Invited Review: A Guide on How to Avoid Common Pitfalls in Model Performance Metrics and Validation

Chun-Peng James Chen

Abstract

This review critically examines the metrics and methodologies used for evaluating prediction models in regression and classification tasks, underscoring the importance of rigorous and standardized approaches in model performance assessment. Modeling, a structured framework for hypothesis formulation and decision-making, relies on the analysis and extrapolation of empirical data. Its advancement is contingent on the accumulation and integrity of prior knowledge within the scientific community.

The review highlights that no single metric suffices to fully grasp model performance, emphasizing the need for understanding the underlying theory of each metric to avoid misleading conclusions. In regression tasks, metrics such as the Pearson Correlation Coefficient, Root Mean Squared Error, and Coefficient of Determination R^2 are discussed, considering their specific applications and limitations. For classification tasks, the focus shifts to metrics including precision, recall, ROC curve, and MCC curve, emphasizing correctly designating the positive class to prevent bias and ensuring label-invariant assessment for a balanced evaluation. Moreover, the paper delves into common pitfalls in cross-validation (CV), a technique crucial for simulating unseen data for model evaluation. Issues such as using the same data for both training and assessment, excluding model selection from CV, and overlooking experimental block effects are explored.

The review is structured into sections covering model performance metrics and validation techniques. Each section provides computational simulations and literature examples to illustrate the concepts from both theoretical and practical perspectives, culminating in a summary of key points and recommendations for future research. This comprehensive approach aims to guide researchers in accurately reporting model performance, especially in scenarios of imbalanced datasets, feature selection, hyperparameter tuning, and limited sample sizes, ultimately contributing to the integrity and progress of scientific research in modeling.

Keywords: cross-validation; performance metrics; model evaluation

1. Overview

Modeling is an essential tool for hypothesis formulation and decision-making. It functions as a structured framework that validates system understanding through the analysis of empirical data. Further, it extends this understanding by enabling the extrapolation of results to novel trials and conditions. The advancement of research is fundamentally built upon the accumulation of prior knowledge within the scientific community. Therefore, evaluating model performance becomes particularly critical, necessitating a rigorous and standardized approach that allows for both reproducibility and comparability. The failure to adhere to these standards, by reporting model performance through ill-defined metrics or non-rigorous procedures, can introduce misinterpretations and miscommunications. Such lapses impede scientific progress and compromise the integrity of the collective body of research in the field.

This review aims to scrutinize commonly used metrics in evaluating a prediction model, covering regression and classification tasks. Since no single metric provides a comprehensive perspective of model performance, understanding the underlying theory of each metric helps avoid misleading conclusions. Additionally, biased or over-optimistic estimations of model performance usually come from inappropriately conducting cross-validation (**CV**), a technique to simulate unseen data for model evaluation. Common pitfalls include using the exact data for both training and model assessment, excluding the model selection process from CV, and neglecting experimental block effects. This review uses a series of hypothetical examples to demonstrate these pitfalls. With the examples, it is anticipated to address further concerns in model evaluation. For instance, how to appropriately report model performance when the dataset is imbalanced in binary classification? How to include feature selection and hyperparameter tuning in a CV? How to unbiasedly estimate a model performance with limited samples?

This review is organized into two key sections that explore the various facets of model performance metrics and validation in depth. In addition to the brief overview in this first section, **section 2** delves into the metrics used for assessing model performance, with **subsection 2. a** focusing on regression metrics and **subsection 2. b** examining classification metrics. **Section 3** is dedicated to the validation techniques essential for ensuring model reliability and includes discussions on bias and variance (**subsection 3. a**), model selection methodologies (**subsection 3. b**), and the application of block cross-validation strategies (**subsection 3. c**). A computational simulation was conducted to provide a hypothetical example in each subsection, accompanied by several literature as real-world examples to illustrate concepts from both theoretical and practical perspectives. Lastly, **Section 4** summarizes key points and recommendations for future research.

2. Performance Metric

Performance metrics serve as quantitative indicators for evaluating model performance. They are critical for benchmarking various modeling approaches and validating hypotheses. Choosing appropriate metrics while testing a hypothesis is crucial, as a wrong selection may lead to an overly optimistic conclusion. This section introduces commonly used performance metrics and highlights potential pitfalls that researchers should be cautious of.

2. a Regression

A regression model aims to predict a continuous variable and is commonly employed in diverse applications, such as estimating body condition scores (Spoliansky et al., 2016; Yukun et al., 2019), body weight (Song et al., 2018; Xavier et al., 2022), milk composition (Mäntysaari et al., 2019; Frizzarin et al., 2021; Rovere et al., 2021; Mota et al., 2022), efficiency in resource usage (Appuhamy et al., 2016; de Souza et al., 2018; Grelet et al., 2020), and early-lactation behavior (van Dixhoorn et al., 2018). This section explores three common metrics for evaluating regression models: Pearson Correlation Coefficient ($\mathbf{r}$), Root Mean Squared Error (**RMSE**), and the Coefficient of Determination ($\boldsymbol{R^2}$).

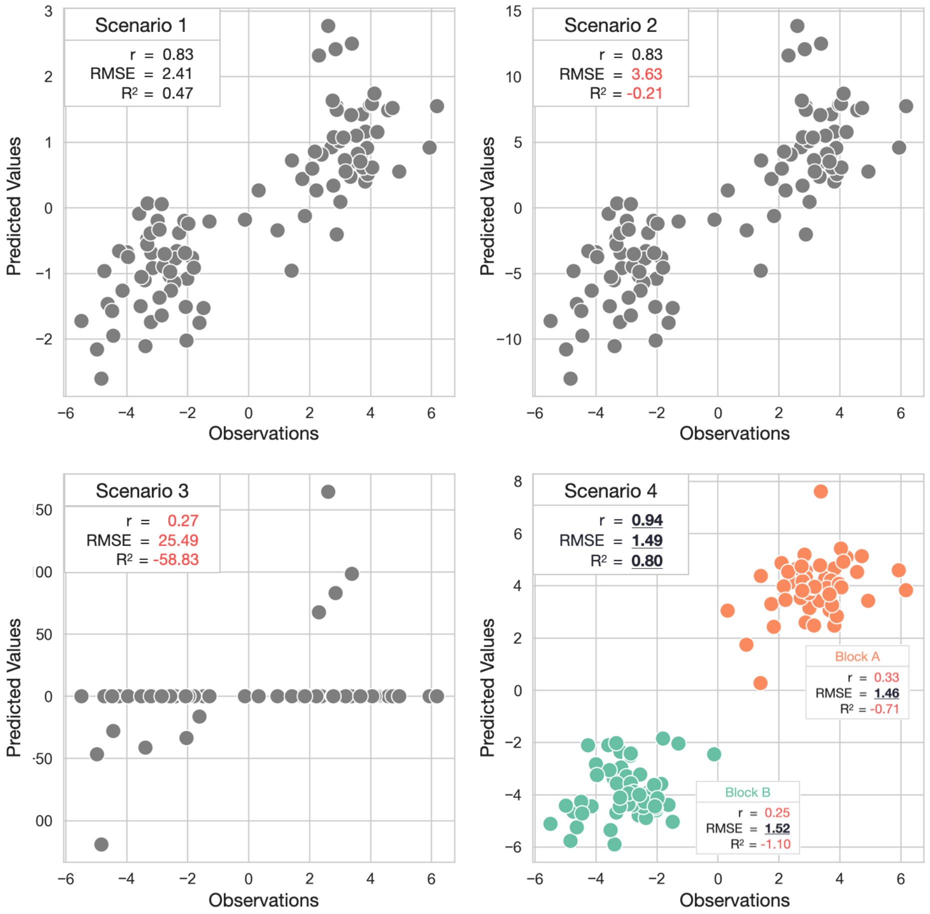


__Figure 1.__ Scatter plots display the same observations against four different prediction scenarios in the given hypothetical example. Scenario 1 serves as a baseline for the metrics, with any metric better than the baseline highlighted in bold and underscored, and any worse metric colored in red.

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

- *Scenario 1*: To establish a correlation relationship, the observations were multiplied by 0.3, followed by the addition of random noise $\mathcal{N}(0, 0.7)$ to introduce prediction errors.
- *Scenario 2:* The prediction outcome from Scenario 1 was multiplied by 5, simulating predictions with a larger variance while maintaining the same relative order as the original predictions.
- *Scenario 3:* only the top 10% of predictions that deviate the most from zero in Scenario 1 were raised to the power of 5. The rest of the predictions were set to zero. This scenario simulates a prediction that focuses solely on the extreme samples.
- *Scenario 4*: Values sampled from two normal distributions, $\mathcal{N}(-3, 1)$ and $\mathcal{N}(3, 1)$, were added respectively to the predictions made in Scenario 1 of Block A (colored orange in **Figure 1**) and Block B (colored green in **Figure 1**). This scenario amplified the original block effects, simulating a model that effectively distinguished between different blocks (e.g., herd or breed) but was less capable of predicting individual variations within each block.

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

<u>*Pearson Correlation Coefficient (r)*</u>

$$
\begin{aligned}
r &= \frac{cov(x, y)}{\sigma_x \sigma_y} \\
&= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}
\end{aligned}
\tag{1}
$$

The Pearson Correlation Coefficient r measures the strength of the linear relationship between two continuous variables, as defined by **Equation (1)** where the observations are denoted by x and the predicted values are represented by y. The numerator of the equation computes the covariance between x and y, which captures how the two variables coordinately deviate from their means. The covariance is then standardized by the product of the standard deviations of x and y, hence the coefficient r is scale-invariant and will always fall within the range of-1 to 1.

When the goal is to rank observations of interest rather than predict the absolute magnitude of the error, this metric is appropriate. The property of this metric was demonstrated in both Scenario 1 and 2, where the coefficient remained consistent despite Scenario 2 having errors five times greater than in Scenario 1. If the absolute error is of interest, this metric should be used in conjunction with other metrics, such as RMSE or R^2. It is also worth noting that this metric can provide a value of 0.27 in Scenario 3, where 90% of the predictions failed to capture the trend and resulted in zero-value predictions. The positive performance of the metric came from the predictions ranking the remaining 10% of the observations in a fairly accurate order, regardless of the large error magnitude. Moreover, one common pitfall of this metric is that block effects can influence it, leading to an inflated performance estimate if individual variation is of greater interest than inter-block variation. This was demonstrated in Scenario 4, where the overall coefficient r was 0.94, but the metric within each block was only 0.33 and 0.25, respectively. Therefore, it is essential to visually inspect regression results through scatter plots or examine them within individual blocks.

