Journal of Dairy Science*, 104(1):501–524, January 2021.

- 622 [14] B. Lahart, S. McParland, E. Kennedy, T.M. Boland, T. Condon, M. Williams, N. Galvin, B. McCarthy, and
623 F. Buckley. Predicting the dry matter intake of grazing dairy cows using infrared reflectance spectroscopy analysis.
624 *Journal of Dairy Science*, 102(10):8907–8918, October 2019.
- 625 [15] Tiago Bresolin and João R. R. Dórea. Infrared Spectrometry as a High-Throughput Phenotyping Technology to
626 Predict Complex Traits in Livestock Systems. *Frontiers in Genetics*, 11, 2020.
- 627 [16] C. Grelet, E. Froidmont, L. Foldager, M. Salavati, M. Hostens, C. P. Ferris, K. L. Ingvartsen, M. A. Crowe, M. T.
628 Sorensen, J. A. Fernandez Pierna, A. Vanlierde, N. Gengler, and F. Dehareng. Potential of milk mid-infrared
629 spectra to predict nitrogen use efficiency of individual dairy cows in early lactation. *Journal of Dairy Science*,
630 103(5):4435–4445, May 2020.
- 631 [17] I. Adriaens, N. C. Friggins, W. Ouweltjes, H. Scott, B. Aernouts, and J. Statham. Productive life span and
632 resilience rank can be predicted from on-farm first-parity sensor time series but not using a common equation
633 across farms. *Journal of Dairy Science*, 103(8):7155–7171, August 2020.
- 634 [18] Lucio F. M. Mota, Diana Giannuzzi, Vittoria Bisutti, Sara Pegolo, Erminio Trevisi, Stefano Schiavon, Luigi Gallo,
635 David Fineboym, Gil Katz, and Alessio Cecchinato. Real-time milk analysis integrated with stacking ensemble
636 learning as a tool for the daily prediction of cheese-making traits in Holstein cattle. *Journal of Dairy Science*,
637 105(5):4237–4255, May 2022.
- 638 [19] Roii Spoliansky, Yael Edan, Yisrael Parmet, and Ilan Halachmi. Development of automatic body condition scoring
639 using a low-cost 3-dimensional Kinect camera. *Journal of Dairy Science*, 99(9):7714–7725, September 2016.
- 640 [20] Sun Yukun, Huo Pengju, Wang Yujie, Cui Ziqi, Li Yang, Dai Baisheng, Li Runze, and Zhang Yonggen. Automatic
641 monitoring system for individual dairy cows based on a deep learning framework that provides identification via
642 body parts and estimation of body condition score. *Journal of Dairy Science*, 102(11):10140–10151, November
643 2019.
- 644 [21] X. Song, E.A.M. Bokkers, P.P.J. Van Der Tol, P.W.G. Groot Koerkamp, and S. Van Mourik. Automated body
645 weight prediction of dairy cows using 3-dimensional vision. *Journal of Dairy Science*, 101(5):4448–4459, May
646 2018.
- 647 [22] C. Xavier, Y. Le Cozler, L. Depuille, A. Caillot, A. Lebreton, C. Allain, J. M. Delouard, L. Delattre, T. Luginbuhl,
648 P. Faverdin, and A. Fischer. The use of 3-dimensional imaging of Holstein cows to estimate body weight and
649 monitor the composition of body weight change throughout lactation. *Journal of Dairy Science*, 105(5):4508–4519,
650 May 2022.
- 651 [23] P. Mäntysaari, E.A. Mäntysaari, T. Kokkonen, T. Mehtio, S. Kajava, C. Grelet, P. Lidauer, and M.H. Lidauer.
652 Body and milk traits as indicators of dairy cow energy status in early lactation. *Journal of Dairy Science*,
653 102(9):7904–7916, September 2019.