This metric is often used to evaluate models that can identify high-performing individuals. De Souza et al. used this metric to demonstrate the ability to identify high-producing dairy cows based on predicted nutrient digestibility (de Souza et al., 2018). This metric was also used in a model selection scenario where multiple models were evaluated based on their ability to rank traits of interest, such as feed intake (Dórea et al., 2018) and milk composition (Rovere et al., 2021) in dairy cows.

Root Mean Squared Error (RMSE)

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

The RMSE serves as a quantitative measure to gauge the average magnitude of prediction errors between observed values y and their predicted values $\hat{y}$. It gives the error in the same units as the observation y, which is particularly useful when the absolute error is of interest. As shown in **Equation (2)**, the RMSE is calculated by first computing the squared error, then summating all squared errors, and finally averaging the sum by the number of observations n.

Distinct from the correlation coefficient, RMSE is sensitive to scale, implying that achieving predictions with a variance akin to the observed values takes precedence over maintaining their order or trend. This property is evident in Scenario 2, where the RMSE inflates from 2.41 to 3.63, despite the fact that the predictions in both scenarios rank the observations identically. Another notable characteristic of RMSE is it weighs more on large errors, which is essential when making a large error is costly and should be prioritized for avoidance. In Scenario 3, where certain predictions deviate substantially from the majority, the squaring operation in **Equation (2)** accentuates these outliers, culminating in an RMSE value of 25.49. It is also worth mentioning that RMSE is impervious to block effects, which was illustrated in Scenario 4. In this scenario, both the complete set of predictions and the intra-block predictions yield similar RMSE values—1.49, 1.46, and 1.52, respectively. This phenomenon emphasizes again that RMSE is affected solely by the magnitude of the error, which neglects the ability of the model to capture relative trends in intra-block or inter-block predictions.

In practice, RMSE is easy to interpret in real-world units of livestock production. For example, monitoring cow body weight is a common practice to aid in the management of dairy cows. Studies by Song et al. and Xavier et al. have utilized RMSE to assess the effectiveness of three-dimensional cameras in estimating dairy cow body weight, yielding RMSE values of 41.2 kg and 12.1 kg, respectively (Song et al., 2018; Xavier et al., 2022). These figures provide a straightforward value for farmers to gauge whether the prediction error is tolerable, considering their specific operational costs and management thresholds. In essence, RMSE translates complex model accuracy into practical insights for productive agricultural units.

Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{SSE}{SST}$$
$$= 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_\imath})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3)$$

The Coefficient of Determination, denoted as R^2, assesses prediction accuracy by comparing the variance explained by the model against the total observed variance from y. It is defined in **Equation (3)**, where **SSE** represents the Sum of Squared Errors and **SST** is the Total Sum of Squares. The R^2 is obtained by subtracting the ratio of SSE to SST from 1. Given that the ratio is non-negative, the maximum R^2 value is 1, which indicates the predictions are the same as the observed values. Besides, there is a special case where R^2 is 0, which indicates a model that performs no better than predicting all samples as the mean of the observed values. Negative R^2 occurs when a model performs worse than the mean-based predictions, suggesting a potentially overfit model with excessive variance.

In Scenario 1, a moderate R^2 of 0.47 indicates that the model explains nearly half of the observed variation. With a five times larger variance compared to Scenario 1, Scenario 2 yields a negative R^2 of -0.21 despite retaining the same prediction order. Similarly to RMSE, where errors are squared, the presence of outliers leads to a dramatic R^2 drop to -58.83, showcasing the sensitivity of R^2 to outlier-induced variance. Lastly, in Scenario 4, the value of R^2 indicated a strong performance by the model with a score of 0.80. This score is statistically reasonable, as the model explained 80% of the total variation, which was mainly contributed by block effects. However, when each block was analyzed individually, the R^2 values decreased to -0.71 and -1.10, respectively, because the model failed to account for intra-block variation. In summary, while both RMSE and R^2 aim to measure prediction errors, R^2 offers additional statistical insights that facilitate a more nuanced evaluation of model performance.

The metric R^2 is useful in comparing multiple regression models. For instance, Xavier et al. used this metric to regress the body weight of dairy cows on a set of morphological traits. The study compared different linear combinations of these traits and used the R^2 values to select the best model for predicting body weight (Xavier et al., 2022) . Another example is the study by Grelet et al., which examined the relationship between the milk spectral profile collected from mid-infrared spectroscopy and nitrogen utilization efficiency in dairy cows. In this study, R^2 was used to determine the response variable that could be explained most by the same set of regressors (Grelet et al., 2020). In both cases, R^2 was used to evaluate the performance of the model and select the best one.

Section Conclusion

All three metrics discussed in this section are effective in examining the performance of regression models. Correlation Coefficient r evaluates the ability of the model to rank the observations. RMSE is a metric that is easy to interpret and is suitable to evaluate the absolute error. The coefficient of determination R^2 measures the proportion of variation that the model can explain. Choosing the appropriate metric hinges on the specific goals of model evaluation. For a thorough analysis, employing multiple metrics is advisable to gain a multifaceted view of the model's performance.

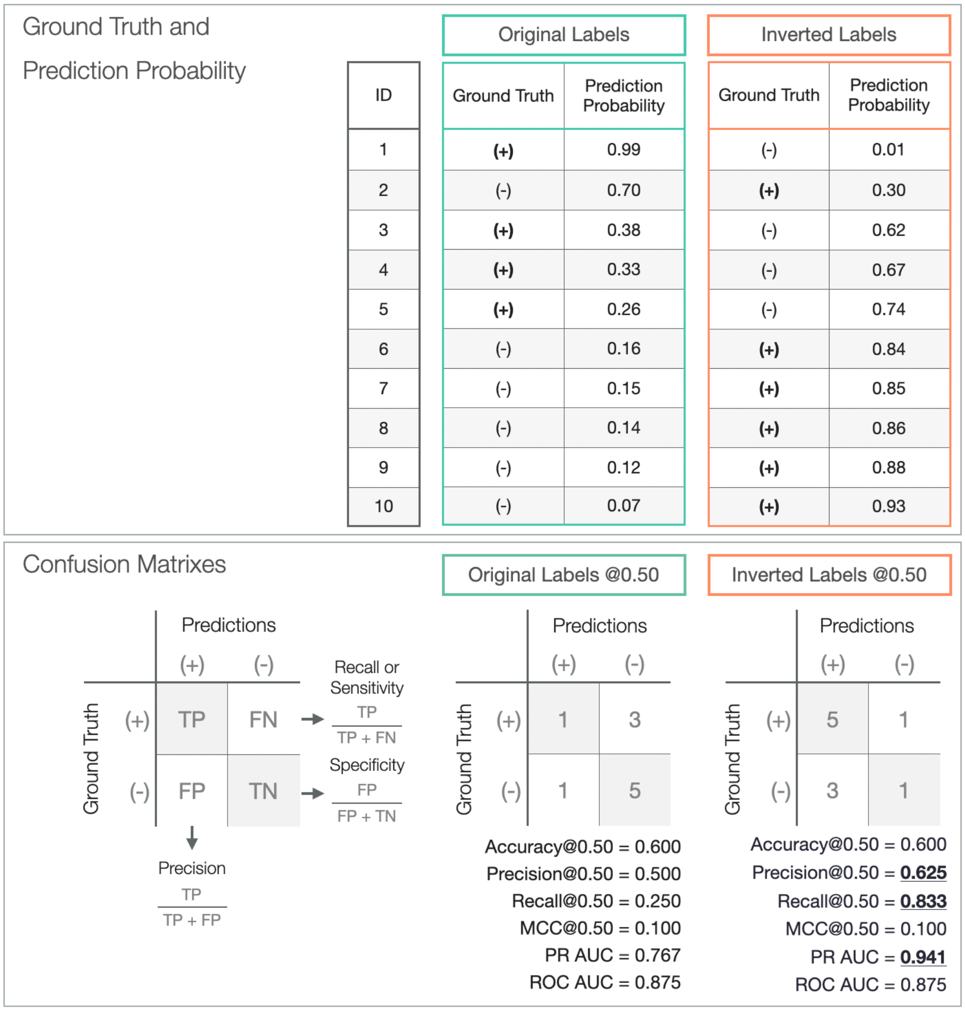
2. b Classification

Classification models aim to predict categorical outcomes such as 'healthy' or 'sick,' 'susceptible' or 'resistant,' and 'high yield' or 'low yield.' This section presents a hypothetical example to highlight how the choice of different performance metrics can lead to different interpretations of a model's effectiveness. The example focuses on binary classification, where the outcome is either positive (Y=1) or negative (Y=0). Suppose a binary classification model always outputs a probability between 0 and 1, indicating the likelihood that a sample belongs to the positive class. It assumes that the model has high confidence in correctly predicting 1 out of 4 positive and 5 out of 6 negative samples. This example intends to illustrate a scenario where the positive outcome is rare, such as predicting the onset of a calving event in dairy cows (Ouellet et al., 2016; Borchers et al., 2017). The simulated predictions can be summarized in the following probability distribution:

$$P(X,Y) = \{ \begin{array}{ll} \mathit{Uniform}(0.8, 1.0) \times \frac{1}{10} & \mathit{if}\ Y = 1 \\ \mathit{Uniform}(0.6, 0.8) \times \frac{1}{10} & \mathit{if}\ Y = 0 \\ \mathit{Uniform}(0.2, 0.4) \times \frac{3}{10} & \mathit{if}\ Y = 1 \\ \mathit{Uniform}(0.0, 0.2) \times \frac{5}{10} & \mathit{if}\ Y = 0 \end{array} \tag{4}$$

Where X is a random variable representing the predicted probabilities sampled from a uniform distribution $\mathit{Uniform(a,b)}$ between a and b, and Y represents the ground truth labels. The simulated result is shown in **Figure 2**. In addition to the original labels, this example also examines a scenario with inverted labels (**Figure 2. Upper**). Since most classification metrics prioritize positive samples, it is generally advisable to designate the event of interest as the positive class in binary classification problems. Inverting the labels illustrates the potential overestimation of model performance when the more common, but less significant, background event is mistakenly marked as the positive class. It is important to note that inverting the labels affects only the interpretation of model performance, not the model configuration or parameters.