- 654 [24] M. Frizzarin, I. C. Gormley, D. P. Berry, T. B. Murphy, A. Casa, A. Lynch, and S. McParland. Predicting cow
655 milk quality traits from routinely available milk spectra using statistical machine learning methods. *Journal of*
656 *Dairy Science*, 104(7):7438–7447, July 2021.
- 657 [25] J. A. D. R. N. Appuhamy, J. V. Judy, E. Kebreab, and P. J. Kononoff. Prediction of drinking water intake by dairy
658 cows. *Journal of Dairy Science*, 99(9):7191–7205, September 2016.
- 659 [26] R. A. de Souza, R. J. Tempelman, M. S. Allen, W. P. Weiss, J. K. Bernard, and M. J. VandeHaar. Predicting
660 nutrient digestibility in high-producing dairy cows. *Journal of Dairy Science*, 101(2):1123–1135, February 2018.
- 661 [27] J. R. R. Dórea, G. J. M. Rosa, K. A. Weld, and L. E. Armentano. Mining data from milk infrared spectroscopy to
662 improve feed intake predictions in lactating dairy cows. *Journal of Dairy Science*, 101(7):5878–5889, July 2018.
- 663 [28] L. I. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268, March
664 1989.
- 665 [29] Edward J. Jones, Thomas F. A. Bishop, Brendan P. Malone, Patrick J. Hulme, Brett M. Whelan, and Patrick
666 Filippi. Identifying causes of crop yield variability with interpretive machine learning. *Computers and Electronics*
667 in Agriculture, 192:106632, January 2022.
- 668 [30] S. J. Denholm, W. Brand, A. P. Mitchell, A. T. Wells, T. Krzyzelewski, S. L. Smith, E. Wall, and M. P. Coffey.
669 Predicting bovine tuberculosis status of dairy cows from mid-infrared spectral data of milk using deep learning.
670 *Journal of Dairy Science*, 103(10):9355–9367, October 2020.
- 671 [31] S.A. Kandeel, A.A. Megahed, M.H. Ebeid, and P.D. Constable. Ability of milk pH to predict subclinical mastitis
672 and intramammary infection in quarters from lactating dairy cattle. *Journal of Dairy Science*, 102(2):1417–1427,
673 February 2019.
- 674 [32] N. W. O’Leary, D. T. Byrne, A. H. O’Connor, and L. Shalloo. Invited review: Cattle lameness detection with
675 accelerometers. *Journal of Dairy Science*, 103(5):3895–3911, May 2020.
- 676 [33] J. Stojkov, G. Bowers, M. Draper, T. Duffield, P. Duivenvoorden, M. Groleau, D. Haupstein, R. Peters,
677 J. Pritchard, C. Radom, N. Sillett, W. Skippon, H. Trépanier, and D. Fraser. Hot topic: Management of cull
678 dairy cows—Consensus of an expert consultation in Canada. *Journal of Dairy Science*, 101(12):11170–11174,
679 December 2018.
- 680 [34] Maher Alsaad, Mahmoud Fadul, and Adrian Steiner. Automatic lameness detection in cattle. *The Veterinary*
681 *Journal*, 246:35–44, April 2019.
- 682 [35] X. Kang, X. D. Zhang, and G. Liu. Accurate detection of lameness in dairy cattle with computer vision: A new
683 and individualized detection strategy based on the analysis of the supporting phase. *Journal of Dairy Science*,
684 103(11):10628–10638, November 2020.
- 685 [36] Dan B. Jensen, Henk Hogeweegen, and Albert De Vries. Bayesian integration of sensor information and a multivariate
686 dynamic linear model for prediction of dairy cow mastitis. *Journal of Dairy Science*, 99(9):7344–7361, September
687 2016.

- 688 [37] P. Delhez, P.N. Ho, N. Gengler, H. Soyeurt, and J.E. Pryce. Diagnosing the pregnancy status of dairy cows: How
689 useful is milk mid-infrared spectroscopy? *Journal of Dairy Science*, 103(4):3264–3274, April 2020.
- 690 [38] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1
691 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, January 2020.
- 692 [39] J. M. Bowen, M. J. Haskell, G. A. Miller, C. S. Mason, D. J. Bell, and C-A. Duthie. Early prediction of
693 respiratory disease in preweaning dairy calves using feeding and activity behaviors. *Journal of Dairy Science*,
694 104(11):12009–12018, November 2021.
- 695 [40] V. Ouellet, E. Vasseur, W. Heuwieser, O. Burfeind, X. Maldague, and É. Charbonneau. Evaluation of calving
696 indicators measured by automated monitoring devices to predict the onset of calving in Holstein dairy cows.
697 *Journal of Dairy Science*, 99(2):1539–1548, February 2016.
- 698 [41] M.R. Borchers, Y.M. Chang, K.L. Proudfoot, B.A. Wadsworth, A.E. Stone, and J.M. Bewley. Machine-learning-
699 based calving prediction from activity, lying, and ruminating behaviors in dairy cattle. *Journal of Dairy Science*,
700 100(7):5664–5674, July 2017.

701 **Appendix**

702 **Cross Validation**

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

$$\begin{aligned}\mathcal{D} &= \{(X, Y)\} \\ &= \{(X_1, Y_1), (X_2, Y_2), \dots, (X_K, Y_K)\}\end{aligned}\tag{17}$$

707 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
 708 variable. The value of n corresponds to the total number of samples, while p represents the number of features. In
 709 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
 710 remaining folds make up the training set $\mathcal{D}_{\text{train}}$ (or \mathcal{D}_{-k}):

$$\begin{aligned}\mathcal{D}_{\text{train}} &= \mathcal{D}_{-k} \\ &= \{(X_1, Y_1), (X_2, Y_2), \dots, (X_{k-1}, Y_{k-1}), (X_{k+1}, Y_{k+1}), \dots, (X_K, Y_K)\} \\ \mathcal{D}_{\text{test}} &= \mathcal{D}_k \\ &= \{(X_k, Y_k)\}\end{aligned}\tag{18}$$

711 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 $f_{\mathcal{D}_{-k}}$.
 712 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
 713 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):

$$\begin{aligned}\hat{g}(f_{\mathcal{D}_{-k}}) &= \mathcal{L}(Y_k, \hat{Y}_k) \\ &= \mathcal{L}(Y_k, f_{\mathcal{D}_{-k}}(X_k))\end{aligned}\tag{19}$$