To evaluate classification performance, one must first establish a **confidence threshold** to dichotomize the prediction probabilities. For instance, if a prediction probability exceeds the threshold, the sample is labeled positive. By default, the threshold is set at 0.5 for its simplicity. Consider the third data row of the simulated outcome, for example: with a prediction probability of 0.38 that falls below the threshold, the sample is deemed negative, resulting in a false negative classification since the ground truth is positive. It is worth mentioning that this threshold is adjustable to fine-tune model performance for particular uses. A **confusion matrix** (**Figure 2. Lower**), effectively encapsulates prediction outcomes. The rows in this 2x2 matrix correspond to ground truth, while its columns reflect predictions. Correct predictions populate the diagonal cells, and errors fill the off-diagonal ones. This matrix serves as the foundation for computing various metrics to assess model performance, which will be explored in the subsequent sections.



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

$\frac{TP}{TP + FN}$ $\frac{FP}{FP + TN}$ $\frac{TP}{TP + FP}$

Original Labels @0.50

Ground Truth \ Predictions	(+)	(-)
(+)	1	3
(-)	1	5

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

Inverted Labels @0.50

Ground Truth \ Predictions	(+)	(-)
(+)	5	1
(-)	3	1

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

Figure 2. *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.*

Accuracy

$$Accuracy = \frac{\text{Total Correct Predictions}}{\text{Total Predictions}} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5)$$

Accuracy quantifies overall model performance by measuring the rate of correct predictions, as defined in **Equation (5)**, with TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives) defining the respective outcomes. With a 0.5 threshold in this example, the accuracy stands at 0.60. This figure might suggest modest efficacy, marginally surpassing random chance, with an accuracy of 0.50. Nonetheless, the same accuracy level could be achieved by classifying every sample as negative in an imbalanced dataset where negatives are predominant. Hence, relying solely on accuracy to assess models can be misleading, indicating the need for additional metrics to ensure robust evaluation.

Precision, Recall, and Precision-Recall Curve

$$\begin{aligned} Precision &= \frac{TP}{\text{Total Predicted Positive}} \\ &= \frac{TP}{TP + FP} \end{aligned} \tag{6}$$

$$\begin{aligned} Recall &= \frac{TP}{\text{Total Actual Positive}} \\ &= \frac{TP}{TP + FN} \end{aligned} \tag{7}$$

Precision and **recall** refine the assessment of a classification model by offering insights that accuracy alone may overlook. Precision calculates the fraction of true positives among all positive predictions, essentially measuring the trustworthiness of positive predictions made by the model (**Equation (6)**). High precision is crucial in scenarios where false positives incur significant costs, and false negatives are more tolerable. For instance, in contexts where clinical treatments and culling are expensive, such as detecting bovine tuberculosis (Denholm et al., 2020) or mastitis (Kandeel et al., 2019) using non-invasive methods, a high-precision model is crucial to minimize unnecessary costs and interventions from false positives. On the other hand, recall, also known as sensitivity, quantifies the ratio of true positives to all actual positives, assessing the model's ability to identify positive cases (**Equation (7)**). High recall is essential where missing a positive case has serious consequences, or where false positives are easily rectifiable. For instance, detecting lameness or abnormal gait is crucial, as these can indicate underlying pathologies (O'Leary et al., 2020) and impact welfare-related transport decisions (Stojkov et al., 2018). An automated detection system (Alsaaod et al., 2019; Kang et al., 2020; O'Leary et al., 2020) 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.

In the hypothetical example, setting a threshold of 0.5 yields precision and recall values of 0.5 and 0.25, respectively. These metrics deliver more interpretable information that only half of the positive predictions are correct, and just a quarter of the actual positives are detected. This contrasts with an accuracy of 0.6, which may appear misleadingly high due to the abundance of negative samples. The chosen confidence threshold significantly impacts precision and recall. While the trade-off between these two metrics is not always linear, it is generally observed that a higher threshold increases precision but decreases recall, and vice versa. A high threshold indicates a conservative approach in predicting positives, reducing false positives, and thus enhancing precision. However, this often leads to missing actual positive cases, lowering recall. Hence, the Precision-Recall (**PR**) curve is an essential tool for evaluating model performance across various thresholds. Plotted with recall on the x-axis and precision on the y-axis, this curve is derived by computing these metrics at different thresholds (**Figure 3, Left**). The Area Under the Curve (**AUC**) provides a summary measure of the PR curve's overall performance. A model's effectiveness is generally indicated by how close a point on the PR curve is to the top-right corner. For example, at a threshold of 0.25, which is positioned near the top-right of the PR curve, the model demonstrates impressive performance with an accuracy of 0.90, precision of 0.80, and recall at 1.00.

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

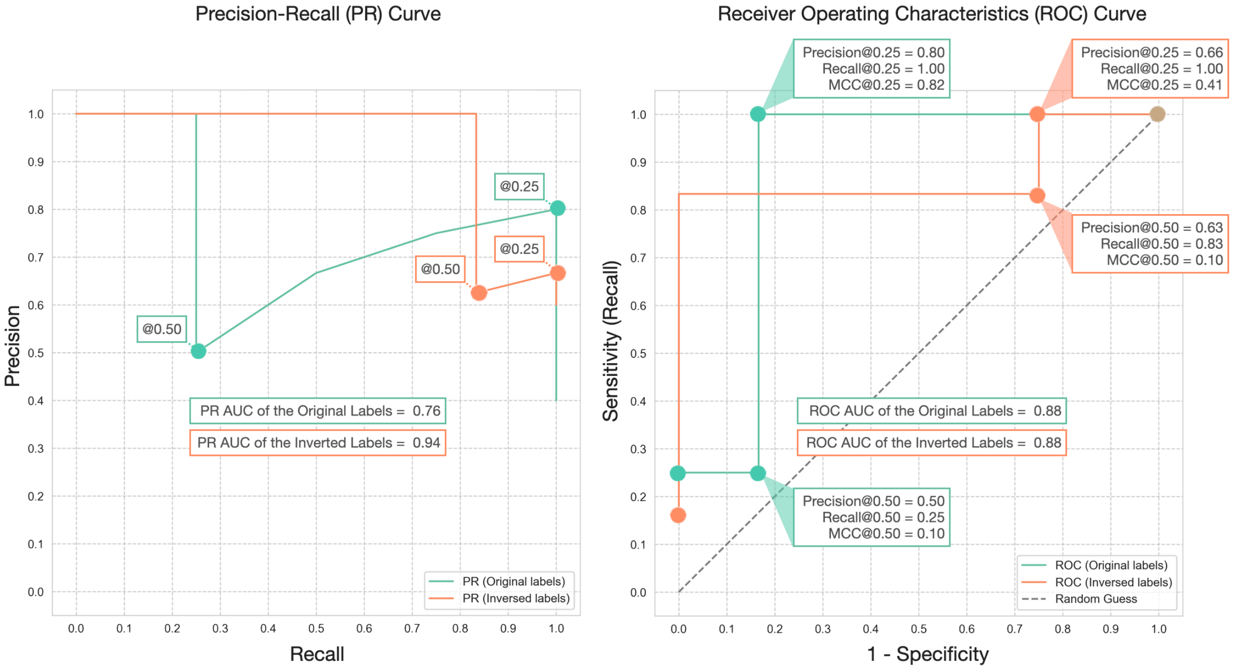


*Figure 3. (Left) Precision-recall (PR) curve and (**Right**) Receiver operating characteristic (ROC) curve for the hypothetical example are displayed. The performance at confidence thresholds of 0.25 and 0.50 is highlighted. Original labels are marked in green, while inverted labels appear in orange. The Area Under the Curve (AUC) is depicted at the center of each curve.*

Receiver Operating Characteristic (ROC) Curve

The Receiver Operating Characteristic (**ROC**) curve is another crucial tool for assessing a model's performance across various thresholds, plotting one minus **specificity** against **sensitivity**. The equations for specificity and sensitivity are as follows:

$$
\begin{aligned}
Specificity &= \frac{TF}{\text{Total Actual Negative}} \\
&= \frac{TF}{FP\ +\ TN} \\
Sensitivity &= Recall \\
&= \frac{TP}{\text{Total Actual Positive}}
\end{aligned}
\tag{8}
$$

$$= \frac{TP}{TP + FN} \tag{9}$$

Unlike metrics focusing solely on positive samples, the ROC curve accounts for both positive and negative samples, making it a label-invariant metric. Specificity is plotted on the x-axis and sensitivity on the y-axis, calculated at different thresholds (**Figure 3, Right**).

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 (Jensen et al., 2016) and diagnosing pregnancy using spectroscopy technology (Delhez et al., 2020). 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.

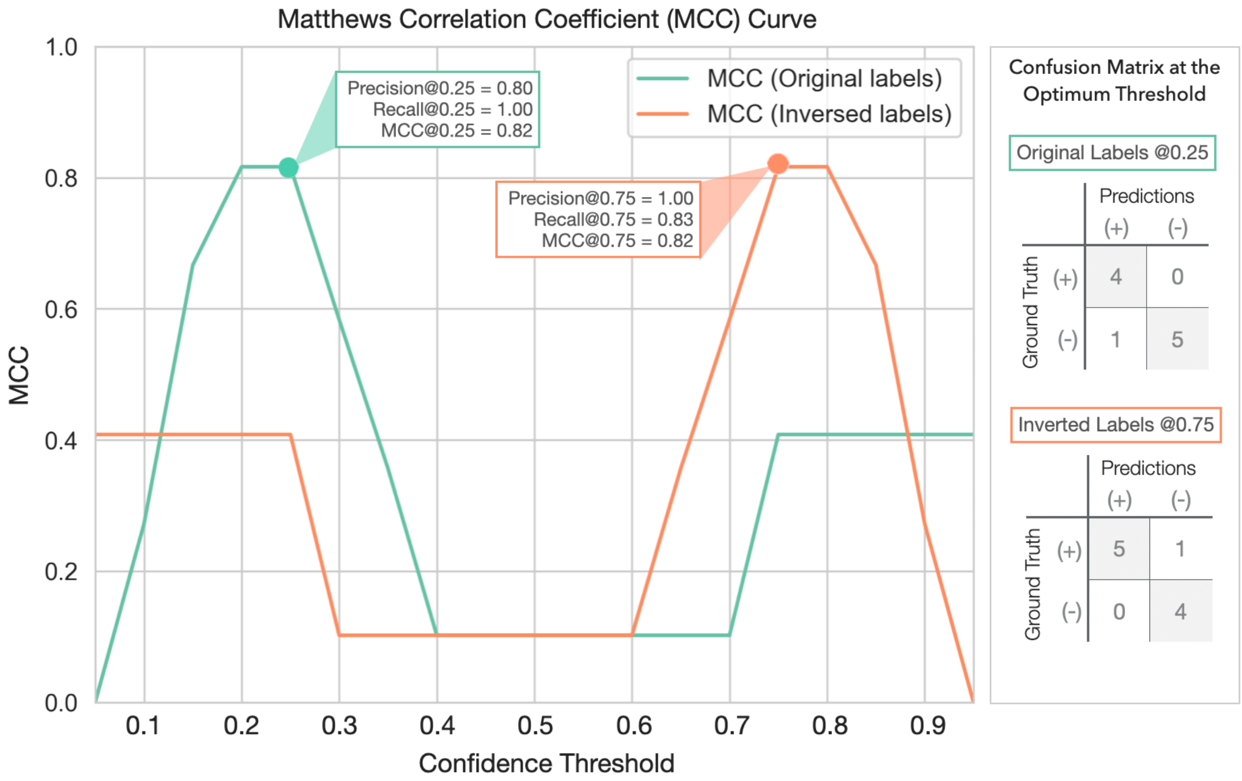
Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (**MCC**) is a robust metric for evaluating binary classification models. Unlike other metrics, the MCC considers both positive and negative samples in the dataset, providing a balanced measure of a model's performance (Chicco and Jurman, 2020). It is defined as:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{10}$$

Equation (10) symmetrically incorporates all four components (i.e., TP, TN, FP, and FN) of the confusion matrix. This symmetry makes MCC invariant to class distribution changes. The coefficient ranges from -1 to 1, where 1 indicates perfect classification, 0 indicates no better performance than random guessing, and -1 signifies total disagreement between prediction and observation. In a case study that used feeding and daily activity behaviors to diagnose Bovine Respiratory Disease in dairy calves, MCC proved particularly insightful (Bowen et al., 2021). The models in this study exhibited strong performance on negative samples (i.e., healthy calves), which were more prevalent, resulting in high specificity. However, sensitivity was relatively low at 0.54. In this context, MCC, with a value of 0.36, provided a more nuanced and representative measure of model performance, especially given the skew towards negative samples.

Considering MCC's balanced approach to evaluating model performance, this review introduces the concept of an MCC curve. This curve, which plots the MCC value against various threshold levels (**Figure 4**), serves as a powerful tool for identifying the optimal confidence thresholds for model predictions. By examining this curve, one can determine the specific threshold at which the MCC value peaks, thereby optimizing the model's performance. For example, when applied to the hypothetical example, the optimum MCC value of 0.82 was attained at a threshold of 0.25. This particular threshold corresponded to accuracy, precision, and recall values of 0.90, 0.75, and 1.00, respectively. Notably, the MCC curve retains its symmetry even when labels are reversed, affirming its status as a label-invariant measure. In scenarios with inverted labels, the maximum MCC value observed was 0.83, achieved at a threshold of 0.75, leading to accuracy, precision, and recall values of 0.90, 1.00, and 0.83, respectively. Such findings underscore the MCC's ability to provide a balanced and comprehensive assessment of both positive and negative samples, thereby reinforcing its utility as a versatile and effective metric for thorough model evaluation.



Original Labels @0.25

Ground Truth \ Predictions	(+)	(-)
(+)	4	0
(-)	1	5

Inverted Labels @0.75

Ground Truth \ Predictions	(+)	(-)
(+)	5	1
(-)	0	4

Figure 4. *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.*

Section Conclusion

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

3. Model Validation

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

$$\begin{aligned}\mathcal{D} &= \{(X, Y)\} \\ &= \{(X_1, Y_1), (X_2, Y_2), \dots, (X_K, Y_K)\}\end{aligned} \tag{11}$$

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

$$\begin{aligned}\mathcal{D}_{train} &= \mathcal{D}_{\text{-}k} \\ &= \{(X_1, Y_1), (X_2, Y_2), \dots, (X_{k\text{-}1}, Y_{k\text{-}1}), (X_{k+1}, Y_{k+1}), \dots, (X_K, Y_K)\}\end{aligned} \tag{12}$$

$$\begin{aligned}\mathcal{D}_{test} &= \mathcal{D}_k \\ &= \{(X_k, Y_k)\}\end{aligned} \tag{13}$$

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

$$\begin{aligned}\hat{g}\left(f_{\mathcal{D}_{\text{-}k}}\right) &= \mathcal{L}(Y_k, \hat{Y}_k) \\ &= \mathcal{L}(Y_k,\ f_{\mathcal{D}_{\text{-}k}}(X_k))\end{aligned} \tag{14}$$

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

$$\mathbb{E}[\hat{g}(f_{\mathcal{D}})] = \mathbb{E}[\hat{g}(f_{\mathcal{D}_{\text{-}k}})] = \frac{1}{K}\sum_{k=1}^{K}\hat{g}\left(f_{\mathcal{D}_{\text{-}k}}\right) \tag{15}$$

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

3. a Validation Bias and Variance

Definition

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

$$Bias = \mathbb{E}[\hat{g}\,(f_{\mathcal{D}})] - G(f_{\mathcal{D}}) \tag{16}$$

For example, if RMSE is used as the performance metric, a positive validation bias suggests that the model validation procedure concludes a pessimistic estimation of the model performance, since the true performance is expected to be lower than the estimated performance.

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

$$\begin{aligned} Variance &= \mathbb{E}[(\hat{g}(f_{\mathcal{D}_{\text{-}k}}) - \mathbb{E}[\hat{g}(f_{\mathcal{D}})])^2] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_{\text{-}k}}) - 2\hat{g}(f_{\mathcal{D}_{\text{-}k}})\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})]] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_{\text{-}k}})] - 2\mathbb{E}[\hat{g}(f_{\mathcal{D}_{\text{-}k}})]\mathbb{E}[\hat{g}(f_{\mathcal{D}})] + \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})] \\ &= \mathbb{E}[\hat{g}^2(f_{\mathcal{D}_{\text{-}k}})] - \mathbb{E}^2[\hat{g}(f_{\mathcal{D}})] \end{aligned} \tag{17}$$

Combining the **Equation (16)** and **(17)**, the mean squared error (**MSE**) of the model validation can be decomposed as:

$$\begin{aligned} MSE &= \mathbb{E}[(\widehat{g}(f_{\mathcal{D}_{\text{-}k}}) - G(f_{\mathcal{D}}))^2] \\ &= \mathbb{E}[\widehat{g}^2(f_{\mathcal{D}_{\text{-}k}})] - 2\mathbb{E}[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})]G(f_{\mathcal{D}}) + G^2(f_{\mathcal{D}}) + \\ &\quad \mathbb{E}^2[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})] - \mathbb{E}^2[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})] \\ &= (\mathbb{E}^2[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})] - 2\mathbb{E}[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})]G(f_{\mathcal{D}}) + G^2(f_{\mathcal{D}})) + \\ &\quad (\mathbb{E}[\widehat{g}^2(f_{\mathcal{D}_{\text{-}k}})] - \mathbb{E}^2[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})]) \\ &= (\mathbb{E}[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})] - G(f_{\mathcal{D}}))^2 + (\mathbb{E}[\widehat{g}^2(f_{\mathcal{D}_{\text{-}k}})] - \mathbb{E}^2[\widehat{g}(f_{\mathcal{D}_{\text{-}k}})]) \\ &= (\mathbb{E}[\widehat{g}(f_{\mathcal{D}})] - G(f_{\mathcal{D}}))^2 \;\; + (\mathbb{E}[\widehat{g}^2(f_{\mathcal{D}_{\text{-}k}})] - \mathbb{E}^2[\widehat{g}(f_{\mathcal{D}})]) \\ &= Bias^2 + Variance \end{aligned} \tag{18}$$

Bias-Variance Trade-off

Equation (18) unveils a trade-off relationship between the bias and variance given a constant validation MSE. When performing K-fold CV with a fixed sample size and model complexity, the choice of K is the pivotal element shaping the model validation. When the K is set to a larger value; each training set $\mathcal{D}_{\text{-}k}$ is larger in size, resulting in a model trained on a more representative subset of the population of interest, leading to lower bias. However, a large K comes with a trade-off: the corresponding test subset $\mathcal{D}_k$ is compressed in size, making the tested model more sensitive to the specific data points, and thus inflating the validation variance. Conversely, a smaller K, along with a minor training set $\mathcal{D}_{\text{-}k}$, reduces their representativeness and increases bias. Nevertheless, a larger size of the test set $\mathcal{D}_k$ leads to more consistent estimations across the folds and, consequently, reduces the validation variance.

Leave-one-out cross-validation (**LOOCV**) is a variant of K-fold CV where K equals the sample size $N$of the complete dataset $\mathcal{D}$. It provides an unbiased estimation of model performance because the training set $\mathcal{D}_{\text{-}k}$ closely resembles the unseen population of interest, given its size of $N - 1$. However, as the trade-off discussion suggested, this method can lead to high validation variance due to the model is evaluated on one sample at a time. The true unbiased nature of LOOCV is fully realized only when all K folds are utilized. Performing an incomplete LOOCV can introduce significant bias because of the inherent high validation variance, which often occurs when training each model iteration is prohibitively time-consuming or computationally demanding. In specific contexts, such as genomic prediction, strategies like the one described by Cheng et al. leverage the matrix inverse lemma, which allows for computational savings by avoiding the inversion of large matrices in each fold. This technique significantly reduces the dependency of computational resources on the sample size (Cheng et al., 2017). Van Dixhoorn et al. exemplify the use of LOOCV with a small dataset, aiming to predict cow resilience with limited data resources (van Dixhoorn et al., 2018). Nevertheless, for large datasets, LOOCV is generally not recommended due to computational inefficiency. Further details of bias-variance trade-off have been extensively explored in the statistical literature (Hastie et al., 2009; Cawley and Talbot, 2010).