714 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
 715 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
 716 performance over all K folds. The model's generalization performance can be expressed as:

$$\begin{aligned}\mathbb{E}[\hat{g}(f_{\mathcal{D}})] &= \mathbb{E}[\hat{g}(f_{\mathcal{D}_k})] \\ &= \frac{1}{K} \sum_{k=1}^K \hat{g}(f_{\mathcal{D}_k})\end{aligned}\tag{20}$$

717 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
 718 all $\hat{g}(f_{\mathcal{D}_k})$ over K folds, which is also the definition of $\mathbb{E}[\hat{g}(f_{\mathcal{D}_k})]$.

719 Cross Validation Bias and Variance

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

$$\text{Bias} = \mathbb{E}[\hat{g}(f_{\mathcal{D}})] - G(f_{\mathcal{D}})\tag{21}$$

723 For example, if RMSE is used as the performance metric, a positive validation bias suggests that the model validation
 724 procedure concludes a pessimistic estimation of the model performance, since the true performance is expected to be
 725 lower than the estimated performance. Another aspect of model validation is the variance of the estimated performance.
 726 For example, in a 5-fold cross-validation, there are five estimates of the model performance. The variance among these
 727 five estimates is known as validation variance. A high validation variance suggests that the performance is sensitive to
 728 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
 729 variance can be defined as:

$$\begin{aligned}\text{Variance} &= \mathbb{E}[(\hat{g}(f_{\mathcal{D}_k}) - \mathbb{E}[\hat{g}(f_{\mathcal{D}})])^2] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k}) - 2\hat{g}(f_{\mathcal{D}_k})\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - 2\mathbb{E}[\hat{g}(f_{\mathcal{D}_k})]\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_k})] - \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]\end{aligned}\tag{22}$$

730 Combining the Equations 21 and 22, the mean squared error (MSE) of the model validation can be decomposed as:

$$\begin{aligned}
\text{MSE} &= \mathbb{E}[(\hat{g}(f_{D_k}) - G(f_D))^2] \\
&= \mathbb{E}[\hat{g}^2(f_{D_k})] - 2\mathbb{E}[\hat{g}(f_{D_k})]G(f_D) + G^2(f_D) + \\
&\quad \mathbb{E}^2[\hat{g}(f_{D_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})] \\
&= (\mathbb{E}^2[\hat{g}(f_{D_k})] - 2\mathbb{E}[\hat{g}(f_{D_k})]G(f_D) + G^2(f_D)) + \\
&\quad (\mathbb{E}[\hat{g}^2(f_{D_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})]) \\
&= (\mathbb{E}[\hat{g}(f_{D_k})] - G(f_D))^2 + (\mathbb{E}[\hat{g}^2(f_{D_k})] - \mathbb{E}^2[\hat{g}(f_{D_k})]) \\
&= (\mathbb{E}[\hat{g}(f_D)] - G(f_D))^2 + (\mathbb{E}[\hat{g}^2(f_D)] - \mathbb{E}^2[\hat{g}(f_D)]) \\
&= \text{Bias}^2 + \text{Variance}
\end{aligned} \tag{23}$$

731 **Hyperparameter**

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

$$\mathcal{L}_{\text{OLS}}(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 \tag{24}$$

$$\mathcal{L}_{\text{ridge}}(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda \sum_{j=1}^p \beta_j^2 \tag{25}$$

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

733 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
 734 denotes the sample size, and β is the coefficient vector. All three models aim to find the optimal β that minimizes their
 735 respective loss function, \mathcal{L} . In the regularized models (i.e., ridge and LASSO regression), the vector length of β is
 736 penalized in the loss function.

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

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

$$\begin{aligned}
\text{cov}(Y, \hat{Y}) &= \text{cov}(\hat{Y} + \epsilon, \hat{Y}) \\
&= \text{cov}(\hat{Y}, \hat{Y}) + \text{cov}(\hat{Y}, \epsilon) \\
&= \text{var}(\hat{Y}) + \text{cov}(\hat{Y}, \epsilon) \\
&= \text{var}(\hat{Y})
\end{aligned} \tag{27}$$

743 With the assumption that $\bar{\hat{Y}} = \bar{Y}$, which typically holds when $\epsilon \sim N(0, \sigma^2)$, the squared correlation coefficient r^2 is
744 expressed as follows:

$$\begin{aligned}
r^2 &= \frac{\text{cov}^2(Y, \hat{Y})}{\text{var}(Y)\text{var}(\hat{Y})} \\
&= \frac{\text{var}(\hat{Y})^2}{\text{var}(Y)\text{var}(\hat{Y})} \\
&= \frac{\text{var}(\hat{Y})}{\text{var}(Y)} \\
&= \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{\hat{Y}})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \\
&= \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{\hat{Y}})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \\
&= \frac{SS_{\text{residual}}}{SS_{\text{total}}} \\
&= R^2
\end{aligned} \tag{28}$$

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