Simulation Objectives and Hypothesis

This simulation study investigated the interplay between sample size and various performance estimators and their collective impact on bias and variance during model validation. It is hypothesized that increasing the sample size will reduce both bias and variance. Additionally, it is expected that the validation variance will increase with the number of folds in the CV, while simultaneously reducing bias. Since K-fold CV employs a fraction (i.e., $K - 1$ folds) of the data for training, it may provide a pessimistic estimate of model performance. This study also seeks to measure the degree of performance underestimation for each cross-validation estimator.

Simulation Design

The design of the simulation assesses 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: r (**Equation (1)**), RMSE (**Equation (2)**), and R^2 (**Equation (3)**). 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 $f_{\mathcal{D}_{\text{-}k}}$) to estimate regression coefficients β, which then predicts the target variable $\widehat{Y}_k$ from the test set $\mathcal{D}_k$. The procedure of K-fold CV can be expressed as:

$$Y_{\text{-}k} = f_{\mathcal{D}_{\text{-}k}}(X_{\text{-}k}) + \epsilon = X_{\text{-}k}\beta + \epsilon \qquad\qquad k = 1,2,\dots,K$$

$$\widehat{Y}_k = f_{\mathcal{D}_{\text{-}k}}(X_k) \qquad = X_k\beta$$

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

$$Y = f_{\mathcal{D}}(X) = X\beta + \epsilon$$

$$\widehat{Y} = f_{\mathcal{D}}(X) = X\beta$$

Where:

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

Estimated performance $\mathbb{E}[\widehat{g}(f_{\mathcal{D}})]$ was derived as previously detailed. To approximate true model performance $G(f_{\mathcal{D}})$, a hundred unseen datasets $\mathcal{D}^*$ were generated identically to $\mathcal{D}$, and the performance $G(f_{\mathcal{D}})$ was estimated by averaging the performance metrics across all $\mathcal{D}^*$. The bias and variance of the validation are computed according to **Equation (16)** and **(17)**, respectively.

Results

The simulation results, depicted in box plots (**Figure 5, 6**), explored the validation bias and variance distribution. Figure 5 examines the bias alterations across various estimators and sample sizes. Independent of the estimator and metric, the bias diminishes with increasing sample sizes. The in-sample estimator consistently overestimates across all metrics and sample sizes, underscoring the necessity of CV for unbiased performance evaluation. In CV estimators, although LOOCV is traditionally viewed as unbiased, it shows underestimation in model performance, especially when the metric is correlation coefficient (r). Comparatively, 2-, 5-, and 10-fold CV provide a more unbiased estimation than LOOCV for all sample sizes. However, for metrics like R^2 or RMSE, LOOCV emerges as the least biased estimator. While K-fold CV exhibits higher bias than LOOCV, this difference dwindles when the sample size exceeds 500. Notably, 10-fold CV, contrary to expectations, demonstrates higher bias than 5-fold CV for small sample sizes (50 and 100) in the R^2 metric, though this disparity also becomes insignificant at larger sample sizes.

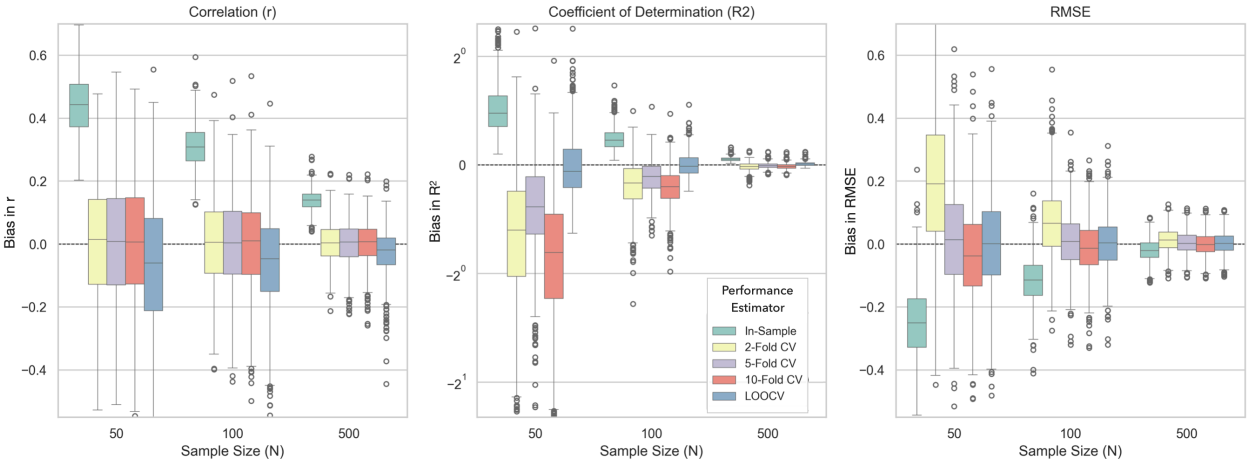


__Figure 5.__ Simulation results of validation bias from 1000 sampling iterations. Multiple performance estimators across different sample sizes were color-coded. Three metrics: r, R^2, and RMSE, were displayed in the column facets.

Considering LOOCV's singular data point testing, its validation variance is pertinent only for RMSE, which permits single data point evaluations. **Figure 6** illustrates the bias and variance in RMSE across different performance estimators as a function of sample size N. Both bias and variance in RMSE decrease as sample size increases, aligning with the hypothesis. LOOCV provides the least biased estimation, while 2-fold CV exhibits the highest bias without significant reduction at larger sample sizes. However, biases across all estimators converge at a sample size of 500. In terms of validation variance, LOOCV consistently shows higher values than other estimators for all sample sizes. Additionally, a lower number of folds K correlates with reduced variance, which is also in line with the hypothesized trend.

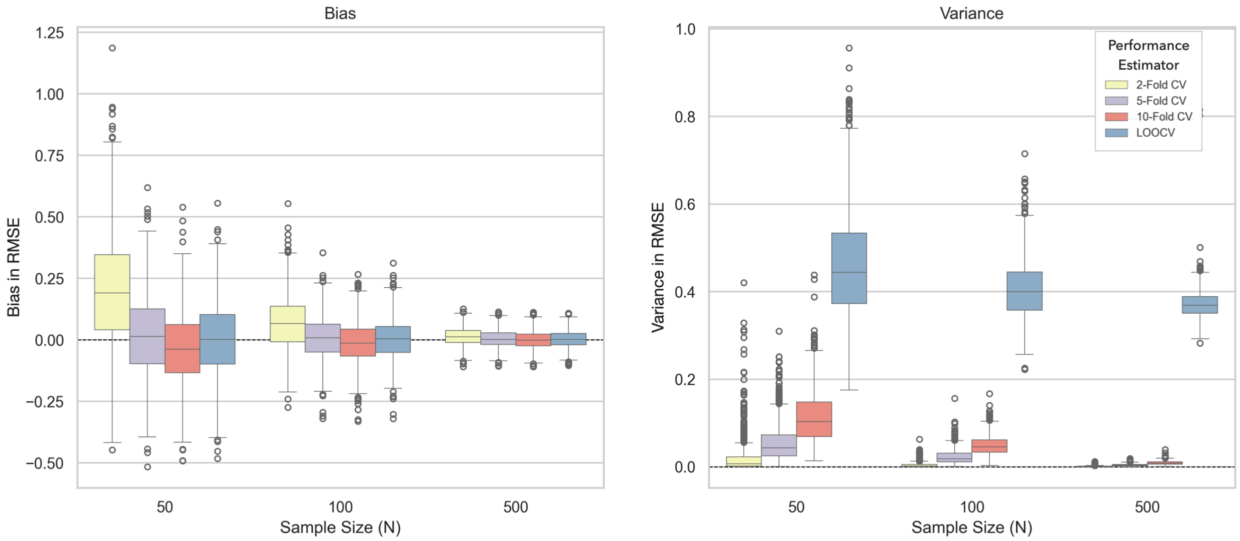


Figure 6. *Simulation results of validation bias and variance from 1000 sampling iterations. Multiple performance estimators across different sample sizes were color-coded. Only RMSE was displayed. Bias and variance were listed in the left and right facets, respectively.*

Section Conclusion

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

3. b Model Selection

Hyperparameter Tuning and Feature Selection

Model selection becomes necessary when models are not entirely determined by the data alone. For example, in a regularized linear regression model such as a ridge regression (Hoerl and Kennard, 1970) or the least absolute shrinkage and selection operator (**LASSO**) (Tibshirani, 1996), it is essential to define a regularization parameter, λ, before fitting the model to the data. A larger λ value yields a more regularized model, which tends to reduce smaller coefficients to negligible values or zero. This approach helps in preventing overfitting noise in the training data. The loss functions for unregularized ordinary least squares (**OLS**), ridge regression, and LASSO regression are given as follows:

$$\mathcal{L}_{OLS}(\beta) = \sum_{i=1}^{n} (y_i - x_i\beta)^2 \tag{19}$$

$$\mathcal{L}_{ridge}(\beta) = \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{20}$$

$$\mathcal{L}_{LASSO}(\beta) = \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right| \tag{21}$$

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

These pre-defined parameters, which influence model fitting and remain constant during the training process, are known as hyperparameters. Beyond regularized models, hyperparameters are crucial in other predictive models, enhancing flexibility and robustness. For example, in the Support Vector Regression (**SVR**) (Drucker et al., 1996), the regressors X are projected onto a linear subspace to approximate the target variable Y. By choosing a suitable kernel function, which transforms the regressors into a non-linear space, as a hyperparameter, SVR can more effectively capture non-linear relationships, thus significantly improving model performance. Another hyperparameter example is the number of latent variables in the Partial Least Square (**PLS**) Regression (Abdi, 2003), which condenses the original regressors into a more manageable set of latent variables, reducing multicollinearity issues. Fewer latent variables might lose significant information from the original regressors, while too many can lead to overfitting. Similarly, in Random Forest (Breiman, 2001), hyperparameters such as tree depth and the number of trees dictate model complexity. The same applies to the number of hidden layers and the size of filters in convolutional neural networks (LeCun, 1989). All these examples highlight the fact that selecting the most suitable hyperparameters, which is known as hyperparameter tuning, is crucial for optimizing model performance.

Feature selection is another crucial aspect of model selection. This process involves fitting the model to a selected subset of the original features, particularly essential in high-dimensional data scenarios where the number of features exceeds the number of observations, leading to poor model generalization. For instance, Ghaffari et al., 2019 sought to predict health traits in 38 multiparous Holstein cows using a metabolite profiling strategy. Out of 170 metabolites, only 12 were identified as effective discriminators between healthy and over-conditioned cows and were thus selected for the predictive model. Therefore, optimizing feature subsets is a vital model selection strategy that significantly affects model performance.

Including the model selection process within the cross-validation is essential to avoid common pitfalls. The risk of inflated model performance arises when model selection is guided by results on the test dataset. Even if the chosen model is subjected to k-fold cross-validation afterward, its selection bias toward the test set can lead to overestimating its efficacy. This issue has been highlighted in statistical literature (such as Hastie et al., 2009). A practical solution is to divide the dataset into training, validation, and test sets. The validation set is then used for model selection, ensuring the test set remains completely unused during the training phase, thereby providing a more accurate measure of model performance. For instance, the study by Rovere et al. exemplifies best practices in hyperparameter tuning and feature selection by employing an independent cross-validation step prior to assessing model performance. This approach enabled the precise selection of relevant spectral bands from the mid-infrared spectrum and the optimal number of latent dimensions in PLS with Bayesian regression for predicting the fatty acid profile in milk (Rovere et al., 2021). Similarly, Becker et al. demonstrated a robust evaluation by using nested cross-validation loops; the inner loop conducted a grid search for the best hyperparameters in logistic regression, while the outer loop was designed to evaluate the performance of the resulting optimized model (Becker et al., 2021). Both examples underscore the importance of separating model selection from performance evaluation to ensure the validity and reliability of the results.

Simulation Objectives and Hypothesis

The objective of this simulation study is to examine the effect of improper model selection implementation on validation bias. The focus will be on the model selection procedures of feature selection and hyperparameter tuning. The study hypothesizes that utilizing the test set inappropriately during any model selection stage will lead to a significant overestimation of model performance.

Simulation Design

This study simulated a regression task using an SVR model, which utilized various kernel functions to project a subset of features, X, to predict a target variable, Y. Both X and Y are drawn from a normal distribution $\mathcal{N}(0,1)$ to establish a baseline null correlation (performance $r = 0$) for assessing validation bias. This study set the sample size and number of features at 100 and 1000, respectively. Feature selection is executed by choosing the top 50 features that correlate most strongly with Y. For hyperparameter tuning, four kernel functions were evaluated: linear, polynomial, radial basis function, and sigmoid.

This study introduces notations **FS** for feature selection and **HT** for hyperparameter tuning, assigning a binary indicator (0 or 1) to denote incorrect (0) or correct (1) implementation of model selection. This yields four possible combinations of model selection strategies: "FS=0; HT=0", "FS=0; HT=1", "FS=1; HT=0", "FS=1; HT=1" (**Figure 7**). When FS=0, feature selection precedes cross-validation splitting. If FS=1, feature selection occurs within each fold of the training set during cross-validation. With hyperparameter tuning, a correct implementation (HT=1) involves splitting the dataset into training (64%), validation (16%), and test (20%) sets. The model is trained and tuned using the training and validation sets, respectively, while the test set is reserved for a single evaluation of model performance. Conversely, with HT=0, only training (80%) and test (20%) sets are used, risking validation bias as the test set informs both training and performance reporting. A 5-fold cross-validation approach was deployed for all strategies.

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

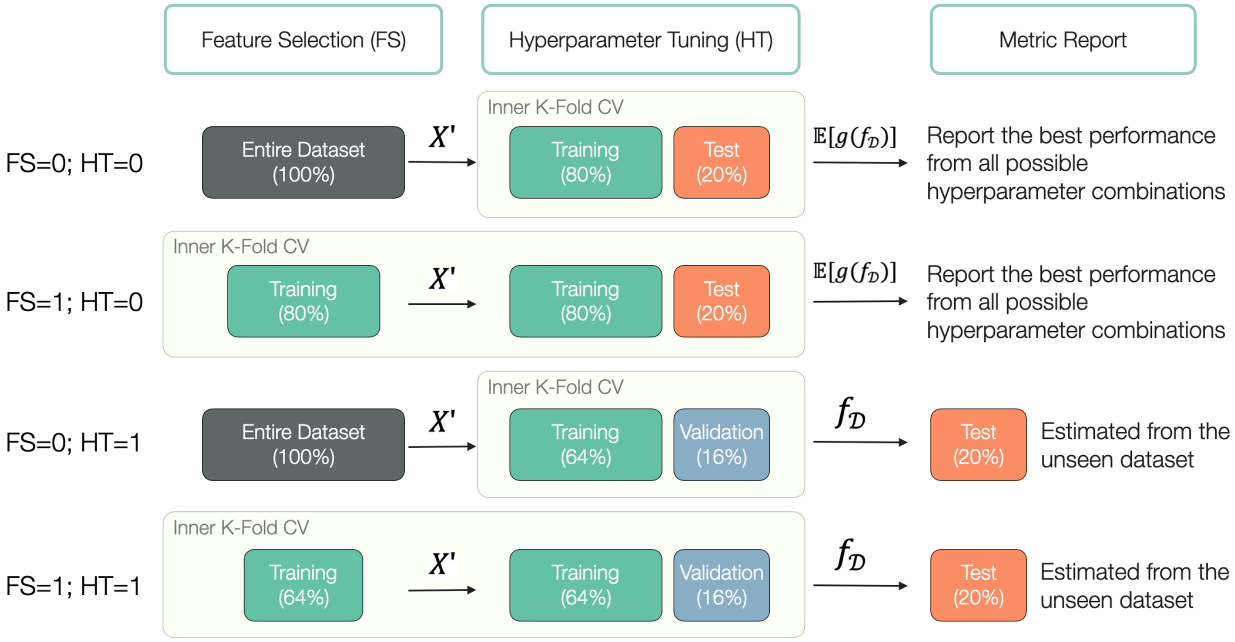


Figure 7. Workflow diagram illustrating four cross-validation strategies of feature selection (FS) and hyperparameter tuning (HT), where 0 denotes incorrect implementation and 1 indicates correct practice.

Result

The validation bias was visualized using box plots (**Figure 8**), with the feature selection factor (FS) on the x-axis and hyperparameter tuning (HT) distinguished by color — green for incorrect and yellow for correct implementation. The y-axis represents the validation bias as measured by the correlation coefficient. The results indicate a clear overestimation of model performance when feature selection is applied to the entire dataset, regardless of hyperparameter tuning. The median biases were 0.797 for “FS=0; HT=0” and 0.761 for “FS=0; HT=1”. Moreover, inappropriate validation in hyperparameter tuning resulted in a significant bias (p-value < 0.001) with a median of 0.113 for “FS=1; HT=0”. The only scenario without bias significantly occurred when both feature selection and hyperparameter tuning were correctly incorporated within the cross-validation process “FS=1; HT=1”, yielding a median bias of-0.008. These findings align with the initial hypothesis and the prevailing literature, reinforcing that model selection must be integrated into the cross-validation workflow to prevent an overestimation of model performance.

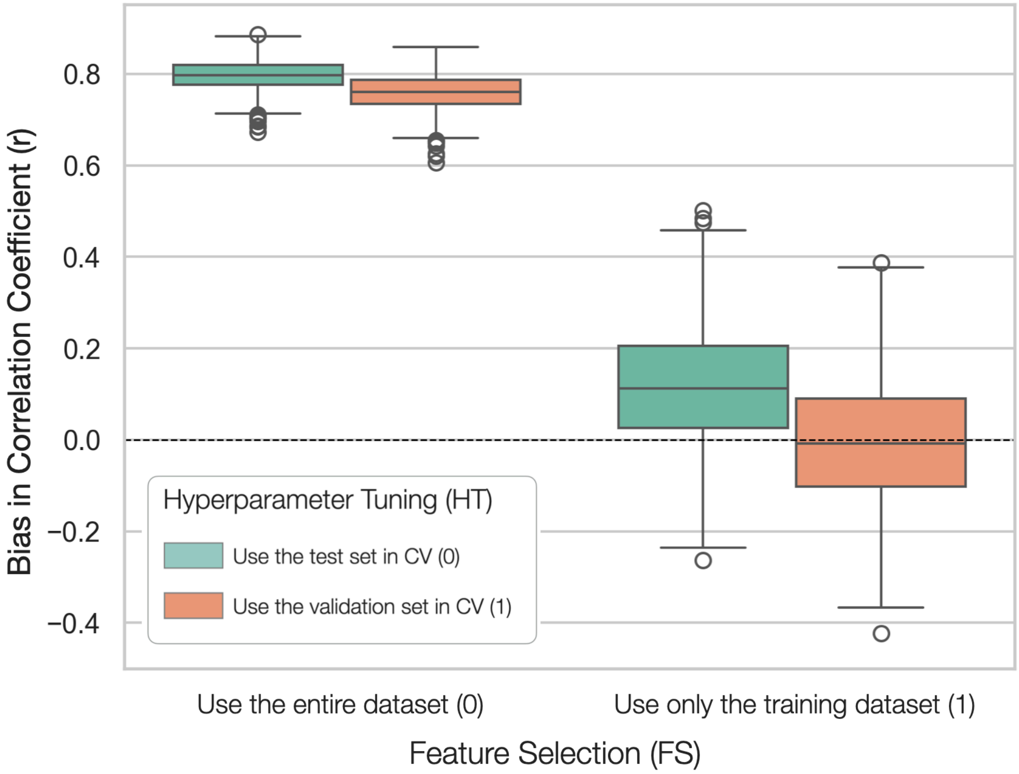


Figure 8. The validation bias of the four model selection strategies

<u>*Section Conclusion*</u>

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

3. c Block Cross Validation

Background

Blocking is an essential approach in experimental design to control for variations that can confound the variable of interest. For instance, Lahart et al., (2019) investigated the dry matter intake of grazing cows using mid-infrared (**MIR**) spectroscopy technology across multiple herds under varying experimental conditions. Given the significant variation between herds, which may contribute to individual differences in both dry matter intake and MIR spectra, it is crucial to consider the herd as a blocking factor before evaluating the predictability of dry matter intake using MIR spectra. This consideration should also extend to model validation. In the cited study, variations in dry matter intake, the primary focus of the prediction model, were observed to exceed one standard deviation among some herds. In cross-validation, if samples from the same herd are assigned to different folds, with one fold used as the test set, the model is likely to achieve high accuracy. This accuracy may largely result from explaining the inter-herd variation rather than individual variations in dry matter intake, leading to an overestimation of model performance. To avoid this pitfall, block cross-validation, where each block (i.e., herd in this example) is used as a fold, is recommended for unbiased model validation.

Literature reviews have indicated that block cross-validation effectively evaluates model performance on external or unseen datasets (Bresolin and Dórea, 2020). In the same study by Lahart et al., three cross-validation strategies were compared: random cross-validation (**Random CV**), which randomly assigns samples to folds; within-herd validation, training and testing the model within each herd; and across-herd validation (**Block CV**), where each herd is used as a fold and tested in turn. The results showed that performance estimates in block CV were noticeably lower than the other two strategies, supporting the hypothesis that ignoring block effects inflates model performance. Other studies considering block effects, including diet (Grelet et al., 2020), herd (Rovere et al., 2021), and farm location (Adriaens et al., 2020; Mota et al., 2022), have shown similar results in cross-validation, demonstrating block CV's effectiveness in evaluating model performance on external datasets.

Objectives and Hypothesis

The objective of the simulation study is to demonstrate how a Random CV, which randomly assigns the samples to folds without considering the block effects, could overestimate the model performance. This study also conducts a block CV, where each block is used as a fold in the cross-validation, as the benchmark. The hypothesis is that the model performance estimated by Random CV is significantly higher than the estimation by block CV.

Simulation Design

This study simulated a regression task with 100 instances across ten features, denoted as X, and one single response variable, Y. Both X and Y are derived from a standard normal distribution. To introduce a block factor, the study groups every 20 observations into a block, with each block incrementally increasing by b units from zero, where b was simulated from 0.5 to 3.0 with an increment of 0.5. Within these ten features, one is substituted as the block level, represented by an integer from 0 to 4, augmented with random noise drawn from a standard normal distribution. This setup aims to simulate a scenario where the predictors primarily capture block variation, given the null expectation in predictability when using ten random variables X to forecast another random variable Y.

The study investigates two model validation strategies: Block CV and Random CV, both utilizing a 5-fold cross-validation method. In block CV, each block serves as a separate fold, while in Random CV, samples are randomly allocated to each fold (**Figure 9**). The predictive model is linear regression, and the performance is evaluated using Pearson's correlation coefficient. This simulation runs for 1000 iterations, with X and Y being resampled in each cycle. A one-tailed t-test assesses if the mean estimated performance significantly exceeds zero. Additionally, an Analysis of Variance (**ANOVA**) table is calculated when b is 0.5 to ascertain if the simulated block variation notably exceeds the assumed individual variation, representing the primary interest.

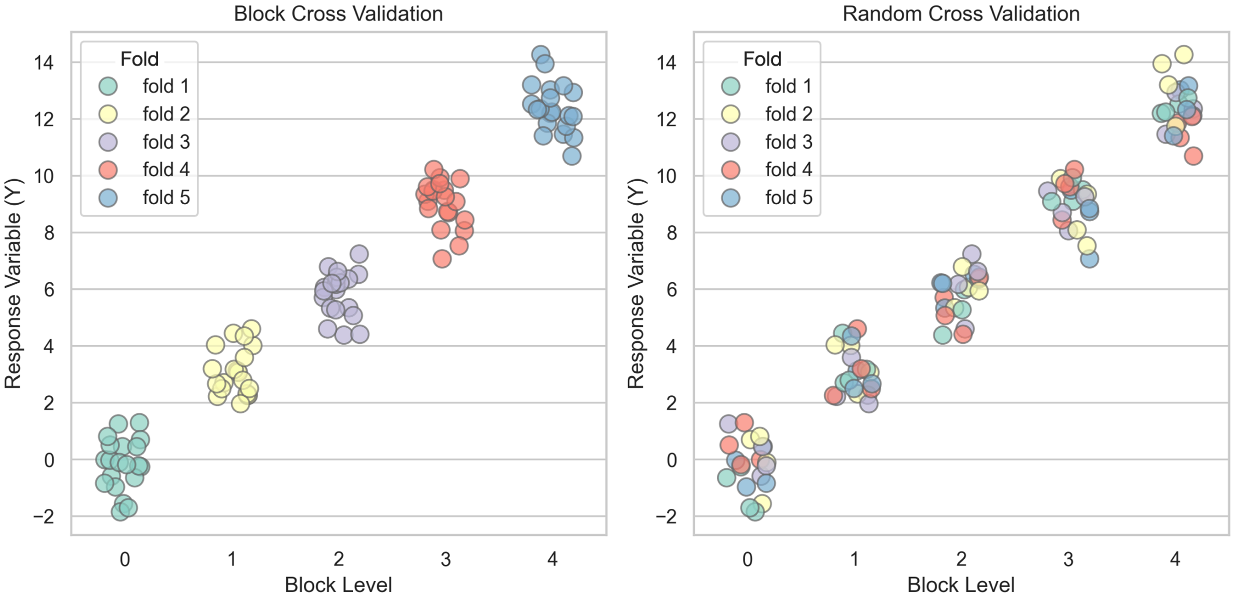


*Figure 9. Illustration of fold assignment in block cross validation (**left**) and random cross validation (**right**). Folds are color-coded, and the block effect is set to 3 in this example.*

<u>*Result*</u>

In this simulation, an ANOVA table (**Table 1**), calculated from a single iteration for illustrative purposes, demonstrates that the simulated data exhibits block variation significantly greater than the residual variance. The result (**Figure 10**) shows that regardless of the amplitude of block effects in this simulation study, the Block CV strategy consistently yields a mean performance estimate close to zero, while the Random CV strategy consistently and significantly overestimates the model performance (p-value < 0.001). This finding supports the hypothesis that Random CV tends to overestimate model performance when block variation predominates over residual variation.

Table 1. 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	131.335	99			

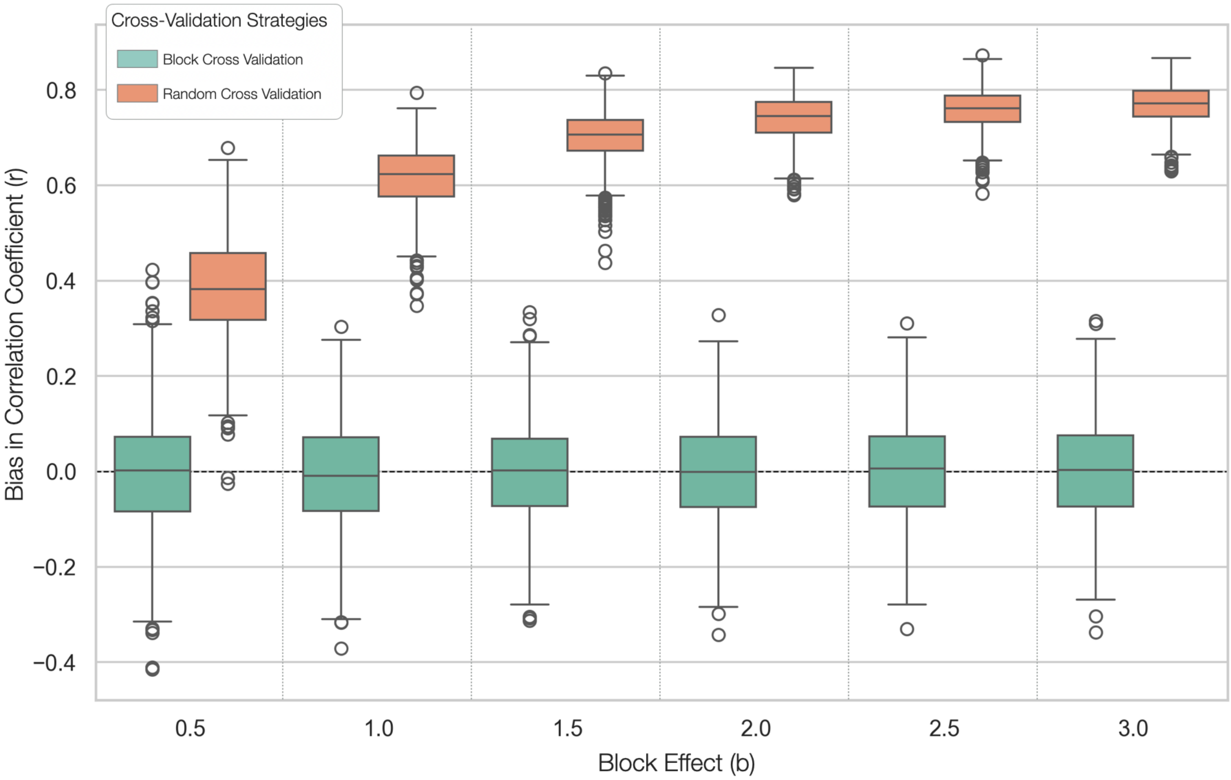


Figure 10. Bias in model performance estimation by Block CV and Random CV across 1000 iterations. The dashed line represents the null hypothesis that the mean performance estimate is zero.

<u>Section Conclusion</u>

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

4. Conclusion

In summary, the review highlights several key considerations for performance assessment and validation in predictive modeling.

When evaluating regression models, the choice of metrics like Correlation Coefficient r, RMSE, and R^2 depends on the specific goals of the model. A comprehensive evaluation should include multiple metrics to understand different aspects of model performance. In binary classification models, precision and recall are crucial, but it is essential to correctly designate the positive class to avoid bias. Label-invariant metrics, such as the ROC curve and the proposed MCC curve, provide a balanced assessment, unaffected by class label choices.

Additionally, the reliability of model validation is significantly influenced by estimator choice and sample size. Larger sample sizes tend to reduce bias and variance, increasing validation reliability. Cross-validation methods, such as K-fold CV and LOOCV, are preferable for unbiased performance estimation, with the number of folds in K-fold CV being particularly influential in smaller datasets. Moreover, the review underscores the importance of correct implementation in model selection processes, as improper techniques can inflate performance estimates. This is especially true in complex models where feature selection and hyperparameter tuning need meticulous cross-validation to avoid overestimation of performance. Finally, the utility of Block CV is emphasized in contexts where block effects are significant. It provides a more realistic assessment of model generalizability and accuracy compared to a Random CV, which tends to overestimate performance in such scenarios.

Overall, the review recommends a thoughtful selection of metrics and validation techniques, tailored to the specific dataset and modeling objectives, to ensure accurate and reliable performance assessments in predictive modeling.

References

Abdi, H. 2003. Partial Least Square Regression PLS-Regression. Encyclopedia of social sciences research methods 792–795.

Adriaens, I., N.C. Friggens, W. Ouweltjes, H. Scott, B. Aernouts, and J. Statham. 2020. Productive life span and resilience rank can be predicted from on-farm first-parity sensor time series but not using a common equation across farms. Journal of Dairy Science 103:7155–7171. doi:10.3168/jds.2019-17826.

Alsaaod, M., M. Fadul, and A. Steiner. 2019. Automatic lameness detection in cattle. The Veterinary Journal 246:35–44. doi:10.1016/j.tvjl.2019.01.005.

Appuhamy, J.A.D.R.N., J.V. Judy, E. Kebreab, and P.J. Kononoff. 2016. Prediction of drinking water intake by dairy cows. Journal of Dairy Science 99:7191–7205. doi:10.3168/jds.2016-10950.

Becker, C.A., A. Aghalari, M. Marufuzzaman, and A.E. Stone. 2021. Predicting dairy cattle heat stress using machine learning techniques. Journal of Dairy Science 104:501–524. doi:10.3168/jds.2020-18653.

Borchers, M.R., Y.M. Chang, K.L. Proudfoot, B.A. Wadsworth, A.E. Stone, and J.M. Bewley. 2017. Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle. Journal of Dairy Science 100:5664–5674. doi:10.3168/jds.2016-11526.

Bowen, J.M., M.J. Haskell, G.A. Miller, C.S. Mason, D.J. Bell, and C.-A. Duthie. 2021. Early prediction of respiratory disease in preweaning dairy calves using feeding and activity behaviors. Journal of Dairy Science 104:12009–12018. doi:10.3168/jds.2021-20373.

Breiman, L. 2001. Random Forests. Machine Learning 45:5–32. doi:10.1023/A:1010933404324.

Bresolin, T., and J.R.R. Dórea. 2020. Infrared Spectrometry as a High-Throughput Phenotyping Technology to Predict Complex Traits in Livestock Systems. Frontiers in Genetics 11.

Cawley, G.C., and N.L.C. Talbot. 2010. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. J. Mach. Learn. Res. 11:2079–2107.

Cheng, H., D.J. Garrick, and R.L. Fernando. 2017. Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction. Journal of Animal Science and Biotechnology 8:38. doi:10.1186/s40104-017-0164-6.

Chicco, D., and G. Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21:6. doi:10.1186/s12864-019-6413-7.

Delhez, P., P.N. Ho, N. Gengler, H. Soyeurt, and J.E. Pryce. 2020. Diagnosing the pregnancy status of dairy cows: How useful is milk mid-infrared spectroscopy?. Journal of Dairy Science 103:3264–3274. doi:10.3168/jds.2019-17473.

Denholm, S.J., W. Brand, A.P. Mitchell, A.T. Wells, T. Krzyzelewski, S.L. Smith, E. Wall, and M.P. Coffey. 2020. Predicting bovine tuberculosis status of dairy cows from mid-infrared spectral data of milk using deep learning. Journal of Dairy Science 103:9355–9367. doi:10.3168/jds.2020-18328.

van Dixhoorn, I.D.E., R.M. de Mol, J.T.N. van der Werf, S. van Mourik, and C.G. van Reenen. 2018. Indicators of resilience during the transition period in dairy cows: A case study. Journal of Dairy Science 101:10271–10282. doi:10.3168/jds.2018-14779.

Dórea, J.R.R., G.J.M. Rosa, K.A. Weld, and L.E. Armentano. 2018. Mining data from milk infrared spectroscopy to improve feed intake predictions in lactating dairy cows. Journal of Dairy Science 101:5878–5889. doi:10.3168/jds.2017-13997.

Drucker, H., C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. 1996. Support vector regression machines. Pages 155–161 in Proceedings of the 9th International Conference on Neural Information Processing Systems. MIT Press, Cambridge, MA, USA.

Frizzarin, M., I.C. Gormley, D.P. Berry, T.B. Murphy, A. Casa, A. Lynch, and S. McParland. 2021. Predicting cow milk quality traits from routinely available milk spectra using statistical machine learning methods. Journal of Dairy Science 104:7438–7447. doi:10.3168/jds.2020-19576.

Ghaffari, M.H., A. Jahanbekam, H. Sadri, K. Schuh, G. Dusel, C. Prehn, J. Adamski, C. Koch, and H. Sauerwein. 2019. Metabolomics meets machine learning: Longitudinal metabolite profiling in serum of normal versus overconditioned cows and pathway analysis. Journal of Dairy Science 102:11561–11585. doi:10.3168/jds.2019-17114.

Grelet, C., E. Froidmont, L. Foldager, M. Salavati, M. Hostens, C.P. Ferris, K.L. Ingvartsen, M.A. Crowe, M.T. Sorensen, J.A. Fernandez Pierna, A. Vanlierde, N. Gengler, and F. Dehareng. 2020. Potential of milk mid-infrared spectra to predict nitrogen use efficiency of individual dairy cows in early lactation. Journal of Dairy Science 103:4435–4445. doi:10.3168/jds.2019-17910.

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

Hoerl, A.E., and R.W. Kennard. 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics 12:55–67. doi:10.2307/1267351.

Jensen, D.B., H. Hogeveen, and A. De Vries. 2016. Bayesian integration of sensor information and a multivariate dynamic linear model for prediction of dairy cow mastitis. Journal of Dairy Science 99:7344–7361. doi:10.3168/jds.2015-10060.

Kandeel, S.A., A.A. Megahed, M.H. Ebeid, and P.D. Constable. 2019. Ability of milk pH to predict subclinical mastitis and intramammary infection in quarters from lactating dairy cattle. Journal of Dairy Science 102:1417–1427. doi:10.3168/jds.2018-14993.

Kang, X., X.D. Zhang, and G. Liu. 2020. Accurate detection of lameness in dairy cattle with computer vision: A new and individualized detection strategy based on the analysis of the supporting phase. Journal of Dairy Science 103:10628–10638. doi:10.3168/jds.2020-18288.

Lahart, B., S. McParland, E. Kennedy, T.M. Boland, T. Condon, M. Williams, N. Galvin, B. McCarthy, and F. Buckley. 2019. Predicting the dry matter intake of grazing dairy cows using infrared reflectance spectroscopy analysis. Journal of Dairy Science 102:8907–8918. doi:10.3168/jds.2019-16363.

LeCun, Y. 1989. Backpropagation Applied to Handwritten Zip Code Recognition.

Mäntysaari, P., E.A. Mäntysaari, T. Kokkonen, T. Mehtiö, S. Kajava, C. Grelet, P. Lidauer, and M.H. Lidauer. 2019. Body and milk traits as indicators of dairy cow energy status in early lactation. Journal of Dairy Science 102:7904–7916. doi:10.3168/jds.2018-15792.

Mota, L.F.M., D. Giannuzzi, V. Bisutti, S. Pegolo, E. Trevisi, S. Schiavon, L. Gallo, D. Fineboym, G. Katz, and A. Cecchinato. 2022. Real-time milk analysis integrated with stacking ensemble learning as a tool for the daily prediction of cheese-making traits in Holstein cattle. Journal of Dairy Science 105:4237–4255. doi:10.3168/jds.2021-21426.

O'Leary, N.W., D.T. Byrne, A.H. O'Connor, and L. Shalloo. 2020. Invited review: Cattle lameness detection with accelerometers. Journal of Dairy Science 103:3895–3911. doi:10.3168/jds.2019-17123.

Ouellet, V., E. Vasseur, W. Heuwieser, O. Burfeind, X. Maldague, and É. Charbonneau. 2016. Evaluation of calving indicators measured by automated monitoring devices to predict the onset of calving in Holstein dairy cows. Journal of Dairy Science 99:1539–1548. doi:10.3168/jds.2015-10057.

Rovere, G., G. de los Campos, A.L. Lock, L. Worden, A.I. Vazquez, K. Lee, and R.J. Tempelman. 2021. Prediction of fatty acid composition using milk spectral data and its associations with various mid-infrared spectral regions in Michigan Holsteins. Journal of Dairy Science 104:11242–11258. doi:10.3168/jds.2021-20267.

Song, X., E.A.M. Bokkers, P.P.J. Van Der Tol, P.W.G. Groot Koerkamp, and S. Van Mourik. 2018. Automated body weight prediction of dairy cows using 3-dimensional vision. Journal of Dairy Science 101:4448–4459. doi:10.3168/jds.2017-13094.

de Souza, R.A., R.J. Tempelman, M.S. Allen, W.P. Weiss, J.K. Bernard, and M.J. VandeHaar. 2018. Predicting nutrient digestibility in high-producing dairy cows. Journal of Dairy Science 101:1123–1135. doi:10.3168/jds.2017-13344.

Spoliansky, R., Y. Edan, Y. Parmet, and I. Halachmi. 2016. Development of automatic body condition scoring using a low-cost 3-dimensional Kinect camera. Journal of Dairy Science 99:7714–7725. doi:10.3168/jds.2015-10607.

Stojkov, J., G. Bowers, M. Draper, T. Duffield, P. Duivenvoorden, M. Groleau, D. Haupstein, R. Peters, J. Pritchard, C. Radom, N. Sillett, W. Skippon, H. Trépanier, and D. Fraser. 2018. Hot topic: Management of cull dairy cows—Consensus of an expert consultation in Canada. Journal of Dairy Science 101:11170–11174. doi:10.3168/jds.2018-14919.

Tibshirani, R. 1996. Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58:267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.

Xavier, C., Y. Le Cozler, L. Depuille, A. Caillot, A. Lebreton, C. Allain, J.M. Delouard, L. Delattre, T. Luginbuhl, P. Faverdin, and A. Fischer. 2022. The use of 3-dimensional imaging of Holstein cows to estimate body weight and monitor the composition of body weight change throughout lactation. Journal of Dairy Science 105:4508–4519. doi:10.3168/jds.2021-21337.

Yukun, S., H. Pengju, W. Yujie, C. Ziqi, L. Yang, D. Baisheng, L. Runze, and Z. Yonggen. 2019. Automatic monitoring system for individual dairy cows based on a deep learning framework that provides identification via body parts and estimation of body condition score. Journal of Dairy Science 102:10140–10151. doi:10.3168/jds.2018-16164.